Smooth Segmentation in Videos: Blind Consistency Over Semantic Segmentations Produced by Fully Convolutional Networks

by Noah Picard

A Thesis submitted in partial fulfillment of the requirements for Honors in the Department of Computer Science at Brown University

> Providence, Rhode Island April 2018

 \bigodot Copyright 2018 by Noah Picard

This thesis by Noah Picard is accepted in its present form by the Department of Computer Science as satisfying the research requirement for the awardment of Honors.

Date _____

James Tompkin, Reader

Date _____

Michael Littman, Reader

Contents

| 1 | Introduction | 1 |
|----------|---|----|
| | 1.1 Abstract | 1 |
| | 1.2 Current Methods | 1 |
| | 1.3 Our Approach | 2 |
| | 1.4 Workflow | 2 |
| 2 | Scene Segmentation | 4 |
| | 2.1 Motivation \ldots | 4 |
| | 2.2 Background | 5 |
| | 2.3 Application of Region-based Convolutional Neural Networks | 5 |
| | 2.4 Results | 6 |
| 3 | Blind Consistency | 8 |
| | 3.1 Motivation | 8 |
| | 3.2 Background | 8 |
| | 3.3 Applications on Scene Segmentation | 10 |
| 4 | Pipeline Results | 12 |
| | 4.1 Current Progress | 12 |
| | 4.2 Results Comparison | 12 |
| | 4.3 Analysis | 12 |
| 5 | Future Directions | 16 |
| 6 | Bibliography | 17 |

Introduction

1.1 Abstract

We apply existing R-CNN semantic segmentation to the frames of input videos, and apply a general temporal consistency algorithm over these segmentations, with the aim of producing smoothed object segmentations. This is required because scene segmentation systems are sensitive to temporal changes between frames, and so, like video editing processes adjusting color, contrast, and tone, produce a flickering effect due to subtle differences in individual frames, which may cause a filter to calibrate differently, or cause one object to be classified as another. We adapt this blind temporal consistency algorithm for scene segmentation to reduce this flickering effect, as it already does with editing processes, and improve recognition, by reducing classification error from obfuscation, lighting, and movement. Expanding from this process, we aim to find improved smoothing properties for use in future video segmentation software. Through this work, we investigate existing papers surrounding R-CNNs and temporal consistency in an attempt to improve consistency over segmentations.

1.2 Current Methods

Semantic video segmentation, while recently spurred into the public consciousness by advances in autonomous vehicle technology, has been an increasingly active research problem since robots were first given cameras to perceive the world around them. Early segmentation approaches applied traditional and probabilistic classification methods [Friedman and Russell, 1997], motivated primarily to find moving objects in an otherwise still scene.

As applications in the industry have diversified, from 3D object identification to motion planning, to medical imaging, to video editing, further contributions have arisen in research to meet these challenges. Methods developed after this approach include the following four primary techniques: superpixels, object proposals, patches, and the bilateral filter. Examples of each follow. Chang et al. [2013] develops temporal superpixels, which, in contrast to supervoxels, move consistently with object parts in different frames, for use in video segmentation. Fan et al. [2016] uses split nearest neighbor fields to detect differences in motion between foreground and background to produce a temporally consistent distinction. Ramakanth and Babu [2014] matches patch seams across frames to propagate temporal labels. Perazzi et al. [2015] prunes and combines many imperfect object proposals in a fully connected random field, and minimizes an energy function to find a best fit for object segmentation. Marki et al. [2016] propagates object masks using an energy formulation in bilateral space, to increase computational efficiency without major loss of accuracy. Optical flow is used in a number of papers to inform propagation of object masks [Perazzi et al. 2017], [Ramakanth and Babu 2014], [Grundman et al. 2010]. Most recently, Caelles et al. [2017] employs fully convolutional neural networks to produce temporally stable foreground segmentations.

However, even as the issue has evolved, current methods within the area of scene segmentation do not efficiently handle video segmentation processing. As scene segmentation developed originally for image segmentation, the techniques used were not designed with efficiency in mind, and therefore perform poorly when applied to larger file sizes.

