Automatic Real-time Background Cut for Portrait Videos

Xiaoyong Shen Ruixing Wang Hengshuang Zhao Jiaya Jia The Chinese University of Hong Kong

{xyshen, rxwang, hszhao, leojia}@cse.cuhk.edu.hk

Abstract

We in this paper solve the problem of high-quality automatic real-time background cut for 720p portrait videos. We first handle the background ambiguity issue in semantic segmentation by proposing a global background attenuation model. A spatial-temporal refinement network is developed to further refine the segmentation errors in each frame and ensure temporal coherence in the segmentation map. We form an end-to-end network for training and testing. Each module is designed considering efficiency and accuracy. We build a portrait dataset, which includes 8,000 images with high-quality labeled map for training and testing. To further improve the performance, we build a portrait video dataset with 50 sequences to fine-tune video segmentation. Our framework benefits many video processing applications.

1. Introduction

Portrait image and video have become conspicuously abundant with the popularity of smart phones [57]. Portrait segmentation thus plays an important role for post-processing such as composition, stylization and editing. High performance automatic portrait video segmentation remains a difficult problem even with recent development on automatic portrait image segmentation and matting [57, 58]. We in this paper tackle this problem starting from following analysis.

Tedious Interaction Problem Previous methods [64, 50, 66, 15] need users to specify samples in key frameworks using strokes and iteratively refine segmentation with more touch-ups, which are actually labor intensive. We tested the best implementation of Rotobrush in Adobe After Effect and eventually used one hour to well segment a one-minute video sequence as shown in Figure 1(a). It would take more time when the video is with more complicated background or along hair boundaries. Thus, improving segmentation efficiency is of great practical value for video processing.

Time and Accuracy of Automatic Methods Although many automatic semantic image segmentation methods

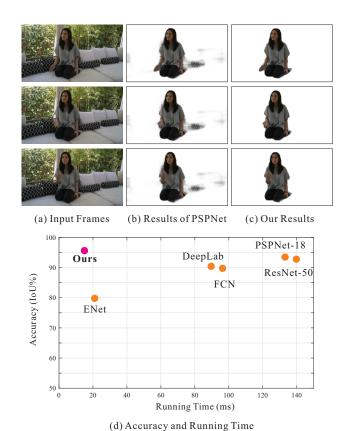


Figure 1. Our automatic real-time background cut method. (a) is input frames in a portrait video. (b) and (c) show the results of state-of-the-art method [74] and ours respectively. (d) shows the accuracy and running time of different segmentation approaches

[39, 8, 74] were developed, they are not real-time processing methods even using GPUs for median-resolution videos, which hinder them from applying to batch and online video editing. As shown in Figure 1(d), representative methods FCN [39], Deeplab [8], ResNet [23] and PSPNet [74] need at least 90ms to process one frame for 720p videos on an Nvidia Titan X GPU card. Fast semantic segmentation such as ENet [47], on the other hand, can only produce lower-quality results.

on our portrait video segmentation dataset.

The major difficulty of accurate automatic background cut stems from the diverse complexity of patterns in foreground and background. It is common that background patterns are brought into foreground estimate even with the powerful PSPNet [74] due to the color-texture similarity. One example is shown in Figure 1(b), where the appearance of the pillow is similar to the foreground patterns and thus misclassified. Given the very high diversity of background patterns in indoor and outdoor scenes, it is challenging to reduce this type of misclassification to a very low level. One focus of this paper is therefore to address this issue.

Our Contributions We propose an automatic real-time portrait video segmentation method, first addressing foreground and background structure ambiguity via deep background attenuation, which incorporates extra video background feature learning to help segmentation. As shown in Figure 1(b) and (c), this background attenuation scheme can greatly reduce boundary and regional errors.

Second, we design a spatial-temporal video segmentation refinement module to efficiently improve results by considering video spatial and temporal coherence. Our framework is an end-to-end trainable convolutional neural network (CNN). We further design Light-ResNet to achieve the real-time performance, which is over 70 frames/second for testing 720p videos on an Nvidia Titan X card. State-of-the-art results are achieved in terms of running time and accuracy as illustrated in Figure 1(d).

