STEm-Seg: Spatio-temporal Embeddings for Instance Segmentation in Videos

Ali Athar*[1], Sabarinath Mahadevan*[1], Aljoša Ošep[2], Laura Leal-Taixé[2], and Bastian Leibe[1]

[1] RWTH Aachen University, Germany
{athar,mahadevan,leibe}@vision.rwth-aachen.de
[2] Technical University of Munich, Germany
{aljosa.osep,leal.taixe}@tum.de

Abstract. Existing methods for instance segmentation in videos typically involve multi-stage pipelines that follow the tracking-by-detection paradigm and model a video clip as a sequence of images. Multiple networks are used to detect objects in individual frames, and then associate these detections over time. Hence, these methods are often non-end-to-end trainable and highly tailored to specific tasks. In this paper, we propose a different approach that is well-suited to a variety of tasks involving instance segmentation in videos. In particular, we model a video clip as a single 3D spatio-temporal volume, and propose a novel approach that segments and tracks instances across space and time in a single stage. Our problem formulation is centered around the idea of spatio-temporal embeddings which are trained to cluster pixels belonging to a specific object instance over an entire video clip. To this end, we introduce (i) novel mixing functions that enhance the feature representation of spatio-temporal embeddings, and (ii) a single-stage, proposal-free network that can reason about temporal context. Our network is trained end-to-end to learn spatio-temporal embeddings as well as parameters required to cluster these embeddings, thus simplifying inference. Our method achieves state-of-the-art results across multiple datasets and tasks. Code and models are available at https://github.com/sabarim/STEm-Seg.

1 Introduction

The task of segmenting and tracking multiple objects in videos is becoming increasingly popular due to a surge in development of autonomous vehicles and robots that are able to perceive and accurately track surrounding objects. These advances are driven by the recent emergence of new datasets [11, 80, 95] containing videos with dense, per-pixel annotations of object instances. The underlying task tackled in these datasets can be summarized as follows: given an input video containing multiple objects, each pixel has to be uniquely assigned to a specific object instance or to the background.
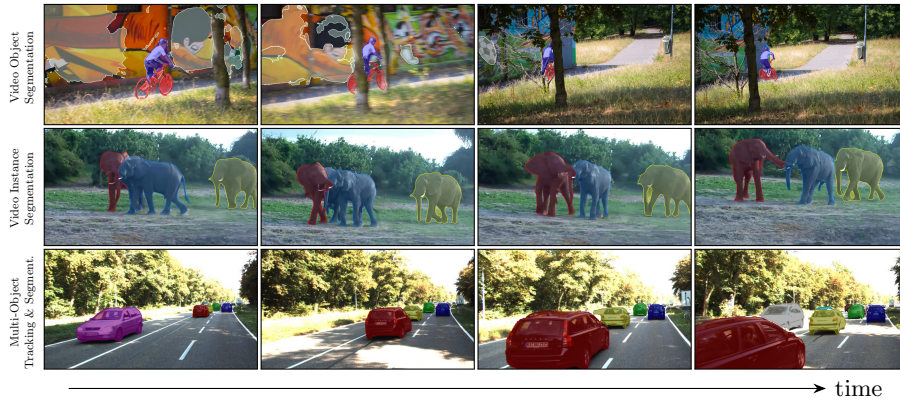
---

* Equal contribution

Fig. 1: Our method is applicable to multi-object segmentation tasks such as VOS *(top)*, VIS *(middle)* and MOTS *(bottom)*.

State-of-the-art methods tackling these tasks [80, 83, 95] usually operate in a *top-down* fashion and follow the *tracking-by-detection* paradigm which is well-established in multi-object tracking (MOT) [8, 37, 47, 62]. Such methods usually employ multiple networks to detect objects in individual images [32], associate them over consecutive frames, and resolve occlusions using learned appearance models. Though these approaches yield high-quality results, they involve multiple networks, are computationally demanding, and not end-to-end trainable.

Inspired by the Perceptual Grouping Theory [66], we learn to segment object instances in videos in a *bottom-up* fashion by leveraging *spatio-temporal* embeddings. To this end, we propose an efficient, single-stage network that operates directly on a 3D spatio-temporal volume. We train the embeddings in a category-agnostic setting, such that pixels belonging to the same object instance across the spatio-temporal volume are mapped to a single cluster in the embedding space. This way, we can infer object instances by simply assigning pixels to their respective clusters. Our method outperforms proposal-based methods for tasks involving pixel-precise tracking such as Unsupervised Video Object Segmentation (UVOS) [9, 11], Video Instance Segmentation (VIS) [95], and Multi-Object Tracking and Segmentation (MOTS) [80].

To summarize, our contributions are the following: (i) We propose a unified approach for tasks involving instance segmentation in videos [11, 80, 95]. Our method performs consistently well under highly varied settings (see Fig. 1) such as automotive driving scenes [80], semantically diverse YouTube videos [95] and scenarios where object classes are not pre-defined [11]. (ii) We propose using spatio-temporal embeddings for the aforementioned set of tasks. To this end, we propose a set of mixing functions (Sec. 3.2) that improve performance by modifying the feature representation of these embeddings. Our method enables a simple inference procedure based on clustering within a 3D spatio-temporal volume, thus alleviating the need for external components for temporal association. (iii) We propose a single-stage network architecture which is able to effectively incorporate temporal context and learn the spatio-temporal embeddings.

2   Related Work

Image-level Instance Segmentation: Several existing methods for image-level instance segmentation, which is closely related to our task, operate in a *top-down* fashion by using a Region Proposal Network to generate object proposals [13, 69, 70, 73], which are then classified and segmented [32]. Other methods operate in a *bottom-up* fashion by grouping pixels belonging to the same object instance [18, 19, 42, 57–59]. Novotny *et al.* [59] introduce an embedding mixing function to overcome appearance ambiguity, and predict a displacement field from the instance-specific center, which is similar to Hough voting based methods for object detection [46, 81]. Recent methods for 3D object detection and instance segmentation also follow this trend [22, 40, 72, 21, 101]. Neven *et al.* [57] extend [18] by training a network to predict object centers and clustering bandwidths, thus alleviating the need for density-based clustering algorithms [16, 23, 54]. This serves as a basis for our work. However, in contrast to our approach, the aforementioned methods are only suitable for image-level segmentation.

Video Segmentation: Temporally consistent object segmentation in videos can benefit several other tasks such as action/activity recognition [29, 34], video object detection [41, 25] and object discovery [44, 64, 82, 91, 17, 92]. Several *bottom-up* methods segment moving regions by grouping based on optical flow [2, 93] or point trajectories [7, 60, 67, 92]. By contrast, our method segments and tracks both moving and static objects. Other methods obtain video object proposals [25, 29, 34, 41] based on cues such as actions [29, 34] or image-level objects of interest [41, 65]. Feature representations are then learned for these proposals in order to perform classification. Instead, we propose a single-stage approach that localizes objects with pixel-level precision.

