Copyright
by
Suyog Dutt Jain
2017

The Dissertation Committee for Suyog Dutt Jain
certifies that this is the approved version of the following dissertation:

Human Machine Collaboration for Foreground Segmentation in Images and Videos

Committee:

Kristen Grauman, Supervisor

Jason Corso

Raymond Mooney

Scott Niekum

Paul Etienne Vouga

Human Machine Collaboration for Foreground Segmentation in Images and Videos

by

Suyog Dutt Jain, B.E.; M.S.Comp.Sci.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2017

Dedicated to my family

Acknowledgments

This work has been truly possible because of the support and guidance I have received over these years from the following special people. You all have been a source of inspiration for me to pursue research and follow my dreams.

First and foremost, I would like to thank my advisor Prof. Kristen Grauman for being a truly inspirational advisor. I feel extremely fortunate for having her as my advisor. Words alone cannot describe my gratitude towards you. Thank you for introducing to the world of computer vision research, believing in me and guiding me at every step. Your exceptional guidance and support over these years is what made this thesis possible. I truly believe that the things I have learned from you, will remain as a source of inspiration for me throughout my life.

I would also like to thank all my committee members Prof. Jason Corso, Prof. Raymond Mooney, Prof. Scott Niekum and Prof. Etienne Vouga for the valuable feedback they provided which helped me in further strengthening this thesis. I am also grateful to Prof. J. K. Aggarwal, Dr. Maria Esteva and Dr. Weijia Xu for providing support and guidance during my early years as a graduate student. The foundation that you helped me build has been immensely beneficial for me throughout these years.

I also thank all my labmates and collaborators for enriching my academic life in Austin in special ways. A special thanks to Josh, Birgi, George and Jong-Taek for all the warmth and friendship during my time as a member of Computer and Vision Research Center. I will always remember our numerous philosophical discussions over lunches. Meeting each one of you gives me great joy every time. I would also like to thank my past lab members for all the valuable discussions and help whenever I needed it. I always remember Jaechul for his sense of humor and incredible wisdom. Thanks to Sung Ju for being the fellow night owl. Thanks to Adriana for your wit and love for beer. I will always remember our amazing discussions at Double Dave's and during conferences. Finally thanks to Chao-Yeh for being so kind, you are a true friend.

Last few years as a graduate student have been memorable because of the company of my current lab members. Thanks Dinesh for your valuable insights, discussions and always being available for our fun table-tennis games. Thanks Aron for trying to bring some sanity to our controversial discussions during TGIFs and conferences and being a great next desk neighbor . I will miss our board game sessions. Special thanks to Bo and Danna, with whom I got an opportunity to collaborate apart from sharing a wonderful friendship. Bo, thanks a lot for sharing my perspective towards research—I will always remember your clarity of thoughts and your natural willingness to solve more challenging problems. Danna, I admire your passion towards research for the greater cause and your endless dedication towards achieving it. I have learnt a great deal from both of you and I look forward to continue working alongside on some exciting problems. Thanks to Yu-Chuan, Ruohan, Wei-Lin, Tushar, Antonino and Ziad for being such good friends.

Beyond academic life, I owe my wonderful time in Austin to my friends who have been like an extended family. Thanks to Yashesh, Nikhil, Akash, Deepak, Sameer, Pooja, Shaival, Akanksha, Vikram, Gauri, Lakshmi, Anushree, Jasjyot, Chetana, Abhishek, Pallavi, Aniruddha, Garima, Siddharth, Poonam, Kartik, Riddhi, Subhamoy, Vinay, Sushil, Gunja, Akhilesh, Akshay and Sucheta. I will always remember our amazing hangouts, game nights and camping trips. Without all your support, none of this would have been possible.

I attribute all my success to the love and blessings of my parents and family. I was truly lucky to have my lovely wife Esha by my side during this entire journey. You make all challenges look easy with your smile, support and love. Being with you has been a blessing for me and you have filled my life with true happiness. Thanks for all your patience, love and kindness. Thanks to my father, who taught me that the best path to success is your own and the only way to achieve it is hard work, dedication and utmost focus. Last but not the least, thank you Ma for all the sacrifices you made for me. You are my first and the best teacher in life. I owe every bit of what I am to you.

Human Machine Collaboration for Foreground Segmentation in Images and Videos

Publication No. __________________

Suyog Dutt Jain, Ph.D.
The University of Texas at Austin, 2017

Supervisor: Kristen Grauman

Foreground segmentation is defined as the problem of generating pixel level foreground masks for all the objects in a given image or video. Accurate foreground segmentations in images and videos have several potential applications such as improving search, training richer object detectors, image synthesis and re-targeting, scene and activity understanding, video summarization, and post-production video editing.

One effective way to solve this problem is human-machine collaboration. The main idea is to let humans guide the segmentation process through some partial supervision. As humans, we are extremely good at perception and can easily identify the foreground regions. Computers, on the other hand, lack this capability, but are extremely good at continuously processing large volumes of data at the lowest level of detail with great efficiency. Bringing these complementary strengths together can lead to systems which are accurate and cost-effective at the same time. However, in any such human-machine

collaboration system, cost effectiveness and higher accuracy are competing goals. While more involvement from humans can certainly lead to higher accuracy, it also leads to increased cost both in terms of time and money. On the other hand, relying more on machines is cost-effective, but algorithms are still nowhere near human-level performance. Balancing this cost versus accuracy trade-off holds the key behind success for such a hybrid system.

In this thesis, I develop foreground segmentation algorithms which effectively and efficiently make use of human guidance for accurately segmenting foreground objects in images and videos. The algorithms developed in this thesis actively reason about the best modalities or interactions through which a user can provide guidance to the system for generating accurate segmentations. At the same time, these algorithms are also capable of prioritizing human guidance on instances where it is most needed. Finally, when structural similarity exists within data (e.g., adjacent frames in a video or similar images in a collection), the algorithms developed in this thesis are capable of propagating information from instances which have received human guidance to the ones which did not. Together, these characteristics result in a substantial savings in human annotation cost while generating high quality foreground segmentations in images and videos.

In this thesis, I consider three categories of segmentation problems all of which can greatly benefit from human-machine collaboration. First, I consider the problem of *interactive image segmentation*. In traditional interactive methods a human annotator provides a coarse spatial annotation (e.g.,

bounding box or freehand outlines) around the object of interest to obtain a segmentation. The mode of manual annotation used affects both its accuracy and ease-of-use. Whereas existing methods assume a fixed form of input no matter the image, in this thesis I propose a data-driven algorithm which learns whether an interactive segmentation method will succeed if initialized with a given annotation mode. This allows us to predict the modality that will be sufficiently strong to yield a high quality segmentation for a given image and results in large savings in annotation costs. I also propose a novel interactive segmentation algorithm called *Click Carving* which can accurately segment objects in images and videos using a very simple form of human interaction—point clicks. It outperforms several state-of-the-art methods and requires only a fraction of human effort in comparison.

Second, I consider the problem of segmenting images in a *weakly supervised image collection*. Here, we are given a collection of images all belonging to the same object category and the goal is to jointly segment the common object from all the images. For this, I develop a stagewise active approach to segmentation propagation: in each stage, the images that appear most valuable for human annotation are actively determined and labeled by human annotators, then the foreground estimates are revised in all unlabeled images accordingly. In order to identify images that, once annotated, will propagate well to other examples, I introduce an active selection procedure that operates on the joint segmentation graph over all images. It prioritizes human intervention for those images that are uncertain and influential in the graph, while also mutually diverse. Building on this, I also introduce the problem of measuring compatibility between image pairs for joint segmentation. I show that restricting the joint segmentation to only compatible image pairs results in an improved joint segmentation performance.

Finally, I propose a *semi-supervised approach for segmentation propagation in video*. Given human supervision in some frames of a video, this information can be propagated through time. The main challenge is that the foreground object may move quickly in the scene at the same time its appearance and shape evolves over time. To address this, I propose a higher order *supervoxel label consistency* potential which leverages bottom-up supervoxels to enforce long-range temporal consistency during propagation. I also introduce the notion of a *generic pixel-level objectness* in images and videos by training a deep neural network which uses appearance and motion to automatically assign a score to each pixel capturing its likelihood to be an "object" or "background". I show that the human guidance in the semi-supervised propagation algorithm can be further augmented with the generic pixel-objectness scores to obtain an even more accurate foreground segmentation in videos.

Throughout, I provide extensive evaluation on challenging datasets and also compare with many state-of-the-art methods and other baselines validating the strengths of proposed algorithms. The outcomes across several different experiments show that the proposed human-machine collaboration algorithms achieve accurate segmentation of foreground objects in images and videos while saving a large amount of human annotation effort.

