An Efficient 3D CNN for Action/Object Segmentation in Video

Rui Hou
houray@gmail.com

Chen Chen
chen.chen@uncc.edu

Rahul Sukthankar
sukthankar@google.com

Mubarak Shah
shah@crcv.ucf.edu

Niantic Labs
Sunnyvale, CA, USA

University of North Carolina at Charlotte
Charlotte, NC, USA

Google Research
Mountain View, CA, USA

University of Central Florida
Orlando, FL, USA

Abstract

Convolutional Neural Network (CNN) based image segmentation has made great progress in recent years. However, video object segmentation remains a challenging task due to its high computational complexity. Most of the previous methods employ a two-stream CNN framework to handle spatial and motion features separately. In this paper, we propose an end-to-end encoder-decoder style 3D CNN to aggregate spatial and temporal information simultaneously for video object segmentation. To efficiently process video, we propose 3D separable convolution for the pyramid pooling module and decoder, which dramatically reduces the number of operations while maintaining the performance. Moreover, we also extend our framework to video action segmentation by adding an extra classifier to predict the action label for actors in videos. Extensive experiments on several video datasets demonstrate the superior performance of the proposed approach for action and object segmentation compared to the state-of-the-art.

1 Introduction

Video object segmentation is a fundamental task in video content analysis. It aims to assign a foreground/background label for each pixel in a video frame. Compared to image segmentation, there are two major differences in video object segmentation. First, the amount of data to be processed in video can be orders of magnitude greater than in image segmentation, placing greater constraints in terms of computational resources. Second, the temporal domain provides additional information about object motion that can judiciously be exploited to improve segmentation performance.

Video object segmentation approaches can be divided into two categories – semi-supervised and unsupervised. The semi-supervised approaches [28, 43] assume the foreground object in the first frame of test video is provided and the task is to segment the specified object in the following frames. On the other hand, the unsupervised approaches [2, 32, 37, 38] segment foreground objects without any prior knowledge, which is more suited for practical use.

In this paper, we approach the video object segmentation in the **unsupervised setting**. We propose an end-to-end encoder-decoder style 3D CNN based method to solve the video object segmentation problem efficiently. Its encoder is composed of a R2plus1D (R2P1D) network [42] and a 3D pyramid pooling module. Its decoder is designed to recover both spatial and temporal dimensions to generate an output of the same size as the input clip. Instead of the popular two-stream framework, we adopt 3D CNN to aggregate spatial and temporal information. To efficiently process video, we propose 3D separable convolution for the pyramid pooling module and decoder, which dramatically reduces the number of operations while maintaining the performance. Additionally, we also extend our framework to video action segmentation by adding an extra classifier to predict the action label for actions in videos. The main contributions of this paper are three-fold:

1. We propose a simple yet efficient 3D CNN framework for action/object segmentation in videos. To the best of our knowledge, this is the first time 3D CNN has been explored for video object segmentation to simultaneously model the spatial and temporal information in a video.

2. We evaluate our method on video object segmentation and action segmentation benchmarks and demonstrate state-of-the-art performance.

3. We conduct a detailed ablation study to identify the relative contributions of the individual components.

2 Related Work

Convolutional neural networks have been demonstrated to achieve excellent results in video action understanding [17, 23, 47]. Video should not be treated as a set of independent frames, since the connection between frames provides extra temporal information for understanding. Simonyan *et al*. [36] propose the two-stream CNN approach for action recognition, which consists of two CNNs taking image and optical flow as input respectively. To avoid computing optical flow separately, Tran *et al*. [41] propose 3D CNN for large scale action recognition. Hara *et al*. [12] apply 3D convolution on ResNet structure. Carreira *et al*. [3] propose I3D by extending Inception network from 2D to 3D and including an extra optical flow stream. Tran *et al*. [42] and Xie *et al*. [44] factorize 3D CNN to treat spatial and temporal information separately to reduce the computational cost while keeping the performance. However, to the best of our knowledge, we are the first ones to exploit 3D CNN for video object segmentation.

