Enhancing Video Object Segmentation with Neural Networks

Jasjeet Dhaliwal
jdhaliwal@cs.umass.edu

Abstract

We address the problem of enhancing unsupervised video object segmentation methods by incorporating objectness cues from deep convolutional neural networks. We first present a fully automatic method that utilizes objectness cues from Sharpmask, an object segmentation neural network.We also discuss our ongoing work in implementing an improved object segmentation system, Mask-RCNN, that has been shown to produce near ground truth object segmentations. Finally, we propose a novel unsupervised video object segmentation method that utilizes objectness cues and Markov Random Fields.

1 Introduction

Unsupervised video object segmentation is the problem of segmenting freely moving non-rigid objects in a video. This means a partly moving object in a given image frame of the video, such as a person preparing to move forward by lifting one foot off the ground, must be fully segmented as a moving object. Many state of the art methods such as [1, 8, 18, 20, 26], are successful in segmenting motion in such a sequence of image frames. However, one of their primary weaknesses is the inability to segment an object, when only a part of it is moving. We aim to address this weakness by using objectness cues from deep Convolutional Neural Networks (CNN).

In general, given a sequence of image frames, the human vision system is excellent at detecting moving objects. This is due, largely in part, to the ability of humans to identify objects, as well as motion in a sequence of images. That is, the human vision system utilizes motion cues, as well as objectness cues to identify freely moving non-rigid objects in a sequence of image frames. However, unlike the human vision system, the systems mentioned above do not utilize direct objectness cues which are crucial to the human vision system. They rely primarily on using motion cues such as optical flow and feature cues such as appearance descriptors. We use this simple observation as the guiding principal behind utilizing Sharpmask [23], and Mask-RCNN (ongoing) [13] to provide objectness cues and enhance the fully automatic video object segmentation method of [1].

Bideau and Learned-Miller update their previous work [1] and utilize the objectness cues from Sharpmask in the second stage of a three stage segmentation process. In the first stage of this process, they use motion cues from the optical flow field of the image frame to assign each pixel the most likely motion model M^j, $j \in \{1, \ldots, k\}$, where every motion model M^j describes a different motion. In the second stage, they use a set of binary object segmentation masks and their associated objectness scores. These masks are the output of Sharpmask for the given image frame. After selecting the top m masks, they assign to each object o, $o \in \{1, \ldots, m\}$, a set of motion models $M^o \in \{M^1, \ldots, M^k\}$ such that $0 \leq |M^o| \leq k$. This set of motion models describes the motion of the object. In the third stage, they reduce the total number of motion models k to the number of moving objects m, and assign the most likely motion model M^o to a pixel. This yields a moving object segmentation for every frame in the video sequence.

The report is structured as follows. In section 2, we cover related work and motivate the use of Sharpmask and Mask-RCNN. In section 3, we describe the updated version of the video segmentation algorithm of [1] that uses objectness cues

from Sharpmask and show its results. In section 4, we describe the architecture of Mask-RCNN and our ongoing efforts in its implementation. In section 5, we propose a novel approach to unsupervised video segmentation using Markov Random Fields. We conclude the paper in section 6.

2 Related Work

Object detection is the problem of detecting objects in images. The goal is to bound an identified object with a box that is as close as possible to the ground truth annotation bounding box. Further, the box must classify the type (or class) of the object. State of the art methods such as [9, 10, 12, 25] have been able to perform very well on the object detection task by generating a large number of object proposals and classifying the proposals as a particular class. [11] developed R-CNN which modularized the process by using Selective Search [28] in order to generate object proposals for a given image. After which, they extracted features from the proposals using a deep CNN and fed these features into a binary linear support vector machine for the classification task. Although, they achieved good results, the training and inference was expensive in space and time. [10] introduced Fast R-CNN which improved the speed training and inference by adding a classification branch to the the CNN used for feature extraction. Fast R-CNN also included a novel pooling layer, ROIPooling, that was able to extract features maps from the object proposals (called Regions of Interest in the paper). Despite its success in the object detection task and its reduced training and inference time, Fast R-CNN was not an end-to-end trainable system. This issue too was resolved with the development of Faster R-CNN [24] that added a trainable CNN called Region Proposal Network (RPN) to generate a proposals (regions of interest) for any given input frame.

