Making a Case for 3D Convolutions for Object Segmentation in Videos

Sabarinath Mahadevan*[1]
mahadevan@vision.rwth-aachen.de
Ali Athar*[1]
athar@vision.rwth-aachen.de
Aljoša Ošep[2]
aljosa.osep@tum.de
Sebastian Hennen[1]
sebastian.hennen@rwth-aachen.de
Laura Leal-Taixé[2]
leal-taixe@tum.de
Bastian Leibe[1]
leibe@vision.rwth-aachen.de

[1] RWTH Aachen University
Aachen, Germany

[2] Technical University of Munich
Munich, Germany

Abstract

The task of object segmentation in videos is usually accomplished by processing appearance and motion information separately using standard 2D convolutional networks, followed by a learned fusion of the two sources of information. On the other hand, 3D convolutional networks have been successfully applied for video classification tasks, but have not been leveraged as effectively to problems involving dense per-pixel interpretation of videos compared to their 2D convolutional counterparts and lag behind the aforementioned networks in terms of performance. In this work, we show that 3D CNNs can be effectively applied to dense video prediction tasks such as salient object segmentation. We propose a simple yet effective encoder-decoder network architecture consisting entirely of 3D convolutions that can be trained end-to-end using a standard cross-entropy loss. To this end, we leverage an efficient 3D encoder, and propose a 3D decoder architecture, that comprises novel 3D Global Convolution layers and 3D Refinement modules. Our approach outperforms existing state-of-the-arts by a large margin on the DAVIS'16 Unsupervised, FBMS and ViSal dataset benchmarks in addition to being faster, thus showing that our architecture can efficiently learn expressive spatio-temporal features and produce high quality video segmentation masks. We have made our code and trained models publicly available at: https://github.com/sabarim/3DC-Seg

1 Introduction

For a given video clip, the task of segmenting salient objects involves generating binary masks for each frame in that clip, such that all pixels belonging to objects that exhibit dominant or salient motion are labeled as foreground. This is challenging in part because the set

© 2020. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

* Equal contribution

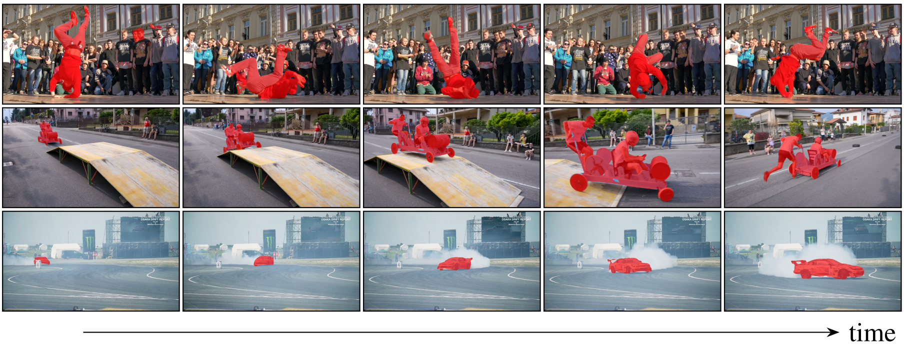

Figure 1: Qualitative results of salient object segmentation masks produced by our network.

of object classes that need to be segmented is not defined a-priori. Therefore, the notion of dominant motion can only be learned by identifying the salient regions based on appearance and motion cues, and capturing the spatial extent and temporal evolution of objects with pixel-level precision over the entire video clip. This is a core task that is directly related to current problems in computer vision and robotics such video object segmentation [31, 32] and object discovery in videos [21, 28, 44, 52].

Some existing methods for segmenting salient objects in videos (*e.g.*, [16, 36]) follow the approach by [34] and first separately process the appearance and motion information before performing a learned fusion of the two streams of information. On the other hand, for the task of video action classification, several methods [10, 12, 38, 39] model videos as 3D volumes and utilize 3D convolutional networks to jointly learn spatial and temporal features and we believe this is a step in the right direction. Applying 3D CNNs for pixel-precise segmentation tasks, however, introduces several challenges. Firstly, these networks are generally slower, and contain significantly more trainable parameters compared to their 2D counterparts of the same architecture and depth. This is especially problematic for segmentation tasks which require higher image resolutions than those used for classification tasks. Secondly, for segmentation tasks it is important (*c.f.*, [4]) to have a network architecture that can capture a large receptive field with respect to each image pixel and effectively leverage multi-scale feature information.