Table of Contents

Acknowledgments **v**

Abstract **viii**

List of Tables **xvi**

List of Figures **xviii**

Chapter 1. Introduction **1**

1.1 Overview of Thesis 8

1.1.1 Interactive image segmentation with active human input 9

1.1.2 Interactive image and video segmentation with point clicks 11

1.1.3 Active segmentation propagation in image collections . . 12

1.1.4 Predicting compatibility for joint segmentation of image pairs 15

1.1.5 Supervoxel consistent foreground propagation in video . 16

1.1.6 Pixel objectness in images and videos 18

1.2 Main Contributions 21

Chapter 2. Related Work **24**

2.1 Segmenting objects in individual images 24

2.1.1 Interactive image segmentation 25

2.1.2 Fully automatic image segmentation 29

2.1.3 Strongly supervised class-specific methods 31

2.2 Segmenting objects in weakly-supervised image collections . . 31

2.2.1 Joint segmentation of weakly-supervised image collections 32

2.2.2 Segmentation propagation from human segmented examples 34

2.2.3 Compatibility for joint segmentation 35

2.3 Segmenting objects in videos 36
2.3.1 Unsupervised video segmentation 37
2.3.2 Interactive video segmentation 38
2.3.3 Semi-supervised video segmentation propagation 39
2.3.4 Supervised video segmentation 41
2.4 Roadmap 43

Chapter 3. Interactive image segmentation with active human input 45
3.1 Interactive segmentation model 48
3.2 Learning segmentation difficulty per modality 50
3.3 Predicting difficulty on novel images 54
3.4 Annotation choices under budget constraints 55
3.5 Results 58
3.5.1 Datasets and baselines 58
3.5.2 Predicting difficulty per modality 61
3.5.3 Annotation choices to meet a budget 64
3.5.4 Application to recognition 66
3.5.5 User study 67
3.6 Conclusion 69

Chapter 4. Interactive image and video segmentation with point clicks 71
4.1 Generating foreground region proposals 75
4.2 Click Carving for discovering an object mask 77
4.3 User clicking strategies 81
4.4 Propagating the mask through the video 84
4.5 Results 85
4.5.1 Datasets and metrics 85
4.5.2 Click Carving for discovering an object mask 88
4.5.3 Click Carving for complete video segmentation 94
4.6 Conclusion 103

Chapter 5. Active segmentation propagation in image collections 108

5.1 Region proposals and descriptors 112
5.2 Semi-automatic joint foreground segmentation 114
5.3 Active selection for propagation 119
5.4 Results 123
    5.4.1 Datasets and baselines 123
    5.4.2 Active segmentation propagation 126
    5.4.3 Weakly supervised foreground segmentation 130
5.5 Conclusion 132

Chapter 6. Predicting compatibility for joint segmentation of image pairs 136

6.1 Single-image segmentation engine 139
6.2 Paired-image cosegmentation engine 141
6.3 Learning cosegmentation compatibility to predict partners . . 144
6.4 Results 150
    6.4.1 Datasets and baselines 150
    6.4.2 Results on MIT Object Discovery dataset 153
    6.4.3 Results on Caltech-28 dataset 157
6.5 Conclusion 159

Chapter 7. Supervoxel consistent foreground propagation in video 160

7.1 Space-time MRF graph structure 165
7.2 Energy function with supervoxel label consistency 166
    7.2.1 Unary potential 167
    7.2.2 Pairwise potential 169
    7.2.3 Higher order potential 169
7.3 Energy minimization and parameters 171
7.4 Results 172
    7.4.1 Datasets and baselines 172
    7.4.2 Results on SegTrack dataset 174
    7.4.3 Results on YouTube-Objects dataset 177
    7.4.4 Results on Weizmann dataset 180
7.5 Conclusion 182

Chapter 8. Pixel objectness in images and videos **184**
8.1 Appearance stream . . . 189
8.2 Motion stream . . . 192
8.3 Fusion model . . . 197
8.4 Semi-supervised extension . . . 198
8.5 Results . . . 199
  8.5.1 Generalization of appearance stream . . . 200
    8.5.1.1 Datasets, baselines and metrics . . . 201
    8.5.1.2 MIT Object Discovery dataset . . . 204
    8.5.1.3 ImageNet-Localization dataset . . . 205
    8.5.1.4 ImageNet-Segmentation dataset . . . 207
    8.5.1.5 Qualitative results . . . 208
  8.5.2 Video Segmentation Results . . . 209
    8.5.2.1 Datasets, baselines and metrics . . . 210
    8.5.2.2 DAVIS dataset . . . 215
    8.5.2.3 YouTube-Objects dataset . . . 218
    8.5.2.4 Segtrack-v2 dataset . . . 219
    8.5.2.5 Qualitative evaluation . . . 220
8.6 Conclusion . . . 221

Chapter 9. Future Work **225**

Chapter 10. Conclusion **228**

Bibliography **231**

Vita **252**

List of Tables

3.1 Application of annotation modality prediction to recognition. . 66

4.1 Quantitative analysis of Click-carving based proposal selection for video frames. 90
4.2 Quantitative analysis of Click-carving based proposal selection for images. 91
4.3 Quantitative results for video segmentation using Click Carving on Segtrack-v2 dataset. 97
4.4 Quantitative results for video segmentation using Click Carving on VSB100 dataset. 99
4.5 Comparison of Click Carving with TouchCut 102

5.1 Quantitative results for weakly supervised segmentation on MIT dataset. 131
5.2 Quantitative results for weakly supervised segmentation on ImageNet dataset. 132

6.1 Overlap accuracy on the MIT Object Discovery dataset. . . . 154
6.2 Comparison to state-of-the-art cosegmentation methods on the MIT Object Discovery dataset. 156
6.3 Overlap accuracy on the Caltech-28 dataset. 157
6.4 Comparison to state-of-the-art cosegmentation algorithms on the Caltech-28 dataset. 158

7.1 Comparison of supervoxel based propagation with existing methods on SegTrack. 175
7.2 Ablation studies for supervoxel based propagation on SegTrack. 176
7.3 Comparison of supervoxel based propagation with existing methods on YouTube-Objects. 178

8.1 Comparison with state-of-the-art methods on MIT Object Discovery dataset. 205

8.2 Comparison with state-of-the-art methods on ImageNet-Localization dataset. 206
8.3 Comparison with state-of-the-art methods on ImageNet-Segmentation dataset. 207
8.4 Video object segmentation results on DAVIS dataset. 216
8.5 Video object segmentation results on YouTube-Objects dataset. 217
8.6 Video object segmentation results on Segtrack-v2 dataset. . . 219

List of Figures

1.1 Overview of the different foreground segmentation problems. . 5
1.2 Illustration of cost versus accuracy trade-off in interactive segmentation. 10
1.3 Overview of the Click Carving algorithm. 11
1.4 Weakly supervised segmentation of an image collection. 13
1.5 Overview of active image segmentation propagation algorithm. 14
1.6 Motivation for predicting compatibility between image pairs for joint segmentation. 15
1.7 Automatic propagation of foreground segmentation in videos from human-labeled frames. 17
1.8 Overview of the generic pixel-level objectness in images and videos. 19

3.1 Illustration of cost versus accuracy trade-off in interactive segmentation. 46
3.2 Possible modes of annotation 49
3.3 Overview of interactive image segmentation datasets. 58
3.4 Difficulty prediction accuracy for within dataset and cross-dataset experiments 61
3.5 Qualitative results for annotation difficulty prediction 62
3.6 Choosing annotation modalities to meet a budget 65
3.7 Annotation choices under a budget with real user data. 68

4.1 Generation of object region proposals using both static and dynamic cues in a video frame. 75
4.2 Click Carving based foreground segmentation. 80
4.3 Example video sequences from Segtrack-v2, VSB100 and iVideoSeg datasets. 86
4.4 Qualitative results for Click Carving. 93
4.5 Cost vs. Accuracy analysis for video segmentation using Click Carving on Segtrack-v2 and VSB100 dataset. 100

4.6 Cost vs. Accuracy analysis for video segmentation using Click Carving on iVideoSeg dataset. 101

4.7 Qualitative results for video segmentation using Click Carving on Segtrack-v2 dataset. 105

4.8 Qualitative results for video segmentation using Click Carving on iVideoSeg dataset. 106

4.9 Qualitative results for video segmentation using Click Carving on VSB100 dataset. 107

5.1 Overview of the proposed active image segmentation propagation algorithm. 112

5.2 Examples from ImageNet dataset. 124

5.3 Example active annotation choices for the MIT dataset. 126

5.4 Active propagation for varying amounts of human annotation for ImageNet dataset. 127

5.5 Active propagation for varying amounts of human annotation for MIT dataset. 128

5.6 Qualitative results for weakly supervised segmentation on MIT and ImageNet datasets. 133

6.1 Feature illustration for predicting cosegmentation compatibility. 147

6.2 Examples from MIT Object Discovery and Caltech-28 datasets. 152

6.3 Qualitative results for predicting cosegmentation compatibility. 155

7.1 Automatic propagation of foreground segmentation in videos from manually labeled frames. 161

7.2 Visual example illustrating supervoxels in videos. 162

7.3 Spatio-temporal graph for supervoxel based segmentation propagation in videos. 164

7.4 Example video sequences from Segtrack-v2, YouTube-Objects and Weizmann datasets. 173

7.5 Qualitative results for segmentation propagation on SegTrack. 175

7.6 Supervoxel based propagation versus pixel flow based propagation. 178

7.7 Qualitative results highlighting the impact of higher order potentials. 179

7.8 Qualitative results for supervoxel based propagation. 180

7.9 Propagation results on Weizmann dataset. 181

8.1 Generic pixel-level objects in images and videos. 185
8.2 Network structure for the two-stream model for generic pixel-level objectness in images and videos. 191
8.3 Filtering steps to generate (pseudo)-ground truth segmentations for training motion stream. 195
8.4 Examples from MIT Object Discovery and ImageNet datasets. 201
8.5 Qualitative results for generic foreground segmentation in images. 209
8.6 Example video sequences from DAVIS, YouTube-Objects and Segtrack-v2 datasets. 210
8.7 Qualitative results for video segmentation using two-stream pixel-objectness network. 224

Chapter 1

Introduction

Computer vision has made rapid progress in recent years. However, even the most advanced computer vision systems cannot come close to the richness of human perception. As humans, we have an unique ability to understand our environment by processing and interpreting high level visual information from low level sensory data at an extremely fast rate. Computers lack this capability, but are extremely good at continuously processing large volumes of data at the lowest level of detail with great efficiency. Human-machine collaborative systems can bring these complementary strengths of the human visual system and computer algorithms together in a way which can be accurate and cost effective at the same time.