Object instance segmentation furthers the object detection task by requiring a pixel wise segmentation of objects in a given frame. Approaches such as [19, 22, 23] used fully convolutional feed forward networks for this task. [19] introduced the Fully Convolutional Networks (FCN) that add deconvolution layers to upsample the extracted features into a segmentation mask and are trainable end to end. Their models, which converted existing object classification systems such as [17] into FCNN, outperformed state of the art systems in segmenting images of arbitrary sizes. [22] introduced Deepmask which utilized the same FCNN approach as [19], but also added a scoring branch that provided an objectness score for every mask. Therefore, their system naturally seemed to be a very good candidate for utilizing object masks and objectness scores as cues to other state of the art unsupervised video segmentation systems. However, this approach was further improved by [23], when they introduced a top-down refinement module which included skip connections from the lower layers of the network and produced much higher quality masks. We utilize this method in enhancing the work of [1] in the task of unsupervised video object segmentation.

3 Video Segmentation using Sharpmask

3.1 Method

Using the terminology of [1], we first review the concepts required to build a motion model. The 3D world can be represented by the axes (X, Y, Z), where Z is the depth axis. An image is 2D, therefore, we represent the projection of a point in 3D world coordinates to the image plane by (x, y). We parametrize the translational motion of a camera in the 3D world by (U, V, W) where U parameterizes the translation along X, V parameterizes the translation along Y, and W parameterizes the translation along Z. We let (u, v) denote the translation motion of point on the 2D image plane. Finally, we let f be the focal length of the camera. We can then calculate the translation motion on the image plane as :

$$u = \frac{-fU + xW}{Z}; \quad v = \frac{-fV + yW}{Z}$$

Then, we can utilize (u, v) to calculate the angle of motion at any given point by:

$$\theta(x, y, f, U, V, W) = \text{atan}(-fV + yW, -fU + xW)$$

Using the above, a motion model M for an $m \times n$ image can be written as:

$$M = \begin{pmatrix} \theta_{11} & \cdots & \theta_{1m} \\ \vdots & \ddots & \vdots \\ \theta_{n1} & \cdots & \theta_{nm} \end{pmatrix}.$$

For a more detailed explanation behind the formation of the model M, the reader is encouraged to read [1]. Note here that the motion model M is depth independent which is also a key contribution of [20] and yields a depth independent segmentation of the video.

We can now describe the fundamental steps of the three stage process in order to illustrate the inclusion of object cues from Sharpmask:

Stage 1: Assigning Motion Models to Pixels
Let the observed flow vector at pixel x be $\vec{v_x}$. Further, let the set of motion models be $\{M^1, \ldots, M^k\}$. Then, let $p(M_x^j \mid \vec{v_x})$ denote the posterior probability of assigning motion model M^j to pixel x given the observed flow vector $\vec{v_x}$. We can calculate the posterior:

$$p(M_x^j \mid \vec{v_x}) \propto p(\vec{v_x} \mid M_x^j) \cdot p(M_x^j).$$

where, the posterior of each pixel in a frame at time t is propagated as the prior for that pixel in the frame at time $t+1$ [1]. Letting L_x denote the most likely motion model at pixel x, we have:

$$L_x = \arg\max_j p(M_x^j \mid \vec{v_t}).$$

Stage 2: Selecting Binary Masks
Sharpmask [23] generates n masks for a given image frame, with an associated objectness score for each of the n masks. The number n is a hyperparameter and can be changed as required. Each mask is the same size as the image frame and contains only a single object. Therefore, we refer to the mask as a binary object proposal mask, i.e. the value for any given pixel in the mask is 1, if the pixel belongs to the object, and 0 otherwise. We utilize these n masks and their objectness scores as cues to the algorithm. In order to choose the masks that correspond to moving objects, the algorithm first generates a motion mask P such that the value of any given pixel i in P is the probability of motion at that pixel. That is:

$$P_i = 1 - p(M^b g_i | \vec{v_i}), \tag{1}$$

where bg is the motion model for the static environment. The n masks are then filtered using the following Algorithm 1. [2]:

