A Single-Stage, Bottom-up Approach for Occluded VIS using Spatio-temporal Embeddings

Ali Athar*
RWTH Aachen University
Aachen, Germany
athar@vision.rwth-aachen.de

Sabarinath Mahadevan*
RWTH Aachen University
Aachen, Germany
mahadevan@vision.rwth-aachen.de

Aljoša Ošep
Technical University of Munich
Munich, Germany
aljosa.osep@tum.de

Laura Leal-Taixé
Technical University of Munich
Munich, Germany
leal-taixe@tum.de

Bastian Leibe
RWTH Aachen University
Aachen, Germany
leibe@vision.rwth-aachen.de

Abstract

*The task of Video Instance Segmentation (VIS) involves segmenting, tracking and classifying all object instances present in a given video clip. Occluded VIS is a more challenging extension of this task which involves longer video sequences where objects undergo significant occlusions over time. Most existing approaches to VIS involve multiple networks which separately handle segmenting, tracking and classifying object instances, and potentially a set of heuristics to combine the individual network outputs. By contrast, we employ just one, single-stage network without any heuristics or post-processing for the end-to-end task. Our approach is called 'STEm-Seg', which is a bottom-up method for **Seg**menting object instances in videos using **S**patio-**T**emporal **Em**beddings. We achieve 3rd place in the Occluded VIS challenge with an mAP score of 21.6% on the test set.*

1. Introduction and Related Work

Occluded Video Instance Segmentation (OVIS) [18] is a newly proposed dataset and benchmark which involves pixel-precise segmentation, classification and tracking of all object instances in a given video clip that belong to a predefined set of 25 classes. This set of object classes consists of humans and various types of vehicles and animals. The primary evaluation metric is mean Average-Precision (mAP) which is first computed separately for each object class and subsequently averaged. The OVIS task is conceptually similar to three other existing datasets/benchmarks which we describe below along with some popular approaches for tackling them.

Video Instance Segmentation (YouTube-VIS). [22] The OVIS task is essentially a more challenging variant of this task since it generally comprises lengthier videos with more occlusions. Another practical difference is that the VIS dataset encompasses 40 object classes which also contain several sports related things (e.g. `skateboard` and `tennis-racket`). As a baseline method to tackle this task, the dataset authors proposed Mask-TrackRCNN [22] which is an extension of the popular image instance segmentation network Mask-RCNN [10] to video. Specifically, an additional embedding branch is added which produces an embedding vector for every image-level object detection. These embeddings are then used to associate per-frame detections over time using Hungarian matching. Luiten *et al*. [13] proposed a multi-stage approach which first uses an ImageNet trained classified and Mask-RCNN networks for classifying and segmenting instances per-image, respectively, followed by temporal association

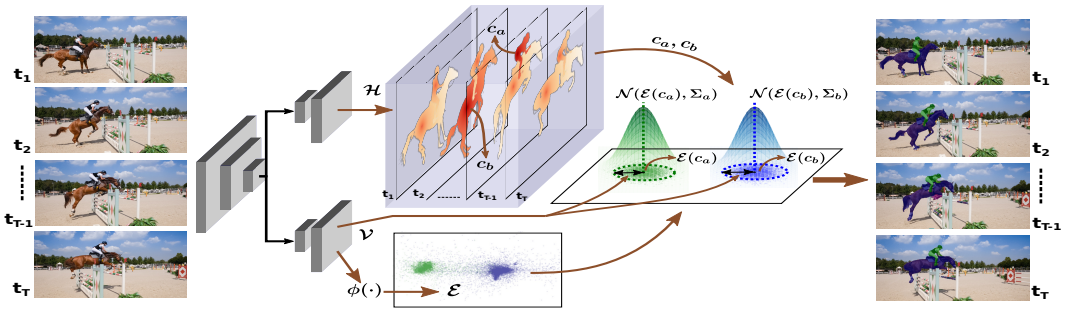

Figure 1: Overview of STEm-Seg [1]. Given an input video clip, STEm-Seg uses 3D decoders to generate embeddings ($\mathcal{E}$), variances ($\mathcal{V}$), instance center heat map ($\mathcal{H}$) and semantic segmentation prediction (not shown here). $\mathcal{E}$ and $\mathcal{V}$ are then used to define the Gaussian distribution for each object. $\mathcal{H}$ encodes a spatio-temporal centre for each object in the input volume, and is used during inference to sample the corresponding distribution parameters.

based on optical flow and ReID. Their approach won first place in the YouTue-VIS 2019 challenge, but is complicated since it involves multiple networks and heuristics. Bertasius *et al*. [5] proposed a single-network solution which uses a Mask-RCNN based network with a novel *Maskprop* module which comprises deformable convolutions [8] and attention to propagate (or, in other words, associate) instances over time.