Pixel-precise Tracking of Multiple Objects: Multi object tracking (MOT) has its roots in robotic perception [38, 68, 76, 86]. Although some works extend MOT with pixel-precise masks [56, 63], a much larger set of works can be found in the domain of Video Object Segmentation (VOS), which encompasses multiple tasks related to pixel-precise tracking. In the *semi-supervised* variant of VOS [10], the ground-truth masks of the objects which are meant to be tracked are given for the first frame. State-of-the-art approaches to this task [9, 36, 61, 77–79, 89, 94, 96, 100] usually involve online fine-tuning on the first frame masks and/or follow the *object proposal generation and mask propagation* approach [53, 100]. Chen *et al.* [14] tackle this problem by learning embeddings from the first-frame masks, and then associating pixels in the remaining frames.

More relevant to our work is the task of *unsupervised* VOS [11, 71]. Here, no ground truth information is provided at test-time, and the goal is to segment and track all dominant "foreground" objects in the video clip. Current state-of-the-art methods for this task are either proposal-based [100, 102] or focus on foreground-background segmentation [35, 39, 74, 75, 77, 78, 97, 98]. Li *et al.* [48] propose an approach that groups pixel embeddings based on *objectness* and optical flow cues. In contrast to ours, this method processes each frame separately, employs K-means clustering for object localization, and cannot separate different

object instances. Wang *et al.* [84] employ an attentive graph neural network [30] and use differentiable message passing to propagate information across time; we compare our results to theirs in Sec. 4.5.

Recently, the task of Multi-Object Tracking and Segmentation (MOTS) [80] and Video Instance Segmentation (VIS) [95] were introduced. Voigtlaender *et al.* [80] extended the KITTI [28] and MOTChallenge [45, 55] datasets with pixel-precise annotations, and proposed a baseline method that adapts Mask R-CNN [32] to associate object proposals across time. Hu *et al.* [1] use a mask network to filter foreground embeddings and perform mean-shift clustering to generate object masks, which are then associated over time using a distance threshold unlike our single-stage end-to-end method.

The YouTube-VIS [95] contains a large number of YouTube videos with per-pixel annotations for a variety of object classes. Compared to MOTS, these videos contain fewer instances, but are significantly more diverse. In addition to adapting several existing methods [6, 31, 79, 85, 96] for this task, Yang *et al.* [95] also proposed their own method (MaskTrack-RCNN) which extends Mask R-CNN with additional cues for data association. Methods such as [20, 26, 51] also rely on object proposals and/or heuristic post-processing to associate objects over time, unlike our end-to-end bottom-up approach.

3   Our Method

We tackle the task of instance segmentation in videos by modeling the video clip as a 3D spatio-temporal volume and using a network to learn an embedding for each pixel in that volume. This network is trained to push pixels belonging to different object instances towards different, non-overlapping clusters in the embedding space. This differs from most existing approaches, which first generate object detections per-frame, and then associate them over time. The following sections explain our problem formulation, the network architecture and loss functions employed, and the inference process.

3.1   Problem Formulation

As input, we assume a video clip with T frames of resolution $H \times W$. Let us denote the set of RGB pixels in this clip with $\mathcal{X} \in \mathbb{R}^{N \times 3}$ where $N = T \times H \times W$. Assuming there are K object instances in this clip (K is unknown), our aim is to produce a segmentation that assigns each pixel in the clip to either the background, or to exactly one of these K instances. We design a network that predicts video instance tubes by clustering pixels simultaneously across space and time based on a learned embedding function. Instead of learning just the embedding function and then using standard density-based clustering algorithms [27, 52, 54] to obtain instance tubes, we take inspiration from Neven *et al.* [57] and design a network which estimates the cluster centers as well as their corresponding variances, thus enabling efficient inference. Formally, our network can be viewed as a function that maps the set of pixels $\mathcal{X}$ to three outputs; (1)

$\mathcal{E} \in \mathbb{R}^{N \times E}$: an E-dimensional embedding for each pixel, (2) $\mathcal{V} \in \mathbb{R}_{+}^{N \times E}$: a positive variance for each pixel and each embedding dimension, and (3) $\mathcal{H} \in [0, 1]^N$: an object instance center heat-map.

Instance representation: The network is trained such that the embeddings belonging to the jth instance in the video clip are modeled by a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$. Assuming that this instance comprises the set of pixel coordinates $\mathcal{C}_j$ with cardinality N_j, we denote the embeddings and variances output by the network at these coordinates using $\mathcal{E}_j \subset \mathcal{E}$, $\mathcal{E}_j \in \mathbb{R}^{N_j \times E}$ and $\mathcal{V}_j \subset \mathcal{V}$, $\mathcal{V}_j \in \mathbb{R}_{+}^{N_j \times E}$, respectively. During training, $\mathcal{C}_j$ (*i.e.*, the ground truth mask tube for instance j) is known. Using it, the mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$ of the distribution are computed by averaging over the per-pixel outputs:

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{\boldsymbol{e} \in \mathcal{E}_j} \boldsymbol{e} \in \mathbb{R}^E, \qquad \boldsymbol{\Sigma}_j = \frac{1}{N_j} \operatorname{diag}\left(\sum_{\boldsymbol{v} \in \mathcal{V}_j} \boldsymbol{v} \right) \in \mathbb{R}^{E \times E} . \qquad (1)$$

This single distribution models all embeddings belonging to instance j across the entire clip (not individually for each frame). We can now use the distribution $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ to compute the probability p_{ij} of any embedding $\boldsymbol{e}_i \in \mathcal{E}$, anywhere in the input clip, of belonging to instance j:

$$p_{ij} = \frac{1}{(2\pi)^{\frac{E}{2}} |\boldsymbol{\Sigma_j}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{e}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{e}_i - \boldsymbol{\mu}_j) \right) . \qquad (2)$$

Using Eq. 2 to compute p_{ij}, $\forall\, i \in \{1, .., N\}$, we can obtain the set of pixels $\widehat{\mathcal{C}}_j$ comprising the predicted mask tube for instance j by thresholding the probabilities at 0.5:

$$\widehat{\mathcal{C}}_j = \left\{ (x_i, y_i, t_i) \,|\, i \in \{1, .., N\},\ p_{ij} > 0.5 \right\} . \qquad (3)$$

Training: This way, the training objective can be formulated as one of learning the optimal parameters $\boldsymbol{\mu}_j^{\text{opt}}$ and $\boldsymbol{\Sigma}_j^{\text{opt}}$ which maximize the intersection-over-union (IoU) between the predicted and ground-truth mask tubes in the clip:

$$\boldsymbol{\mu}_j^{\text{opt}}, \boldsymbol{\Sigma}_j^{\text{opt}} = \operatorname*{argmax}_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j} \frac{\widehat{\mathcal{C}}_j \cap \mathcal{C}_j}{\widehat{\mathcal{C}}_j \cup \mathcal{C}_j}, \qquad (4)$$

that is, all pixels in the ground truth mask tube $\mathcal{C}_j$ should have probability larger than 0.5, and vice versa. A classification loss such as cross-entropy could be used to optimize the pixel probabilities. However, we employ the Lovàsz hinge loss [3, 4, 57, 99] (details in supplementary material), which is a differentiable, convex surrogate of the Jaccard index that directly optimizes the IoU. This formulation allows $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ to be implicitly learned by the network.