To train and evaluate our framework, a portrait segmentation dataset with 8,000 images whose size are 1200 * 800 are built. We also collect a 50 portrait video segmentation sequences, each with 200 frames. The two datasets are labeled with high-quality segmentation maps. Our high performance portrait segmentation system can be the fundamental tool for video processing and benefit all applications requiring high-quality segmentation maps.

2. Related Work

We in this section review the most related image and video segmentation schemes.

Graph based Object Segmentation Approaches Image and video object segmentation can be a graph-model based problem. For image segmentation, many methods are based on the graph-cut schemes [4], such as Lazing Snapping [34], Grabcut [53], and Paint Selection [36]. These methods need user interaction to specify different object samples. Besides the graph-cut framework, dense CRF is also applied to object segmentation as explained in [29].

Image segmentation can be directly extended to videos by considering the temporal pixel/object correspondence. Most of the methods pay attention to how to build graph models [67, 37, 70]. Approaches of [24, 30, 20, 13, 41,

68, 11, 76, 71, 49] introduced different schemes to estimate class object distributions. The geodesic distance [3] was used to model the pixel relation more accurately. To efficiently solve graph based models, the bilateral space is applied in [42]. Energy or feature propagation schemes were also presented in [54, 77]. To reduce user interaction, Nagaraja *et al.* [44] proposed a framework that only needs a few strokes and Lee *et al.* [31] found key-segments automatically.

Temporal coherence is another important issue in video segmentation. Optical flow [61, 27], object/trajectory tracking [18, 5], parametric contour [40], long/short term analysis [46, 24], etc. are applied to address the temporal coherence issue. Many previous methods handle bilayer segmentation [12]. Tree-based classifier was presented in [69] and locally competing SVMs were designed in [21] for better bilayer segmentation. To evaluate video segmentation quality, benchmarks [48, 19] were proposed. Compared with these graph based methods, our method is real-time and without any interaction.

Learning based Semantic Segmentation Previous work focus in part on learning feature for video segmentation. Price *et al.* [50] learned multiple cues and integrated them into an interactive segmentation system. Tripathi *et al.* proposed learning early- and mid-level features to improve performance. To handle training data shortage, weakly-supervised and unsupervised learning frameworks were developed in [63], [73] and [72] respectively. An one-shot learning method was proposed in [6] only needing one example for learning. Drayer *et al.* proposed a unified framework including object detection, tracking and motion segmentation for object-level segmentation. To reduce errors during propagation, Wang *et al.* [66] developed segmentation rectification via structured learning.

In recent years, CNNs have achieved great success in semantic image segmentation. Representative work exploited CNNs in two ways. The first is to learn important features and then apply classification to infer pixel labels [2, 43, 16]. The second way is to directly learn the model from images. Long *et al.* [39] introduced fully convolutional networks. Following it, DeepLab [8] and CRFasRNN [75] were developed using CRF for label map refinement. Recent PSPNet [74] is based on ResNet [23], which performs decently.

These frameworks can be directly applied to videos in a frame-by-frame fashion. To additionally deal with temporal coherence, spatial-temporal FCN [17] and recurrent FCN [62, 59, 45] were proposed. Shelhamer *et al.* [56] proposed Clockwork Convnets driven by fixed or adaptive clock signals that schedule processing of different layers. To use the temporal information, Khoreva *et al.* [28] predicted per-frame segmentation guided by the output of previous frameworks. These approaches aim at general object segmentation. They have difficulty to achieve real-time per-



(a) Input Frame

rame (b) Result of PSPNet









(c) Material Ambiguity

(d) Appearance Ambiguity







(e) Complex Motions

Figure 2. Difficulty of portrait video segmentation. (a) is an input frame and (b) shows the segmentation result by state-of-the-art method [74]. (c) and (d) show the ambiguities stemming from material and appearance similarity. (e) shows the complex motion in portrait videos.

formance for good quality portrait video segmentation.