CNN-based segmentation. The success of CNN-based approaches for image classification [13, 22] have led to dramatic advances in image segmentation [26, 29]. Many of the segmentation approaches leverage recognition models trained on ImageNet and replace the fully-connected layers with 1×1 kernel convolutions to generate dense (pixel-wise) labels. Recently, the encoder-decoder style network architecture, such as SegNet [1] and U-Net [33], has been the main stream design for semantic segmentation. Moreover, pyramid pooling [5, 48] and dilated convolution [5, 6, 46] are effective techniques to improve the segmentation accuracy. Our 3D CNN also builds upon the encoder-decoder structure for video object segmentation.

Video object segmentation. Video object segmentation [19, 45] aims to delineate the foreground object(s) from the background in each frame. Semi-supervised segmentation

pipelines [28, 43] assume that the segmentation mask of the first frame in the sequence during testing is given, and exploit temporal consistency in video sequences to propagate the initial segmentation mask to subsequent frames.

In the more challenging unsupervised setting, as we address in this paper, no object mask is provided as initialization during the test phase. Unsupervised segmentation has been addressed by several variants of CNN-based models, such as the two-stream architecture [2, 32], recurrent neural networks [37, 38] and multi-scale feature fusion [37, 39]. These approaches generally perform much better than traditional clustering-based pipelines [4]. The core idea behind such approaches involves leveraging motion cues explicitly (via optical flow) using a two-steam network [8, 38, 39], and/or employing a memory module to capture the evolution of object appearance over time [37, 38].

Action segmentation. Action segmentation provides pixel-level localization for actions (*i.e.* action segmentation maps), which are more accurate than bounding boxes for action localization. Lu *et al.* [27] propose supervoxel hierarchy to enforce the consistency of the human segmentation in video. Gavrilyuk *et al.* [11] infer pixel-level segmentation of an actor and its action in video from a natural language input sentence.

While we take inspiration from these works, we are the first to present a 3D CNN based deep framework for video object segmentation in a fully automatic manner. Moreover, our proposed method is designed with computational efficiency in mind to enable practical applications for video segmentation.

3   Generalizing CNN for Dense Prediction from 2D to 3D

Generalizing CNN framework from images (2D) to videos (spatio-temporal 3D) involves more work than simply adding one more dimension. A key challenge is due to the asymmetry between space and time. To address the change in apparent size of an object due to perspective, image pipelines crop and reshape images to a fixed size. One cannot do the same with videos since input videos and the duration of an object track or action in an untrimmed video can vary widely in the temporal dimension. Since an entire video cannot be rescaled to a fixed size, approaches typically process video as sequences of short (*e.g.* 8-frame) clips. In this section, we delve into the details of our network design to address these challenges.

3.1   3D separable convolution with dilation

3D CNN training is much less efficient than its 2D counterpart, since 3D kernels and feature maps have more parameters. To address pixel-level prediction in video more efficiently, we propose to use 3D separable convolution in lieu of the standard 3D convolution. 3D separable convolution factorizes a standard 3D convolution into a channel-wise 3D convolution and a point-wise 3D convolution with $1 \times 1 \times 1$ kernel. The channel-wise convolution applies a specific filter to each input channel, then point-wise convolution combines the outputs of the channel-wise convolution via a $1 \times 1 \times 1$ convolution. This factorization drastically reduces computation and model size.

A standard 3D convolutional operator takes a $H \times W \times T \times M$ feature map F as input and produces a $H \times W \times T \times N$ feature map G, where H, W and T are the height, width and number of frames (duration) in a clip, respectively. M and N are the number of input/output channels. 3D convolution kernel has shape $K_h \times K_w \times K_t$, where K_h, K_w and K_t are the spatial and temporal dimensions of the kernel. The computation complexity of a standard
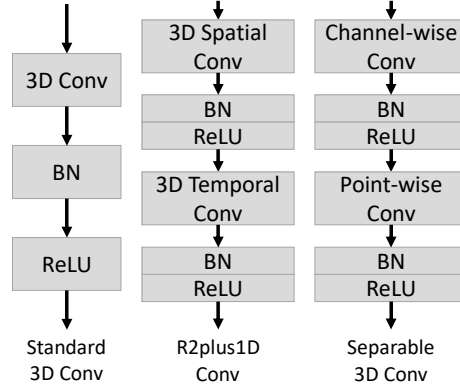
Figure 1: Comparison between standard 3D convolution, R2plus1D and 3D separable convolution. R2plus1D factorizes 3D convolution into spatial and temporal convolutions. 3D separable convolution is composed of two convolution modules – channel-wise convolution and point-wise convolution.