1.3 Our Approach

Our approach differs from existing techniques in that it decomposes the problem of video scene segmentation into two distinct problems: single frame segmentation, which has already been researched deeply, as discussed later, and frame segmentation recombination, where we use information from the original video to inform the merging of frames to create a smoothly segmented video. The approach is also novel in its use of a general consistency algorithm to resolve temporal artifacts. This process makes the assumption that frame segmentations will grant us most of the information needed to segment the entire video, and that the missing knowledge can be inferred from the motion in the video itself. By making this assumption, we can reach great leaps in the processing of these videos: batches of frames can be processed in parallel, and the consistency algorithm used to smooth segmentations is very efficient, as discussed in the Results section.

This temporal consistency, or smoothing, is required because scene segmentation systems are sensitive to temporal changes between frames, similar to photo processing effects such as autotone, color balance, auto-contrast. This sensitivity sometimes produces flickering effects due to subtle differences in individual frames, obfuscating an object just enough to render it unclassified, or diverting the edges of an objects mask. In particular, we aim to fix this issue using the blind temporal consistency algorithm created in Boneel et al. [2015]. This algorithm should reduce this flickering effect, as it already does with editing processes, and improve video-wide recognition.

1.4 Workflow

Our final pipeline [Figure 1.1] outlines the complete flow of information. When given a video to be processed, we first separate the video into frames. We then run each of these frames through a

scene segmentation algorithm, which produces individual masks for each object type found within each frame. The frames for each object mask are then combined to form a complete video for each object type. Then, we run these object videos in parallel through a blind consistency algorithm, as developed in Boneel et al. [2015], to gain smoothed mask videos for each object type. Finally, these object videos can be recombined to form a full video segmentation.



Figure 1.1. Our Segmentation Consistency Pipeline

Scene Segmentation

2.1 Motivation

Scene segmentation provides an automated method to delineate an image into various objects. For example, a street scene might be parsed into cars, people, signs, etc. These segmentations also provide masks in the shape of the classified objects, as in the following scene segmentation [Figure 2.1].



Figure 2.1. A Segmented Street Scene. Contains people, traffic signs, cars, and a motorcycle at the center frame. Text at the corners of bounding boxes state object class and confidence of classification.

These segmentations have a variety of applications: identifying objects in robotic vision, image description for the visually impaired, image editing tools, etc. Producing scene segmentations for over an entire video is generally expensive because of the size of the data being processed at once.

However, by splitting the video into frames and processing these frames in parallel, we can gain most of the information needed for segmentation, and take advantage of existing, well-developed systems for batched image segmentation. These frame segmentations provide a rough description of the full video segmentation, which can be later interpolated.

2.2 Background

Image segmentation, or the partitioning of pixels into discrete groups, is a research problem dating back to 1965, and has spurred research into many widely used techniques, including edge detection, thresholding, graph search and clustering. Early applications include medical imaging, where region growing [Pavlidis 1972] and successive partitioning [Brice and Fennema 1970] provided an initial framework for approaches. Prior to the success of convolution neural networks in 2012, these frameworks diversified into many categories: snake and balloon approaches [Kass et al. 1988], region based approaches [Zhu and Yuille 1996], formulations by graph partitioning [Boykov et al. 1999], [Shi and Malik 2000], [Felzenszwalb and Huttenlocher 2004], contour detection [Arbelez et al. 2011], and more. These approaches each address segmentation through edges detection and isolation, found by processes like the Canny edge detector, or by similarities between pixels and groups of pixels in more region-based methods. Zhu and Yuille [1996] combines several earlier approaches, forming a novel region competition method, which draws from balloon approaches to minimize a Bayesian/MDL criterion. Felzenszwalb and Huttenlocher [2004] forms a graph representation of the image, and provides a minimum spanning tree-like algorithm to balance the coarseness of segmentations. Arbelez et al. [2010] performs high-performance contour detection over the image, and provides a segmentation algorithm which transforms the contours into a hierarchical region tree, thereby reducing segmentation to contour detection. These methods apply local and global information to aid in the segmentation of images, but rely entirely on visual representations, rather than apply learning about object structure. Convolutional neural networks (CNNs) went on to fill this gap and dramatically improve segmentation through semantic segmentation, segmentation that identifies the objects presented in the region, and learns the structure of those objects to better inform their regions.