To the best of our knowledge, the work by Hou *et al.* [15] was the first attempt to apply a fully 3D CNN for video object segmentation. While promising, it is out-performed by state-of-the-art methods that use 2D CNNs [47, 57]. This is mainly due to two reasons. Firstly, to keep the computational load manageable, they employ a shallow 3D ResNet-34 [38] as the encoder network. Secondly, they follow the commonly employed design choice of using a smaller stride in the backbone to preserve feature localization, and additionally use atrous convolutions to maintain a large receptive field [4, 5]. As a result, their approach propagates large feature maps through the full depth of the network, which in turn significantly increases the memory footprint and run-time during both inference and training.

In this paper, we propose a network architecture that mitigates the aforementioned issues and outperforms both [15] as well as existing state-of-the-art methods based on 2D CNNs [16, 36, 47, 57]. We argue that a better approach for applying 3D CNNs to such tasks involves using a lightweight encoder network with nominal stride. Doing so frees up computational budget which can then be better utilized in enhancing the decoder. In particular, we use a computationally efficient channel-separated network [39] pre-trained on large-scale video action classification datasets as the encoder. In the decoder, we use novel 3D variants of Global Convolutions [29] and Refinement modules [30, 51]. These enable us to capture a large receptive field and learn high-quality segmentation masks from multi-scale encoder features, respectively. To validate the effectiveness of our network, we apply it to three dataset benchmarks related to salient object segmentation in videos: DAVIS'16 Unsupervised [32], Freiburg-Berkeley Motion Segmentation (FBMS) [27] and ViSal [49]. We show that our network is both faster than existing state-of-the-arts, and also outperforms them by a large margin. Moreover, we perform several ablation experiments to quantitatively justify our various design choices.

To summarize, in this paper, we (i) demonstrate that 3D CNNs can significantly outperform existing (2D CNN based) methods for tasks involving pixel-precise video segmentation; (ii) propose novel 3D variants of Global Convolutions [29] and Refinement modules [30, 51] which significantly improve the decoder's performance; and (iii) establish a new state-of-the-art on three datasets. We believe that our results will motivate others to utilize similar network architectures for other tasks involving pixel-precise video understanding, *e.g.*, discovery of novel object classes [21, 28, 44, 52], semi-supervised Video Object Segmentation [2], Video Instance Segmentation [56] and Multi-Object Tracking and Segmentation [43]. Fig. 1 shows some qualitative results produced by our network.

2 Related Work

3D CNNs for Video Action Classification Early works [17, 18, 37, 41] which applied 3D CNNs to video human action classification used shallow, often custom network architectures, similar to their 2D CNN counter-parts at the time. To overcome the lack of annotated video data, [3, 8] proposed novel ways of leveraging 2D image data for training 3D CNNs. Later, with the emergence of larger video datasets (*e.g.* [19]), it became possible to effectively train deep 3D CNNs from scratch. [12] extended the ResNet [13] architecture to 3D by inflating the 3x3 convolutional kernels to 3x3x3. Doing so, however, significantly increases the computational overhead. [53] proposed mixing 2D and 3D convolutions to improve speed and performance whereas [38] proposed *R(2+1)D* convolutions which factorize 3D convolutions into spatial and temporal convolutions. Inspired by the success of 2D CNNs with channel-separated convolutions [6], [39] proposed a 3D channel-separated ResNet which both performed better and had fewer parameters than existing networks. [10] improved 3D CNN performance through weakly supervised pre-training on large-scale video data. We show that such pre-training is also beneficial for dense pixel-precise segmentation tasks.

Unsupervised Video-Object Segmentation The task of *unsupervised* Video Object Segmentation is to estimate a binary segmentation mask for objects in the video clip that exhibit dominant motion. FusionSeg [16] and LVO [36] process optical flow and appearance in separate streams before performing a learned fusion of the two. [58] generate per-frame object proposals using super-pixels and associate them over time followed by a filtering step to obtain dominant objects. Different from these works, our 3D CNN approach inherently learns to reason about appearance and motion in a unified manner. In the same spirit, [35] use Convolutional LSTMs [54] to leverage the sequential nature of video data and jointly learn spatio-temporal features. However, they use a CRF-based model on top to obtain binary segmentation masks.

