Learning to Segment Instances in Videos with Spatial Propagation Network

Jingchun Cheng[1,2] Sifei Liu[2] Yi-Hsuan Tsai[2] Wei-Chih Hung[2] Shalini De Mello[3]
Jinwei Gu[3] Jan Kautz[3] Shengjin Wang[1] Ming-Hsuan Yang[2,3]
[1]Tsinghua University [2]University of California, Merced [3]NVIDIA Research

Abstract

We propose a deep learning-based framework for instance-level object segmentation. Our method mainly consists of three steps. First, We train a generic model based on ResNet-101 for foreground/background segmentations. Second, based on this generic model, we fine-tune it to learn instance-level models and segment individual objects by using augmented object annotations in first frames of test videos. To distinguish different instances in the same video, we compute a pixel-level score map for each object from these instance-level models. Each score map indicates the objectness likelihood and is only computed within the foreground mask obtained in the first step. To further refine this per frame score map, we learn a spatial propagation network. This network aims to learn how to propagate a coarse segmentation mask spatially based on the pairwise similarities in each frame. In addition, we apply a filter on the refined score map that aims to recognize the best connected region using spatial and temporal consistencies in the video. Finally, we decide the instance-level object segmentation in each video by comparing score maps of different instances.

1. Introduction

In this work, we focus on the problem of multiple instance segmentation in videos. Specifically, given each object mask in the first frame, we seek to predict segmentations for this instance throughout the video sequence. The task is challenging when dealing with non-rigid objects (e.g., human, animals) because these objects often have their individual movements with various perspectives, poses. Occlusions can also pose significant challenges for tracking based methods since the foreground objects could be fully occluded in some frames. With the multiple instance setting, occlusions between different instances also introduce further difficulties to keep tracking each instance separately.

Most state-of-the-art approaches tackle the problem with convolutional neural networks (CNNs) [13, 2]. Intuitively, CNNs are trained to output the foreground/background segmentation maps following the structure of the fully convolution networks (FCN) [17] for every frame in the video sequence. In the unsupervised setting, a general foreground model is learned with the training set. Under the semi-supervised setting, one can further fine-tune the model using the segmentation mask in the first frame of the test video to focus on the particular foreground region. To extend this pipeline to the multiple instance setting, we decompose this task into foreground segmentation and instance recognition. While foreground segmentation can be trained on the training set with respect to all foreground objects, instance recognition can be trained on each specific instance to separate the foreground mask into multiple instances.

We observe that similar to most FCN based segmentation methods [17, 3, 33], segments generated by the network are often not aligned to the actual object boundaries because of the pooling operations during forward propagation. To address this issue, many existing methods on image-level semantic segmentation task apply the conditional random field (CRF) as the post-processing module to refine object boundaries [3, 33]. However, densely connected CRF requires sophisticated designs of potential functions and fine-tuned hyper-parameters. There are end-to-end trainable CRFs such as [33], but they often introduce much memory and computational overhead.

To address this issue, we model the boundary refinement task as a spatial propagation problem with pixel-wise affinity prediction. Specifically, we propose a spatial propagation network (SPN) that propagates the segmentation probabilities using the learned pixel-wise affinity as guidance with a linear 2D propagation module. To further refine the segments that are not consistent in the temporal domain, we propose the connected region-aware filter (CRAF) to eliminate inconsistent labels. Figure 1 illustrates the overview of the proposed algorithm.

To evaluate the proposed methods, We carry out extensive experiments and ablation studies on the DAVIS 2017 challenge dataset [23]. We show that the proposed SPN improves the object boundaries, while the proposed CRAF eliminates segments with inconsistent instance labels.

The contributions of this work are summarized below:

- We extend the segmentation network to handle multiple instances simultaneously by decomposing the task into foreground segmentation and instance recognition.
- We propose the spatial propagation network to refine object segments through learning the spatial affinity.
- We develop the connected region-aware filter to eliminate inconsistent segments.