3D convolution is:

$$H \times W \times T \times M \times N \times K_h \times K_w \times K_t. \tag{1}$$

On the other hand, 3D separable convolution is a two-step process. First channel-wise convolution generates the intermediate feature map by applying a specific filter to each input channel (M channels in total). The computational complexity of channel-wise convolution is:

$$H \times W \times T \times M \times K_h \times K_w \times K_t. \tag{2}$$

where the kernel of channel-wise convolution has size $K_h \times K_w \times K_t$. Then point-wise convolution projects intermediate feature map to the final output with computational complexity:

$$H \times W \times T \times M \times N. \tag{3}$$

The computational cost of the standard 3D convolution is the multiplication of the feature map dimension (H, W, T), input channel dimension M, output channel dimension N and kernel dimension (K_h, K_w, K_t); while in 3D separable convolution, channel-wise convolution is irrelevant to output filters and point-wise convolution is isolated from kernel dimension.

Therefore, the computational cost reduction of 3D separable convolution is:

$$\frac{H \times W \times T \times M \times K_h \times K_w \times K_t + H \times W \times T \times M \times N}{H \times W \times T \times M \times N \times K_h \times K_w \times K_t} = \frac{1}{N} + \frac{1}{K_h \times K_w \times K_t}. \tag{4}$$

For example, a 3D separable convolution with $3 \times 3 \times 3$ kernel dimension and 512 input/output size only needs about $1/25$ of computational resource compared to that of a standard 3D convolution, leading to significant inference computation reduction.

Meanwhile, in R2plus1D convolution, the standard 3D convolution is factorized into spacial and temporal convolutions. The input feature map with M channels passes through spatial convolution with kernel shape $K_h \times K_w \times 1$ and generates a M' channels intermediate feature map. Then, the intermediate result goes through the temporal convolution with kernel
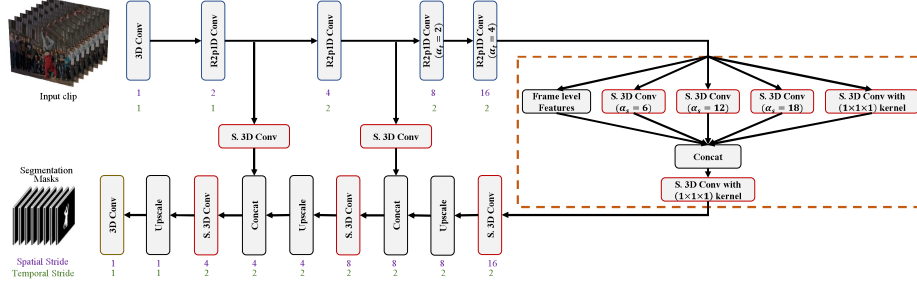
Figure 2: The network architecture of our method for video object segmentation. It has three components: an encoder (feature extractor), a pyramid pooling module and a decoder to recover the spatial-temporal details gradually. The encoder, shown in the top-left, takes pixel values as input, extracts features layer by layer, and generates rich contextual feature as the final output. Moreover, its intermediate results are merged with the decoder module. The pyramid pooling module connects the encoder and decoder modules. It varies the receptive field size by modifying the spatial stride. "S. 3D Conv" indicates the 3D separable convolution. "Frame level features" block is composed by a reduced average pooling and duplication to generate the frame level features with the same dimension as the other branches.

shape $1 \times 1 \times K_t$ and generates the final output. Compared to standard 3D convolution, the computational cost reduction of R2plus1D convolution is:

$$\frac{H \times W \times T \times M \times M' \times K_h \times K_w + H \times W \times T \times M' \times N \times K_t}{H \times W \times T \times M \times N \times K_h \times K_w \times K_t} = \frac{M'}{N \times K_t} + \frac{M'}{M \times K_h \times K_w}. \quad (5)$$

The computational reduction is depends on the number of intermediate filters M'. According to [42], the number of intermediate filters is set as $\frac{M \times N \times K_t \times K_h \times K_w}{N \times K_t + M \times K_h \times K_w}$ to keep the number of parameters the same as standard convolution.