2.3 Application of Region-based Convolutional Neural Networks

Developing semantic segmentations requires recognition of groups of pixels in an image as objects, and classification of those objects. This analysis over an image matches the process naturally used in Region-based Convolutional Neural Networks (R-CNN), and thus R-CNNs provide a strong solution for semantic segmentation problems. The R-CNN system works in three parts: first, by proposing regions of interest within the image using the Selective Search algorithm; second, by identifying and returning the most likely candidates for regions in the image which contain a distinct classifiable object; third, by refining the bounding boxes for each object region. R-CNNs were first developed in Girshick et al. [2013], following the success of AlexNet in 2012. This process is simplified and sped up in the Fast R-CNN system [Girshick 2015] using Region of Interest pooling. RoI pooling was developed to prevent each region of the image from requiring a separate pass through AlexNet (which required approx 2000 calls in Girshick et al. [2013]), and instead passes the entire image through once, pooling over regions to provide features for each region simultaneously. Girshick [2015] also merges the process steps into one network, which provides classifications and bounding box refinements for all regions in one pass. The Faster R-CNN [Ren et al. 2015] approach develops a Region Proposal Network, which combines the calculations for feature detection for the RoI pooling with the feature detection for region proposals using a sliding window.

However, these approaches focus on scene segmentation with bounding boxes, whereas more fine-grained pixel segmentations were still needed. Then, in 2017, the Mask R-CNN system [He et al. 2017] was created at Facebook AI. The Mask R-CNN system attaches another branch alongside the classification and bounding box branches, to produce a binary matrix representing the mask of the object, using a Fully Connected Network. With some refinements [He et al. 2017], this process is extremely successful for image segmentations, and is what we use to produce the frame segmentations in our pipeline.

2.4 Results

The following images are the visualizations of using the Mask R-CNN repository, trained on the MSCOCO segmentation dataset, to classify frames from example videos from the DAVIS challenge dataset.

As seen in these images, the Mask R-CNN generally produces well-bounded, if overly smoothed masks of the objects within the frames [Figure 2.1], [Figure 2.2], [Figure 2.3]. However, in [Figure 2.4], [Figure 2.5], and [Figure 2.6] objects were not identified in some frames, incorrectly identified in others, and identified with poor masks. We aim to mitigate these issues using our consistency algorithm in the next section.



(a) Mask/Frame Overlay (b) Mask R-CNN output for cars (c) Ground truth mask for car

Figure 2.1: Classification results for one frame of Car Roundabout video



(a) Mask/Frame Overlay (b) Mask R-CNN output for person(c) Ground truth mask for girl/cat

Figure 2.2: Classification results for one frame of Cat Girl video



(a) Mask/Frame Overlay

(b) Mask R-CNN output for bear (c) Ground truth mask for bear

Figure 2.3: Classification results for one frame of Bear video



- (a) Mask/Frame Overlay
- (b) Mask R-CNN output for cars (c) Ground truth mask for car

Figure 2.4: Poor classification results. There is only one real person in the car.



(a) Mask/Frame Overlay (b) Mask R-CNN output for person(c) Ground truth mask for girl/cat

Figure 2.5: Poor classification results. The cat is identified as both a dog and a pizza.



(a) Mask/Frame Overlay (b) Mask R-CNN output for bear (c) Ground truth mask for bear

Figure 2.6: Poor classification results. The plant is not identified in this frame.

Blind Consistency

3.1 Motivation

Blind consistency algorithms are algorithms that enforce a temporal consistency over processed frames in a video, while preserving the quality of the video. As previously alluded to, many image editing processes are sensitive to temporal differences between frames, such as luminance and color, causing sharp differences in adjacent edited frames, producing flicker. However, using temporal consistency algorithms can smooth these differences. Most of these algorithms have been tailored to particular types of effects (e.g, reflectance reduction, color correction). However, if a temporal consistency algorithm can be developed without knowledge of the underlying effect applied to the video, then its use can be extended over a variety of effects, even to segmentations or probability fields.

3.2 Background

Initial temporal consistency algorithms, such as those in [Aydin et al. 2014] and [Boneel et al. 2013], focus on techniques to address inconsistencies caused by specific image filters. For example, [Aydin et al. 2014] develops a technique to reduce jumps in tone caused by HDR, by splitting into spatiotemporal base-detail layers, and strongly filtering base layers. However, these techniques do not extend to enforcing consistency in more general settings, because they require specific knowledge of the effect applied over the video.