Video Matting Schemes Similar to image matting, video matting computes the alpha matte in each frame. A survey of matting techniques can be found in [65] and an evaluation benchmark is explained in [14]. Most video matting methods extend the image one by adding temporal consistency. Representative schemes are those of [78, 55, 32, 9, 3, 1, 10]. Since the matting approaches need user specified trimaps, methods of [26, 22] applied segmentation to improve trimap quality. Our method automatically achieves portrait segmentation and generates trimaps for further video matting.

3. Our Framework

Our end-to-end trainable framework is illustrated in Figure 3. It addresses two main challenges. The first is the ambiguity between foreground and background patterns. As shown in Figure 2, material and appearance in (c) and (d) are very similar, making even state-of-the-art semantic segmentation method [74] fail to cut out foreground accurately. We design a deep background attenuation model to address this challenge.

The second challenge is on complex motion as shown in Figure 2(e) that may cause correspondence estimation to fail. Also, fast motion could blur the content. We address this challenge with a spatial-temporal refinement module. These modules are implemented with the in-depth consideration of short running time and high quality.

As shown in Figure 3, our framework takes successive 2n+1 frames $\{I^{t-n}...I^{t+n}\}$ as input and outputs the seg-

mentation map of I^t ,where I^t denotes the tth frame of the video.

3.1. Global Background Attenuation

Our global background attenuation is to collect a few background samples around the same scene without the requirement of alignment or fixing cameras, and use them to globally attenuate background pixels. It is a rather easy setting taking a few seconds prior scene capture for following live video processing. The background samples can be also got by manually cropping out the video background. It is different from the method of [60] where the latter requires stationary background.

Segmentation with Global Attenuation After collecting a few frames shooting the background from arbitrary locations and angles, we process them through a network. As shown in Figure 3, our overall network includes two paths. The first extracts feature maps of the input frames $\{I^{t-n}...I^{t+n}\}$ and the second path computes features of the background samples. Both of branches are specially designed light-weight ResNet, which we will detail later. Similar to previous segmentation frameworks [8, 74], the extracted feature maps in our first path are further processed by one convolution layer to obtain the score maps, corresponding to the segmentation result after Softmax operation as illustrated in Figure 3.

Our second path plays the role of background attenuation. Since these background samples are not aligned with the input frames, directly apply extracted background feature to attenuate the segmentation feature maps is vulnerable to errors caused by object/region discrepancy. We address this issue by estimating the global background features. It starts from the extracted background feature by adding global average pooling and upsampling. The upsampling step makes global feature map keep the same size as the inputs. The final background global features are concatenated with the segmentation feature maps from the first path as illustrated in Figure 3.

The features adopted in previous segmentation frameworks [38, 74] are used to enlarge the receptive fields. They are similar to our first path. In contrast, the way to extract global features for background samples for our special task of attenuation is new and empirically effective.

Effectiveness of Our Attenuation Our background attenuation can quickly reduce segmentation errors. To demonstrate it, we shown an example in Figure 4 where foreground and background both have clothes for segmentation. Directly applying the segmentation CNNs in our upper branch in Figure 3 cannot estimate correct foreground – results are shown in (b). With our background attenuation, the network outputs much higher quality results as shown in (c) and the background samples are shown in (a)

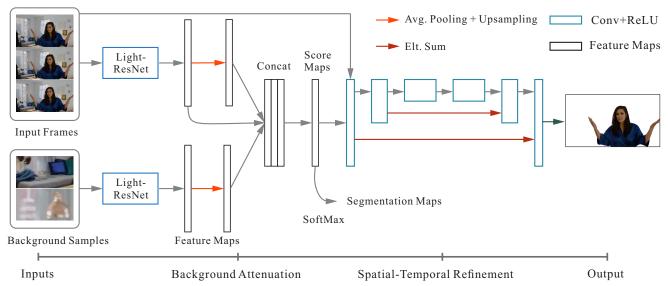


Figure 3. Our automatic high-quality real-time portrait video segmentation framework.



Figure 4. Effectiveness of our background attenuation. (a) Input frames. (b)-(c) Results without and with background attenuation respectively. Background samples are inside rectangles in (a).

in the highlighted rectangles. Note that in this example the video background is not stationary and not aligned with input images. Yet its usefulness in our framework is clearly exhibited.