Algorithm 1: Choosing the best set of object proposal masks of moving objects

Input: n_{obj}: number of objects
M_{obj}: set of binary object proposal masks
s_{obj}: set of objectness scores
P: motion mask $1 - p(M_{bg}|\vec{v_t})$
Output: M_{mov}: set of binary object proposal masks of moving objects only

$e \leftarrow \infty$;
for $T \leftarrow$ *1 to iterations* **do**
  $\hat{n}_{obj} \leftarrow$ draw from Gauss dist. $\mathcal{N}(n_{obj}, \sigma_1)$;
  $\hat{M}_{mov} \leftarrow$ choose $\hat{n}_{obj}$ different object proposal masks from M_{obj}, each with probability $\mathcal{N}(s_{obj}; \mu_2, \sigma_2)$;
  $\Delta \leftarrow P - \frac{\cup \hat{M}_{mov}}{\hat{n}_{obj}+2}$;
  $\Delta \leftarrow \min(0, \Delta)$;
  ▷ $\cup \hat{M}_{mov}$ is divided by the estimated number of objects $\hat{n}_{obj}$ plus the static background component plus one component for a possible new motion, since P is at least $\frac{1}{\hat{n}_{obj}+2}$ if a moving object is observed
  if $sum(\Delta) < e$ **then**
    $e \leftarrow sum(\Delta)$;
    $M_{mov} \leftarrow \hat{M}_{mov}$;
  end
end

The results of the algorithm can be seen in Figure 1.[3]

Stage 3: Object Segmentation
Given the m moving object proposal masks, each object $o \in \{1, \ldots, m\}$ is assigned a set of motion models $M^o \in \{M^1, \ldots, M^k\}$ such that

[1] The reader is referred to [1] for a more thorough explanation of the motion and location priors.

[2] This algorithm has been written by Pia Bideau for the paper submitted to ICCV 2017

[3] This image has been created by Pia Bideau for the paper submitted to ICCV 2017

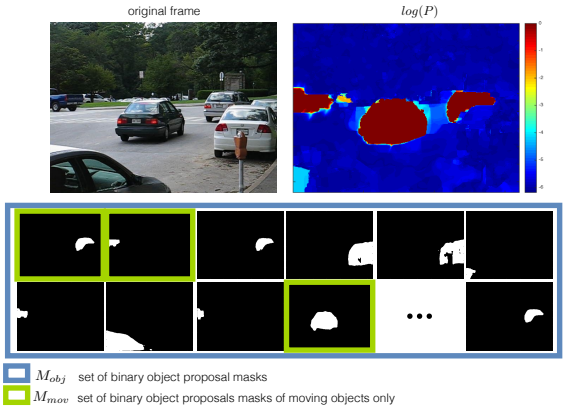

Figure 1: **Selecting motion masks from Sharpmask** Top left: Original Image Frame. Top Right: The logarithm of the motion mask P. Bottom: M_{obj} is a subset of the binary object proposal masks generated by Sharpmask. M_{mov} is the set of binary object proposal masks that are remaining after the selection algorithm completes.

.

$0 \leq |M^o| \leq k$. In order to decide the whether to include a motion model in the set M^o, the object must cover the motion model, and the probability of assigning that motion model to object o must be higher that the probability of assigning the motion model to any other object. Where we can calculate the probability of assigning the motion model to the object as:

$$p(M^j \in M^o) = \frac{\text{\# times } M^j \in M^o}{\text{\# Frames processed}} \tag{2}$$

Therefore, the total number of motion models is reduced from k (number of original motion models) to m (number of moving object proposal masks). Letting O^i denote the motion model assigned to object i where $i \in \{1, \dots m\}$, we calculate the posterior of O^i given the flow vector $\vec{v_x}$:

$$p(O^i_x|\vec{v_x}) = \frac{p(\vec{v_x}|O^i_x) \cdot p(O^i_x)}{p(v_x)} \tag{3}$$

Finally, each pixel x is re-assigned the most likely motion model:

$$L_x = \arg\max_i (p(O^i_x|\vec{v_x})) \tag{4}$$

Hence, each pixel is segmented as a moving object from one of the m moving objects, or static background.

3.2 Results

Using the multi-label scoring scheme from [21], the algorithm outperforms [16, 27] by a significant margin on FBMS-59 [5], and by 5% on the complex background data set [20]. The complex background dataset shows videos with a high variance in depth. This is in particular challenging for trajectory based motion segmentation approaches such as [16] as well as for occlusion based object segmentation approaches [27].