Recent work has produced options for more general temporal consistency systems. Lang et al. [2012] develops a system that uses optical flow to inform the temporal paths of Gaussian convolutions. This, combined with a specialized edge aware filter, allows their colorization to move smoothly while respecting edges. In further edge-preserving video applications, Paris [2008] applies the Gaussian kernel in order to transfer classical algorithms like bilateral filtering a mean-shift segmentation onto video streams.

The algorithm used in this work was developed by Boneel et al. [2015] to enact blind temporal consistency, wherein the algorithm has no prior knowledge of the effect used, and uses only the raw and processed video to smooth temporal issues. This method uses optical flow to calculate baseline temporal consistency from the original video in order to stabilize effects on the processed video, and observes an energy function in the gradient domain to preserve edges throughout the temporal smoothing. This algorithm was tested on a variety of image processing effects, such as color harmonization, style transfer, HDR compression, and dehazing, and was found to produce cutting edge results on temporal smoothing while preserving high frequency detail. Specifically, given input frames $\{V_0, V_1, ...\}$ and processed frames $\{P_0, P_1, ...\}$, and with the aim of iteratively producing temporally consistent frames $\{O_0, O_1, ...\}$, general temporal smoothing can be formulated as "averaging the current frame with the previous frame, i.e, $O_n = (P_n + \mu_0 \operatorname{warp}(O_{n-1}))/(1 + \mu_0)$ where μ_0 controls the smoothing strength [Paris 2008]" [Boneel et al. 2015]. Here, warp() refers to optical flow applied to morph the previous frame towards the current frame. This equation also corresponds to averaging in the frequency domain, which implies that temporal smoothing affects all frequencies equally. By contrast, Boneel et al. [2015] aims to minimize both temporal consistency and scene dynamics, and therefore must treat low and high frequencies differently. They achieve this by minimizing the least squared error over both attributes simultaneously, as follows:

$$\int || \bigtriangledown O_n - \bigtriangledown P_n ||^2 + w(x) ||O_n - \operatorname{warp}(O_{n-1})||^2 dx \qquad (1a)$$
with: $w(x) = \lambda \exp(-\alpha ||V_n - \operatorname{warp}(V_{n-1})||^2) \qquad (1b)$

$$O_0 = P_0 \qquad (1c)$$

As seen in (1a), the first least squares statement encodes the scene dynamics by maintaining similarities between the gradients of the processed and output video, while the second encodes the temporal consistency, minimizing the amount of warp between the current and previous frame. These two are balanced by a weight function w(x), which reduces the need for temporal consistency when the input frames are not temporally consistent (e.g., sudden motion or scene change) (1b). Finally, this equation constrains the first output frame to be the same as the first processed frame (1c). Because this equation balances between scene dynamics and temporal consistency, it tends to strongly reduce low frequency artifacts (such as global or local flickering), while leaving high frequency details intact.

This algorithm was expanded upon by Boneel et al. [2017] to address both temporal and spatial inconsistencies between processed views in camera arrays, the product of which is the exact algorithm incorporated in our pipeline.



a) Original Image b) Reflectance frames (Bell et al. [2014]) c) Boneel et al. [2015]
Figure 3.1: Example figure taken from Boneel et al. [2015] applied over reflectance reduction.
Notice how the temporal artifacts between the red squares are removed by the algorithm.

3.3 Applications on Scene Segmentation

While flicker for color correction may mean a difference from blue to yellow in low frequency color, flicker in segmentations may be produced when an object is recognized in one frame, but not in the next. Therefore, if an algorithm can smooth over these inconsistencies while preserving the detail in each frame, it may be able to fill in the gaps of misclassification or poor masks (caused by obfuscation, lighting, movement, etc.) in the frame segmentations, thereby improving the overall video segmentation.

To demonstrate, consider a sequence of five adjacent frames that, when classified, produce masks outlining the objects correctly only in frames 1, 3 and 5. In frame 2 the masks are not quite fit to the objects (perhaps they are in motion), and in frame 4 the masks are gone completely. Were this sequence to be combined into a video, the result would shudder and flicker. However, when a blind consistency algorithm is applied to the data, it can use the information from the surrounding frames, along with the pixel flow in the original video, to fill in these gaps- classifications can be moved to when pixels have relocated to fill out the partial mask in frame 2, and in frame 4 the mask can be transferred using the scene dynamics from the surrounding frames 3 and 5. This demonstrates the amount of information that can be transferred from surrounding frames: enough to produce a smoothed mask over the entire 5 frame sequence.