Light-ResNet for Real-time Performance To achieve real-time performance, we designed a Light-ResNet to balance accuracy and efficiency as illustrated in Figure 3. Our network is based on ResNet-18 [23] and goes through model compression to accelerate the network. Since the running time bottleneck is mostly related to the large number of output channels of convolution layers, we prune convolution filters in each layer [33]. We gradually reduce the low-response filters and fine-turn the compressed model.

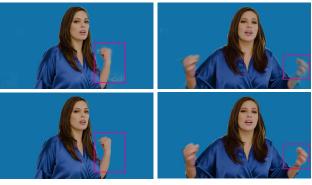
For each step, we keep 90% of filters based on previous fine-turned model and finally retain only 20% of filters compared to the original ResNet-18 model. Our system only needs 14.99ms to test one 720p color image, while the original structure takes 110.91ms. The accuracy is not much sacrificed on our testing dataset, discussed later in experiments. We note that gradually pruning filters is essential for balancing performance and speed.

3.2. Spatial-temporal Refinement

To enforce temporal coherence and deal with remaining spatial segmentation errors, we propose the spatial-temporal refinement network as shown in Figure 3 to further improve result quality.

Network Design In our refinement network, the input is the segmentation score maps of the input frames $\{I^{t-n}...I^{t+n}\}$. The original images are taken as guidance for refinement. The output is updated segmentation of frame I^t as shown in Figure 3. We first utilize three convolution layers to extract spatial-temporal features [39, 52]. Considering inevitable displacement, we use stride 2 for downsampling to reduce this effect. Then three corresponding deconvolution layers are employed to upsample previously downsampled feature maps. We further fuse the same-size feature maps between feature extraction and deconvolution layers by summation to improve results (Figure 3).

Compared with previous refinement network [35], which did not apply downsampling or pooling operations to preserve image details, ours applies sampling to address the displacement problem and accelerate computation. Moreover, our network does not need to regress image details.



(a) Small Motion

(b) LargeMotion

Figure 5. Effectiveness of refinement using our spatial-temporal networks. (a) shows the refinement on a small motion case and (b) is with large motion. The top row shows results without refinement and the bottom includes our final results.

Analysis of the Refinement Our spatial-temporal refinement can further uplift edge accuracy and reduce small-segment errors as shown in Figure 5. For small-motion cases shown in (a), our refinement sharps edges for better quality. Our network has the reasonable ability to handle large motion as shown in Figure 5(b). The reason is that the input color images are applied as guidance in our network for it to learn the way to combine temporal and spatial information.

4. Model Implementation

Our network can be trained via an end-to-end scheme. In addition to the details below, more are included in our supplementary file.

4.1. Data Preparation

We prepare two datasets to train our model. The first is the portrait image segmentation dataset, which includes 8,000 portrait images, each with labeled masks. The second dataset is with 50 portrait sequences, each with 200 frames and labels. For the first dataset, we split the data into training and testing sets with 7,670 and 330 images respectively. We also use 45 portrait videos in training and 5 in testing.

Examples of our portrait images are shown in Figure 6. We first collect a large number of images from Flickr by searching keywords "portrait", "human", "person", etc. Then we process each downloaded image using the person detector [51] to crop out persons and adjust each image to resolution 1200×800 as shown in Figure 6. Finally, we consider the variety in person age, appearance, pose, accessories etc. and keep the most diverse 8,000 portraits in our final dataset. To get the segmentation ground truth, we label each portrait by the Adobe Photoshop quick selection tool.

The portrait videos are from 5 different places with 10 people inside. Complex poses are required for each person.



Figure 6. Data examples in our portrait dataset.

We also captured the background samples for our model training – each video is only with 20 frames of background *not* aligned with any following person-involved frames. We labeled each video using Adobe After Effect's Rotobrush by iteratively refining each frame. Examples are included in our supplementary file.

The reason that collect two datasets is to cover diverse persons and have data to learn temporal coherence. We note it is still very difficult to collect and label a large portrait video dataset covering a greater quantity of persons.