We display the results of the algorithm as compared to other state of the art video object segmentation models in Figure 2. [4]

4 Mask-RCNN

In order to improve the objectness cues provided to any video object segmentation algorithm, we need better object proposal masks and Mask R-CNN [13] outputs much higher quality object segmentation masks than Sharpmask. Mask R-CNN, extends Faster R-CNN by adding a FCN branch for predicting segmentation masks on each Region of Interest (RoI). The segmentation branch works in parallel to the classification and the bounding box regression branch. The two primary changes made to Faster-RCNN in order to develop Mask-RCNN are:

1. Addition of a Fully Convolutional Network (FCN) Branch for Object Segmentation

2. Replacing the ROIPooling Layer with the ROIAlign Layer

4.1 FCN Branch

Mask R-CNN uses the same two-stage training procedure as Faster-RCNN with the same first stage (RPN). However, in the second stage, in parallel to predicting the class and bounding box targets, Mask R-CNN also outputs a binary mask for each class, for every RoI generated by the RPN. This is one of the keys to its success as the FCN segmentation branch produces k binary segmentation masks for k classes. Hence, there is no competition among classes. The loss during training is defined as a multi-task loss on each sampled RoI. That is $L = L_{cls} + L_{box} + L_{mask}$. L_{cls} and L_{box} are the same as in Faster-RCNN [24]. However, the segmentation branch applies an average cross-entropy loss applied only to the mask generated

[4] This image has been created by Pia Bideau for the paper submitted to ICCV 2017

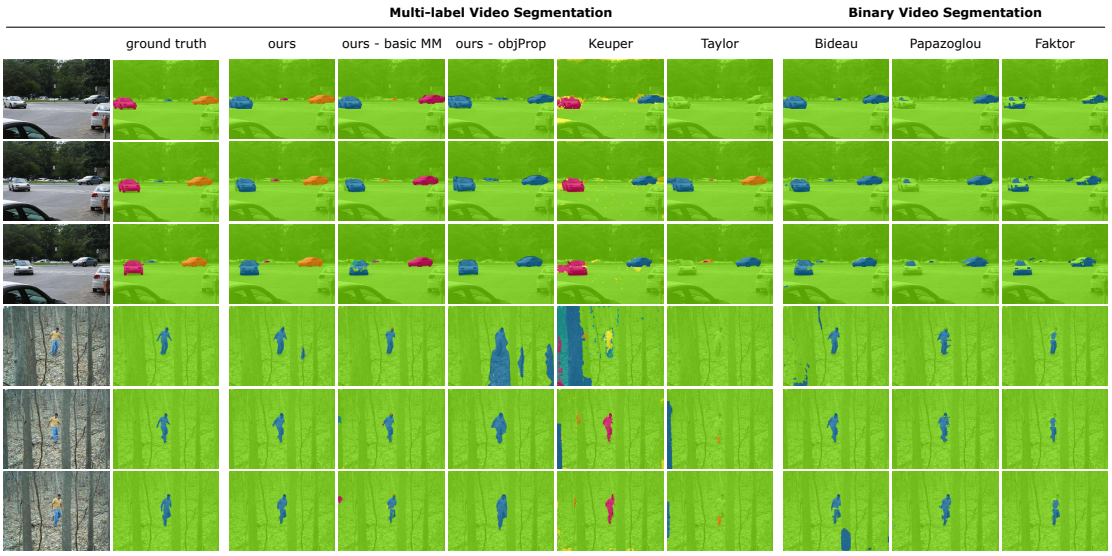

Figure 2: **Sample results.** Top to bottom: The first three rows show the video *cars5* from the FBMS-59 test set. Rows four to six show results on the video *forest* of the complex background data set. For both videos we show frames 1, 10 and 20. We show results on our final version of our algorithm ("ours") as well as for intermediate results of our final algorithm ("ours - basic MM" and "ours - objProp"). Comparisons to state of the art methods on multi-label segmentation and binary segmentation are shown in rows 6-10 [1, 7, 16, 18, 27].

.

for the ground truth class (i.e. for k classes, the network outputs k binary masks, and applies loss only to the mask that contains the ground truth object).