Unsupervised Video Object Segmentation (DAVIS). [6] This task differs from OVIS in that it does not require classification of object instances. Rather, the set of object classes is not known a-priori and the task if to segment and track all 'salient' foreground objects. Here, saliency is subjectively defined as those objects which capture the attention of the human eye. The main evaluation metrics are the Jaccard Index and F1-score, which are averaged into a single "J&F" metric. The current state-of-the-art solution for this task, UnOVOST [14], also uses separate networks and heuristics for per-instance segmentation and temporal association.

Multi-object Tracking and Segmentation (MOTS). [20] This task is based on existing Multi-Object Tracking (MOT) benchmarks [16, 9] which require object instances in a video to be given bounding boxes which are tracked over time. The MOTS dataset extends this by instead requiring pixel-precise object labels (similar to OVIS). But whereas OVIS (and VIS) focus on generic videos (obtained from YouTube), MOTS focuses on autonomous driving scenarios since it comprises videos of street scenes and only requires objects belonging to the `person` and `car` to be segmented and tracked. The primary evaluation metric for this task was initially sMOTSA, an extension of the CLEAR MOT metrics [4] for pixel-precise labels, but it was later changed to the recently proposed HOTA metrics [12]. The dataset authors proposed a baseline method called TrackRCNN [20] which is conceptually very similar to Mask-TrackRCNN in that it adds an embedding output to Mask-RCNN for performing temporal association.

In general, we can see that existing methods for segmenting object instances in videos follow the well-known tracking-by-detection paradigm, where objects are first segmented in each image frame individually, followed by a second temporal association step. To this end, methods such as TrackRCNN [20], Mask-TrackRCNN [22] and the one proposed by Bertasius *et al*. [5] comprise two-stage networks that are extensions of Mask-RCNN [10], whereas UnOVOST [14] and the winning approach for the YouTube-VIS 2019 challenge [13] use entirely separate networks and heuristics for per-image segmentation and temporal association.

STEm-Seg. In an earlier work [1], we proposed STEm-Seg - a single-stage, end-to-end network that segments and tracks objects across videos in a single step. Instead of considering videos to be a sequence of individual images, we treat them as a single 3D (spatio-temporal) volume and learn per-pixel embeddings that can then be used to associate pixels over space and time in a single step. Furthermore, in contrast to typical clustering-based approaches, we formulate the problem in a way that offloads the prediction of clustering parameters onto the network, thus eliminating the need for slow clustering algorithms (e.g. Mean-Shift Clustering, HDBScan [15], etc.) during inference.

In our earlier work [1], we already applied STEm-Seg to the three above mentioned benchmarks (*i.e*. DAVIS Unsupervised [6], YouTube-VIS [22] and KITTI-MOTS [20]) and showed that it achieved state-of-the-art performance. In this work, we applied STEm-Seg to OVIS [18] and show that it achieves an mAP of 21.6% on the test set, putting it in 3rd place in the leaderboard.

2. Method

As already mentioned, we use the STEm-Seg [1] architecture to solve the task of Occluded Video Instance Seg-

mentation (OVIS) [18]. STEm-Seg takes a video clip of T frames as input, and generates a set of clustering parameters, which can then be used to perform clustering in the pixel space within a spatio-temporal volume. The output of STEm-Seg is hence a set of temporally coherent instance segmentation masks.

Fig. 1 shows the overview of the STEm-Seg pipeline. Given an input clip of size $T \times H \times W \times 3$, where T is the number of frames in the clip, $H \times W$ is the frame resolutions and 3 represents the RGB channels, STEm-Seg generates a set of K instance tubes by clustering pixels across space and time. The clustering is performed based on a learned embedding function and is defined in such a way that no external clustering algorithm or post-processing is necessary to obtain the segmentation masks. In this regard, STEm-Seg predicts for each of the N pixels in the clip (*i.e.* $N = T \times H \times W$) an E-dimensional embedding vector $\mathcal{E} \in \mathbb{R}^{N \times E}$, an associated variance value $\mathcal{V} \in \mathbb{R}_{+}^{N \times E}$, and an object centre heat map $\mathcal{H} \in [0, 1]^{N}$.