Dilated convolution. Dilation rate γ_a is an attribute of convolution operation. It has the ability to increase the receptive field size, while maintaining the computational cost by skipping $\gamma_a - 1$ entities per valid one in one of dimensions (x, y, t). Adding dilation rates helps capture multi-scale feature representations. Specifically in 3D separable convolution, the dilation rate is only applied to channel-wise convolution, since kernel size of point-wise convolution is fixed at $1 \times 1 \times 1$. Moreover, due to the asymmetry between space and time in video, dilation rate in 3D convolution can be divided into two parts – spatial rate γ_s and temporal rate γ_t. By using the dilation rate, the channel-wise convolution can be expressed as follows:

$$\hat{G}_{h,w,t,m} = \sum_{i,j,k} \hat{K}_{i,j,k,m} \cdot F_{h+\gamma_s \cdot i, w+\gamma_s \cdot j, t+\gamma_t \cdot k, m}. \quad (6)$$

As shown in Eq. 6, the computational cost does not change by adding the dilation rate.

3.2 Network architecture

The proposed 3D CNN architecture for action/object segmentation in video is illustrated in Figure 2. The network builds upon an encoder-decoder structure for image semantic segmentation. A video is divided into 8-frame clips as input to the network. In the encoder module, 3D convolution are performed. To capture higher level information, the spatial and

temporal sizes are reduced. In order to generate the pixel-wise segmentation map for each frame in the original size, 3D upsampling is used in the decoder module to increase the resolution of feature maps. To capture spatial and temporal information at different scales, a concatenation with the corresponding feature maps from the encoder module is employed after each 3D upsampling layer. Finally, a softmax cross entropy loss layer is used for pixel-wise prediction (*i.e.* background or object foreground) for each frame in a clip.

In the encoder, we adopt R2plus1D [42] as feature extractor to leverage its pre-trained model on large scale action recognition dataset. We modify the last three groups of convolutional layers by adjusting the dilation rate to keep temporal stride as 4. We also insert a 3D pyramid pooling before the decoder. In the decoder, the up-sampling layer group is composed of a tri-linear interpolation layer, a 3D separable convolution layer as well as a feature concatenation layer incorporating encoder feature.

3D Pyramid Pooling. Compared to the 2D counterpart [6], pyramid pooling in 3D is more challenging. In spatial domain, multi-scale information should be captured. In temporal domain, detailed motion information should be preserved as well. Therefore, our 3D pyramid pooling has 5 branches. The details are listed as follows:

1. 3D separable convolutions with kernel size $3 \times 3 \times 3$ and different spatial dilation rates $\gamma_s = 6, 12, 18$. The convolutions with multiple spatial dilation rates are able to capture multi-scale information.

2. 3D separable convolution with kernel size $1 \times 1 \times 1$ – a default layer for dense prediction.

3. There are two steps to get the frame feature. First, input is average pooled with kernel $H \times W \times 1$. For the average pooling output, its spatial dimension is 1, while its temporal length and filter size are the same as those of the input. Then the average pooling output is upscaled to the input size.

By concatenating the 5 branches together, and applying a $1 \times 1 \times 1$ convolution, the final output is obtained and fed into the decoder module.

4 Experiments

Implementation details. To verify the effectiveness of the proposed 3D CNN framework for action/object segmentation in videos, we evaluate our approach on two video object segmentation datasets – DAVIS'16 [31] and Segtrack-v2 [25], and a video action segmentation dataset – J-HMDB [15]. We adopt R2plus1d encoder pretrained on Kinetics dataset [18] and leverage the Adam optimizer with initial learning rate 10^{-4}, exponential decay rate 0.95 and decay step 1 epoch. The model is trained for 100 epochs with exponential decay. In 3D spatial pyramid pooling module, a large crop size is required to ensure the convolution kernel with spatial dilation rate is effective; otherwise, the filter weights with large dilation rate are mostly applied to the padded zero region. Therefore, We employ a spatial crop size of 384 during both training and testing. During training, we randomly scale the resolution of input clips with ratio $[0.5, 2]$ and apply a random horizontal flip.