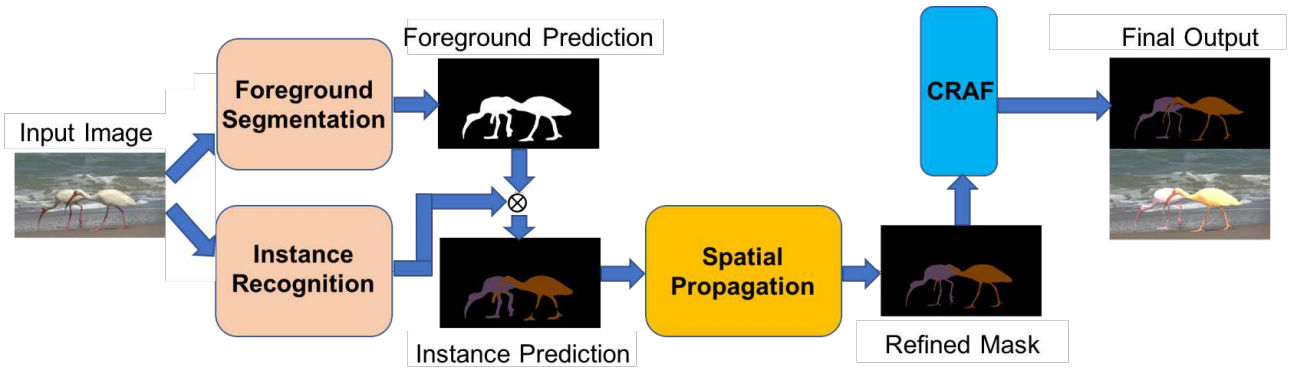

Figure 1. Framework of the proposed method.

2. Related Work

Video Object Segmentation. There are two problem settings for video object segmentation: unsupervised and semi-supervised ones. Unsupervised methods aim to segment foreground objects without any knowledge of the object during testing (e.g., an initial object mask). Several methods have been proposed to generate object segmentation via superpixel [10, 30, 8], saliency [24, 7, 28], or optical flow [1, 20]. To incorporate higher level information such as objectness, object proposals are used to track object segments and generate consistent regions through the video [14, 15]. However, these methods usually have heavy computational loads to generate region proposals and associate thousands of segments, making such approaches only feasible to offline applications.

Semi-supervised approaches [9, 31, 19] assume an object mask in the first frame is known, and the objective is to track the object mask throughout the video. To achieve this, existing approaches focus on propagating superpixels [12], constructing graphical models [18, 27] or utilizing object proposals [22]. Recently, CNN based methods [13, 2] combine offline and online training on static images.

Instance Segmentation. Our work is also related to instance segmentation in the image level, especially to the subtasks including occlusion handling and boundary refinement. Most state-of-the-art approaches tackle this task using region proposals [25] followed by object mask prediction. In [5], a multiple stage network is used to predict bounding box proposals, mask proposals, and class score iteratively. However, the performance of instance segmentation often suffers from heavy occlusions. To handle occlusions, one can apply dense CRF on the patch level to generate instance masks [32]. In a non-parametric approach, exemplar segments from the training set are utilized to help the occlusion handling [4].

To further obtain boundary accuracies between instances, probabilistic models can be applied as a post-processing module to refine object boundaries and enforce spatial smoothness. A straightforward approach is to apply fully-connected CRF in the testing phase [3]. In [33], CRF is formulated as an RNN, resulting in an end-to-end trainable network with the spatial smoothness constraint. In our work, the proposed SPN directly learns the pixel affinity in an end-to-end manner from the data itself. It results in a light-weighted, computationally efficient refinement module.

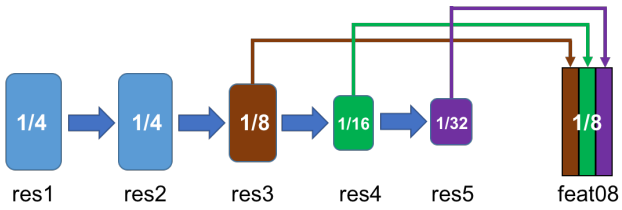

Figure 2. Network architecture for foreground segmentation and instance recognition. The first five blocks (res1 to res5) are adapted from the ResNet-101.

3. Learning to Segment Instances

Given the object mask of the first frame at the instance level, our goal is to segment this instance throughout the entire video. Toward this end, we first train a generic foreground/background segmentation model to localize objects and then fine-tune this generic model to learn instance-level models.

3.1. Foreground Segmentation

Inspired by the effectiveness of FCN in image segmentation [17] and the deep structure in image classification [11], we construct our foreground/background segmentation network based on the ResNet-101 architecture [11] with modifications for pixel-wise segmentation predictions as follows: 1) the fully-connected layer for classification is removed, and 2) features of convolution modules in different levels are fused together for obtaining more details during up-sampling.