4.2. Training and Testing Details

We train our model using the two datasets and implement our model using Caffe [25] with one Nvidia Titan X GPU card. We train our Light-ResNet using the 7,670 portraits. We only apply the Softmax loss similar to semantic segmentation [74]. During training, the batch size is set to 16 and each sample is randomly cropped to 569×569 . The initial learning rate is set to 1e-3. We change it using the "poly" policy with gamma 0.9. 40 epochs are conducted for our model training.

Then, we train the whole network by the portrait video data – the 45 sequences include 9,000 frames and also corresponding separate unaligned background samples. The two Light-ResNet modules are initialized by our previously trained Light-ResNet model. The Softmax loss is added similar to the first step and the L_2 loss is added for the refinement network. For each input frame, we randomly select one background sample and we set the batch size to 16 and n to 2 – it means we use five frames for each-pass spatial-temporal refinement during training. The learning rate in the refinement network is set to 10 times of the Light-ResNet because no pre-trained weights are provided. The base learning rate and the changing policy remain the same.

To test our model, we first pre-compute the background global features using the provided background samples.

Methods	Accuracy (Mean IoU%)
FCN [39]	91.62
DeepLab [8]	91.59
PSPNet-50 [74]	93.51
PSPNet-18 [74]	93.33
ENet [47]	82.58
Ours w/o Atten. w/o Refin.	93.04
Our with background training	94.13
Ours with Atten. w/o Refin.	96.49
Ours	96.74

Table 1. Comparisons of different video segmentation methods on our portrait video dataset.

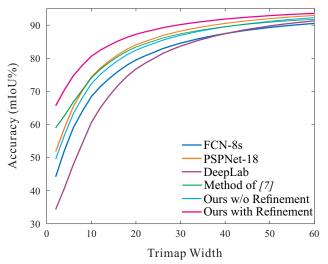


Figure 7. Comparisons of boundary accuracy of different methods.

Then we test the neighboring five frames as a batch using the background attenuation model and finally we apply the spatial-temporal refinement via sliding windows. This scheme makes each frame only needs to be computed once. For the 720p video, our whole network only takes 17.57ms where 14.99ms is for the background attenuation segmentation model and 2.58ms is for spatial-temporal refinement. All our experiments are conducted with one graphics card as explained above.

5. Evaluation and Applications

We in this section evaluate our methods and also show the applications.

5.1. Evaluations and Comparisons

Portrait Video Segmentation Evaluation We first compare different methods on our portrait video segmentation dataset. The mean Interaction-over-Union (IoU) metric is applied to measure accuracy on our 1,000 video frames on our testing data. We compare our method with the representative semantic segmentation methods of FCN [39],

Methods	Accuracy (Mean IoU%)	Time (ms)
FCN [39]	89.68	94.25
DeepLab [8]	89.72	90.61
PSPNet-50 [74]	93.56	139.7
PSPNet-18 [74]	92.81	110.9
ENet [47]	79.86	21.00
Our Light-ResNet	91.34	14.99

Table 2. Evaluations of the Light-ResNet on our portrait segmentation dataset.

DeepLab [8], PSPNet [74] and ENet [47]. For each method, we re-train the portrait segmentation model and fine turn it from corresponding public models.

The number of output channels is set to 2 for the twoclass segmentation problem. All models are trained with our portrait image segmentation and portrait video segmentation datasets that contain 7,670 portrait images and 9,000 video frames respectively. The training policy follows the original paper. We test each method on our video testing sequences frame-by-frame and the results are reported in Table 1.

Among all these methods, state-of-the-art semantic segmentation PSPNet [74] achieves the best performance on our portrait video segmentation dataset, complying with results on general object segmentation on datasets of ImageNet and Cityscapes. Because of simplicity of the model, ENet [47] does not perform similarly well.

We then evaluate our method based on results reported in Table 1. Our system achieved the best performance. High importance of background attenuation and spatial-temporal refinement is also revealed from Table 1. The background attenuation can greatly improve the quality because it can reduce errors caused by the ambiguity between foreground and background.