Instance Representation: Every object instance j is modelled by a multi-variate Gaussian distribution $\mathcal{N}(\vec{\mu}_j, \boldsymbol{\Sigma}_j)$ with mean $\vec{\mu}_j$ and variance $\boldsymbol{\Sigma}_j$. Given the ground truth mask $\mathcal{C}_j$ for instance j, the mean $\vec{\mu}_j$ and variance $\boldsymbol{\Sigma_j}$ can be computed by averaging the network outputs at each pixel:

$$\vec{\mu}_j = \frac{1}{N_j} \sum_{\vec{e} \in \mathcal{E}_j} \vec{e} \in \mathbb{R}^E,$$

$$\boldsymbol{\Sigma}_j = \frac{1}{N_j} \text{diag} \left(\sum_{\vec{v} \in \mathcal{V}_j} \vec{v} \right) \in \mathbb{R}^{E \times E} .$$

Here N_j is the number of pixels that belong to instance mask $\mathcal{C}_j$. Since the distribution $\mathcal{N}(\vec{\mu}_j, \boldsymbol{\Sigma}_j)$ represents all the pixels that belong to instance j across the input video clip, it can be directly used to compute the probability p_{ij} of each embedding $e_i \in \mathcal{E}$ belonging to instance j:

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

Using Eq. 1, we can compute all the pixels in the input volume that belongs to object instance j by simply thresholding the probability map at 0.5.

Embedding Representation: Generally, the embeddings that are learnt by a network can take arbitrary representations. In STEm-Seg however, we fix this representation by using a so-called *spatio-temporal coordinate mixing function*, where the embeddings $\mathcal{E}$ are modified using a mixing function $\phi : \mathbb{R}^E \rightarrow \mathbb{R}^E$. More formally, $\mathcal{E} \leftarrow \{ \phi(\vec{e}), \mid \vec{e} \in \mathcal{E} \}$. The baseline version of our network generates 2D embeddings ($E = 2$); thus ϕ enhances the embedding representation by adding the spatial coordinates: $\phi_{\text{xy}}(\vec{e}_i) = \vec{e}_i + [x_i, y_i]$.

In addition, we also add two extra dimensions to the output embeddings that are left to learn arbitrary representations, which gives the network additional degrees of freedom. This is useful to generate better clustering results in most of the cases. Hence, with these additional free dimensions, the final mixing function which we use for our OVIS experiments can be written as: $\phi_{\text{xyff}}(\vec{e}_i) = \vec{e}_i + [x_i, y_i, 0, 0]$, where $E = 4$. For these extra dimensions, we fix the variances to a constant value v_{free} to ensure that the network does not predict very large variances.

Network Architecture: The STEm-Seg network uses an encoder to learn feature representations from the input clip, and two decoders - the first produces the embeddings $\mathcal{E}$, variances $\mathcal{V}$, and instance centre heat map $\mathcal{H}$, while the second one produces per-pixel class predictions. The encoder comprises a wide ResNeXt-101 [21] backbone with Feature Pyramid Network (FPN). This is a stronger backbone compared to the one used in our first work [1] paper, and helps in learning richer features that are necessary to represent the challenging scenarios present in OVIS. Each decoder consists of 3D convolutions and pooling, and learns the temporal context by first squashing the temporal dimension to a smaller size, and then gradually expanding it back to the original resolutions. Skip-connections are used at regular intervals and trilinear up-sampling is used to incorporate high-level features from the encoder. We refer to this novel decoder as a *Temporal Squeeze-Expand (TSE)* decoder.

Augmented Image Sequences: Since the number of densely annotated video sequences is limited, we diversify the training set by adding fake video clips generated by applying random affine and perspective transforms to image instance segmentation datasets. For the model which we use to evaluate on the OVIS test, 65% of the training clips were generated from the OVIS training set, and 35% were generated by randomly augmenting annotated images from the COCO dataset [11].

Training: The training objective of STEm-Seg is to optimise the probability distribution for each of the object instances in the input video clip. This is done by regressing the probability heat map defined in Eq. 1 using a Lovàsz hinge loss [2, 3, 17, 23], which is a differentiable, convex surrogate of the Jaccard index and it directly optimizes the IoU between the prediction and the ground-truth masks. The network is trained end-to-end by optimizing the following loss function:

$$L_{\text{total}} = L_{\text{lov}} + L_{\text{smooth}} + L_{\text{center}} + L_{\text{sem}} \tag{2}$$

Here, L_{lov} is the Lovàsz hinge loss that was mentioned above, L_{smooth} is a variance smoothness loss that ensures that the variance of pixels belonging to the same object are