The ResNet-101 has five convolution modules, where each of them consists of several convolutional layers. Specifically, we draw feature maps from the 3-th to 5-th convolution modules after the pooling operations, where these maps are with sizes of 1/8, 1/16, 1/32 of the input image size, respectively. Then these maps are up-sampled and concatenated together for predicting the final output (see Figure 2 for an illustration).

A pixel-wise cross-entropy loss with the softmax func-

tion $\mathbb{E}$ is used during optimization. To overcome imbalanced pixel numbers between foreground and background regions, we use the weighted version as adopted in [29], and the loss function is defined as:

$$\mathcal{L}_s(I_t) = -(1-w) \sum_{i,j \in fg} \log \mathbb{E}(y_{ij} = 1; \theta) \\ -w \sum_{i,j \in bg} \log \mathbb{E}(y_{ij} = 0; \theta), \quad (1)$$

where i, j denotes the pixel location of foreground fg and background bg, y_{ij} denotes the binary prediction of each pixel of the input image I at frame t, and w is computed as the foreground-background pixel-number ratio.

3.2. Instance-level Recognition

After discovering foreground segmentations, the next step is to further segment instance-level objects. To achieve this, we adopt the same model and loss function in (1) for foreground segmentation, and fine-tune it for instance segmentation. As a result, for each instance we train a model, where the softmax function in (1) has two channels for the object instance and the background.

Since each video may have multiple object instances and different instance-level models may not agree to each other, we develop a method to solve such confusions, e.g., two objects are close to each other. We compute a pixel-wise score map for each object from the output of the instance-level model, in which this score map indicates the likelihood of instance segmentation. To take advantage of the foreground model and reduce noisy segments, we also enforce the score map being non-zero only within the foreground segmentation. Once we have score maps from different instances, we determine the final instance-level segmentation results by labeling the one with the maximum score for each pixel.

3.3. Network Implementation and Training

To train the foreground generic model, we first use annotations from the DAVIS training set, and then fine-tune on foreground masks with augmentations in the first frame of the DAVIS test set. When training the foreground generic model, we use weights from ResNet-101 [11] as initializations. We use stochastic gradient descent (SGD) optimizer with batch size 1 and learning rate 1e-8 for 100,000 iterations. During training instance-level models, we then fine-tune this generic model by using augmented instance-level annotations on the test set. For instance-level models, we use batch size 1, starting from learning rate 1e-8 and decreasing it by half for every 10,000 iterations with a total of 30,000 iterations. Since the number of total training samples is relatively small, we adopt affine transformations (i.e., shifting, rotation, flip) to generate one thousand samples for each frame.

4. Mask Refinement

In this section, we refine the mask in a frame-wise manner. This is done by a spatial propagation network (SPN) that improves the object shapes from a coarse shape to a finer one under the guidance of the original frame, and a connected region-aware filter (CRAF) that eliminates inconsistent regions. We note that these two refinement processes are independent to instances, in which a learned SPN can be applied to any instances.

4.1. Spatial Propagation Network

The SPN contains a deep CNN that learns the affinity entities, and a spatial linear propagation module that refines a coarse mask. The coarse mask is refined under the guidance of the affinity, the learned pairwise relations, for any pairs of pixels. All modules are differentiable and jointly trained using the SGD method. The spatial linear propagation module is computationally efficient for inference due to the linear time complexity of the recurrent architecture.

Method. The SPN contains a propagation module that builds a learnable graph through linear propagation over a 2D map. Let $\mathbf{H} \in \mathbb{R}^{m \times n}$ denotes a propagation hidden layer on top of a $m \times n$ feature map, h_{ij} and x_{ij} be the pixel at (i, j) for the hidden layer and the feature map, respectively. We use $\left\{p_{ij}^K\right\}_{K \in \mathbb{N}(ij)}$ to represent a group of weights for (i, j), where K is a neighboring coordinate of (i, j), denoted as $\mathbb{N}(ij)$. The 2D linear propagation along one direction (e.g., left-to-right) is:

$$h_{ij} = \left(1 - \sum_{K \in \mathbb{N}(ij)} p_{ij}^K\right) x_{ij} + \sum_{K \in \mathbb{N}(ij)} p_{ij}^K h_K, \quad (2)$$