In general, methods that employ optical flow, object proposal association, or RNNs struggle with establishing long-range connections. To remedy this, AD-Net [57] learns to associate regions of a reference image frame with those in arbitrary query frames. However, such an approach cannot effectively leverage context from several frames. AGNN [47] uses Graph Neural Networks to pass messages between frames in order to model long-range temporal connections. STEm-Seg [1] uses an encoder-decoder like architecture with 3D convolutions in the decoder to learn temporal context; however, their encoder network is fully 2D. [15] is the most similar to our method as it proposes a fully 3D encoder-decoder network, however, our proposed network architecture differs from theirs and achieves significantly higher performance.

Video Salient Object Detection Several other works tackle the same problem using various nomenclatures involving video salient object detection. Non deep learning based methods [9, 26, 45, 49] generally use handcrafted features to create separate intra-frame and inter-frame saliency maps. The task of merging these maps into a coherent sequence of segmentation masks is then formulated as an optimization problem. [22] learn a saliency model by using optical flow based motion cues in conjunction with LSTMs, whereas [50] use a CNN to learn single-frame saliency and then apply a dynamic saliency model to handle temporal connections. Different from all these works, we use 3D CNNs to jointly learn a saliency model over both spatial and temporal domains.

3 Method

Our method for segmenting salient object regions in videos is based on an encoder-decoder architecture that leverages 3D convolutions to jointly learn spatio-temporal features. As mentioned in Sec. 1, pixel-precise segmentation tasks benefit from higher image resolutions and networks with large receptive fields, which is computationally challenging when working with 3D CNNs. We mitigate these challenges by employing an efficient channel-separated encoder network [39], and a decoder comprising (i) novel 3D Global Convolutions (GC3D) which can capture a large receptive field, and (ii) novel 3D Refinement modules which effectively refine multi-scale encoder features into high quality segmentation masks.

3.1 Encoder

The encoder of our network is a computationally efficient 3D ResNet with channel-separated convolutions which has been successfully used for video action classification [39]. In particular, we use the reduced interaction (ir-CSN) variant of their model in which every 3x3x3 convolution in the *bottleneck* block of the ResNet is replaced with a 3x3x3 depth-wise separable convolution, while the pre-existing 1x1x1 convolutions in the *bottleneck* block capture channel interactions. The reduced memory footprint of this architecture enables a 152 layer variant of this network to be feasibly applied to pixel-precise segmentation tasks in conjunction with our proposed decoder architecture (Sec. 3.2). To justify this design decision, we provide a quantitative analysis of the computational overhead of various backbones used in recent works in Tab. 1. Despite being a *deeper*

Backbone Comparison

Architecture	Type	# Params ($\times 10^6$)	Runtime (sec)
ResNet-101 [13]	2D	42.5	0.173
DeepLabV3 ResNet-101 [4]	2D	42.6	0.793
ResNet-34 R2+1D * [38]	3D	63.5	0.891
ResNet-152 (ir-CSN) [39]	3D	28.7	0.213

Table 1: Comparison of various backbones. Runtime is for generating feature maps for an 8-frame clip with 854 × 480 resolution on an Nvidia GTX-1080Ti. *: lower stride [15]

backbone, the ResNet-152 based ir-CSN has significantly fewer parameters compared to other shallower networks. In terms of runtime, only the vanilla 2D ResNet-101 is slightly faster, however, such 2D networks are inherently unable to learn temporal context.

State-of-the-art methods [47, 57] either employ a 2D network such as DeepLabV3's [4] ResNet-101 backbone, or a shallow 3D network with atrous convolutions and reduced stride [15]. Though this strategy improves performance in segmentation tasks, a major drawback is that it also significantly increases the memory footprint and run-time. We argue that a better approach is to use a computationally efficient channel-separated backbone with nominal stride. Not only does this enable us to have a deeper backbone which can generally learn better features for the end-task, but more importantly, it frees up valuable computational budget that can be used to enhance the decoder's efficacy.

3.2 Decoder

For an input video clip, the encoder produces feature maps at 4 different scales. The decoder architecture comprises a series of 3D convolutions and up-sampling layers which refine these feature maps into the final segmentation mask. To capture a large receptive field in the encoder features, Chen *et al.* [5] proposed using an encoder with reduced stride (8x or 16x) in combination with an Atrous Spatial Pyramid Pooling (ASPP) module which applies multiple parallel atrous convolutions with different dilation rates to a feature map. [15] also proposed a 3D variant of ASPP and used it in their network.