Using Eqs. 1-4, we can define and optimize the distribution for every instance (*i.e.* $\forall j \in \{1, ..., K\}$). Note that only a single forward pass of the network is required regardless of the number of instances in the clip. This is in contrast

to common approaches for Video Object Segmentation, which only process one instance at a time [61, 90].

We further remark that ours is a *bottom-up approach* which detects and tracks objects in a single step, thus mitigating the inherent drawback of *top-down* approaches that often require different networks/cues for single-image object detection and temporal association. A further advantage of our approach is that it can implicitly resolve occlusions, insofar as they occur within the clip.

3.2   Embedding Representation

Under the formulation described in Sec. 3.1, the network can learn arbitrary representations for the embedding dimensions. However, it is also possible to fix the representation by using a mixing function $\phi : \mathbb{R}^E \to \mathbb{R}^E$ that modifies the embeddings $\mathcal{E}$ as follows: $\mathcal{E} \leftarrow \{\, \phi(\boldsymbol{e}), \mid \boldsymbol{e} \in \mathcal{E} \,\}$.

In [57], for the task of single-image instance segmentation, 2D embeddings were used ($E = 2$) in conjunction with a spatial coordinate mixing function $\phi_{\mathrm{xy}}(\boldsymbol{e}_i) = \boldsymbol{e}_i + [x_i, y_i]$ [3]. With this setting, the embeddings could be interpreted as offsets to the (x, y) coordinates of their respective locations. The network thus learned to cluster the embeddings belonging to a given object towards some object-specific point on the image. It follows that the predicted variances could be interpreted as the network's estimate of the size of the object along the x and y axes. We postulate that the reason for this formulation yielding good results is that the (x, y) coordinates already serve as a good initial feature for instance separation; the network can then enhance this representation by producing offsets which further improve the clustering behavior. Furthermore, this can be done in an end-to-end trainable fashion which allows the network to adjust the clustering parameters, *i.e.*, the Gaussian distribution parameters, for each instance. In general, it has been shown that imparting spatial coordinate information to CNNs can improve performance for a variety of tasks [50, 59].

Compared to single-image segmentation, the task of associating pixels across space and time in in videos poses additional challenges, *e.g.*, camera ego-motion, occlusions, appearance/pose changes. To tackle these challenges we propose (and experimentally validate in Sec. 4.3) the following extensions:

Spatio-temporal coordinating mixing: Since we operate on video clips instead of single images, a logical extension is to use 3D embeddings ($E = 3$) with a spatio-temporal coordinate mixing function $\phi_{\mathrm{xyt}}(\boldsymbol{e}_i) = \boldsymbol{e}_i + [x_i, y_i, t_i]$.

Free dimensions: In addition to the spatial (and temporal) coordinate dimensions, it can be beneficial to include extra dimensions whose representation is left for the network to decide. The motivation here is to improve instance clustering quality by allowing additional degrees of freedom in the embedding space. From here on, we shall refer to these extra embedding dimensions as *free dimensions*. For example, if $E = 4$ with 2 spatial coordinate dimensions and 2 free dimensions, the mixing function is denoted as $\phi_{\mathrm{xyff}}(\boldsymbol{e}_i) = \boldsymbol{e}_i + [x_i, y_i, 0, 0]$.

---

[3] This notation denotes element-wise addition between $\boldsymbol{e}_i$ and $[x_i, y_i]$
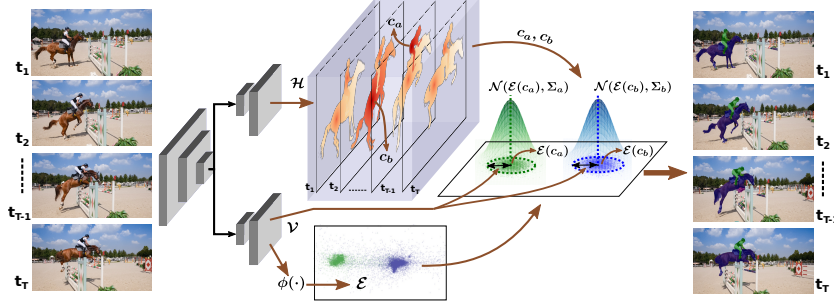
Fig. 2: For an input video clip, our network produces embeddings ($\mathcal{E}$), variances ($\mathcal{V}$), and instance center heat map ($\mathcal{H}$). $\mathcal{H}$ contains one peak per object for the entire spatio-temporal volume (c_a for the rider, c_b for the horse). $\mathcal{E}(c_a)$, $\mathcal{E}(c_b)$ and $\mathcal{V}(c_a)$, $\mathcal{V}(c_b)$ are the corresponding embeddings and variances at c_a and c_b, respectively. These quantities are then used to define the Gaussian distribution for each object.

There is, however, a caveat with free dimensions: since the spatial (and temporal) dimensions already achieve reasonable instance separation, the network may converge to a poor local minimum by producing very large variances for the free dimensions instead of learning a discriminative feature representation. Consequently, the free dimensions may end up offering no useful instance separation during inference. We circumvent this problem at the cost of introducing one extra hyper-parameter by setting the variances for the free dimensions to a fixed value v_{free}. We justify our formulation quantitatively using multiple datasets and different variants of the mixing function $\phi(\cdot)$ in Sec. 4.3.