Large scale image and video annotation is one area where human-machine collaboration has high potential impact. Image and video annotation can include a variety of tasks: e.g., listing all objects in the scene, drawing boundaries of objects or describing the scene. While automatic computer vision algorithms exist for each of these tasks, they are not accurate enough to be relied upon completely. On the other hand, employing humans alone to do a task will be prohibitively expensive. Designing computer vision algorithms that can actively request human supervision as and when needed have the potential to achieve far greater accuracies than completely automatic systems and at a cost far less than employing humans alone. Moreover, partial human supervision can remove ambiguity for the vision system, and thus can greatly simplify the design of algorithms, which otherwise have to make several complex decisions regarding scene perception on their own. With the advent of modern crowd-sourcing techniques, human supervision can be obtained on-demand and in an economical fashion. Human-machine collaborative systems can therefore now operate at an extremely large scale, opening up new possibilities of research in this direction.

In recent years, major strides have been made in computer vision by leveraging large scale annotated datasets to learn powerful predictive models, most notably for object classification in images [47, 65, 77, 123, 128, 129]. Before the arrival of crowdsourcing, it was not even practical to create datasets with millions of annotated examples to facilitate learning. Modern crowd-powered datasets (e.g, ImageNet [29]) have been instrumental in the success of current deep learning systems which need a large amount of supervised data to excel.

However, even with good crowdsourcing tools, image and video annotation for large datasets remains a rather costly undertaking in terms of both time and money. In particular, gathering high quality *spatial annotations*—pixel-level foreground masks—is challenging. First of all, the physical mousing actions required to delineate objects from background are time intensive (e.g., compared to simply labeling which object is present). Furthermore, non-expert annotators exhibit inconsistencies in how precisely they mark object boundaries, which means leveraging the crowd typically requires some finessing and “re-dos”.

As a result, datasets with spatial annotations lag seriously behind their category-labeled counterparts. For example, while ImageNet [29] is comprised of an impressive 14M labeled images, there are orders of magnitude fewer spatial annotations—only 1M images (7% of the dataset) offer bounding box annotations, and only 4K images (0.03%) have foreground segmentation masks [44]. More recently, another dataset Microsoft COCO[89] with 328,000 images was collected containing more difficult object instances than ImageNet. Collecting spatial annotations for 2,500,000 object instances in these images turned out to be a very time consuming task requiring over 22 worker hours per 1,000 segmentations and about $400,000 in cost. The problem gets even more challenging for video datasets. The sheer volume of video data available on the internet can be an incredible source of training data for learning richer object representations. However, the cost of annotating them is prohibitively large; hence, no large-scale video dataset with spatial annotations for objects currently exists to facilitate this direction of research.

This scarcity of foreground-labeled image and video collections is problematic given their high potential utility. Apart from being useful in building training sets for object detectors, good foreground segmentation of objects can improve visual search by focusing on the region of interest. Furthermore, several interesting computer graphics applications such as data-driven image synthesis, 3D reconstruction, and image re-targeting can directly use a well-segmented image database. Similarly, a good foreground segmentation in video can be very helpful in activity recognition, video summarization, and post-production video editing.

In this thesis, I explore the idea of human-machine collaboration for the problem of foreground object segmentation in images and videos. In my work, I have developed algorithms which effectively and efficiently make use of human guidance for accurately segmenting foreground objects in images and videos. The segmentation problems that I explore in this thesis can be broadly divided into three categories:

1. **Interactive image segmentation:** In interactive segmentation algorithms, the human annotator is asked to provide a coarse spatial annotation (e.g., draw a bounding box, scribbles, or point clicks) on the object of interest. This coarse input is then used to guide the underlying segmentation algorithm which converts the coarse human input into a fine-grained final segmentation for the object (see Figure 1.1 (top) for an example).

2. **Weakly supervised segmentation of image collections:** In this setting, a pool of images known to contain the same object category (weak supervision) is considered. Such a collection of images can be easily obtained from the Web using a simple keyword search. A joint

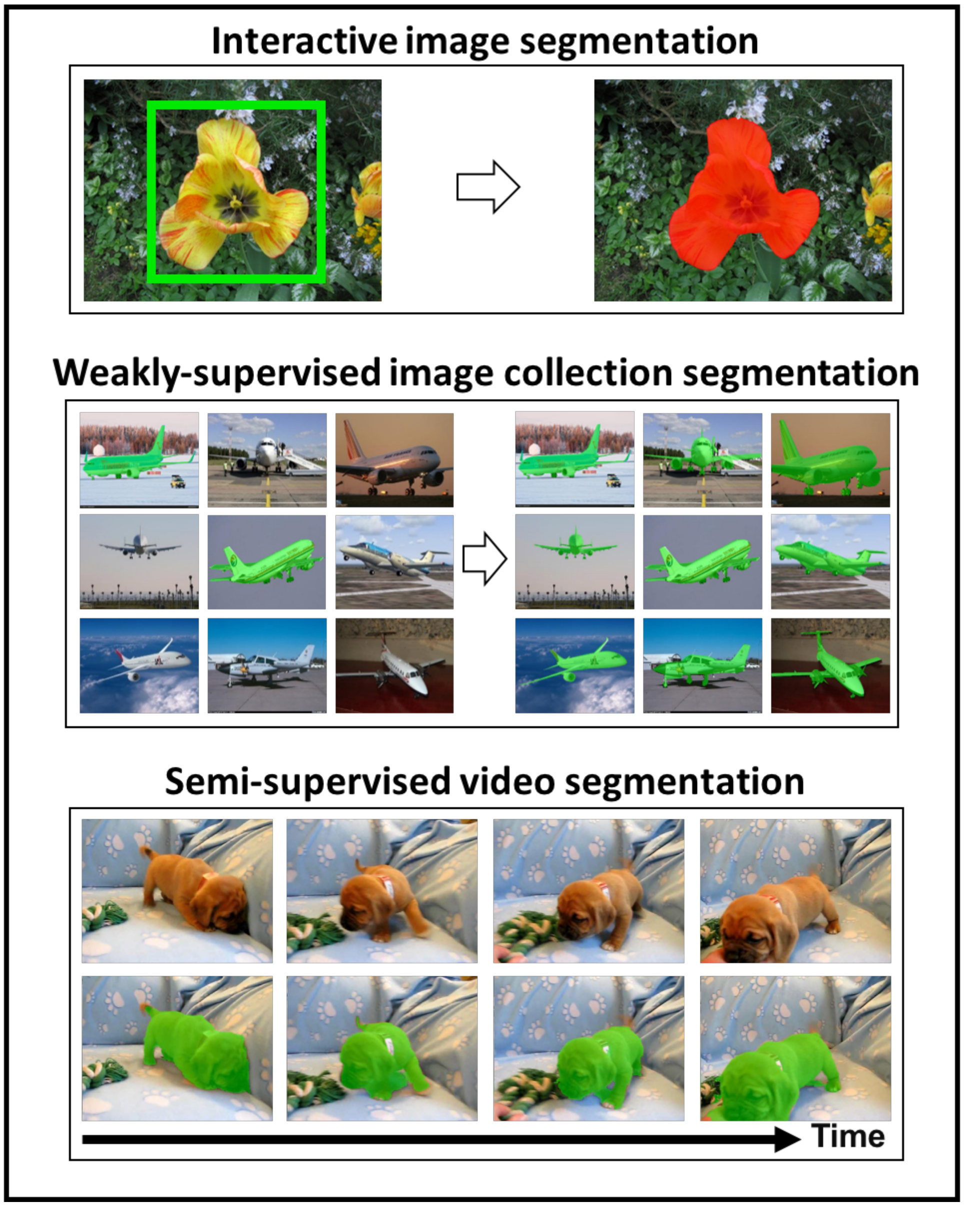

Interactive image segmentation
Weakly-supervised image collection segmentation
Semi-supervised video segmentation
Time

Figure 1.1: Overview of the different segmentation problems addressed in this thesis. Best viewed in color.

segmentation approach which discovers common patterns in the collection is typically used (see Figure 1.1 (middle) for an example). A semi-supervised variant of this problem assumes that human annotations on a subset of images in the collection are provided.

3. **Semi-supervised video segmentation:** Given a video as input and some subset of frames segmented by human annotators, the goal here is to propagate these region segmentations to all other unsegmented frames, to obtain a segmentation for the entire video (see Figure 1.1 (bottom) for an example).