By contrast, we learn encoder features at the nominal 32x stride commonly used for classification tasks. To capture wide spatial context, we propose a 3D variant of *Global Convolutional Networks* which were introduced for semantic segmentation in images [29]. The idea here is that a large $k \times k$ convolution can be replaced with a series of row and column convolutions with kernel sizes $1 \times k$ and $k \times 1$, respectively. This yields the same effective receptive field while having fewer parameters. Our 3D Global Convolution module (GC3D) comprises 3D convolutions with unity kernel size along the temporal dimension. This is because the temporal dimension of the input video clip is usually much smaller than the spatial dimensions.

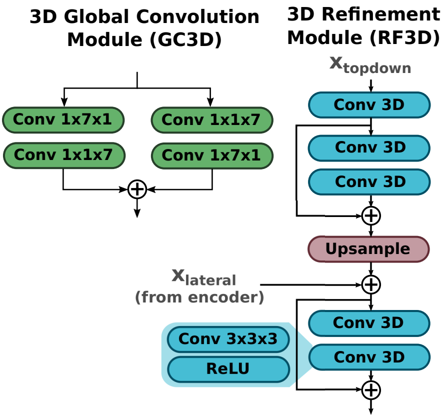

Figure 2: 3D Global Convolution (GC3D) and 3D Refinement modules (RF3D) illustrated.

To combine and upsample the multi-scale feature maps, we additionally propose a 3D variant of the *Refinement module* introduced in [30] for object proposal generation in images. The basic idea here is to apply two 3x3x3 convolutions to a given feature map with a skip connection, followed by trilinear upsampling and addition with the corresponding encoder feature map at that scale. This is followed by two further convolutions with a skip connection. Both GC3D and 3D Refinement modules are illustrated in Fig. 2, and the overall network architecture is illustrated in Fig. 3.

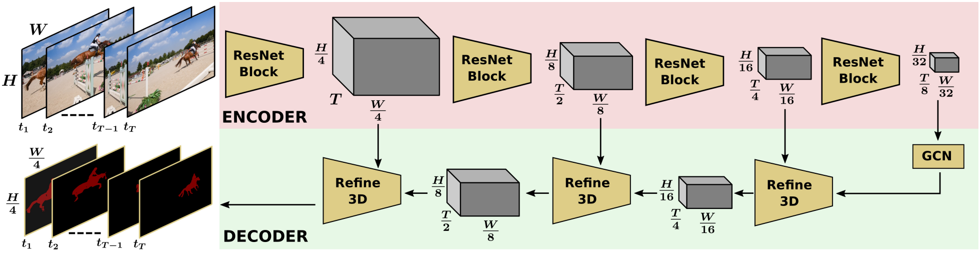

Figure 3: Illustration of our encoder-decoder network architecture

3.3 Video Clip Sampling

For optimal network performance, the input clip's temporal length should be consistent between training and inference. Therefore, to apply the network to videos of arbitrary length, we divide the input video into clips of length T_c with an overlap of T_o between successive clips. For overlapping frames, the mask probabilities are averaged to produce the final segmentation masks. Generally, our method is therefore *near-online*, because given a new frame, the segmentation mask for it becomes available after at most $T_c - T_o - 1$ time-steps (except for the very first T_c frames in the video stream). Note that an online variant can be realized if $T_o \leftarrow T_c - 1$.

4 Experiments

4.1 Datasets and Evaluation

DAVIS'16. Densely Annotated Video Instance Segmentation (DAVIS) is a popular set of benchmarks for video object segmentation related tasks. We evaluate on the DAVIS'16 *unsupervised* benchmark [32] which contains 30 videos for training and 20 for validation/testing. The task is to produce a segmentation mask that captures the dominant objects in the video. Note that the *unsupervised* task differs from the more popular *semi-supervised* task in which ground-truth annotations of the first frame are known during inference. The evaluation metrics used are (i) $\mathcal{J}$-mean, which is the intersection-over-union of the predicted and ground truth masks, and (ii) $\mathcal{F}$-mean, which measures the accuracy of the predicted mask boundaries. These measures can be averaged to give an overall $\mathcal{J}\&\mathcal{F}$ score.

FBMS. The Freiburg-Berkeley Motion Segmentation dataset [27] contains 59 videos which include 12 videos from the Hopkins-155 dataset [40]. The ground truth annotation for every 20$^{\text{th}}$ frame is provided resulting in a total of 720 annotated frames in the entire dataset.