Experiments on DAVIS'16. Densely Annotated Video Segmentation 2016 (DAVIS'16) dataset is a benchmark dataset for video object segmentation. It consists of 50 videos with 3455 annotated frames. Consistent with most prior work, we conduct experiments on the 480p videos with a resolution of 854×480 pixels. 30 videos are used for training and 20

for validation. We adopt the same evaluation setting in [31]. There are three parts. **Region Similarity** $\mathcal{J}$, which is obtained by IoU (Intersection over Union) between the prediction and the ground-truth segmentation map. **Contour Accuracy** $\mathcal{F}$ measures the contours accuracy. **Temporal Stability** $\mathcal{T}$ tracks the temporal consistency in a video. For the first two evaluation, we report the mean, recall and decay. For the third one, we report the average.

Measure		MotAdapt [35]	LSMO [40]	PDB [37]	ARP [21]	FSEG [8]	LMP [39]	FST [20]	CUT [9]	NLC [10]	Ours
	Mean ↑	77.2	78.2	77.2	76.2	70.7	70.0	55.8	55.2	55.1	**78.3**
$\mathcal{J}$	Recall ↑	87.8	89.1	90.1	**91.1**	83.5	85.0	64.9	57.5	55.8	**91.1**
	Decay ↓	5.0	4.1	0.9	7.0	1.5	1.3	**0.0**	2.2	12.6	2.3
	Mean ↑	**77.4**	75.9	74.5	70.6	65.3	65.9	51.1	55.2	52.3	77.2
$\mathcal{F}$	Recall ↑	84.4	**84.7**	84.4	83.5	73.8	79.2	51.6	61.0	51.9	**84.7**
	Decay ↓	3.3	3.5	**-0.2**	7.9	1.8	2.5	2.9	3.4	11.4	4.9
$\mathcal{T}$	Mean ↓	27.9	**21.2**	29.1	39.3	32.8	57.2	36.6	27.7	42.5	22.0

Table 1: Overall results of region similarity ($\mathcal{J}$), contour accuracy ($\mathcal{F}$) and temporal stability ($\mathcal{T}$) for different approaches. ↑ means the higher the better, and ↓ means the lower the better.

We compare our results with several unsupervised approaches, since our approach does not require any manual annotation or prior information about the object to be segmented. We cannot compare directly to semi-supervised approaches that require the ground truth segmentation map in the first frame of each test video to be given. Table 1 summarizes the performance of our method against the state-of-the-art unsupervised approaches on DAVIS'16. Our method achieves the best performance in all performance metrics. Compared to ARP [21], the previous state-of-the-art unsupervised approach, our method achieves 5% gain in contour accuracy ($\mathcal{F}$) and 15% gain in temporal stability ($\mathcal{T}$), demonstrating that 3D CNN can effectively take advantage of the temporal information in video frames to achieve temporal segmentation consistency. We also present the qualitative results on four video sequences in Figure 3.

Figure 3: Qualitative results of the proposed (Ours) approach (red), ARP (yellow), LVO (cyan) and FSEG (magenta) on selected frames from DAVIS dataset.

Experiments on Segtrack-v2. SegTrack-v2 [25] contains 14 video clips with 24 objects and 947 frames. Pixel-level object mask on each frame is provided. We adopt the same

Method	FST [30]	KEY [24]	LSMO [38]	FSEG [8]	Ours
mean IoU	53.5	57.3	53.7	61.4	**62.1**

Table 2: Video object segmentation results on Segtrack-v2 dataset. We compare the performance of our approach with other state-of-the-art unsupervised approaches.

evaluation setting in [32] and report the mean Intersection over Union. For videos with multiple instances with individual ground-truth segmentation mask, we group them as a single foreground for evaluation. Compared with other unsupervised video object segmentation methods, our proposed approach outperforms all of them (see table 2).