The novel contributions in this thesis for each of the above mentioned problems come from addressing one or more of the following questions which naturally arise in a human-machine collaboration system for segmentation:

- **How to annotate?** Human annotators can provide partial supervision in many ways. For example in interactive segmentation, the annotator can initialize the algorithm using a bounding box or a rough outline around the object. Different forms of human input come with different costs. Their effectiveness is a function of the image content. Depending on the complexity of the object of interest, some images may need more detailed human involvement and some less. While previous methods assume a fixed form of human input, I show that this cost versus accuracy trade-off can be exploited by actively choosing the mode of human input for interactive segmentation.

Moreover, I also show how can we go beyond the traditional and more involved forms of human interaction (i.e., bounding boxes and scribbles). I do this through a novel formulation of the interactive segmentation problem which results in accurate foreground segmentations yet only requires human guidance in form of point clicks, which are very simple to provide and also are very cost effective.

- **What to annotate?** Depending on the annotation budget, it might be possible to get annotation only on a subset of data. I show that in instances where information can be propagated (e.g., weakly supervised image collections) this subset can be chosen in an active manner depending on its utility for other unsegmented instances. I show that this active selection results in large savings in human annotation costs when compared with other naive methods of making these decisions (e.g., random selection).
- **How to propagate?** Given some human annotations on a subset of data (e.g., some frames in a video or a subset of images in a collection), I show that the structural patterns in the data collection (e.g., temporal continuity in video, similarity in related images) can be used to propagate information to other unsegmented instances. I show that these propagation algorithms effectively exploit the structural similarities within the data to transfer information which allows us to restrict human annotation to only a few select data points, thus substantially reducing annotation costs.

Addressing the aforementioned questions in each segmentation problem category will lay the foundations of building computer vision systems which make an effective use of human machine collaboration to achieve high performance but remain cost effective at the same time. Throughout the chapters in my thesis, I introduce novel contributions of my work in the context of these important questions and how the proposed algorithms address them. I compare with existing state-of-the-art algorithms and establish the advantages our proposed methods have over existing techniques. Next, I provide a brief overview of the main components of my thesis.

1.1 Overview of Thesis

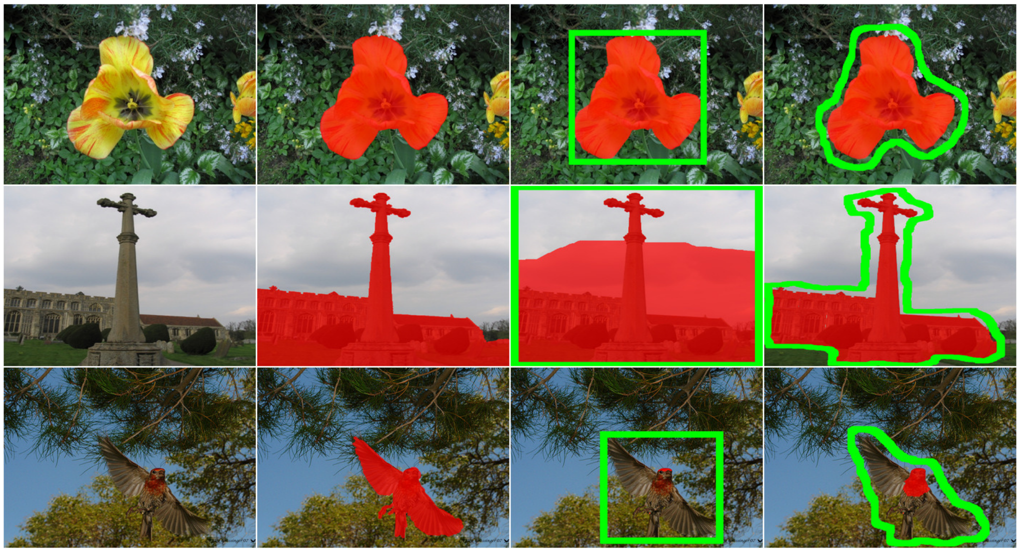
In this section, I will provide a brief summary of the main ideas and insights from my thesis. I will first present a technique for actively making annotation choices for the problem of interactive image segmentation (Sec. 1.1.1). I will then discuss a novel formulation for interactive image segmentation which only makes use of a simple point click based human interaction (Sec. 1.1.2). Moving on to *collections* of images or frames, I will present my algorithm for active annotation and segmentation propagation for segmenting objects in weakly supervised image collections (Sec. 1.1.3). Building on this concept, I will also present a technique which enables a data-driven way of predicting compatibility between image pairs for joint segmentation (Sec. 1.1.4). Finally, I will introduce my novel algorithms for doing foreground segmentation propagation in videos (Sec. 1.1.5 and 1.1.6).

1.1.1 Interactive image segmentation with active human input

Research on *interactive segmentation* considers how a human can work in concert with a segmentation algorithm to efficiently identify the foreground region [9, 13, 45, 66, 82, 100, 119]. The idea is to leverage the respective strengths of both the human and the algorithm. Humans can easily identify the foreground, hence provide high-level guidance—in the form of coarse spatial annotations. Meanwhile, the algorithm can easily assign pixels to objects based on their low-level properties, converting the high level guidance into a fine-grained segmentation. Often this is done by constructing a foreground color model from the user-indicated regions, then optimizing foreground/background labels on each pixel (e.g., using graph cuts [13, 119]).

Existing methods assume that the user always gives input in a particular form (e.g., a bounding box or a scribble), and so they focus on how to use that input most effectively. However, fixing the input modality in advance leads a suboptimal tradeoff in human and machine effort. Each input type has its own degree of precision, but also has a proportional cost associated with it. At the same time, depending on its content, an image may be better served by one form or another. Figure 1.2 illustrates this with an example.

The tradeoffs are clear, but what is a system to do about it? A system which can determine what tool works best for an image can result in large savings in human effort. The problem is that it needs to do it before the human uses the tool! To address this problem, I propose an algorithm which can leverage image properties to predict how successful a given form of user



(a) Image (b) Ground Truth (c) Bounding Box (d) Sloppy Contour

Figure 1.2: Interactive segmentation results (shown in red) for three images using various annotation strengths (marked in green). Note how the most effective mode of input depends on the image content. My method in Chapter 3 predicts the easiest input modality that will be sufficiently strong to successfully segment a given image. Best viewed in color.

input will be, once handed to an interactive segmentation algorithm [52]. Using these predictions, we can optimize the mode of input requested on new images a user wants segmented. Whether given a single image that should be segmented as quickly as possible, or a batch of images that must be segmented within a specified time budget, the proposed algorithm can be used to select the easiest modality that will be sufficiently strong to yield high quality segmentations. Chapter 3 introduces the complete approach and also provides experimental results.

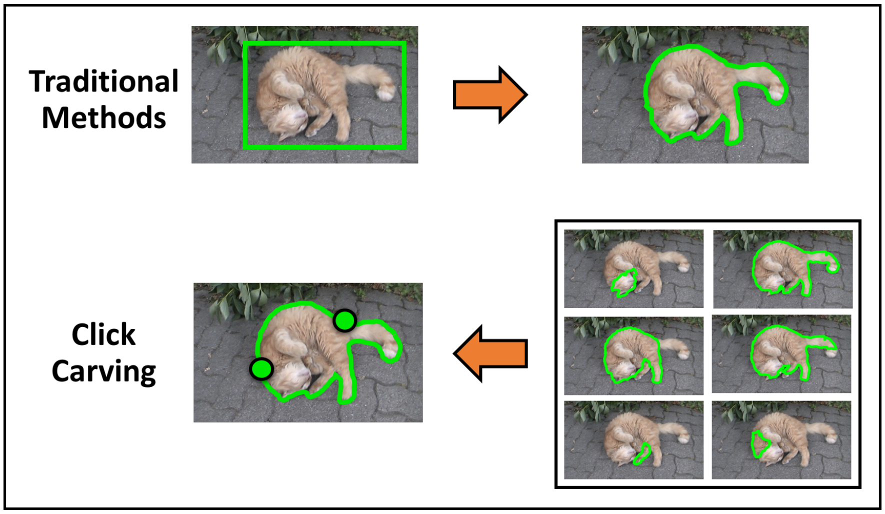


Figure 1.3: Overview of the ClickCarving algorithm (Chapter 4) for interactively segmenting objects using point clicks. Best viewed in color.

1.1.2 Interactive image and video segmentation with point clicks

Thus far, I have discussed a method using which we can optimize for the modality of human interaction in the traditional interactive segmentation pipeline. Regardless of the exact input modality, the common assumption in all existing methods is to completely rely on the user's input to learn about the object's appearance to generate a segmentation output. Reliably learning about an object's appearance requires a reasonable number of data points on the object. This seemingly precludes the use of simple human interactions such as point clicks, which provide very little information to learn a complex appearance model. In this sense, in traditional interactive segmentation algorithms, information flows first from the user to the system and thus far forms the bottleneck for using simple human interactions such as point clicks.

To address this problem, I propose a novel formulation [58] of the interactive segmentation problem which reverses this standard flow of information. The key idea is for the system itself to first hypothesize plausible object segmentations in a given image, and then allow the user to efficiently and interactively prioritize those hypotheses. Such an approach stands to reduce human annotation effort, since the user can use very simple feedback to guide the system to its best hypotheses, often just a couple of clicks on the boundary of the true object (see Figure 1.3). This algorithm called *Click Carving* essentially uses the clicks to "carve" away erroneous hypotheses whose boundaries disagree with the clicks. This process iterates (typically 2-3 times), and each time the system revises the top ranked hypotheses set, until the user is satisfied and chooses a final segmentation mask. Chapter 4 introduces the complete approach and also provides experimental results. Click Carving can also be effectively used to segment objects in videos. This is achieved by first segmenting a video frame using Click Carving and then propagating it to all other frames using the algorithm that I will describe in Chapter 7.