4.2 ROIAlign Layer

In order to solve the segmentation task, Mask R-CNN replaces the ROIPooling Layer of Fast R-CNN [10] with the ROIAlign Layer. The ROIAlign layer maintains the pixel-to-pixel alignment with the feature maps and hence outputs better segmentations. The ROIPooling layer looses this alignment while extracting a fixed size feature map from a RoI because it quantizes the floating number scale values of the RoI to the discrete integer values For instance, the mapping from an RoI to a feature map requires computing the relationship between cooridinates of the feature map and those of the RoI. For a feature map stride of 16, this value is calculated by computing [x/16], where 16 is the feature map stride and [] is rounding. This RoI is then subdivided into spatial bins whose size also calculated by performing quantization. Finally, the feature values covered by each bin are aggregated into the feature map by max pooling. These quantizations introduce misalignments between the RoI and the extracted feature map and negatively impact the segmentation mask predictions.

The RoIAlign layer removes both the quantizations of RoIPooling layer and uses bilinear interpolation [14] to compute the exact values of the input features at four regularly sampled locations in each RoI bin. These values are then aggregated using max pooling.

The authors of Mask-RCNN have not released the code yet.Therefore, we have implemented the FCN segmentation branch of Mask-RCNN and ROIAlign layer in the Caffe framework. [15].

4.3 Training

Since the segmentation task is much harder than the classification and bounding box regression task, we have not yet been able to find the right set of hyper parameters required to output high quality segmentation masks. Further, the authors of Mask-RCNN do not provide the implementation details of the ROIAlign layer, hence, the hyper-parameters provided in [13] do not work. We are currently in the process of setting up training procedure on a GPU cluster that will perform a thorough grid search in the hyper parameter space.

We aim to train the Mask R-CNN network to output high quality segmentation masks and use these masks to further enhance the work in [1].

5 Using Markov Random Fields for Video Object Segmentation

In this section, we describe a novel approach to unsupervised video segmentation system using Markov Random Fields. We begin by noting that in order to achieve good performance, an unsupervised video object segmentation system must be able to utilize two sets of cues:

1. Motion cues
2. Object cues

The motion cues we will provide to our systems will be based on the estimated motion field (optical flow field) of two temporally consecutive images in a video. Based on the work of [1, 20], we will calculate the depth independent angle difference of the observed translational optical flow field and ideal translational optical flow field, as well as the difference in the ideal translational optical flow field and the observed optical flow field. We will combine these two results into a motion cue input for our systems via a pixelwise conjunction of the two results as done by [1]. In order to calculate the parameters of the camera motion, we will use the minimization method developed by [6] with modified projection error function defined by [1].

For the object cues, we will use the output of Mask R-CNN for each frame. Here, we rely on Mask R-CNN to provide a segmentation which will be refined by our systems into an accurate motion based segmentation. A crucial assumption here is that the Mask R-CNN does not undersegment frames. This assumption may be violated once we test our system and hence lead to various changes. We now describe our system in more detail.

Given, a video sequence $V = \{I_1, \ldots, I_T\}$, let I_t be the image frame corresponding to the video sequence at time t with entire video sequence containing T frames. Let each frame I_t be of size $x \times y$. Further, let the set of all pixel locations in the frame I_t be S_1 such that $|S_1| = x \times y$. Next, we define the set S_2 as the set of doubleton clique tuples (neighboring pixels) such that $(s, r) \in S_2 \iff (s, r)$ is a tuple of adjacent pixels in I_t.

Now, for a given frame I_t, we will provide a motion cue M_t of size $x \times y$, and and object cue O_t of size $x \times y$ as well. M_t^M refers to the set of pixels in M_t that we consider as moving independently of the camera motion and M_t^S be the set of pixels in M_t that we consider as not having moved independently of the camera motion. Therefore, $|M_t^M + M_t^S| = x \times y$. We let $\Gamma_t \in \mathbb{N}$ be the finite set of objects labels in O_t as segmented by the Mask R-CNN. Let ω_s be the label at pixel location s, such that $\omega_s \in \Gamma_t$ and $\omega_s = 0$ means background. Then, we let ω refer to a complete labeling of every pixel in the frame.