where h_K is an adjacent pixel of (i, j) in the hidden layer. Taking the example of the left-to-right direction, the neighborhood $\mathbb{N}(ij)$ contains three nodes $\{(i-1, j-1); (i-1, j); (i-1, j+1)\}$ from the previous column. Each p_{ij}^K represents the weight between two adjacent pixels. In this way, one direction of propagation enables each pixel to receive the information from a triangular 2D plane, where the integration of four different directions (e.g., top-to-bottom and the other two with the reverse directions) enables it to receive information from all the other pixels of the image/feature map. The propagation in (2) is performed as column-wise transitions, which can be expressed by the following linear operation:

$$H_i = (1 - P_{i-1,i}) X_i + P_{i-1,i} H_{i-1}, \quad (3)$$

where H_i and X_i are the i-th column for the hidden layer and the feature map, $P_{i-1,i} \in \mathbb{R}^{n \times n}$ is a linear transition matrix, and the key factor for the transport of information from column $i-1$ to i. Corresponding to the three-neighbor connection, $P_{i-1,i} \in \mathbb{R}^{n \times n}$ is a tridiagonal matrix, a simple form whose system stability can be easily controlled through the Gershgorin's theorem [34]. In the back propagation pass, the derivative $\sigma_{i+1,i}$ with respect to $P_{i+1,i}$ is

$$\sigma_{i+1,i} = \theta_i \cdot (H_{i+1} - X_i), \quad (4)$$

where θ_i is the error for H_i flowing back from the top layer.

We use the guidance network, a regular deep CNN with symmetric downsample and upsample parts, to output all the elements of $P_{i-1,i}$. The error signal flows in the reverse direction along the hidden layer and then propagates to the guidance network such that the entire network can be trained end-to-end.

Network Implementation. We describe the implementation of the SPN, which contains two separate branches: 1) a deep CNN based guidance network that outputs all elements of the transformation matrix, and 2) the linear propagation module that outputs the refined segmentations. In this work, we use the VGG-16 [26] pre-trained network from the *conv1* to *pool5* as the downsampling part of the guidance network. The upsampling part adopts the exactly symmetric architecture and is learned from scratch. The layers with the same dimension between the downsampling and upsampling part are connected with skipped links to leverage the features of different levels.

The guidance network typically takes in RGB images. It outputs all the connection weights w.r.t each pixel, where each has 3×4 parameters to learn. The propagation module takes a coarse segmentation mask produced by the previous step, and the weights generated by the guidance network. Suppose that we have a map of size $n \times n \times c$ that inputs into the propagation module, the guidance network needs to output a weight map with the dimensions of $n \times n \times c \times (3 \times 4)$, i.e., each pixel in the input map is paired with 3 scalar weights per direction, and 4 directions in total. The propagation module contains 4 independent hidden layers for different directions, where each layer combines the input map with its respective weight map using (2). Similar to [16], we use the node-wise max-pooling to integrate the hidden layers and obtain the final propagation result.

Network Training. There are two requirements to train the SPN. First, the SPN processes the two-class mask refinement. Second, for each training image with the ground truth annotation, a coarse mask is required. Therefore, we train our SPN on the training set of PASCAL VOC 2012 [6], where the coarse mask is generated by the FCN [17]. For each image, we randomly pick a valid label according to the annotations, while treating all the other pixels as the background, in order to generate a two-class training sample out of the original 21 classes. During training, we randomly crop 256×256 square patches from the image, the binary label, and the coarse mask. We note that there is only one single SPN as a general refinement module, no finetuning is carried out on any frame from Davis 2017 dataset.

4.2. Connected Region-aware Filter

After applying the SPN, we observe that there exit many confusions between instances since we only do the segmentation in a one-shot manner without considering temporal information in other frames. To improve the instance mask, we use a connected region-aware filter (CRAF), which considers the consistency between two frames and helps rectify some instance confusions (see column (b) and (c) in Figure 3 for illustrations).