1.1.3 Active segmentation propagation in image collections

There has been a lot of recent interest in jointly segmenting a pool of images known to contain the same object category (e.g., a collection of "airplane" images, see Figure 1.4) [2, 30, 44, 64, 121, 122, 124, 131, 140]. While interactive segmentation (discussed above) works well for segmenting images individually, a joint segmentation approach that can directly use this weak

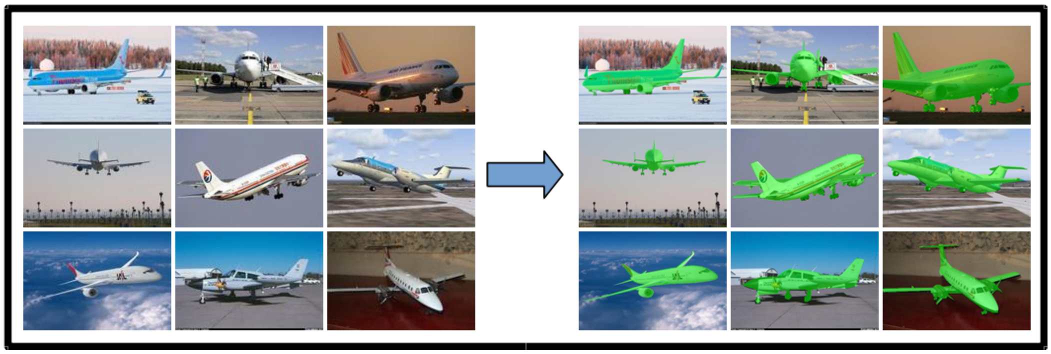


Figure 1.4: Weakly supervised segmentation of an image collection. On the left side is a collection of "airplane" images and on the right the desired segmentation of the common object "airplane" is shown with a green overlay. Best viewed in color.

supervision may be more effective here. The main idea is to leverage the weak supervision by exploiting the repeated patterns to jointly segment out the foreground per image.

On the one hand, this paradigm is attractive for its low manual effort, especially since such weakly labeled pools of images are often readily available on the Web from keyword search. On the other hand, the resulting fully automatic segmentations are necessarily imperfect. No matter the method, the foreground masks will hit a ceiling of accuracy since the segmentation task is underconstrained even with weak supervision.

For this transductive setting, where the goal is to collect spatial annotations for every image in the collection, I propose an intermediate solution. Rather than relying solely on human-provided segmentations (accurate but too expensive) or automatic segmentations (inexpensive but too inaccurate), I propose a *semi-automatic segmentation propagation* approach [54]. The key

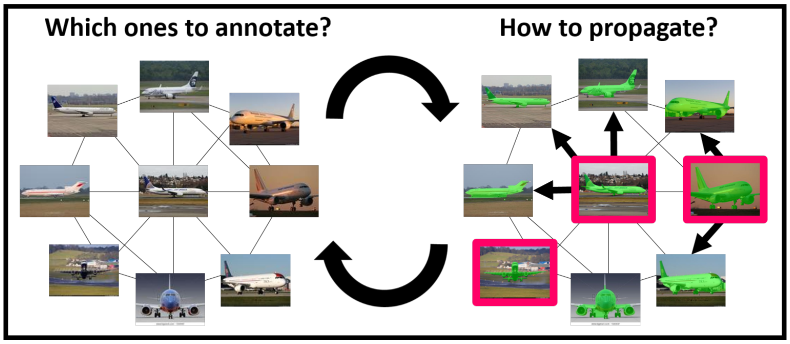


Figure 1.5: The proposed active image segmentation propagation method (Chapter 5) alternates between: (1) Actively choosing images which once annotated by humans will likely be most useful in propagating segmentations to other images and (2) Given human annotations on actively chosen images (marked in pink), propagating them (dark arrows) to generate segmentations for other unlabeled images. Best viewed in color.

idea is to develop a stagewise active approach: in each stage, the system actively determines the images that appear most valuable for human annotation, and then revises the foreground estimates in all unlabeled images by propagating information from the labeled ones (see Figure 1.5). In order to identify images that, once annotated, will propagate well to other examples, I introduce an active selection procedure that operates on the joint segmentation graph over all images. The edges in this graph capture inter-image similarities by computing distances in a predefined feature space. The active selection algorithm prioritizes human intervention for those images that are uncertain and influential in the graph, while also mutually diverse. In this way, we neither restrict ourselves to the saturation point of the fully automatic methods, nor do we get large volumes of data labeled by humans. Chapter 5 introduces the complete approach and also provides experimental results.

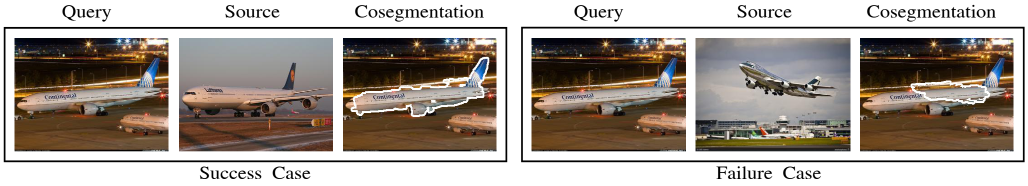


Figure 1.6: Motivation for predicting compatibility between image pairs for joint segmentation (Chapter 6). When an image pair share strong foreground similarity, their joint segmentation is successful (left). However, when incompatible images are used—even from the same object category—joint segmentation fails (right).

1.1.4 Predicting compatibility for joint segmentation of image pairs

Thus far, I have discussed how to jointly segment images in a weakly supervised image collection by building joint segmentation graphs over images to perform segmentation propagation and active selection. There, to build the joint segmentation graph, distances between image features were used to capture inter-image similarities and assign edge weights. The underlying assumption is that images which look similar in this feature space are structurally similar and thus should be compatible for joint segmentation.

Nonetheless, this assumption does not hold strongly in all cases. Intra-class appearance variation remains a major obstacle to accurate joint segmentation. This is problematic, since coupling the "wrong" images together, where the foreground objects may look quite different can actually deteriorate the joint segmentation performance (see Figure 1.6 for examples). Instead of assuming that all images are compatible for joint segmentation or simply relying on global image similarity, I propose to *predict* which pairs of images are likely to be most compatible when paired together for joint segmentation [53].

Given an input image and a pool of candidate images sharing the same weak label (e.g., a batch of “airplane” images like above), the goal is to find the candidate that, when coupled with the input image, will most boost its foreground accuracy if they are jointly segmented. To this end, I develop a learning-to-rank approach that identifies good partners, based on paired descriptors capturing the amenability to joint segmentation of an image pair. I show that pairing with the right partners results in an improved segmentation performance as opposed to pairing with random partners or simply relying on image similarity. Chapter 6 introduces the complete approach and also provides experimental results.

1.1.5 Supervoxel consistent foreground propagation in video

Previous sections gave an overview of my proposed approach for actively seeking human annotation for segmenting single images or a collection of images. I will now preview my work on semi-supervised segmentation propagation in videos. Different from the algorithms for segmenting images, a video segmentation algorithm can directly benefit from the temporal continuity in video data. This temporal prior facilitates propagation of information (e.g., human annotations) through time.

In video, the *foreground object segmentation* problem consists of identifying those pixels that belong to the primary object(s) in every frame. A resulting foreground object segment is a space-time “tube” whose shape may deform as the object moves over time. In the semi-supervised foreground prop-

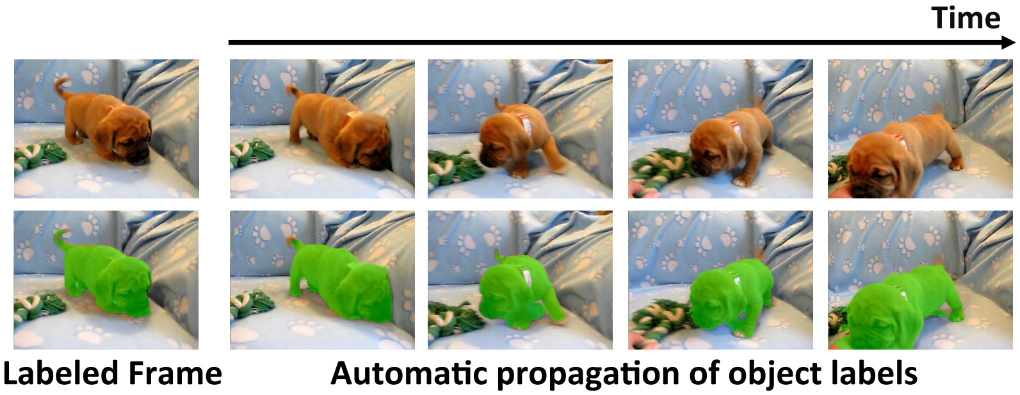


Figure 1.7: Automatic propagation of foreground segmentation in videos from a single/multiple labeled frame(s). Here we see human drawn segmentation on a single frame being propagated to all the other frames in the video using my supervoxel based propagation (Chapter 7) algorithm [57]. Best viewed in color.

agation task, the goal is to take the foreground object segmentation drawn on few frames by human annotators and accurately propagate it to the remainder of the frames (see Figure 1.7).