ViSal. The Video Saliency dataset [49] is a collection of 17 videos with a diverse set of objects and backgrounds, varying in length from 30 to 100 frames. For both FBMS and ViSal, the evaluation measures are (i) F-measure, which is the harmonic mean of the per-pixel precision and recall scores, and (ii) the Mean Absolute Error (MAE) $\in [0,1]$ between the predicted and ground truth segmentation masks.

4.2 Training

As mentioned in Sec. 3.1, our encoder is an *ir-CSN* variant of a 3D ResNet-152. We initialize its weights from a model pre-trained on IG-65M [10] and Kinetics [19]. The decoder weights

Backbone	Pre-training	Fine-tuning	$\mathcal{J}\&\mathcal{F}(\%)$
ir-CSN-152	Sports1M + Kinetics	COCO	61.8
ir-CSN-152	Sports1M + Kinetics	DAVIS	82.2
ir-CSN-152	Sports1M + Kinetics	YT-VOS	69.5
ir-CSN-152	Sports1M + Kinetics	COCO, YT-VOS	72.6
ir-CSN-152	Sports1M + Kinetics	YT-VOS, DAVIS	79.4
ir-CSN-152	Sports1M + Kinetics	COCO, DAVIS	82.8
ir-CSN-152	Sports1M + Kinetics	COCO, YT-VOS, DAVIS	83.6
ir-CSN-152	IG-65M + Kinetics	COCO, YT-VOS, DAVIS	84.1
R(2+1)D-34	IG-65M + Kinetics	COCO, YT-VOS, DAVIS	79.6

(a)

(b)

Module	$\mathcal{J}\&\mathcal{F}$
C3D	80.3
NL3D	81.7
ASPP	81.0
GC3D	84.1

(c)

Temporal Length	$\mathcal{J}\&\mathcal{F}$
4	83.4
8	84.1
16	83.7
24	81.5

Table 2: Ablation studies on DAVIS'16 `val`: (a): Comparison of different backbones and the impact of training data; (b): Effect of bridging modules; (c): Performance study on different temporal window size. C3D: 3D convolution, NL3D: Non-local 3D, GC3D: 3D Global convolution. Scores higher than the existing state-of-the-art are highlighted in blue.

are initialized randomly. The network's inputs are video clips of length $T_c = 8$ (the same clip length is used for inference). We sample training clips on the fly from a video sequence of length L by first choosing a random frame $t \in \{1,...,L\}$, and then sampling the remaining $T_c - 1$ frames randomly from $\{t+1,\ldots,\min(t+S,L)\}$. Here, $S = 32$ is a hyper-parameter which limits the maximum temporal span of a training clip. If $t + T_c < L$, the video is padded with its last frame until $t + T_c = L$.

The network is trained end-to-end using the Adam [20] optimizer with an initial learning rate of 10^{-5} which is decayed exponentially after every epoch. The network is first trained on synthetically generated video clips from the COCO Instance Segmentation dataset [23], followed by a second stage with video data from the YouTube-VOS [55] and DAVIS'16 [32] datasets.

Pre-training on Images: Similar to [1, 50, 51], our network generalizes better if we train on static image datasets in addition to video data. For this, we synthesize video clips from static images by augmenting them using a combination of random affine and piecewise-affine transformations. These transformations are applied sequentially to mimic video motion. To obtain the ground truth masks, we combine all object instances into a single foreground mask before applying the same set of transformations.

4.3 Ablations

We perform several ablation experiments on the DAVIS'16 unsupervised validation set to justify our design choices.

Backbone and Training Data: Tab. 2a shows the $\mathcal{J}\&\mathcal{F}$ scores for different encoder networks and the corresponding datasets used for (pre-)training. It can be seen that our model performs consistently well regardless of the encoder network depth and the amount of (pre-)training data. Using a shallow ResNet-34 *R(2+1)D* encoder network [38] (last row), our network achieves 79.6% $\mathcal{J}\&\mathcal{F}$. This is only 1.5% behind the current state-of-the-art method AD-Net [57] (see Tab. 4) which uses a low stride ResNet-101 backbone from DeepLabV3 [4] in addition to heuristic post-processing.