3.3 Inference

Since the ground truth mask tube is not known during inference, it is not possible to obtain μ_j and Σ_j using Eq. 1. This is where the instance center heat map $\mathcal{H}$ comes into play. For each pixel $c_i = (x_i, y_i, t_i)$, the value $\mathcal{H}(c_i) \in [0, 1]$ in the heat map at this location gives us the probability of the embedding vector $\mathcal{E}(c_i)$ at this location being an instance center. The sequential process of inferring object instances in a video clip is described in the following algorithm:

1. Identify the coordinates of the most likely instance center $c_j = \text{argmax}_i \, \mathcal{H}(c_i)$.
2. Find the corresponding embedding vector $\mathcal{E}(c_j)$ and variances $\mathcal{V}(c_j)$.
3. Using $\mu_j \leftarrow \mathcal{E}(c_j)$ and $\Sigma_j \leftarrow \text{diag}(\mathcal{V}(c_j))$, generate the 3D mask tube $\widehat{\mathcal{C}}_j$ for this instance by computing per-pixel probabilities using Eq. 2, and then thresholding them as in Eq. 3.
4. Since the pixels in $\widehat{\mathcal{C}}_j$ have now been assigned to an instance, the embeddings, variances and heat map probabilities at these pixel locations are masked out and removed from further consideration:

$$\mathcal{E} \leftarrow \mathcal{E} \setminus \widehat{\mathcal{E}}_j \,, \qquad \mathcal{V} \leftarrow \mathcal{V} \setminus \widehat{\mathcal{V}}_j \,, \qquad \mathcal{H} \leftarrow \mathcal{H} \setminus \widehat{\mathcal{H}}_j. \tag{5}$$

5. Repeat steps 1-4 until either $\mathcal{E} = \mathcal{V} = \mathcal{H} = \emptyset$, or the next highest probability in the heat map falls below some threshold.

Even though this final clustering step (Fig. 2) depends on the number of instances in a clip, in practice, the application of Eq. 2 and 3 carries little computational overhead and its run-time is negligible compared to a forward pass.

Video clip stitching: Due to memory constraints, the clip length that can be input to the network is limited. In order to apply our framework to videos of arbitrary length, we split the input video into clips of length T with an overlap of T_c frames between consecutive clips. Linear assignment [43] is then used to associate the predicted tracklets in consecutive clips. The cost metric for this assignment is the IoU between tracks in overlapping frames. Our approach is currently *near online* because, given a new frame, the delay until its output becomes available is at most $T - T_c - 1$.

3.4   Losses

Our model's loss function is a linear combination of three terms:

$$L_{\text{total}} = L_{\text{emb}} + L_{\text{smooth}} + L_{\text{center}} \qquad (6)$$

Embedding loss L_{emb}: As mentioned in Sec. 3.1, we use the Lovàsz hinge loss to optimize the IoU between the predicted and ground truth masks for a given instance. The embedding loss for the entire input clip is calculated as the mean of the Lovàsz hinge loss for all object instances in that clip.

Variance smoothness loss L_{smooth}: To ensure that the variance values at every pixel belonging to an object are consistent, we employ a smoothness loss L_{smooth} similar to [57]. This regresses the variances $\mathcal{V}_j$ for instance j to be close to the average value of all the variances for that instance, *i.e.* $\text{Var}[\mathcal{V}_j]$.

Instance center heat map loss L_{center}: For all pixels belonging to instance j, the corresponding outputs in the sigmoid activated heat map $\mathcal{H}_j$ are trained with an L_2 regression loss to match the output of Eq. 2. The outputs for background pixels are regressed to 0. During inference, this enables us to sample the highest values from the heat map which corresponds to the peak of the learned Gaussian distributions for the object instances in an input volume, as explained in Sec 3.3.

3.5   Network Architecture

The network (Fig. 3) consists of an encoder with two decoders. The first decoder outputs the embeddings $\mathcal{E}$ and variances $\mathcal{V}$, while the second outputs the instance center heat map $\mathcal{H}$. The encoder comprises a backbone with Feature Pyramid Network (FPN) that produces feature maps at 4 different scales for each image in the input clip. The feature maps at each scale are then stacked along the temporal dimension before being input to each of the decoders.

Our decoder consists of 3D convolutions and pooling layers which first compress the feature maps along the temporal dimension before gradually expanding them back to the input size. The underlying idea here is to allow the network to learn temporal context in order to enable pixel association across space and
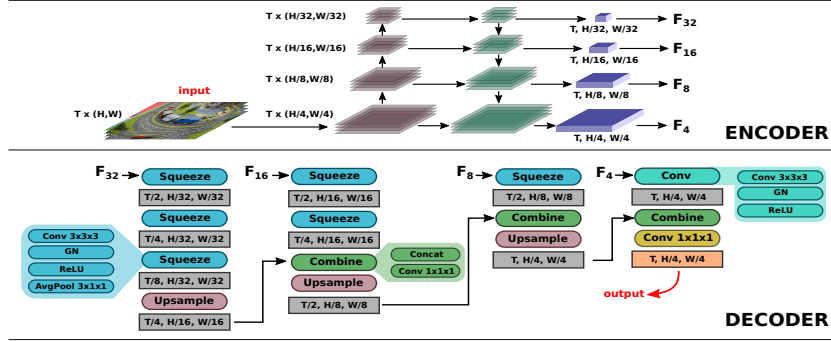
Fig. 3: The network has an encoder-decoder structure. GN: Group Normalization [87].

time. To reduce the decoder's memory footprint and run-time, the large sized feature maps undergo a lower degree of temporal pooling (*i.e.*, fewer convolution/normalization/pooling layers). We call our decoder a *temporal squeeze-expand decoder* (abbreviated as TSE decoder). In Sec. 4.5, we experimentally validate our network's stand-alone ability to learn spatio-temporal context.

3.6   Category Prediction

Our task formulation is inherently category-agnostic. However, some datasets [80, 95] require a category label for each predicted object track. For such cases we add an additional TSE decoder to the network that performs semantic segmentation for all pixels in the input clip, and that is trained using a standard cross-entropy loss. During inference, the logits for all pixels belonging to a given object instance are averaged, and the highest scoring category label is assigned. Note that this is merely a post-processing step; the instance clustering still happens in a category-agnostic manner.

4   Experimental Evaluation

To demonstrate our method's effectiveness and generalization capability, we apply it to three different tasks and datasets involving pixel-precise segmentation and tracking of multiple objects in videos.

4.1   Training

For all experiments, we use a ResNet-101 [33] backbone initialized with weights from Mask R-CNN [32] trained for image instance segmentation on COCO [49]. The temporal squeeze-expand decoders are initialized with random weights. The network is optimized end-to-end using SGD with momentum 0.9 and an initial learning rate of 10^{-3} which decays exponentially.

Augmented Images. Since the amount of publicly available video data with dense per-pixel annotations is limited, we utilize image instance segmentation

datasets by synthesizing training clips from single images. We apply on-the-fly random affine transformations and motion blur and show that this technique is effective (Sec. 4.3) even though such augmented image sequences have little visual resemblance to video clips.