Graph-based methods are commonly used for propagating foreground regions in video [6, 36, 118, 135, 141]. The general idea is to decompose each frame into spatial nodes for a Markov Random Field (MRF), and seek the foreground-background label assignment that maximizes both appearance consistency with the supplied labeled frame(s) as well as label smoothness in space and (optionally) time. Despite encouraging results, these methods face an important technical challenge. In video, reliable foreground segmentation requires capturing *long-range* connections as an object moves and evolves in shape over time. However, current methods restrict the graph connectivity to local cliques in space and time, thus offer only a myopic view of consistency and can be misled by inter-frame optical flow errors.

To alleviate these problems, I propose a foreground propagation approach using *supervoxel* higher order potentials [57]. Supervoxels—the space-time analog of spatial superpixels—provide a bottom-up volumetric segmentation that tends to preserve object boundaries [26, 40, 43, 157, 158]. To leverage their broader structure in a graph-based propagation algorithm, the proposed method augments the usual adjacency-based cliques with potentials for supervoxel-based cliques. These new cliques specify soft preferences to assign the same label (foreground or background) to superpixel nodes that occupy the same supervoxel. This allows us to enforce long-range temporal constraints while propagating segmentations in videos. Chapter 7 introduces the complete approach and also provides experimental results.

1.1.6 Pixel objectness in images and videos

In the previous section, foreground segmentation in the video was primarily driven by the "video-specific" information which was learned from the human segmented frame. In this section, I explore the idea of a "generic" pixel level objectness in the context of image and video segmentation that generalizes across large number of object categories. More specifically, I explore whether it is possible to learn a model which can assign a score to every pixel in an image or a video frame measuring its likelihood to be a pixel belonging to any foreground object. The key intuition lies in the idea that there are

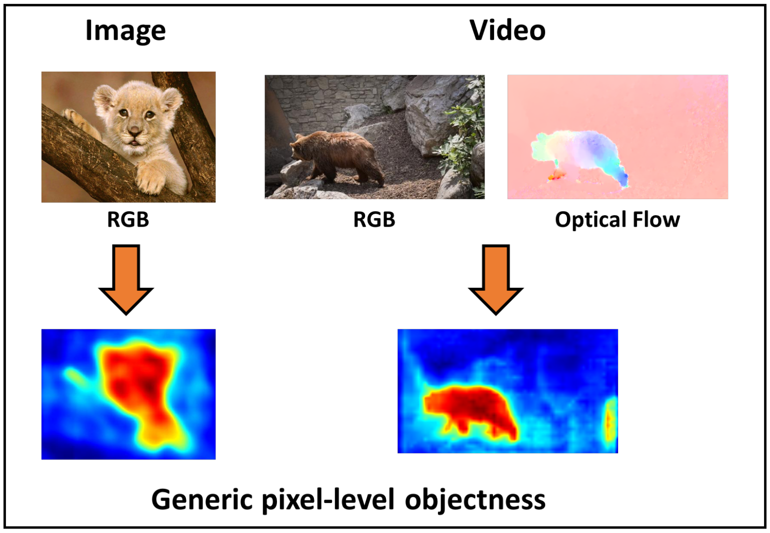


Figure 1.8: Overview of the generic pixel-level objectness (Chapter 8) in images and videos. The heatmaps below show the per-pixel objectness scores assigned to the example image and video frame. The red values reflect high objectness, the blue reflect low objectness. These non-category-specific pixel objectness scores provide a strong prior on the foreground objects in images and videos that can be incorporated in other human-machine collaboration algorithms presented in this thesis. Best viewed in color.

some inherent properties of an object's appearance and motion which allow us to separate it from the background. However, the generic objectness signal from both appearance and motion is complex. Hand-designing rules which capture these rich signals and generalize to thousands of object categories is non-trivial.

Instead, I propose an end-to-end trainable model that draws on the respective strengths of generic (non-category-specific) object appearance and motion in a unified framework [55, 56]. Specifically, I develop a novel two-stream fully convolutional deep segmentation network, where individual streams encode generic appearance and motion cues and can be trained to predict per-pixel objectness maps. For images, naturally we rely only on the appearance. For videos, we rely both on the appearance derived from the video frame and its corresponding optical flow (see Figure 1.8). The two streams are fused in the network to produce a per-pixel objectness map for each frame. This allows us to learn from both the signals in a unified manner, leading to a true synergy between appearance and motion for segmenting objects in video. The per-pixel objectness maps can naturally be thresholded to obtain a binary foreground-background segmentation as well.

Finally, I also introduce a semi-supervised extension to the generic pixel-level objectness approach. The key idea here is to combine the respective strengths of the generic pixel-level objectness with the video specific information learned from the human segmented frames. This is especially useful when there is ambiguity about what exactly is the object of interest or the object undergoes significant changes across time and propagation alone is not sufficient. This is done by incorporating generic pixel-level objectness output as additional unaries in my supervoxel-based propagation algorithm, augmenting the unaries derived from the human annotation. Together, it results in an even better performance than what can be achieved individually by each method. Chapter 8 introduces the complete approach and also provides experimental results.

1.2 Main Contributions

My thesis makes several contributions in bringing humans and machines together to effectively and efficiently solve the problem of segmenting foreground objects in images and videos. In particular,

- a method to actively select the input modality which is best suited for a given image when using traditional interactive segmentation algorithms (Chapter 3). This acknowledges that a variable amount of manual effort is required for different inputs and accounting for it leads to a substantial savings in human annotation costs.

- a batch-extension of the previous method, which when given a fixed annotation budget for a group of images, can make a collective decision depending on each image's suitability for each annotation modality (Chapter 3).

- a novel formulation of the interactive image segmentation problem called *Click Carving* which allows images to be interactively segmented using a very simple form of human interaction—point clicks (Chapter 4). It is much more efficient and requires much less annotation effort than existing algorithms.

- a joint segmentation propagation method for weakly supervised image collections (Chapter 5). It gives state-of-the-art results in a pure weakly supervised setting and can also effectively propagate when a subset of images is already labeled by humans.

- an active selection algorithm, which accounts for influence, diversity, and uncertainty while making annotation choices for joint segmentation of weakly supervised image collections (Chapter 5).

- a novel learning-to-rank based algorithm for measuring compatibility between image pairs for joint segmentation (Chapter 6).

- a semi-supervised segmentation propagation method for videos, which uses supervoxels to define a higher order potential in order to enforce *long term temporal consistencies* in the propagation (Chapter 7).

- an end-to-end trainable two-stream fully convolutional deep segmentation model which captures the generic notion of pixel-level objectness in images and videos. This results in a state-of-the-art automatic image and video object segmentation system (Chapter 8).

- a semi-supervised extension to the two-stream model which combines its generic pixel level objectness (Chapter 8) with the semi-supervised supervoxel-based segmentation propagation method (Chapter 7).

Overall my thesis realizes the potential of human-machine collaboration for the problem of segmenting objects in images and videos. The algorithms presented in this thesis have many potential real-world applications especially in enabling the large-scale collection of image and video segmentation annotations much more economically along with improving search, summarization, high level understanding of scenes, and several aspects of computer graphics.

Throughout, I test the methods on challenging benchmark datasets and show that the proposed methods outperform several state-of-the-art methods and relevant baselines.

In the following chapter, I will discuss the background material and related work for my thesis.

Chapter 2

Related Work

In this chapter, I review the literature and discuss existing techniques related to the research presented in this thesis. I group them into three main categories. First, I overview the existing work on segmenting objects in individual images (Section 2.1). Next, I describe existing methods which work with weakly supervised image collections to jointly segment the common objects among them (Section 2.2). Finally, I provide an overview of existing methods for segmenting objects in video (Section 2.3). In each section, I describe different aspects of the problem and how existing methods try to address them. Simultaneously, I also discuss important similarities and differences that exist between my work and existing methods.

2.1 Segmenting objects in individual images

In this section, I review the existing work that tries to address the problem of object segmentation in individual images. Here, I only consider those methods which segment a single image at one time. There is no propagation of information across images or videos as we will see in the remaining two sections. Existing methods which fall under this category occupy a wide spectrum. These include: **1) Interactive methods:** which require a human-in-the-loop to provide guidance while segmenting an object of interest, **2) Fully automatic methods:** which aim to segment objects without any human interaction, and **3) Strongly supervised methods:** which require a large amount of training data and can only segment objects from a predefined set of categories.

Next, I provide a brief review of the existing methods from these categories and compare them with my proposed work for actively tailoring human input (Chapter 3) and also the use of point clicks (Chapter 4) for interactively segmenting individual images.

2.1.1 Interactive image segmentation

Research on *interactive segmentation* in images considers how a human can work in concert with a segmentation algorithm to efficiently identify the foreground region [9, 13, 45, 66, 82, 100, 119]. Early interactive segmentation methods include active contours [66] and intelligent scissors [100], where a user draws loose contours that the system snaps to a nearby object. Alternatively, a user can indicate some foreground pixels—often with a bounding box or mouse scribble—and then use graph cuts to optimize pixel label assignments based on a foreground likelihood and local smoothness prior [13, 119]. Building on this idea, recent work develops co-segmentation [9], topological priors [82], shape constraints [45], and simulated human user models [72].