In the interest of comparing against existing 2D CNN approaches which typically use ImageNet [7] pre-training, we conducted several ablations using Sports1M [18] + Kinetics [19] pre-trained weights. The number of data samples in Sports1M + Kinetics (1.8M video clips) is comparable to ImageNet (1.2M images). However, we stress that a direct comparison is difficult since (i) the video datasets have more images, but unlike ImageNet, the image frames within a video are highly correlated, and (ii) ImageNet contains 1000 highly diverse object classes, but Kinetics and Sports-1M are restricted to human action classes. With this setting, fine-tuning on DAVIS alone yields 82.2% $\mathcal{J}\&\mathcal{F}$ (row 2) which outperforms the existing state-of-the-art by 1.1% (see Tab. 4). However, since DAVIS is a small dataset with only 30 training sequences, we obtained further improvements by training on multiple datasets. With additional static image training using COCO [23], as in the (COCO, DAVIS) setting, the score improves to 82.8% (row 6). By further adding YouTube-VOS [55] to the training set (COCO, DAVIS, YT-VOS), the score improves to 83.6% (row 7). Finally, using pre-trained weights from the much larger IG-65M dataset [10] and fine-tuning on all three datasets yields the best score of 84.1% (row 8).

We conclude that even though using more (pre-)training data improves performance, our model achieves state-of-the-art scores even with training data settings that are comparable to existing methods. Secondly, the efficacy of static image training is evident from the fact that the (COCO, DAVIS) setting yields 82.8% $\mathcal{J}\&\mathcal{F}$ which is 3.4% higher than the 79.4% obtained with video data only (YT-VOS, DAVIS). For the sake of completeness, we also report results with models fine-tuned only on COCO (row 1) and YT-VOS (row 3).

Decoder: To justify our decoder architecture, we ablate its two major components: the 3D Global Convolution (GC3D) and 3D Refinement modules. In our network, the GC3D module is applied to the final (smallest) output feature map of the encoder to capture a large receptive field. In Tab. 2b, we compare the network's performance when the GC3D module is replaced by (i) a Non-Local 3D (NL3D) block [48], (ii) an Atrous Spatial Pyramid Pooling (ASPP) module, and (iii) a vanilla 3x3x3 convolution baseline (C3D). It can be seen that ASPP (81.0%) outperforms C3D (80.3%) by 0.7% $\mathcal{J}\&\mathcal{F}$, and NL3D further improves this by another 0.7% (81.7%), but GC3D outperforms all of these modules (84.1%) yielding a 3.8% improvement over the baseline C3D. This highlights the effectiveness of using the GC3D module in our network.

The second major component of our decoder is the 3D Refinement module which helps the network in recovering the spatial and temporal resolution from the feature maps generated by the encoder. Tab. 3 compares our 3D refinement module (RF3D) against a baseline *Upsampling* block which contains two 3x3x3 convolutions followed by a concatenation with encoder features and subsequent trilinear upsampling. As it can be seen, the 3D Refinement module (84.1%) improves performance on DAVIS'16 by 3.9% $\mathcal{J}\&\mathcal{F}$ compared to the *Upsampling* baseline (80.2%), thereby showing its effectiveness in recovering the spatio-temporal resolution.

Module	$\mathcal{J}\&\mathcal{F}$
Upsampling	80.2
RF3D	84.1

Table 3: Analysis of different decoder modules on DAVIS'16.

Input Clip Length: Finally, we ablate the effect of varying the input clip length (T_c) and report the results in Tab. 2c. [38] noted that 3D CNNs can be initially trained with a lower T_c followed by fine-tuning on the target T_c without sacrificing performance. Following this, we first train with $T_c = 8$ on COCO and YouTube-VOS, followed by fine-tuning with the reported T_c on DAVIS. As can be seen, our method is robust to large variations of T_c between 4 and 16. For $T_c > 16$ however, the performance decreases. This highlights our architecture's limitation in coping with very large temporal dimensions, which we leave for future work.

DAVIS 2016 Unsupervised							
Method	OF	CRF	MS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$-mean	$\mathcal{F}$-mean	Time (s/frame)
OnAVOS [42]				-	72.7	-	-
ARP [58]	✓			73.4	76.2	70.6	-
LVO [36]	✓	✓		74.0	75.9	72.1	-
PDB [35]		✓		75.9	77.2	74.5	-
MotAdapt [33]				77.3	77.2	77.4	-
3D-CNN [15]				77.8	78.3	77.2	0.38
AD-Net [57]			✓	78.8	79.4	78.2	0.38
AGNN [47]		✓	✓	79.9	80.7	79.1	2.96
STEm-Seg [1]			✓	80.6	80.6	80.6	1.42
AD-Net + Inst-Pruning [57]*			✓	81.1	81.7	80.5	2.94
Ours				84.1	83.9	84.3	**0.22**
Ours - Dense				**84.5**	**84.3**	**84.7**	0.84

Table 4: DAVIS'16 validation set results for the unsupervised track. OF: Optical Flow, MS: Multi-Scale inference, CRF: CRF post-processing. Runtime was computed on an Nvidia GTX-1080Ti. * Uses heuristic post-processing. Best performance scores are highlighted in bold.