In practice, because the number of different classes of object in a scene can be high, each objects mask (i.e, an image that is black everywhere except on the object, and lighter on the object, according to the confidence of the objects classification) is processed separately through the consistency algorithm, and then combined with the other masks to form the complete video scene segmentation. These masks produced by Mask R-CNN define sharp gradients between when the object is and is not predicted to be, which increases the difficulty of moving them to improve consistency. To combat this, our processing may first apply a Gaussian blur to the masks, to lubricate temporal motion.

However, this technique could be improved if instead of defining sharp masks, a process were used which produced a probability field for each object over the entire image, lighter on pixels where the object was likely to be and darker where the object was not. This extra information would provide much easier movement for object predictions across the frame, intelligently smoothing the gradients.

Pipeline Results

4.1 Current Progress

To date, we have implemented the primary components listed prior, and produced preliminary results for pure and smoothed masks. As described in the workflow pipeline [Figure 1.1], we have the process of splitting input video, using Mask R-CNN to produce frame by frame masks for each object classified, combining the mask frames into mask videos, and piping the original videos and mask videos through the blind consistency algorithm to produce more temporally consistent object frames. We have also inserted the optional Gaussian filter, applied to the object masks to aid the consistency algorithm. Using this, we demonstrate below the application of this pipeline on three videos from the DAVIS challenge dataset, showing results with and without Gaussian filtering. We have preliminarily experimented with some of the techniques described in Section 5, but these have not yet yielded results.

4.2 **Results Comparison**

Shown below are the results of several variations of our pipeline applied in the three previous DAVIS videos, compared with the original frames, the Mask R-CNN masks, and the ground truth masks. [Figure 4.1] shows results without Gaussian blurring. [Figure 4.2] shows results with medium spatiotemporal Gaussian blurring. [Figure 4.3] shows the results with strong spatial Gaussian blurring.

4.3 Analysis

As visible in the sample results [Figure 4.1], [Figure 4.2], [Figure 4.3] the blind consistency data subtly smooths the confidences of masks over a set of frames in a temporal edge aware manner, but does not easily adjust the edges of the mask frames produced by the semantic segmentation, producing only subtle perturbations in accordance with optical flow. One possible reason for this is



Figure 4.1: Results of pipeline without Caussian blurring. Note that the blind consistency algorithm

Figure 4.1: Results of pipeline without Gaussian blurring. Note that the blind consistency algorithm has a minor smoothing effect on confidence, but does not effect edges in any meaningful way.

described in the original Boneel et al. [2015] paper. They find that their approach "does not work well on matting because instabilities occur on fuzzy object boundaries which is also where optical flow techniques tend to fail" When we present our spatiotemporally blurred images, these fuzzy edges likely cause the same effect as the fuzzy matting edges. On the other hand, when our edges are sharp and defined, their edges might not correspond to the original edges of the object, due to imperfect masks produced by the Mask R-CNN system, or they may be too strong to be shifted by our blind consistency implementation. This sharpness, along with the single confidence over the mask, is a product of the way Mask R-CNN is designed, which only produces a single confidence value and binary mask within the bounding box, rather than a probability field over the bounding box, or even better, the entire image. Were this smooth probability field to be used instead of the sharp edges of the mask and bounding box, more movement would likely occur.

Another note is that missing masks are not filled in during the blind consistency process for sharp image definitions, which we posit is because the blind consistency algorithm maintains sensitivity to high frequencies, and because, unlike the confidence smoothing in other frames, there is no underlying mask to apply a smoothing to in an edge-aware way. This is mitigated by the use of the Gaussian spatiotemporal blurring, which provides an interpolated mask to be confidence-adjusted by the blind consistency algorithm, thereby producing temporally smooth results by sacrificing some granular flow details. As detailed in the next section, further work is needed to ascertain exactly why these issues are occurring, and to improve the system as a whole.