Method	Validation Set					Test Set				
	AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
FEELVOS [19]	9.6	22.0	7.3	7.4	14.8	10.8	23.4	8.7	9.0	16.2
IoUTracker+ [22]	7.3	17.9	5.5	6.1	15.1	9.5	18.8	10.0	6.6	16.5
MaskTrack R-CNN [22]	10.8	25.3	8.5	7.9	14.9	11.8	25.4	10.4	7.9	16.0
SipMask [7]	10.2	24.7	7.8	7.9	15.8	11.7	23.7	10.5	8.1	16.6
CSipMask [18]	14.3	29.9	12.5	9.6	19.3	14.5	31.1	13.5	9.0	19.4
CMaskTrack-RCNN [18]	15.4	33.9	13.1	9.3	20.0	15.1	31.6	13.2	9.8	20.5
STEm-Seg	**21.3**	**43.9**	**18.8**	**13.3**	**28.5**	**21.6**	**39.8**	**20.2**	**12.6**	**27.4**

Table 1: Results for various methods on the OVIS [18] validation and test sets.

consistent, L_{center} is the instance centre heat map loss, and L_{sem} is the semantic loss. L_{smooth} regresses the variances $\mathcal{V}_j$ for instance j to be close to the mean variances of that instance, L_{center} regresses the instance centre heat map $\mathcal{H}_j$ to the corresponding probabilities obtained using Eq. 1 for instance j and L_{sem} is a cross-entropy loss which regresses the predicted class labels against the ground-truth.

We optimize the network using a batch size of 8 clips using SGD with 0.9 momentum and an initial learning rate of 10^{-3} for 60k iterations. After this, we exponentially decay the learning rate to 10^{-5} over another 60k iterations. The training is carried out on 4x Nvidia V100 GPUs.

Inference: Since the ground truth map is not available during inference, the means $\vec{\mu}$ and variances $\boldsymbol{\Sigma}$ have to be sampled using the instance centres that are available from the instance centre heat map. The overall inference process remains the same as described in the STEm-Seg [1] paper, which is referenced below.

1. Identify the coordinates of instance centre $\vec{c}_j = \text{argmax}_i\ \mathcal{H}(\vec{c}_i)$.
2. Find the corresponding embedding vector $\mathcal{E}(\vec{c}_j)$ and variances $\mathcal{V}(\vec{c}_j)$.
3. Using $\vec{\mu}_j \leftarrow \mathcal{E}(\vec{c}_j)$ and $\boldsymbol{\Sigma}_j \leftarrow \text{diag}\,(\mathcal{V}(\vec{c}_j))$, generate the 3D mask tube $\widehat{\mathcal{C}}_j$ for this instance by computing per-pixel probabilities using 1, and then thresholding them..
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:

$$\mathcal{E} \leftarrow \mathcal{E} \backslash \widehat{\mathcal{E}}_j\,, \qquad \mathcal{V} \leftarrow \mathcal{V} \backslash \widehat{\mathcal{V}}_j\,, \qquad \mathcal{H} \leftarrow \mathcal{H} \backslash \widehat{\mathcal{H}}_j. \quad (3)$$

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.

Clip stitching: Since large videos cannot fit into GPU memory at once, we divide every video into overlapping sub-clips of length T. Hungarian matching is then performed on the overlapping frames, with the cost metric being the IoU between tracklets, to associate the tracklets for the entire video.

3. Experimental Results

The results on the OVIS validation and test set for STEm-Seg and various other methods are given in Table 1. The results for other methods are identical to those reported by Qi *et al*. [18].

It can be seen that STEm-Seg outperforms all existing baselines by a significant margin. Specifically, we outperform the second-best performing method (CMaskTrack-RCNN [18]) by an absolute margin of 5.4% on the validation set and 6.5% on the test set. The end-to-end runtime for our approach on the test set is $\sim$ 7.5 frames per second on a single Nvidia RTX3090 GPU.

4. Conclusion

We have successfully used the end-to-end trainable *bottom-up* approach STEm-Seg to generate temporally consistent instance segmentation masks on the Occluded Video Instance Segmentation (OVIS) dataset and have attained the third best performance in the corresponding benchmark. This result shows the generalization capability of STEm-Seg, and its efficacy on heavily occluded scenes.

References

[1] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020.

[2] Maxim Berman and Matthew B. Blaschko. Optimization of the jaccard index for image segmentation with the lovász hinge. *CVPR*, 2018.

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018.

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

[5] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020.

[6] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv*, 2019.

[7] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020.

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11), 2013.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[11] 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 *ECCV*, 2014.

[12] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 129(2), 2021.

[13] Jonathon Luiten, Philip Torr, and Bastian Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In *ICCV Workshops*, 2019.

[14] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *WACV*, 2020.

[15] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11), 2017.

[16] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv*, 2016.

[17] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, 2019.

[18] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation. *arXiv*, 2021.

[19] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[20] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[21] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CVPR*, pages 5987–5995, 2017.

[22] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *CVPR*, 2019.

[23] Jiaqian Yu and Matthew Blaschko. Learning submodular losses with the lovász hinge. In *International Conference on Machine Learning (ICML)*, 2015.