4.4 Benchmark results

DAVIS 2016 Unsupervised: We apply our network to the DAVIS'16 unsupervised benchmark [32] and report the results in Tab. 4. Our 3D CNN achieves 84.1% $\mathcal{J}\&\mathcal{F}$, which is a substantial 3% improvement over the existing state-of-the-art of 81.1%. It also performs better in terms of the individual $\mathcal{J}$-mean and $\mathcal{F}$-mean measures. This shows that our no-bells-and-whistles encoder-decoder network is able to produce high quality segmentation masks by jointly learning the salient objects' appearance and motion models. By contrast, several competing methods perform inference at multiple input scales and/or apply post-processing techniques such as CRFs to improve performance which imposes additional computational overhead and renders the method non-end-to-end trainable. The best performing existing method, AD-Net [57], applies an *instance pruning* post-processing step that additionally requires inference with a separate image instance segmentation network which is trained on COCO [23]. Under this setting, AD-Net uses ImageNet and DAVIS for (pre-)training its primary network and COCO [23] for post-processing. Without this post-processing step, AD-Net achieves 79.4% $\mathcal{J}\&\mathcal{F}$ which is 4.7% lower than our score of 84.1%. STEm-Seg [1], the second-best performing existing method, uses a backbone network initialized from Mask-RCNN [14] weights, and then further fine-tunes jointly on COCO [23], YouTube-VIS [56] and DAVIS [32]. It also employs multi-scale inference whereas we do not. The existing 3D CNN approach [15] fine-tunes only on DAVIS and achieves 77.8% $\mathcal{J}\&\mathcal{F}$ which is 3.4% lower than our comparable ablation score of 82.2% (see Tab. 2a, row 2). Due to our decision of using a nominal stride and an efficient encoder network, our method is also the fastest among recent works. It runs at 0.22 s/frame (4.5fps) on an Nvidia GTX-1080Ti which is 42% faster than the two tied second fastest methods (*AD-Net* and *3D-CNN* with 0.38 s/frame or 2.6fps).

We use a frame overlap of $T_o = 3$, however, as mentioned in Sec. 3.3, an online version can be realized if $T_o \leftarrow T_c - 1$. The results for this setting are given as *Ours - Dense* and the performance is slightly better, but this setting is slower.

Video Object Saliency: This task involves segmenting pixels in a video which belong to *salient* objects, and is similar to unsupervised video object segmentation. Our method can therefore be directly evaluated on it.

In addition to evaluating our DAVIS'16 [31] results using the F-measure and MAE, we also evaluate on the FBMS [27] and ViSal [49] datasets without any additional dataset-specific training. The scores for all three datasets are reported in Tab. 5. Our method outperforms the state-of-the-art on all these datasets for both evaluation measures, thus signifying its generalization capability. Note that the performance improvement on DAVIS'16 is particularly high compared to the second-best method.

	DAVIS '16		FBMS		ViSal	
Method	F↑	MAE↓	F↑	MAE↓	F↑	MAE↓
FGRNE [11]	78.6	0.043	77.9	0.083	85.0	0.040
FCNS [46]	72.9	0.053	73.5	0.100	87.7	0.041
SGSP [25]	67.7	0.128	57.1	0.171	64.8	0.172
GAFL [49]	57.8	0.091	55.1	0.150	72.6	0.099
SAGE [45]	47.9	0.105	58.1	0.142	73.4	0.096
STUW [9]	69.2	0.098	52.8	0.143	67.1	0.132
SP [24]	60.1	0.130	53.8	0.161	73.1	0.126
AD-Net [57]	80.8	0.044	81.2	0.064	90.4	0.030
Ours	**91.8**	**0.015**	**84.5**	**0.048**	**92.2**	**0.019**

Table 5: F-measure and MAE for DAVIS, FBMS and ViSal datasets. ↑: Higher is better, ↓: Lower is better.