4.2   Benchmarks

DAVIS Unsupervised: DAVIS [11] is a popular video object segmentation dataset with 90 videos (60 for training and 30 for validation) containing multiple moving objects of diverse categories. Several DAVIS benchmarks have been introduced over the years; we evaluate our method on the 2019 *Unsupervised Video Object Segmentation* (UVOS) benchmark in which the salient "foreground objects" in each video have to be segmented and tracked. The evaluation measures employed are (i) $\mathcal{J}$-score (the IoU between predicted and ground truth mask tubes), and (ii) $\mathcal{F}$-score (accuracy of predicted mask boundaries against ground truth). The mean of those measures, $\mathcal{J}\&\mathcal{F}$, serves as the final score.

YouTube-VIS: The YouTube Video Instance Segmentation (YT-VIS) [95] dataset contains 2,883 high quality YouTube videos with ∼131k object instances spanning 40 known categories. The task requires all objects belonging to the known category set to be segmented, tracked and assigned a category label. The evaluation measures used for this task are an adaptation of the Average Precision ($\mathcal{AP}$) and Average Recall ($\mathcal{AR}$) metrics used for image instance segmentation.

KITTI-MOTS: Multi-Object Tracking and Segmentation (MOTS) [80] extends the KITTI multi-object tracking dataset [28] with pixel-precise instance masks. It contains 21 videos (12 for training and 9 for validation) of driving scenes captured from a moving vehicle. The task here is to segment and track all *car* and *pedestrian* instances in the videos. The evaluation measures used are an extension of the CLEAR MOT measures [5] to account for pixel-precise tracking (details in [80]). We report these measures separately for each object class.

4.3   Ablation Studies

Embedding formulation: We first ablate the impact of using different mixing functions for the embeddings (Sec. 3.2) on the DAVIS'19 `val` dataset with clip length $T = 8$, and summarize the results in Tab. 1(a). Compared to the identity function baseline ϕ_{identity}, imparting a spatial coordinate offset (ϕ_{xy}) improves the $\mathcal{J}\&\mathcal{F}$ from 57.3% to 61.6%. Adding another embedding dimension with a temporal coordinate offset, as in ϕ_{xyt}, yields a further improvement to 62.6%.

Comparing the mixing function pairs (ϕ_{xyf}, ϕ_{xyt}) where $E = 3$, and (ϕ_{xyff}, ϕ_{xytf}) where $E = 4$, we note that having a free dimension is slightly better than having a temporal dimension since there is a difference of 0.2% $\mathcal{J}\&\mathcal{F}$ for both pairs of functions. This is the case for both DAVIS'19 and YT-VIS,[4] where ϕ_{xyff} yields the best results. For KITTI-MOTS however,[4] temporal and

---

[4] results for various $\phi(\cdot)$ on YT-VIS and KITTI-MOTS are given in supplementary

Mixing Function	$\mathcal{J}\&\mathcal{F}$	E
ϕ_{identity}	57.3	2
ϕ_{xy}	61.6	2
ϕ_{xyt}	62.6	3
ϕ_{xyf}	62.8	3
ϕ_{xytf}	64.2	4
ϕ_{xyff}	64.4	4
ϕ_{xyfff}	62.4	5

(a)

Clip Length (T)	$\mathcal{J}\&\mathcal{F}$
4	62.2
8	64.4
16	64.7
24	63.1

(b)

(c)

Training Data	$\mathcal{J}\&\mathcal{F}$
Images	57.1
Video	60.7
Images + Video	64.4

(d)

Object Category	$\mathcal{AP}$	$\mathcal{AR}$
Oracle	33.0	34.5
Predicted	24.7	31.8

Table 1: Ablation studies on DAVIS'19 `val`: (a): Impact of different embedding mixing functions; (b): Effect of temporal context; (c): Analysis of training data; (d): Impact of Semantic head on a custom validation split of YT-VIS.

free dimensions yield roughly the same performance. Our intuitive explanation is that for DAVIS and YT-VIS, object instances normally persist throughout the video clip. Therefore, in contrast to the spatial coordinates which serve as a useful feature for instance separation, the temporal coordinate provides no useful separation cue, thus rendering the temporal embedding dimension less effective. On the other hand, using a free dimension enables the network to learn a more discriminative feature representation that can better aid in separating instances. By contrast, objects in KITTI-MOTS driving scenes undergo fast motion and often enter/exit the scene midway through a clip. Thus, the temporal dimension becomes useful for instance separation.

Having additional embedding dimensions is beneficial, but only up to a certain point (*i.e.*, $E = 4$). Beyond that, test-time performance drops, as can be seen by comparing ϕ_{xyff} and ϕ_{xyfff}. We conclude by noting that our proposed formulations for $\phi(\cdot)$ improve performance on video-related tasks compared to existing formulations for single-image tasks [57, ?]. For further discussion and results we refer to the supplementary material.

Temporal Window Size: Next, we investigate the effect of varying the input clip length on DAVIS'19 `val` (ϕ_{xyff} is used throughout). As shown in Tab. 1(b), larger temporal length helps the TSE decoder to predict better quality mask tubes. Increasing the input clip length from $T = 4$ to $T = 16$ improves the $\mathcal{J}\&\mathcal{F}$ from 62.2% to 64.7%, respectively. Above $T = 24$, the performance decreases.

Training Data: For the DAVIS Unsupervised task, we train on image datasets (COCO [49] and Pascal VOC [24]) in conjunction with video datasets (YT-VIS [95] and DAVIS [11]). As shown in Tab. 1(c), this combination yields 64.4% $\mathcal{J}\&\mathcal{F}$ compared to 60.7% when using only video datasets, and 57.1% when using only image datasets. This highlights the benefit of using a combination of image-augmented and video data. For this ablation, $T = 8$ and ϕ_{xyff} were used.