In reference to run-time, our pipeline is quite efficient. A single video can be quickly processed into frames, and those frames can be batched in parallel through the Mask R-CNN classifier at about 15 seconds per frame. Then the object mask frames can be quickly recombined into video, and then



Figure 4.2: Results of pipeline with medium spatiotemporal Gaussian blurring. Again the blind consistency algorithm has a minor contrasting effect on patches of confidence, and slightly pushes and pulls edges.

video can be passed in with the input video to the blind consistency algorithm, which, due to its iterative processing, is lightweight in memory and takes about 2 seconds per frame. These results are promising for rapid video editor processing, but do not approach video management in real-time.



Figure 4.3: Results of pipeline with strong spatial Gaussian blurring. Note that the blind consistency algorithm has little effect on confidence, and posterizes edges.

Future Directions

From these results, we can continue in numerous directions. First, we would like to resolve the issues inherent in directing the blind consistency algorithm over our processed segmentations, which does not regularly guarantee edge matching. One approach would be to improve our spatiotemporal Gaussian blurring by following the motion aware techniques formalized by Lang et al. [2012], allowing us to maintain more granular flow detail while filling in temporal gaps. Another solution would be to replace Mask R-CNN with another system which can produce probability fields over the entire image (e.g., non region-based fully convolutional networks), to produce smoother prediction images, with edges more likely to align with existing edge movement in the input video. This would also require removing a lower confidence bound to predictions, to get the maximum amount of information to smooth over. As demonstrated by the results in Boneel et al. [2015] on full image depth prediction, full image grayscale fields with better edges correspondence perform much better than black/white image matting, and the results can later be cut off at a confidence bound to provide distinct classifications. However, the closest the existing Mask R-CNN system gets to producing a probability field is at a 14x14 resolution, before upsampling to produce a final mask. This lowresolution probability field could be scaled up to fit the bounding box, and applied to the mask frame in a number of ways to provide more granular confidence. Applying over the entire bounding box and mask frame will likely cause the mask to be pulled out towards the full box shape, while applying the upscaled field solely inside the mask shape may achieve more granular probability on the mask, but will likely cause the blind consistency algorithm to pull the mask inwards. Alternatively, it may be possible to downsample our original video, and use the blind consistency to adjust the 14x14 probability field before upsampling to produce a final, adjusted mask.

If we experiment in these ways and find that the results substantially improve current methods using temporal data to improve individual frame segmentation and overall video segmentation quality, then the process could be utilized to improve autonomous vehicle perception through temporally adjusted 3D model approximation; or to apply as a temporally aware module into any neural network trying to produce segmentations over video, not just with the use of Mask RCNN; or to allow for many image processing techniques to be applied in an object specific manner over videos.

Bibliography

- Arbelez, P, et al. Contour Detection and Hierarchical Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, 2011, pp. 898916., doi:10.1109/tpami.2010.161.
- 2. Aydin, Tunc Ozan, et al. Real-Time Temporally Coherent Local HDR Tone Mapping. IEEE International Conference on Image Processing (ICIP), 2014, doi:10.1109/icip.2016.7532485.
- Bell, Sean, et al. Intrinsic Images in the Wild. ACM Transactions on Graphics, vol. 33, no. 4, 2014, pp. 112., doi:10.1145/2601097.2601206.
- Bonneel, Nicolas, et al. Blind Video Temporal Consistency. ACM Transactions on Graphics, vol. 34, no. 6, 2015, pp. 19., doi:10.1145/2816795.2818107.
- Bonneel, Nicolas, et al. Consistent Video Filtering for Camera Arrays. Computer Graphics Forum, vol. 36, no. 2, 2017, pp. 397407., doi:10.1111/cgf.13135.
- Bonneel, Nicolas, et al. Example-Based Video Color Grading. ACM Transactions on Graphics, vol. 32, no. 4, Jan. 2013, p. 1., doi:10.1145/2461912.2461939.
- Boykov, Y., et al. Fast Approximate Energy Minimization via Graph Cuts. Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, doi:10.1109/iccv.1999.791245.
- Brice, Claude R., and Claude L. Fennema. Scene Analysis Using Regions. Artificial Intelligence, vol. 1, no. 3-4, 1970, pp. 205226., doi:10.1016/0004-3702(70)90008-1.
- Caelles, S., et al. One-Shot Video Object Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, doi:10.1109/cvpr.2017.565.
- Chang, Jason, et al. A Video Representation Using Temporal Superpixels. 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, doi:10.1109/cvpr.2013.267.