Based on the work of [2], we now define the energy $E(\omega)$ for a configuration ω as :

$$E(\omega) = \sum_{s \in S_1} \psi(\omega_s) + \sum_{s,r \in S_2} \phi(\omega_s, \omega_r)\delta(\omega_s, \omega_r)$$

where,

$$\begin{aligned}\psi(\omega_s) &= -\log \text{P (pixel s has label } = \omega_s) \\ &= -\log \left\{\mathcal{N}(\mu_{\omega_s}, \sigma^2_{\omega_s})\right\} \\ &= \left\{\log\left(\sqrt{2\pi}\sigma_{\omega_s}\right) + \frac{(M_t^s - \mu_{\omega_s})^2}{\sigma^2_{\omega_s}}\right\}\end{aligned}$$

$$\phi(\omega_s, \omega_r) = \exp\left\{-\frac{|C_s - C_r|^2}{2\beta^2}\right\}$$

$$\delta(\omega_s, \omega_r) = \begin{cases} 1 & \omega_s \neq \omega_r \\ 0 & \omega_s = \omega_r \end{cases}$$

where,
C_s = Color intensity at pixel s in I_t
β = Penalizes pixels with similar color intensities. i.e. $|C_s - C_r| < \beta$

$$\mu_{\omega_s} = \begin{cases} \frac{1}{|M_t^M|}\sum_{m \in M_t^M} M_t^m & \text{if } \omega_s \neq 0 \\ \frac{1}{|M_t^S|}\sum_{s \in M_t^S} M_t^s & \text{if } \omega_s = 0 \end{cases}$$

$$\sigma^2_{\omega_s} = \begin{cases} \frac{1}{|M_t^M|}\sum_{m \in M_t^M}(M_t^m - \mu_{\omega_s}) \\ \qquad \text{if } \omega_s \neq 0 \\ \frac{1}{|M_t^S|}\sum_{s \in M_t^S}(M_t^s - \mu_{\omega_s}) \\ \qquad \text{if } \omega_s = 0 \end{cases}$$

Next, we can convert the energy function $E(\omega)$ into a probability as :

$$P(\omega) = \frac{1}{Z}\ exp(-E(\omega))$$

Note, that we can use any (or a combination of any/all) of the following features for the pairwaise potentials: {color, intensity, texture}. In order to find the most probable segmentation, we will use the graph based min-cut alpha-swap algorithm developed in [4]. The solution provided by the algorithm is an approximation global optimum in the multi-label case. However if we consider the binary segmentation case (i.e. background and foreground only), the graph min-cut algorithm developed in [3] provides the global optimal under submodularity constraints of the energy function. Therefore, the configuration ω that minimizes the energy function $E(\omega)$ and maximizes the probability.

Thus, we can get a completely unsupervised video object segmentation of a given frame.

6 Conclusion

In this report, we presented an enhancement to the method of [1] by using objectness cues from Sharpmask. We also motived the reason for using Mask R-CNN in order to further improve the results obtained with Sharpmask. We described the architecture of Mask R-CNN and the changes made to Faster R-CNN. Finally, we proposed a novel method for unsupervised video object segmentation using Markov Random Fields.

7 Acknowledgements

We would like to thank Professor Erik Learned-Miller for giving us the opportunity to work on this project and his guidance throughout this work. We would also like to thank Pia Bideau for her helpful input, guidance in this work and for generously providing content from her recently submitted paper. Lastly, we would like to thank Professor Maji for being the second advisor on the project.

References

[1] Pia Bideau and Erik G. Learned-Miller. It's moving! A probabilistic model for causal motion segmentation in moving camera videos. *CoRR*, abs/1604.00136, 2016. URL `http://arxiv.org/abs/1604.00136`.

[2] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70 (2):109–131, 2006.

[3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–1137, 2004.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[5] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010.

[6] Anna R Bruss and Berthold KP Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20, 1983.

[7] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, volume 2, page 8, 2014.

[8] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, volume 2, page 8, 2014.

[9] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual

recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[16] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3271–3279, 2015.

[17] 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.

[18] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011.

[19] 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.

[20] Manjunath Narayana, Allen Hanson, and Erik Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1577–1584, 2013.

[21] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2014.

[22] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

[23] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[25] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[26] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015.

[27] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4268–4276, 2015.

[28] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.