We note that directly adding the background images for model training can only yield improvement of about 1% IoU. Compared with our attenuation that improves 3.7% IoU, the straight-forward background sample training is obviously not optimal. Our spatial-temporal refinement further improve the accuracy. Since the improvement is mainly on edges, it cannot be accurately measured by IoU and we analyze it more below.

Effectiveness of the Spatial-temporal Refinement Our spatial-temporal refinement plays an important role to improve boundary accuracy for video segmentation. To evaluate it, we compute the IoU only near the ground truth boundary. Similar to [7], we generate different width trimaps centered at the ground truth boundary and then compute the mean IoU only on the trimap in our portrait video segmentation testing datasets. The changes between the mean IoU and the trimap width of different methods are shown in Figure 7.



Figure 8. Comparison of different methods. (a) is the input and (b) is the result of FCN [39]. (c) is the result from DeepLab [8] and (d) is computed by PSPNet [74]. (e) is ours.

Since the methods FCN [39], DeepLab [8], PSPNet [74] and ours without refinement do not have special boundary post-processing, the boundary accuracy is not that satisfying. Method of [7] learns domain transform filter to refine the score map in the network and thus improves boundary accuracy. As shown in Figure 7, our method with spatial-temporal refinement can effectively improve accuracy near boundary.

Evaluation of the Light-ResNet for Portraits We compare our Light-Resnet for segmentation with representative segmentation schemes of FCN [39], DeepLab [8], PSPNet [74] and ENet [47]. Similar to the evaluation in portrait video segmentation dataset, we first update the network to binary-label output and use our portrait image dataset to train it. Each method is trained using the originally released code following authors instruction.

The accuracy and running time of each method are reported in Table 2. Methods of FCN and DeepLab take over 90ms to test an image and the mean interaction-over-union (IoU) is near 90%. Although the PSPNet achieved the best performance, its running time is over 110ms. The ENet is very efficient; but the segmentation accuracy is an issue.

As shown in Table 2, our method only takes 14.99ms to test a color image with size 1200×800 and the accuracy is comparable to state-of-the-art PSPNet on our portrait image testing dataset.

Visual Comparisons We show more comparisons in Figures 8 and 9. The two examples are complex scenes. FCN [39], DeepLab [8] and PSPNet [74] results shown in (b-d) cannot well distinguish between foreground and background in pixel level. Ours with background attenuation can well cut out foreground. Background samples we apply for attenuation is the background of the first frame. As shown in (e), the spatial-temporal refinement further improves boundary accuracy. More results are shown in our supplementary material.

5.2. Other Applications

Our method provides the suitable solution to edit portrait video background in real-time. In Figure 10, we show an example for automatically video background change. With the high-quality segmentation map, the portrait can be blended into new background images. More applications such as video stylization, depth-of-field are exhibited in our



Figure 9. More comparisons. (a) is the input and (b) is the result of FCN [39]. (c) is the result from DeepLab [8] and (d) is computed by PSPNet [74]. (e) is ours.

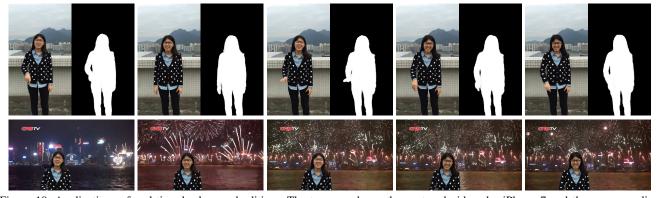


Figure 10. Applications of real-time background editing. The top row shows the captured videos by iPhone 7 and the corresponding segmentation maps by our method. The bottom row is the blending results on new background.

supplementary material.

6. Conclusion

We have presented the automatic real-time background cut method for portrait videos. The segmentation quality is greatly improved by the deep background attenuation model and spatial-temporal refinement. The limitations of our methods are that our approach may fail for severe blur caused by fast motion and very dynamic background. Addressing these limitations will be our future work.

References

[1] N. Apostoloff and A. W. Fitzgibbon. Bayesian video matting using learnt image priors. In *CVPR*, pages 407–414, 2004.