- Fan, Qingnan, et al. JumpCut: Non-Successive Mask Transfer and Interpolation for Video Cutout. ACM Transactions on Graphics, vol. 34, no. 6, 2015, pp. 110., doi:10.1145/2816795.2818105.
- Felzenszwalb, Pedro F., and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, vol. 59, no. 2, 2004, pp. 167181., doi:10.1023/b:visi.0000022288.19776.77.
- Friedman, Nir, and Stuart Russell. Image Segmentation in Video Sequences: A Probabilistic Approach. UAI'97 Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, 1997, pp. 175181.
- Girshick, Ross, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, doi:10.1109/cvpr.2014.81.
- Girshick, Ross. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, doi:10.1109/iccv.2015.169.
- Gong, Yunchao, et al. Deep Convolutional Ranking for Multilabel Image Annotation.[1312.4894] Deep Convolutional Ranking for Multilabel Image Annotation, 14 Apr. 2014, arxiv.org/abs/1312.4894.
- Grundmann, Matthias, et al. Efficient Hierarchical Graph-Based Video Segmentation. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, doi:10.1109/cvpr.2010.5539893.
- He, Kaiming, et al. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, doi:10.1109/iccv.2017.322.
- Kass, Michael, et al. Snakes: Active Contour Models. International Journal of Computer Vision, vol. 1, no. 4, 1988, pp. 321331., doi:10.1007/bf00133570.
- Lang, Manuel, et al. Practical Temporal Consistency for Image-Based Graphics Applications. ACM Transactions on Graphics, vol. 31, no. 4, Jan. 2012, pp. 18., doi:10.1145/2185520.2335385.
- Long, Jonathan, et al. Fully Convolutional Networks for Semantic Segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, doi:10.1109/cvpr.2015.7298965.
- Marki, Nicolas, et al. Bilateral Space Video Segmentation. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.87.
- Masci, Jonathan, et al. A Fast Learning Algorithm for Image Segmentation with Max-Pooling Convolutional Networks. A Fast Learning Algorithm for Image Segmentation with Max-Pooling Convolutional Networks, 7 Feb. 2013, arxiv.org/abs/1302.1690.

- Paris, Sylvain. Edge-Preserving Smoothing and Mean-Shift Segmentation of Video Streams. Lecture Notes in Computer Science Computer Vision - ECCV 2008, 2008, pp. 460473., doi:10.1007/978-3-540-88688-4_34.
- Pavlidis, Theodosios. Segmentation of Pictures and Maps through Functional Approximation. Computer Graphics and Image Processing, vol. 1, no. 4, 1972, pp. 360372., doi:10.1016/0146-664x(72)90021-4.
- Perazzi, Federico, et al. Fully Connected Object Proposals for Video Segmentation. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, doi:10.1109/iccv.2015.369.
- Perazzi, Federico, et al. Learning Video Object Segmentation from Static Images. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, doi:10.1109/cvpr.2017.372.
- Pont-Tuset, Jordi, et al. The 2017 DAVIS Challenge on Video Object Segmentation. Computing Research Repository, 2017, arxiv.org/abs/1704.00675.
- Ramakanth, S. Avinash, and R. Venkatesh Babu. SeamSeg: Video Object Segmentation Using Patch Seams. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, doi:10.1109/cvpr.2014.55.
- Ren, Shaoqing, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, doi:10.1109/tpami.2016.2577031.
- Shi, Jianbo, and J. Malik. Normalized Cuts and Image Segmentation. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2000, doi:10.1109/cvpr.1997.609407.
- Yao, Jian, et al. Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, doi:10.1109/cvpr.2012.6247739.
- Zeiler, Matthew D., and Rob Fergus. Visualizing and Understanding Convolutional Networks. Computer Vision ECCV 2014 Lecture Notes in Computer Science, 2014, pp. 818833., doi:10.1007/978-3-319-10590-1_53.
- Zhu, S.c., and A.l. Yuille. Region Competition: Unifying Snakes, Region Growing, Energy/Bayes/MDL for Multi-Band Image Segmentation. Proceedings of IEEE International Conference on Computer Vision, 1996, doi:10.1109/iccv.1995.466909.