Semantic Head: Since our network does not produce semantic labels for objects, we adapt it to tasks requiring such labels by adding a semantic segmentation decoder as explained in Sec. 3.6. Here, we compare the quality of our semantic output to an *oracle* by training our network on a custom train-validation

DAVIS 2019 Unsupervised										
Method	P/D	OF	RI	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$ Mean	$\mathcal{J}$ Recall	$\mathcal{J}$ Decay	$\mathcal{F}$ Mean	$\mathcal{F}$ Recall	$\mathcal{F}$ Decay	fps
KIS* [15]	✓		✓	59.9	-	-	-	-	-	-	-
UnOVOST* [102]	✓	✓	✓	67.0	65.6	75.5	0.3	68.4	75.9	3.7	< 1
RVOS [78]				41.2	36.8	40.2	0.5	45.7	46.4	1.7	20+
AGNN [84]	✓	✓		61.1	58.9	65.7	11.7	63.2	67.1	14.3	-
OF-Tracker	✓	✓		54.6	53.4	60.9	-1.3	55.9	63	1.1	∼1
RI-Tracker	✓		✓	56.9	55.5	63.3	2.7	58.2	64.4	6.4	< 1
Ours				**64.7**	**61.5**	**70.4**	**-4**	**67.8**	**75.5**	**1.2**	7

Table 2: DAVIS'19 validation results for the unsupervised track. P/D: Proposals/Detections, OF: optical flow, RI: Re-Id, * : uses heuristic post-processing.

split of YT-VIS. This was done because ground truth annotations for the official validation set are not publicly available. The results presented in Tab. 1(d) show that using *oracle* category labels improves $\mathcal{AP}$ performance by 8.3 from 24.7 to 33.0. This suggests that our results on the official validation set could be further improved by using a better semantic classifier. We leave this for future work.

4.4   Comparison with state of the art

Video Object Segmentation: Tab. 2 summarizes our results on the DAVIS'19 unsupervised validation set. OF-Tracker and RI-Tracker are our own optical flow and re-id baselines which use proposals from a Mask-RCNN [32] network trained with the same backbone and data as our method (see supplementary for details). Our method (64.7% $\mathcal{J}\&\mathcal{F}$) outperforms these baselines and the other published methods by a significant margin, even though we use a single, proposal-free network. AGNN [84], with the second best score of 61.1%, uses object proposals from Mask R-CNN [32] on the salient regions detected by their network, and associates them over time using optical flow. We also list the top entries of the DAVIS'19 Challenge Workshop in gray. UnOVOST [102] achieves a higher score (67.0%), but (i) uses several networks along with heuristic-based post-processing, (ii) is an order of magnitude slower (1 vs. 7FPS), and (iii) is highly tailored to this benchmark. To validate this, we adapted UnOVOST to KITTI-MOTS by re-training its networks and optimizing the post-processing parameters with grid search (for further details and analysis of this experiment, we refer to the supplementary). As can be seen in Tab. 4, it does not generalize well to the task of Multi-object Tracking and Segmentation.

Video Instance Segmentation (VIS):   This task requires object instances to be segmented, tracked and also assigned a category label. We therefore adapt our network to this setting as explained in Sec. 3.6. We train jointly on YT-VIS [95] and augmented images from COCO [49] (for COCO, we only use the 20 object classes which overlap with the YT-VIS class set). Since this is a new task with few published works, we compare our method to various baselines and adaptions of existing works from [95] in Tab. 3. As can be seen, our method performs best with respect to all evaluation metrics. Compared to MaskTrack-RCNN [95], we improve the $\mathcal{AP}$ from 30.3 to 34.6, even though MaskTrack-RCNN uses a two stage object detector and incorporates additional cues during

YouTube Video Instance Segmentation								
Method	FF	P/D	OF	$\updownarrow\mathcal{AP}$	$\mathcal{AP}$@50	$\mathcal{AP}$@75	$\mathcal{AR}$@1	$\mathcal{AR}$@10
OSMN MaskProp [96]	✓			23.4	36.5	25.7	28.9	31.1
FEELVOS [79]	✓			26.9	42.0	29.7	29.9	33.4
IoUTracker+ [95]		✓		23.6	39.2	25.5	26.2	30.9
OSMN [96]		✓		27.5	45.1	29.1	28.6	33.1
DeepSORT [85]		✓		26.1	42.9	26.1	27.8	31.3
MaskTrack R-CNN [95]			✓	30.3	51.1	32.6	31.0	35.5
SeqTracker [95]				27.5	45.7	28.7	29.7	32.5
Ours (ResNet-50)				30.6	50.7	33.5	31.6	37.1
Ours				**34.6**	**55.8**	**37.9**	**34.4**	**41.6**

Table 3: YouTube-VIS results. P/D: Proposals/Detections, FF: First Frame Proposals, OF: Optical Flow.

KITTI MOTS											
				Car				Pedestrian			
Method	P/D	OF	RI	sMOTSA	MOTSA	MOTSP	IDS	sMOTSA	MOTSA	MOTSP	IDS
UnOVOST [102]	✓	✓	✓	50.7	60.2	85.6	151	33.4	47.7	76.0	68
MaskRCNN+maskprop [80]	✓	✓		75.1	86.6	87.1	-	45.0	63.5	75.6	-
TrackRCNN [80]	✓		✓	**76.2**	**87.8**	**87.2**	93	46.8	65.1	75.7	78
Ours				72.7	83.8	**87.2**	**76**	**50.4**	**66.1**	**77.7**	**14**

Table 4: KITTI MOTS validation set results for Car and Pedestrian class. P/D: Proposals/Detections, OF: optical flow, RI: Re-Id.

post-processing. Since MaskTrack-RCNN uses a ResNet-50 backbone, we also applied this backbone to our network and still improve the $\mathcal{AP}$ from 30.3 to 30.6.

Multi-Object Tracking and Segmentation (MOTS): Here we again adapt the network for category prediction as required by the task formulation. We outline the results of comparing our method with TrackR-CNN [80] and Un-OVOST [102] in Tab. 4. Current top-performing method on MOTS is the two-stage network that extends Mask-RCNN with a re-id head, trained to learn an appearance embedding vector for each object detection, used for data association. Our method achieves the highest sMOTSA score (50.4) on the *pedestrian* class, but TrackR-CNN performs better on the *car* class. However, for both classes, STEm-Seg suffers significantly fewer ID switches (IDS) compared to TrackR-CNN (76 vs. 93 for the *car* class and 14 vs. 78 for the *pedestrian* class). This measure is of particular interest to the tracking community since it directly reflects temporal association accuracy.

Similar to UnOVOST [102], TrackR-CNN does not generalize well to other tasks. To validate this, we retrained TrackR-CNN on the YT-VIS dataset. However, the resulting $\mathcal{AP}$ was less than 10. We assume this is due to TrackR-CNN that is forced to use a 14×14 ROI-Align layer [32] due to memory constraints. This results in coarse segmentation masks which are heavily penalized by the $\mathcal{AP}$ measure. Furthermore, the ReID-based embeddings can only learn an ap-

pearance model, which is a limitation in YT-VIS where similar looking objects often occur in the same scene.

4.5   Segmentation of Salient Regions in Videos

Finally, we apply our method to the DAVIS'16 unsupervised benchmark [71], where the task is to produce a binary segmentation for the salient regions in a given video clip. Since separating object instances is not required, we simplify our network to one decoder with two output channels trained for binary segmentation using a bootstrapped cross-entropy loss [88] on randomly selected clips from the YT-VIS and DAVIS datasets. Although competing methods are specifically engineered for this task, our simple setup obtains state-of-the-art results (Tab. 5).