In all prior methods, the user's annotation tool is fixed. No matter what is the input image that needs to be segmented, the annotator uses the same mode of human interaction for all of them. This is sub-optimal, since different inputs may have varying degrees of complexity, hence may require human input at different granularities. In my work (Chapter 3), I show that the user's input modality can be tailored to the image to achieve best graph cut segmentation results with minimal effort. In other words, I show that depending on the content of the image the mode of human interaction can be actively chosen, which results in large savings for annotation costs.

Active learning also helps in minimizing the annotation costs by reducing the amount of labeled examples needed to train a recognition system. Most active learning systems are tied to a particular classifier of interest and typically try to get class labels for sequentially selected samples based on how they reduce category uncertainty. In some such cases, region labels [127, 139, 142–144] have also been explored. In particular Vijayanarasimhan et al. [143, 144] also considers different levels of granularity of human annotation to build a reliable classifier. In contrast, my work in Chapter 3 on adapting the granularity of user interaction depending on the input image is class-independent i.e., it is not limited to a fixed set of categories, and it is able to segment arbitrary images and addresses interactive segmentation, not recognition.

Actively optimizing annotation requests for individual images has also been studied in various other settings. For instance, in video segmentation, the most useful frames to annotate are found with tracking uncertainty measures [141, 145, 146]. In object recognition, a human is asked to click on object parts, depending on what seems most informative [147]. In interactive co-segmentation, the system guides a user to scribble on certain areas of certain images to reduce foreground uncertainty [9, 148]. Like my work in Chapter 3, all these methods also try to reduce human effort. However, whereas prior work predicts *which images should be annotated* (and possibly where) to minimize uncertainty, in my work I predict *what strength of annotation will be sufficient* for interactive segmentation to succeed.

The way a user interacts with an interactive segmentation system is the key behind its performance. While traditional interactive segmentation methods [9, 13, 45, 66, 82, 100, 119] have progressed a lot over the years, the underlying premise remains the same. They all require the human to first provide input, before they can generate any segmentation output. In that sense, the output segmentation is very tightly coupled with the human input and thus it requires the human to provide a sufficient amount of data points on the object for these methods to work. Even my proposed method in Chapter 3 which tailors the input granularity based on the image content relies on bounding boxes as the fastest mode of human interaction.

However, in reality even faster and simpler modes of human interaction such as *point clicks* are available. Only limited work explores click supervision for image annotation. Clicks on objects in images can remove ambiguity to help train a convolution neural network (CNN) for semantic segmentation from weakly labeled images [10], or to spot object instances in images for dataset collection [89]. Clicks on patches are used to obtain ground truth material types in [11]. However for interactive segmentation, the tight coupling between human input and segmentation output in traditional methods [9, 13, 45, 66, 82, 100, 119] thus far precludes the use of point clicks as an annotation modality. It will require a large number of point clicks for these existing models to work reliably, which defeats the purpose of using a simpler mode of human interaction.

In my work (Chapter 4), I show that simple point clicks can be effectively used for interactively segmenting objects in images and also video frames. In my work, I flip the underlying premise that exists in traditional interactive methods [9, 13, 45, 66, 82, 100, 119], by pre-generating thousands of possible segmentation outputs and then using the human guidance to quickly find the most accurate ones. This decoupling between human input and object segmentation allows for an effective use of point clicks which was not possible to do in existing methods.

There have been only two prior efforts for using clicks to do interactive segmentation, and their usage is quite different than ours. In one, a click and drag user interaction is used to segment objects [114]. A small region is first selected with a click, then dragged to traverse up in the hierarchy until the segmentation does not bleed out of the object of interest. In contrast, my proposed user-interaction is much simpler (jut a few mouse clicks or taps on the touchscreen) and the boundary clicks that I use are discriminative enough to quickly filter good segmentations.

In the other, the TouchCut system uses a single touch to segment the object using level-set techniques [151]. The object contour is grown from the initial click made by the user. Strong image boundaries can act as false positives and restrain the evolution of the object contour to reach object boundaries. In contrast, my proposed method does not have this disadvantage and significantly outperforms [151] in experiments.

2.1.2 Fully automatic image segmentation

Next, I discuss a set of methods which aim to segment objects in images without any human supervision. Note that these methods are generic in nature and are expected to work on any object category. This makes it different from the class-specific image segmentation methods, which I describe in the next section. Today there are two main strategies for generic foreground object segmentation in images: saliency and object proposals. Both strategies capitalize on properties that can be learned from images and generalize to unseen objects (e.g., well-defined boundaries, differences with surroundings, shape cues, etc.).

Saliency methods identify regions likely to capture human attention. They yield either highly localized attention maps [12, 78, 92, 105] or a complete segmentation of the prominent object [61, 86, 88, 93, 110, 162, 163]. Saliency focuses on regions that stand out, which is not the case for all foreground objects. Alternatively, *object proposal* methods learn to localize all objects in an image, regardless of their category [5, 22, 33, 51, 74, 113, 137, 165]. Proposal methods aim to obtain high recall at the cost of low precision, i.e., they must generate a large number of object proposals (typically 1000s) to accurately cover all objects in an image. This usually involves a multi-stage process: first bottom-up segments are extracted, then they are scored by their degree of "objectness". The ideas is that a downstream processing step, such as an object detector, can look at only the top scored segments and ignore the rest.

The methods in my proposed work for segmenting objects in images presented in Chapter 3 and 4 have some key advantages over both these techniques. The interactive nature of my proposed algorithms for segmenting foreground objects eliminates the need of enforcing these other priors such as the ones used in saliency methods (i.e., the object stands out from the background). As long as the human provides the required input, they can segment any foreground object whether "salient" or not.

For object proposal methods, generating thousands of hypotheses helps ensure high recall, but at the same time, it makes it difficult to automatically filter out accurate proposals from this large hypothesis set without class-specific knowledge. This is limiting where only a single hypothesis for a foreground object is desired, which my proposed methods for interactive segmentation provide. In fact, I show that this inherent disadvantage of the region proposal methods turns out to be an advantage for my point click based interactive segmentation method in Chapter 4. Object proposal methods on their own cannot effectively filter out the accurate segmentations from the noisy ones. However combining it with my point-click based human interaction results in a system that can filter out the accurate segmentations efficiently.

2.1.3 Strongly supervised class-specific methods

Finally, I discuss class-specific segmentation methods which require a large amount of training data from a pre-defined set of object categories. This is commonly known as *semantic segmentation*. It refers to the task of jointly *recognizing* and segmenting objects, classifying each pixel into one of k fixed categories. Prior methods in this area have studied this problem in the context of segmenting objects [126] as well as parsing entire scenes [90, 161]. Recent advances in deep learning have fostered increased attention to this task. Most deep semantic segmentation models include fully convolutional networks that apply successive convolutions and pooling layers followed by upsampling or deconvolution operations in the end to produce pixel-wise segmentation maps [23, 94, 102, 164].

All these methods are limited to segmenting objects from only those categories which were present during training. These methods do not generalize to other unseen categories of objects. Again, in contrast, since our proposed methods in Chapter 3 and 4 are interactive in nature, they are not limited to segmenting a fixed set of object categories and can segment any object based on the human input.

2.2 Segmenting objects in weakly-supervised image collections

Having discussed the existing methods for segmenting objects in individual images, I next review the existing work that addresses the problem of segmenting objects in weakly supervised image collections, where all images contain objects from the same category. A common method for utilizing this weak supervision is to jointly segment the weakly supervised collection, where images can mutually benefit from each other [2, 25, 30, 64, 67, 121, 122, 124, 140]. Another popular approach is to propagate segmentations from a subset of human segmented images, which can then benefit the unsegmented images [44, 122].

Next, I provide a brief review of the existing methods from these categories and compare them with my proposed work on active image segmentation propagation (Chapter 5) and also predicting compatibility (Chapter 6) for joint segmentation in weakly supervised image collections.

2.2.1 Joint segmentation of weakly-supervised image collections

Here, I discuss several existing methods which jointly segment an image collection from a known object category. Early works in this area referred to this problem as *co-segmentation* and worked with an assumption that a strong agreement in the foregrounds of all images exists, i.e., that the images in the collection contain the same exact object against differing backgrounds [120]. This setting continues to be developed, e.g., for greater efficiency [48] and multi-image collections with interactive user input [9]. However the models developed in these methods [9, 48, 120] enforce strong constraints about the appearance similarities across images. This is a restrictive setting. Typically, it is more likely to have an image collection that contains objects from the same class, but with large variations in shapes, sizes and appearance. This scenario is well-motivated by keyword image search on the Internet, which can readily return a set of likely candidates containing a named object, albeit amidst variable backgrounds and scenes.

More recent weakly supervised methods including the work proposed in this thesis in Chapter 5 are more suitable for jointly segmenting such a group of images. They segment the foreground object(s) while exploiting the fact that all input images contain instances of the same object category to discover repeated patterns [2, 25, 30, 64, 67, 121, 122, 124, 140].[1] Depending on the method, the output segmentation might be pixel-level masks [2, 64, 67, 121, 124, 140] or bounding boxes [30, 131]. Recent advances include ways to accommodate noisily labeled inputs [121, 131], multi-class data [64, 67], and object proposal regions [1, 30, 140]. While most methods use only bottom-up saliency and pairwise matching to discover the common foreground, some recent work bootstraps an appearance model in an iterative localization-learning procedure [25, 30].