5 Conclusion

In this paper, we proposed a simple and fast network architecture consisting entirely of 3D convolutions that is capable of effectively learning spatio-temporal features without additional bells and whistles. To this end, we employed a deep yet computationally efficient 3D ResNet pretrained for video action classification as an encoder, and a novel decoder architecture inspired by existing 2D convolutional networks. Our experiments show that in addition to being faster than existing state-of-the-art methods, our network also out-performs them on three different datasets by a large margin. We believe that our findings will encourage other researchers to employ 3D convolutions for a variety of tasks involving pixel-precise video scene understanding, and that our proposed network architecture can serve as a useful starting point for their endeavours.

Acknowledgements. This project was funded, in parts, by ERC Consolidator Grant DeeVise (ERC-2017-COG-773161), EU project CROWDBOT (H2020-ICT-2017-779942) and the Humboldt Foundation through the Sofja Kovalevskaja Award. Computing resources for several experiments were granted by RWTH Aachen University under project 'rwth0519'. We thank Paul Voigtlaender and István Sárándi for helpful discussions.

References

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

[2] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[3] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *ECCV*, 2018.

[9] Yuming Fang, Zhou Wang, Weisi Lin, and Zhijun Fang. Video saliency incorporating spatiotemporal cues and uncertainty weighting. *Trans. Image Proc.*, 2014.

[10] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

[11] Li Guanbin, Xie Yuan, Wei Tianhao, Wang Keze, and Lin Liang. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, 2018.

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[15] Rui Hou, Chen Chen, Rahul Sukthankar, and Mubarak Shah. An efficient 3d CNN for action/object segmentation in video. In *BMVC*, 2019.

[16] Suyog Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017.

[17] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 2012.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[21] Suha Kwak, Minsu Cho, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, 2015.

[22] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, 2018.

[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 *ECCV*, 2014.

[24] Zhi Liu, Xiang Zhang, Shuhua Luo, and Olivier Le Meur. Superpixel-based spatiotemporal saliency detection. *IEEE Trans. Circ. and Syst. Vid. Tech.*, 2014.

[25] Zhi Liu, Junhao Li, Linwei Ye, Guangling Sun, and Liquan Shen. Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation. *IEEE Trans. Circ. and Syst. Vid. Tech.*, 2016.

[26] Zhi Liu, Junhao Li, Linwei Ye, Guangling Sun, and Liquan Shen. Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation. In *IEEE Trans. Circ. and Syst. Vid. Tech.*, 2017.

[27] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 2013.

[28] Aljoša Ošep, Paul Voigtlaender, Jonathon Luiten, Stefan Breuers, and Bastian Leibe. Large-scale object mining for object discovery from unlabeled video. *ICRA*, 2019.

[29] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, 2017.

[30] P.H.O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016.

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

[32] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo ArbelÃąez, Alex Sorkine-Hornung, and Luc Van Gool. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[33] Mennatullah Siam, Chen Jiang, Steven Weikai Lu, Laura Petrich, Mahmoud Gamal, Mohamed Elhoseiny, and Martin Jägersand. Video segmentation using teacher-student adaptation in a human robot interaction (HRI) setting. In *ICRA*, 2018.

[34] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[35] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018.

[36] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017.

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[38] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[39] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019.

[40] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007.

[41] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *PAMI*, 2017.

[42] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

[43] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, B.B.G Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *CVPR*, 2019.

[44] Le Wang, Gang Hua, Rahul Sukthankar, Jianru Xue, and Nanning Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, 2014.

[45] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015.

[46] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. *Trans. Image Proc.*, 2017.

[47] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J. Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 2019.

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[49] Ling Shao Wenguan Wang, Jianbing Shen. Consistent video saliency using local gradient flow optimization and global refinement. *Trans. Image Proc.*, 2015.

[50] Ling Shao Wenguan Wang, Jianbing Shen. Video salient object detection via fully convolutional networks. *Trans. Image Proc.*, 2018.

[51] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018.

[52] Fanyi Xiao and Yong Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016.

[53] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[54] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[55] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

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

[57] Zhao Yang, Qiang Wang, Luca Bertinetto, Weiming Hu, Song Bai, and Philip H. S. Torr. Anchor diffusion for unsupervised video object segmentation. In *ICCV*, 2019.

[58] Chang-Su Kim Yeong Jun Koh. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017.