DAVIS 2016 Unsupervised					
Method	OF	CRF	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$-mean	$\mathcal{F}$-mean
LVO[75]	✓	✓	-	75.9	72.1
PDB[75]		✓	-	77.2	74.5
MotAdapt[74]			-	77.2	77.4
3D-CNN[35]			-	78.3	77.2
FusionSeg[39]	✓		-	70.7	65.3
LVO[77]	✓	✓	-	75.9	72.1
ARP[98]	✓		-	76.2	70.6
PDB[75]	✓		-	77.2	74.5
AD-Net[97]			78.8	79.4	78.2
AGNN[84]		✓	79.9	**80.7**	79.1
Ours			**80.6**	80.6	**80.6**

Table 5: Results on DAVIS'16 val. for the unsupervised track. OF: optical flow, CRF: post-processing using CRF.

We note that while AGNN [84] and AD-Net [97] use a stronger DeepLabv3 [12] backbone, we use additional video data for training. This is needed as we work with 3D input volumes. Since we do not perform post-processing, we report results from AD-Net [97] without their DAVIS-specific post-processing for a fair comparison.

5   Conclusion

We have proposed a novel *bottom-up* approach for instance segmentation in videos which models video clips as 3D space-time volumes and then separates object instances by clustering learned embeddings. We enhanced the feature representation of these embeddings using novel mixing functions which yield considerable performance improvements over existing formulations. We applied our method to multiple, diverse datasets and achieved state-of-the-art results under both category-aware and category-agnostic settings. We further showed that, compared to existing dataset-specific state-of-the-arts, our approach generalizes much better across different datasets. Finally, we validated our network's temporal context learning ability by performing a separate experiment on video saliency detection and showed that our good results also generalize there.

References

1. Anthony Hu, Alex Kendall, R.C.: Learning a spatio-temporal embedding for video instance segmentation. arxiv preprint arXiv:1912:08969v (2019)
2. Van den Bergh, M., Roig, G., Boix, X., Manen, S., Van Gool, L.: Online video seeds for temporal window objectness. In: ICCV (2013)
3. Berman, M., Blaschko, M.B.: Optimization of the jaccard index for image segmentation with the lovász hinge. CVPR (2018)
4. Berman, M., Rannen Triki, A., Blaschko, M.B.: The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: CVPR (2018)
5. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The clear mot metrics. JIVP **2008**, 1:1–1:10 (2008)
6. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: AVSS (2017)
7. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV (2010)
8. Butt, A.A., Collins, R.T.: Multi-target tracking by lagrangian relaxation to min-cost network flow. In: CVPR (June 2013)
9. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017)
10. Caelles, S., Montes, A., Maninis, K.K., Chen, Y., Van Gool, L., Perazzi, F., Pont-Tuset, J.: The 2018 davis challenge on video object segmentation. arXiv preprint arXiv:1803.00557 (2018)
11. Caelles, S., Pont-Tuset, J., Perazzi, F., Montes, A., Maninis, K., Gool, L.V.: The 2019 DAVIS challenge on VOS: unsupervised multi-object segmentation. arXiv arXiv:1905.00737 (2019)
12. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
13. Chen, X., Girshick, R., He, K., Dollár, P.: Tensormask: A foundation for dense object segmentation. In: ICCV (2019)
14. Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR (2018)
15. Cho, D., Hong, S., amd J. Kim, S.K.: Key instance selection for unsupervised video object segmentation. The 2019 DAVIS Challenge on Video Object Segmentation - CVPR Workshops (2019)
16. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI **24**(5), 603–619 (2002)
17. Dave, A., Tokmakov, P., Ramanan, D.: Towards segmenting everything that moves. arXiv preprint arXiv:1902.03715 (2019)
18. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation for autonomous driving. In: CVPR Workshops (2017)
19. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation with a discriminative loss function. arXiv preprint arXiv:1708.02551 (2017)
20. Dong, M., Wang, J., Huang, Y., Yu, D., Su, K., Zhou, K., Shao, J., Wen, S., Wang, C.: Temporal feature augmented network for video instance segmentation. In: ICCV Workshops (2019)
21. Elich, C., Engelmann, F., Schult, J., Kontogianni, T., Leibe, B.: 3D-BEVIS: Birds-Eye-View Instance Segmentation. In: German Conference on Pattern Recognition (GCPR) (2019)

22. Engelmann, F., Bokeloh, M., Fathi, A., Leibe, B., Nießner, M.: 3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
23. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: ACM Conference on Knowledge Discovery and Data Mining (KDD) (1996)
24. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. IJCV **88**(2), 303–338 (2010)
25. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: ICCV (2017)
26. Feng, Q., Yang, Z., Li, P., Wei, Y., Yang, Y.: Dual embedding learning for video instance segmentation. In: ICCV Workshops (2019)
27. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on information theory **21**(1), 32–40 (1975)
28. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR (2012)
29. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR (2015)
30. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: IJCNN (2005)
31. Han, W., Khorrami, P., Paine, T.L., Ramachandran, P., Babaeizadeh, M., Shi, H., Li, J., Yan, S., Huang, T.S.: Seq-nms for video object detection. arXiv preprint arXiv:1602.08465 (2016)
32. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
34. Hou, R., Chen, C., Shah, M.: Tube convolutional neural network (t-cnn) for action detection in videos. In: ICCV (2017)
35. Hou, R., Chen, C., Sukthankar, R., Shah, M.: An efficient 3d CNN for action/object segmentation in video. In: BMVC (2019)
36. Hu, Y., Huang, J., Schwing, A.: Maskrnn: Instance level video object segmentation. In: NIPS (2017)
37. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: ECCV (2008)
38. Jain, R., Nagel, H.H.: On the analysis of accumulative difference pictures from image sequences of real world scenes. PAMI **1**, 206 – 214 (1979)
39. Jain, S., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In: CVPR (2017)
40. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: CVPR (2020)
41. Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., Wang, X.: Object detection in videos with tubelet proposal networks. In: CVPR (2017)
42. Kong, S., Fowlkes, C.C.: Recurrent pixel embedding for instance grouping. In: CVPR (2018)
43. Kuhn, H.W., Yaw, B.: The hungarian method for the assignment problem. Naval Res. Logist. Quart pp. 83–97 (1955)
44. Kwak, S., Cho, M., Laptev, I., Ponce, J., Schmid, C.: Unsupervised object discovery and tracking in video collections. In: ICCV (2015)