- [2] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, pages 3378–3385, 2012.
- [3] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal on Computer Vision*, 82(2):113–132, 2009
- [4] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, volume 1, pages 105–112, 2001.
- [5] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833–840, 2009.
- [6] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. One-shot video object segmentation. *CoRR*, abs/1611.05198, 2016.
- [7] L. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In CVPR, pages 4545–4554, 2016.
- [8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2014.
- [9] I. Choi, M. Lee, and Y. Tai. Video matting using multi-frame nonlocal matting laplacian. In ECCV, pages 540–553, 2012.
- [10] Y. Chuang, A. Agarwala, B. Curless, D. Salesin, and R. Szeliski. Video matting of complex scenes. In *ACM Trans. Graph.*, pages 243–248, 2002.
- [11] C. Couprie, C. Farabet, and Y. LeCun. Causal graph-based video segmentation. *CoRR*, abs/1301.1671, 2013.
- [12] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In CVPR, pages 53–60, 2006.
- [13] R. Cucchiara, A. Prati, and R. Vezzani. Object segmentation in videos from moving camera with mrfs on color and motion features. In CVPR, pages 405–412, 2003.
- [14] M. Erofeev, Y. Gitman, D. Vatolin, A. Fedorov, and J. Wang. Perceptually motivated benchmark for video matting. In *BMVC*, pages 99.1–99.12, 2015.
- [15] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. ACM Trans. Graph., 34(6):195:1–195:10, 2015.
- [16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1915–1929, 2013.
- [17] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, and R. Klette. STFCN: spatio-temporal FCN for semantic video segmentation. *CoRR*, abs/1608.05971, 2016.
- [18] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In CVPR, pages 1846–1853, 2012.
- [19] F. Galasso, N. S. Nagaraja, T. J. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, pages 3527–3534, 2013.
- [20] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato. Superpixel-based video object segmentation using percep-

- tual organization and location prior. In *CVPR*, pages 4814–4822, 2015.
- [21] M. Gong. Foreground segmentation of live videos using locally competing 1svms. In CVPR, pages 2105–2112, 2011.
- [22] M. Gong, Y. Qian, and L. Cheng. Integrated foreground segmentation and boundary matting for live videos. *IEEE Trans*actions on Image Processing, 24(4):1356–1370, 2015.
- [23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.
- [24] W. Jang and C. Kim. Streaming video segmentation via short-term hierarchical segmentation and frame-by-frame markov random field optimization. In *ECCV*, pages 599– 615, 2016.
- [25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In ACM MM, pages 675–678, 2014.
- [26] J. Ju, J. Wang, Y. Liu, H. Wang, and Q. Dai. A progressive tri-level segmentation approach for topology-changeaware video matting. *Comput. Graph. Forum*, 32(7):245– 253, 2013.
- [27] J. R. Kender and B. Yeo. Video scene segmentation via continuous video coherence. In CVPR, pages 367–373, 1998.
- [28] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. *CoRR*, abs/1612.02646, 2016.
- [29] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In NIPS, pages 109–117, 2011.
- [30] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In CVPR, pages 3168–3175, 2016.
- [31] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011.
- [32] D. Li, Q. Chen, and C. Tang. Motion-aware KNN laplacian for video matting. In *ICCV*, pages 3599–3606, 2013.
- [33] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *CoRR*, abs/1608.08710, 2016.
- [34] Y. Li, J. Sun, C. Tang, and H. Shum. Lazy snapping. ACM Trans. Graph., 23(3):303–308, 2004.
- [35] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video superresolution via deep draft-ensemble learning. In *ICCV*, pages 531–539, 2015.
- [36] J. Liu, J. Sun, and H. Shum. Paint selection. ACM Trans. Graph., 28(3), 2009.
- [37] S. Liu, G. Dong, C. H. Yan, and S. H. Ong. Video segmentation: Propagation, validation and aggregation of a preceding graph. In *CVPR*, 2008.
- [38] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. CoRR, abs/1506.04579, 2015.
- [39] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, pages 3431– 3440, 2015.
- [40] Y. Lu, X. Bai, L. G. Shapiro, and J. Wang. Coherent parametric contours for interactive video object segmentation. In CVPR, pages 642–650, 2016.