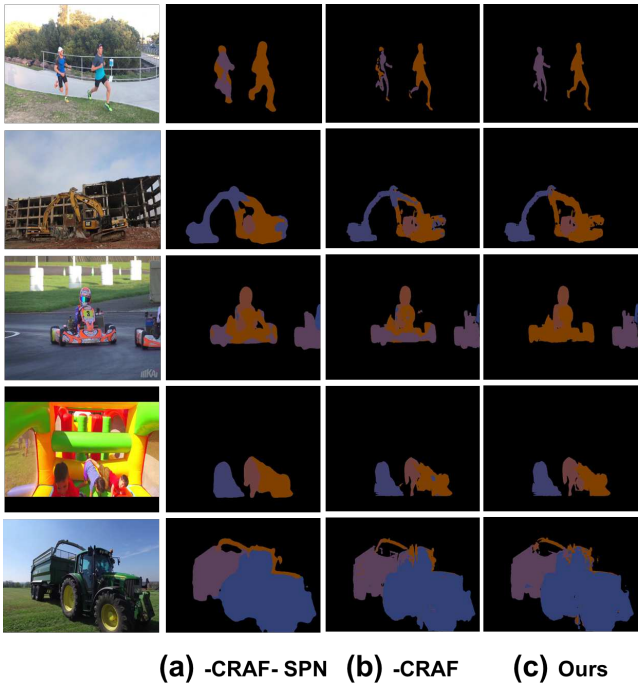

Figure 3. Segmentation results after applying each step. (a) is the result without using any refinements, while (b) and (c) are the results without using CRAF and our final outputs, respectively.

In CRAF, we select a best connected region for each object using the following criteria.

Step 1. For an object i, we extract connected regions $(CR_1, CR_2, ...)$ on its score map after applying the SPN. Then a jaccard similarity $(J_1, J_2, ...)$ of each connected region compared to the region in the previous frame is computed, as well as the area of each region $(A_1, A_2, ...)$. If $J_m = max(J_1, J_2, ...)$ and $A_m / max(A_1, A_2, ...) > \alpha$, we pick CR_m as the best connected region for object i, denoted by CRS_i. If there is more than one jaccard similarities equal to the maximum value, we choose the connected region with the largest area to be CRS_i.

Step 2. We calculate the coverage rate between CRS of every two objects using the following formula:

$$Coverage(i, j) = |CRS_i \cap CRS_j| / |CRS_j|. \quad (5)$$

If $Coverage(i, j) > \beta$, we remove the regions of CRS_j within CRS_i, i.e., $CRS_i = CRS_i - CRS_j$.

Step 3. For an object i, if the region of $CRS_i < \gamma$, we check all the connected regions that are not selected as CRS. Then, if one of these regions has a jaccard similarity greater than δ, we pick this connected region as CRS_i and repeat Step 2.

Table 1. Comparisons of instance segmentation models.

Method	Global Mean	J Mean	J Recall	F mean	F Recall
Per-video model	0.460	0.442	0.513	0.478	0.501
Per-object model	0.481	0.457	0.536	0.504	0.526

In this work, $\alpha, \beta, \gamma, \delta$ are empirically set as 0.2, 0.9, 0.1, 0.4 respectively. Note that, we set the scores outside each selected connected region as zero. To obtain the final instance segmentation, we determine the label for each pixel by considering the maximum score from all instances.

5. Experiments

5.1. Dataset and Evaluation Metrics

The DAVIS benchmark [23, 21] is a recently-released high-quality video object segmentation dataset. It consists of videos with multiple objects and instance-level pixel-wise annotations. In total, there are 150 sequences (60 in training set, 30 in each of the validation, test-dev and test-challenge sets), with 10459 annotated frames and 376 objects.

We first use the training set to train our models and evaluate on the validation set. The best models are then trained on training and validation set and tested on the 2017 DAVIS Challenge for competition. The challenge uses the mean of region similarity (J mean) and contour accuracy (F mean) over all object instances as the performance metrics. The same algorithm (evaluation code from the DAVIS website) is used in the validation set to validate our method.

5.2. Comparisons of Instance-level Recognition

We compare two different models for instance-level segmentation: per-video and per-object settings. Initialized from weights of the foreground model, the per-video object recognition network has a softmax layer with a N+1 dimensional score map as the output, where N denotes the number of objects in the video with an additional one for the background. Each score map denotes the probability of one object. In prediction, the pixel that has the maximum score is considered as the label of that pixel.

For the per-object model, only one object is considered as the foreground each time. The network for each per-object model has a 2-dimensional output that contains the background and one object instance. In prediction, score maps of different objects are concatenated together, and if the maximum score is below 0.5, the pixel belongs to the background. Otherwise, the object label of each pixel is determined by the instance with the maximum score.

Comparisons of these two methods are shown in Table 1, where the models are trained on the training set and tested on the validation set. The results show that the per-object model outperforms the per-video model by 2.1% in Global Mean. Therefore, we choose to use the per-object model for the following experiments.