Experiments on J-HMDB. The J-HMDB dataset consists of 928 videos with 21 different actions. There are three train-test splits and the evaluation is done on the average results over the three splits. We leverage the mask annotation provided from J-HMDB dataset and train our semantic segmentation pipeline to segment the foreground action. Then the region with maximum area is selected through connected component [14]. On the final feature map of each frame, we crop a box to tightly surround the selected foreground region and resize the box into a fixed shape. Finally, the cropped feature map goes through a classifier to predict action classes. We use softmax cross entropy loss to back propagate gradients for all the weights. For evaluation, we followed the metrics in [11]. Mean IoU is computed as the average over the IoU of each test sample. In addition, frame-level mean Average Precision (mAP) is evaluated as well. Since bounding boxes detection can be obtained by selecting the tightest rectangle region enclosing the segmentation mask, we are also able to compare with the state-of-the-art action detection approaches.

	mean IoU	frame-mAP ($\alpha = 0.5$)
Lu *et al.* [27]	48.8	–
Gavrilyuk *et al.* [11]	54.2	–
Kalogeiton *et al.* [16]	–	65.7
Duarte *et al.* [7]	–	64.6
Hou *et al.* [34]	–	61.3
Ours	**68.1**	**68.9**

Table 3: Comparison with the state-of-the-art action segmentation and detection approaches on J-HMDB. The first part of the table shows the mean Intersection-over-Union (IoU) of action segmentation approaches and the second part shows the frame level mean Average Precision (mAP) of CNN based action detection approaches.

Table 3 reports the action segmentation/detection results of our method and the state-of-the-art approaches. It is evident that our method outperforms these methods considerably in evaluation metrics. Figure 4 presents both action segmentation (in red) and bounding boxes detection (in yellow) results on several video sequences from J-HMDB.

4.1 Ablation Study

To better understand the contribution of each component in our proposed approach, we conduct video object segmentation on DAVIS'16 with different settings, summarized in Table 4 and Table 5.

Dilation rate. We adopt R2Plus1D pre-trained model in our approach, and increase the feature map's temporal size of last two convolution layer group by specifying the dilation rate. As shown in first section of Table 4, the performance is boosted by increasing the
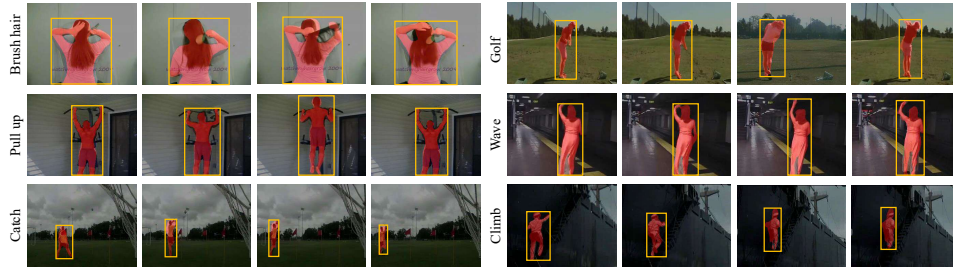
Figure 4: Action segmentation and detection results obtained by our method on the J-HMDB dataset. Red pixel-wise segmentation maps show the predictions, and the yellow boxes show the bounding boxes generated from the segmentation maps.

number of frames in the final feature map. However, too large feature map (8 frames) will cause out of memory error, that is the reason we stop at 4 frames.

3D pyramid pooling. As shown in the second section of Table 4, inserting 3D pyramid pooling module improves the segmentation accuracy by over 5%. We experiment multiple branches with different spatial dilation rates (as noted in the bracket). According to the experimental results, including more branches with larger receptive field (4 branches) or larger temporal stride is not helpful. When the receptive field size is close to or larger than the feature map size, most of the filters cannot capture any useful information, since they only cover padded zeros instead of valid area. Segmentation accuracy further improves when frame-level features are added.

Final feature map dim.	3D Pyramid Pooling	$\mathcal{J}$-Mean
$1 \times 20 \times 20$	–	64.5
$2 \times 20 \times 20$	–	69.1
$4 \times 20 \times 20$	–	71.3
$4 \times 20 \times 20$	(6, 12, 18)	78.1
$4 \times 20 \times 20$	(6, 12, 18) + FF	78.3
$4 \times 20 \times 20$	(6, 12, 18, 24)	77.9
$4 \times 20 \times 20$	(8, 16, 24)	74.9

Table 4: Ablation study of our method for video object segmentation on DAVIS-16. In the first horizontal section of the Table, we investigate various temporal size of the final feature map by increasing the temporal dilation rate in the last two convolutional layer groups. In the second section of the Table, we fix feature map dimensions (including temporal dimension) and explore different settings in 3D pyramid pooling module (shown in the second column), which includes various receptive fields and whether frame-level features (FF) are included or not. Specifically, the numbers in the parentheses () indicate spatial dilation rates of branches. We compare performance with different sizes of feature maps, with or without 3D pyramid pooling layer. Mean IoU is used as evaluation metric as shown in third column.