- [41] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In CVPR, pages 670–677, 2012.
- [42] N. Marki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In CVPR, pages 743–751, 2016.
- [43] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In CVPR, 2014.
- [44] N. S. Nagaraja, F. R. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*, pages 3235–3243, 2015.
- [45] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *CoRR*, abs/1612.08871, 2016.
- [46] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1187–1200, 2014.
- [47] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *CoRR*, abs/1606.02147, 2016.
- [48] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. J. V. Gool, M. H. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016.
- [49] F. Perazzi, O. Wang, M. H. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, pages 3227–3234, 2015.
- [50] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786, 2009.
- [51] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.
- [52] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MIC-CAI*, pages 234–241, 2015.
- [53] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309–314, 2004.
- [54] O. Sener, K. Ugur, and A. A. Alatan. Efficient MRF energy propagation for video segmentation via bilateral filters. *CoRR*, abs/1301.5356, 2013.
- [55] E. Shahrian, B. Price, S. Cohen, and D. Rajan. Temporally coherent and spatially accurate video matting. *Comput. Graph. Forum*, 33(2):381–390, 2014.
- [56] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clock-work convnets for video semantic segmentation. *CoRR*, abs/1608.03609, 2016.
- [57] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. L. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. *Comput. Graph. Forum*, 35(2):93–102, 2016.
- [58] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia. Deep automatic portrait matting. In *ECCV*, pages 92–107, 2016.
- [59] M. Siam, S. Valipour, M. Jägersand, and N. Ray. Convolutional gated recurrent networks for video segmentation. *CoRR*, abs/1611.05435, 2016.

- [60] J. Sun, W. Zhang, X. Tang, and H. Shum. Background cut. In ECCV, pages 628–641, 2006.
- [61] Y. Tsai, M. Yang, and M. J. Black. Video segmentation via object flow. In CVPR, pages 3899–3908, 2016.
- [62] S. Valipour, M. Siam, M. Jägersand, and N. Ray. Recurrent fully convolutional networks for video segmentation. *CoRR*, abs/1606.00487, 2016.
- [63] H. Wang, T. Raiko, L. Lensu, T. Wang, and J. Karhunen. Semi-supervised domain adaptation for weakly labeled semantic video object segmentation. *CoRR*, abs/1606.02280, 2016.
- [64] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. ACM Trans. Graph., 24(3):585– 594, 2005.
- [65] J. Wang and M. F. Cohen. Image and video matting: A survey. Foundations and Trends in Computer Graphics and Vision, 3(2):97–175, 2007.
- [66] J. Wang, S. Yeung, J. Wang, and K. Zhou. Segmentation rectification for video cutout via one-class structured learning. *CoRR*, abs/1602.04906, 2016.
- [67] Y. Wang, T. Tan, and K. Loe. Video segmentation based on graphical models. In CVPR.
- [68] S. Yi and V. Pavlovic. Multi-cue structure preserving MRF for unconstrained video segmentation. In *ICCV*, pages 3262– 3270, 2015.
- [69] P. Yin, A. Criminisi, J. M. Winn, and I. A. Essa. Tree-based classifiers for bilayer video segmentation. In CVPR, 2007.
- [70] C. Yu, H. Le, G. J. Zelinsky, and D. Samaras. Efficient video segmentation using parametric graph partitioning. In *ICCV*, pages 3155–3163, 2015.
- [71] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In CVPR, pages 628–635, 2013.
- [72] K. Zhang, X. Li, and Q. Liu. Unsupervised video segmentation via spatio-temporally nonlocal appearance learning. *CoRR*, abs/1612.08169, 2016.
- [73] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In CVPR, pages 3641–3649, 2015.
- [74] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.
- [75] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. *ICCV*, 2015.
- [76] F. Zhong, X. Qin, and Q. Peng. Transductive segmentation of live video with non-stationary background. In *CVPR*, pages 2189–2196, 2010.
- [77] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. *CoRR*, abs/1611.07715, 2016.
- [78] D. Zou, X. Chen, G. Cao, and X. Wang. Video matting via sparse and low-rank representation. In *ICCV*, pages 1564– 1572, 2015.