Table 2. Ablation study on the DAVIS validation set. We show comparisons of our results with different components removed, i.e., foreground model fine-tuning (FT), spatial propagation network (SPN), connected region-aware filter (CDAF).

Method	Ours	-CRAF	-CRAF-SPN	-CRAF-SPN-FT
J Mean	0.540	0.506	0.457	0.442
J Recall	0.601	0.582	0.536	0.528
F Mean	0.611	0.568	0.504	0.453
F Recall	0.683	0.602	0.526	0.501
Global Mean	0.576	0.537	0.481	0.448

Table 3. Runtime Analysis. We show the runtime in foreground segmentation (FS), instance-level recognition (IR), spatial-propagation network (SPN) and connected region-aware filter (CRAF). The runtime is calculated on average over all frames and objects.

step	FS	IR	SPN	CRAF
test time	0.3s	0.3s	0.08s	0.1s

5.3. Ablation Study

To analyze the necessity and importance of each step in our proposed method, we carry out extensive ablation studies on the validation set. Results are summarized in Table 2. We validate our method by comparing our final results to the ones without fine-tuning (**-FT**), spatial propagation network (**-SPN**), and connected region-aware filter (**-CRAF**). The detailed settings are as follows:
-FT: train the foreground segmentation model without fine-tuning on first frames.
-CRAF: apply the SPN after instance segmentation without the use of the connected region-aware filter.
-SPN: the results from instance segmentation network without using spatial propagation network for refinement.

Table 2 shows that the SPN and CRAF post-processing steps play an important role in generating better results, and improve the Global mean by 5.6% and 3.9% respectively. It also demonstrates that the foreground mask prediction network needs a fine-tuning step on the first frame for more accurate segmentations on the specific object (-CRAF-SPN vs. -CRAF-SPN-FT). Some example results after applying different steps are shown in Figure 3.

5.4. Runtime Analysis

For the model trained offline, the proposed method runs at the speed of 0.78 seconds per object per frame on a Titan X GPU with 12 GB memory. Detailed analysis in each step is shown in Table 3. When taking the fine-tuning time into account, our system runs at about 10 seconds per frame per object on the DAVIS validation set. The table shows that the SPN and CRAF steps improve performance significantly without adding much computational costs.

Table 4. Overall results on the DAVIS 2017 Challenge

Method	Ours (cjc)	lixx	apdata	vantam299	haamooon	voigtlaender	lalalafine123	YXLKJ	wasidennis	Fromandtozh
Global Mean	0.569	0.699	0.678	0.638	0.615	0.577	0.569	0.558	0.548	0.539
J mean	0.536	0.679	0.651	0.615	0.598	0.548	0.548	0.538	0.516	0.507
F Mean	0.602	0.729	0.706	0.662	0.632	0.605	0.591	0.578	0.579	0.571

5.5. Results in DAVIS 2017 Challenge

In the DAVIS 2017 challenge, we test our method without and with CRAF on the test-challenge set. Without CRAF, the J mean and F mean are 51.6% and 57.9%, and they are improved by 2% and 2.3% with CRAF. The final comparisons of top-10 teams are listed in Table 4. As shown in the table, our method is at the 6th place in the competition.

6. Conclusion

In this work, we propose to use the spatial propagation network (SPN) and connected region-aware filter (CRAF) to refine the instance segmentation in both spatial and temporal domains. We show that on the challenging DAVIS 2017 dataset, the proposed methods achieve competitive performance for multiple instance segmentations in videos.

References

[1] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 2

[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. 1, 2

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. 2015. 1, 2

[4] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, 2015. 2

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 2

[6] M. Everingham, S. A. Eslami, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 4

[7] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 2

[8] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012. 2

[9] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013. 2

[10] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[12] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 2

[13] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 1, 2

[14] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 2

[15] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2

[16] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016. 4

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 4

[18] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 2

[19] N. S. Nagaraja, F. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*, 2015. 2

[20] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 2

[21] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 5

[22] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *CVPR*, 2015. 2

[23] 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:1704.00675*, 2017. 1, 5

[24] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, 2010. 2

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556:1187–1200, 2014. 4

[27] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 2

[28] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 2

[29] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[30] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012. 2

[31] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 2

[32] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation with deep densely connected mrfs. In *CVPR*, 2016. 2

[33] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015. 1, 2

[34] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber. Recurrent highway networks. *arXiv:1607.03474*, 2016. 3