3D separable convolution. The proposed pipeline leverages 3D separable convolution in pyramid pooling and decoder. We perform an experiment by replacing all the 3D separable convolutions with R2plus1D convolutions and standard 3D convolutions. The comparison of performance and computational cost is shown in Table 5. All the experiments are carried out on a workstation with a NVIDIA Titan XP GPU and PyTorch. We only take 3D pyramid

pooling and decoder part during inference into computation cost, since all of them share the same pre-trained model. "Operations" counts the total number of additions and multiplications. "GPU Mem." is the size of allocated GPU memory. With the same input (a 8 frames clip with resolution 320×320), 3D separable convolution greatly reduces the computational cost without sacrificing the performance.

Conv. Type	Operations	GPU Mem.	$\mathcal{J}$-Mean
Standard 3D Conv.	136 Billion	255 MB	77.1
R2plus1D Conv.	136 Billion	256 MB	77.6
3D Separable Conv.	6 Billion	11 MB	77.4

Table 5: The comparison of performance and computational cost of different 3D convolutions. 3D separable convolution is able to reach the similar accuracy with only 5% of parameters. In R2plus1D, we adopt the settings in [42], which sets the number of intermediate filters to be the same as the number of parameters of the standard 3D convolution.

5 Conclusion

This paper proposes a fully convolutional 3D CNN pipeline for action/object segmentation in video. The approach leverages a model pre-trained on large-scale action recognition task as an encoder to enable us to perform unsupervised video object segmentation (*i.e.* generate pixel-level object masks without initialization). We also use separable filters to significantly reduce the computational burden of the standard 3D convolutions. Extensive experiments on several benchmark datasets demonstrate the strength of our approach for spatio-temporal action segmentation as well as video object segmentation compared with the state-of-the-art approaches.

6 Acknowledgements

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[4] Jason Chang, Donglai Wei, and John W. Fisher III. A video representation using temporal superpixels. In *IEEE Conference on Computer Vision Pattern Recognition*, 2013.

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

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[7] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems*, pages 7610–7619, 2018.

[8] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3664–3673, 2017.

[9] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *British Machine and Vision Conference*, volume 2, 2014.

[10] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[11] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d CNNs retrace the history of 2d CNNs and ImageNet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Lifeng He, Yuyan Chao, Kenji Suzuki, and Kesheng Wu. Fast connected-component labeling. *Pattern Recognition*, 42(9):1977–1987, 2009.

[15] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[16] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[17] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[18] 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.

[19] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event detection in crowded videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[20] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[21] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553), 2015.

[24] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[25] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[27] Jiasen Lu, Jason J Corso, et al. Human action segmentation with hierarchical supervoxel consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[28] Kevis Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, and Luc Van Gool. Video object segmentation without temporal information. *IEEE transactions on pattern analysis and machine intelligence*, PP(99):1–1, 2018.

[29] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[30] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[31] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[34] Hou Rui, Chen Chen, and Mubarak Shah. Tube convolutional neural network (T-CNN) for action detection in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[35] Mennatullah Siam, Chen Jiang, Steven Lu, Laura Petrich, Mahmoud Gamal, Mohamed Elhoseiny, and Martin Jagersand. Video segmentation using teacher-student adaptation in a human robot interaction (HRI) setting. *arXiv preprint arXiv:1810.07733*, 2018.

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

[37] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[38] Pavel Tokmakov and Karteek Alahari. Learning Video Object Segmentation with Visual Memory. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[39] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *International Journal of Computer Vision*, 127(3):282–301, 2019.

[41] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[42] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[43] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 DAVIS challenge on video object segmentation. In *The 2017 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, 2017.

[44] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[45] Pingkun Yan, Saad M Khan, and Mubarak Shah. Learning 4d action feature models for arbitrary view action recognition. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008.

[46] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[47] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.