45. Leal-Taixé, L., Milan, A., Reid, I., , Roth, S., Schindler, K.: MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
46. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV **77**(1-3), 259–289 (2008)
47. Leibe, B., Schindler, K., Cornelis, N., Gool, L.V.: Coupled object detection and tracking from static cameras and moving vehicles. PAMI **30**(10), 1683–1698 (2008)
48. Li, S., Seybold, B., Vorobyov, A., Fathi, A., Huang, Q., Kuo, C.C.J.: Instance embedding transfer to unsupervised video object segmentation. In: CVPR (2018)
49. 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)
50. Liu, R., Lehman, J., Molino, P., Such, F.P., Frank, E., Sergeev, A., Yosinski, J.: An intriguing failing of convolutional neural networks and the coordconv solution. In: NIPS (2018)
51. Liu, X., Ye, T.: Spatio-temporal attention network for video instance segmentation. In: ICCV Workshops (2019)
52. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
53. Luiten, J., Voigtlaender, P., Leibe, B.: Premvos: Proposal-generation, refinement and merging for video object segmentation. In: Asian Conference on Computer Vision (2018)
54. McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. Journal of Open Source Software **2**(11), 205 (2017)
55. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
56. Milan, A., Leal-Taixé, L., Schindler, K., Reid, I.: Joint tracking and segmentation of multiple targets. In: CVPR (2015)
57. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: CVPR (2019)
58. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: NIPS (2017)
59. Novotny, D., Albanie, S., Larlus, D., Vedaldi, A.: Semi-convolutional operators for instance segmentation. In: ECCV (2018)
60. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: CVPR (2012)
61. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)
62. Okuma, K., Taleghani, A., De Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: ECCV (2004)
63. Ošep, A., Mehner, W., Voigtlaender, P., Leibe, B.: Track, then decide: Category-agnostic vision-based multi-object tracking. ICRA (2018)
64. Ošep, A., Voigtlaender, P., Luiten, J., Breuers, S., Leibe, B.: Large-scale object mining for object discovery from unlabeled video (2019)
65. Ošep, A., Voigtlaender, P., Weber, M., Luiten, J., Leibe, B.: 4d generic video object proposals. In: ICRA (2020)
66. Palmer, S.E.: Organizing objects and scenes. Foundations of cognitive psychology: Core readings pp. 189–211 (2002)
67. Palou, G., Salembier, P.: Hierarchical video representation with trajectory binary partition tree. In: CVPR (2013)

68. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. PAMI **22**, 266–280 (2000)
69. Pinheiro, P., Lin, T., Collobert, R., Dollár, P.: Learning to refine object segments. In: ECCV (2016)
70. Pinheiro, P., Collobert, R., Dollr, P.: Learning to segment object candidates. In: NIPS (2015)
71. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbelez, P., Sorkine-Hornung, A., Gool, L.V.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
72. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: CVPR (2019)
73. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
74. Siam, M., Jiang, C., Lu, S.W., Petrich, L., Gamal, M., Elhoseiny, M., Jägersand, M.: Video segmentation using teacher-student adaptation in a human robot interaction (HRI) setting. In: ICRA (2018)
75. Song, H., Wang, W., Zhao, S., Shen, J., Lam, K.M.: Pyramid dilated deeper convlstm for video salient object detection. In: ECCV (September 2018)
76. Teichman, A., Levinson, J., Thrun, S.: Towards 3D object recognition via classification of arbitrary object tracks. In: ICRA (2011)
77. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: ICCV (2017)
78. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marqués, F., Gir'o i Nieto, X.: RVOS: end-to-end recurrent network for video object segmentation. CVPR (2019)
79. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen., L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: CVPR (2019)
80. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B., Geiger, A., Leibe, B.: MOTS: Multi-object tracking and segmentation. In: CVPR (2019)
81. Wang, H., Luo, R., Maire, M., Shakhnarovich, G.: Pixel consensus voting for panoptic segmentation. In: CVPR (2020)
82. Wang, L., Hua, G., Sukthankar, R., Xue, J., Zheng, N.: Video object discovery and co-segmentation with extremely weak supervision. In: ECCV (2014)
83. Wang, Q., He, Y., Yang, X., Yang, Z., Torr, P.: An empirical study of detection-based video instance segmentation. In: In ICCV Workshops (2019)
84. Wang, W., Lu, X., Shen, J., Crandall, D.J., Shao, L.: Zero-shot video object segmentation via attentive graph neural networks. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
85. Wojke, N., Bewley, A., Paulus., D.: Onboard contextual classification of 3D point clouds with learned high-order markov random fields. In: ICIP (2017)
86. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. PAMI **19**, 780–785 (1997)
87. Wu, Y., He, K.: Group normalization. In: ECCV (2018)
88. Wu, Z., Shen, C., Hengel, A.v.d.: Wider or deeper: Revisiting the resnet model for visual recognition. arXiv preprint arXiv:1611.10080 (2016)
89. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018)
90. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018)

91. Xiao, F., Jae Lee, Y.: Track and segment: An iterative unsupervised approach for video object proposals. In: CVPR (2016)
92. Xie, C., Xiang, Y., Harchaoui, Z., Fox, D.: Object discovery in videos as foreground motion clustering. In: CVPR (2019)
93. Xu, C.: Evaluation of super-voxel methods for early video processing. In: CVPR (2012)
94. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: YouTube-VOS: Sequence-to-sequence video object segmentation. In: ECCV (2018)
95. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
96. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: CVPR (2018)
97. Yang, Z., Wang, Q., Bertinetto, L., Hu, W., Bai, S., Torr, P.H.S.: Anchor diffusion for unsupervised video object segmentation. In: ICCV (2019)
98. Yeong Jun Koh, C.S.K.: Primary object segmentation in videos based on region augmentation and reduction. In: CVPR (2017)
99. Yu, J., Blaschko, M.: Learning submodular losses with the lovász hinge. In: International Conference on Machine Learning (ICML) (2015)
100. Zeng, X., Liao, R., Gu, L., Xiong, Y., Fidler, S., Urtasun, R.: Dmm-net: Differentiable mask-matching network for video object segmentation. In: ICCV (2019)
101. Zhang, D., Chun, J., Cha, S.K., Kim, Y.M.: Spatial semantic embedding network: Fast 3d instance segmentation with deep metric learning. arXiv preprint arXiv:2007.03169 (2020)
102. Zulfikar, I.E., Luiten, J., Leibe, B.: UnOVOST: Unsupervised Offline Video Object Segmentation and Tracking for the 2019 Unsupervised DAVIS Challenge. The 2019 DAVIS Challenge on Video Object Segmentation - CVPR Workshops (2019)