The joint segmentation algorithm proposed in this thesis (Chapter 5) builds on this rich body of work. The proposed joint segmentation method uses object-like regions (instead of pixels [2, 64, 121, 122, 124]) as a building block for segmentation for scalability and efficient propagation. This along with several

---

[1] This class of techniques can also be described as *co-segmentation* or *joint segmentation* or *object discovery* or *co-localization* methods; in all cases, a set of related images is used to discover the common foreground.

other refinements improves the state-of-the-art when applied even without any manual foreground labels. At the same time, the proposed algorithm can further request human annotation on images which are most likely to improve the segmentation performance by propagating new information. This is a departure from previous methods: existing weakly supervised methods above use no human intervention.

2.2.2 Segmentation propagation from human segmented examples

Different from the works discussed in the previous section which operate without any human input, several methods including my work in Chapter 5 have explored the utility of injecting human input during the joint segmentation process. Most closely related are methods for *segmentation propagation*, which use labeled seeds (human drawn segmentations on some seed images) to propagate foreground masks to other images in the weakly labeled set [44, 122].

In comparison, my proposed active image segmentation propagation method (Chapter 5) has two key novel aspects. First, it actively selects which images should next receive foreground labels from human annotators. In contrast, existing methods are either opportunistic (and hence passive) about the labeled seeds, using only existing labeled data [44], or else select them in a one-shot manner without reacting to the impact of previously annotated examples [122]. Second, the stagewise procedure constantly re-evaluates the impact of new labels, revising the current foreground estimates on all images. In contrast, Guillaumin et al. [44] assume that propagation will proceed best among the closest semantically related classes in an external object hierarchy (ImageNet), and Rubinstein et al. [122] assume that propagation will proceed best among each image's neighbors in a global image descriptor space.

Traditional active learning also relies on several selection strategies such as reducing the classifier's expected error [4, 139, 142] or maximizing the diversity among the selected images [17, 32, 49]. However, all such methods are closely coupled to their classifier of interest, and they aim to find good images to label by category (even those using regions [127, 139, 144]). In contrast, our task in is to select images from which *segmentation will propagate well*, and the aim of my technique in Chapter 5 is to find good images to annotate with foreground masks.

2.2.3 Compatibility for joint segmentation

Prior methods assume that all the input images are amenable to be jointly segmented together. In the strict same-object joint segmentation setting [9, 48, 120], this is assured by manually selecting the input pair (or set). For example, a designer may supply a set of images to be rotoscoped [120], or an analyst may gather aligned brain images from which to segment pathologies [48], or a consumer may group a burst of photos at an event (e.g., a soccer game) into a mini-album [9].

In the weakly supervised setting, the related images often originate from Internet search for an object's name. In this case, the majority of methods assume that all images are mutually amenable to a joint segmentation [2, 21, 63, 64, 67, 133, 140, 154]. However, the intra-class appearance and viewpoint variations make this assumption rather strong in practice. Some methods aim to limit the influence of joint segmentation to closely related images, whether by selecting nearest neighbors [121, 122] or discovering subcategory clusters [25]. However in all these cases, this is done on the basis of a manually defined (i.e., non-learned) image similarity metric. The assumption is that image similarity alone is sufficient to predict joint segmentation success. In contrast, Chapter 6 in this thesis proposes an approach which learns the behavior of the joint segmentation algorithm from the training data generated directly from the joint segmentation algorithm. It does this by developing a learning to rank approach which predicts the compatibility of image pairs to be jointly segmented together.

2.3 Segmenting objects in videos

Having discussed the existing approaches for segmenting objects in individual images and in weakly supervised image collections, I next review the existing work that addresses the problem of object segmentation in videos. This is a well studied problem in computer vision and several existing methods have tried to address the various challenges involved in segmenting objects from video. There has been a wide range of methods that have been proposed for solving this problem including: **1) Unsupervised methods:** which try to segment the video without any human supervision, **2) Interactive methods:** which require a human in the loop to constantly guide a segmentation algorithm, **3) Semi-supervised propagation methods:** which require some frames a video to be segmented by humans, which are then propagated to other unsegmented frames and, **4) Supervised methods:** which require training data to learn segmentation models that can classify each pixel in a video as object versus background.

Next, I provide a brief review of the existing methods from these categories and compare them with my proposed work for semi-supervised supervoxel based video propagation (Chapter 7) and also the proposed end-to-end learning approach for video segmentation (Chapter 8).

2.3.1 Unsupervised video segmentation

Fully automatic or unsupervised video segmentation methods assume no human input on the video. First we have the unsupervised methods which simply segment the videos in coherent space-time tubes. They can be grouped into two broad categories: region based and tracking based. Region based supervoxel methods [43, 158] oversegment the video volume into space-time blobs with cohesive appearance and motion. Others group superpixels using spectral clustering [40] or novel tracking techniques [16, 138]. Distinct from the region-based methods, tracking methods use point trajectories to detect cohesive moving object parts [20, 83]. Any such bottom-up method tends to preserve object boundaries, but "oversegment" them into multiple parts. Their goal is to generate mid-level video regions useful for downstream processing, whereas in this thesis the goal for the proposed methods in Chapters 7 and 8 is to produce space-time tubes which accurately delineate complete object boundaries.

Next we have the fully automatic methods that generate thousands of "object-like" space-time segments [38, 103, 155, 156, 159], typically by learning the category-independent properties of good regions, and employing some form of tracking. While useful in accelerating object detection, it is not straightforward to automatically select the most accurate one when a single hypothesis is desired. Methods that do produce a single hypothesis [35, 50, 81, 96, 107, 130, 136, 160] strongly rely on motion to identify the foreground objects, either by seeding appearance models with moving regions or directly reasoning about occlusion boundaries using optical flow. This limits their capability to segment static objects in video.

In comparison, the semi-supervised video propagation algorithms proposed in my thesis (Chapters 7 and 8) receive human guidance on a subset of frames and as output produce a single hypothesis in the form of a space-time object tube. While they do require human guidance, it enables the propagation methods to be more accurate than the fully automatic methods which also makes them useful for video data annotation.

2.3.2 Interactive video segmentation

Interactive methods for video segmentation have also been proposed in the literature. They typically require a human annotator to be constantly in the loop to correct the algorithm's mistakes [7, 87, 116, 125, 149], either by monitoring the results closely, or by responding to active queries by the system [36, 141, 145] until the video is adequately segmented. While such intensive supervision is warranted for some applications, particularly in graphics [7, 87, 116, 149], it may be overkill for others. The interactive video segmentation methods usually have the advantage of greater precision, but at the disadvantages of greater human effort and less amenability to crowdsourcing.

Hence in this thesis, I focus on using human guidance for video segmentation without requiring the human to be constantly in the loop monitoring the algorithm. The video segmentation algorithms developed in this thesis (Chapters 7 and 8) require the humans to only provide few labeled frames for initialization and everything else remains automated. No human involvement is required beyond the initialization part. This is much more scalable from crowdsourcing perspective. One can simply choose to upload video frames on crowd platforms to collect human segmentations. Uploading entire videos and creating interfaces which allow for efficient interactive video segmentation on such crowd platforms is challenging.

2.3.3 Semi-supervised video segmentation propagation

Semi-supervised propagation methods, which are a focus in this thesis, accept some manually labeled frames with the foreground region and propagate them to the remaining clip [6, 36, 118, 135, 141]. While differing in their optimization strategies, most prior methods use the core graph based Markov Random Field (MRF) structure, with i) unary potentials determined by the labeled foreground's appearance/motion and ii) pairwise potentials determined by nodes' temporal or spatial adjacency. Pixel-based graphs can maintain very fine boundaries, but suffer from high computational cost and noisy temporal links due to unreliable flow [6, 141]. Superpixel-based graphs form nodes by segmenting each frame independently [36, 118, 135]. Compared to their pixel counterparts, they are much more efficient, less prone to optical flow drift, and can estimate neighbors' similarities more robustly due to their greater spatial extent. Nonetheless, their use of per-frame segments and frame-to-frame flow links confines them to short range interactions.

In contrast, the key idea in my supervoxel based segmentation propagation algorithm (Chapter 7) is to impose a supervoxel higher order potential to encourage consistent labels across broad spatio-temporal regions. The proposed approach is inspired by higher order potentials (HOP) for multi-class static image segmentation [71]. There, multiple over-segmentations are used to define large spatial cliques in the Robust P^n model, capturing a label consistency preference for each image segment's component pixels. I extend this idea to handle video foreground propagation with supervoxel label consistency.

Two existing unsupervised methods also incorporate the Robust P^n model for video segmentation, but with important differences from my approach. In [26], the spatial cliques of [71] are adopted for each frame, and 3-frame temporal cliques are formed via optical flow. The empirical impact is shown for the former but not the latter, making its benefit unclear. In [138], the Robust P^n model is used to prefer consistent labels in temporally adjacent superpixels within 5-frame subsequences. Both prior methods [26, 138] rely on traditional adjacency criteria among spatial superpixel nodes to define HOP cliques, and they restrict temporal connections to a short manually fixed window (3 or 5 frames). In contrast, I propose *supervoxel* cliques and HOPs that span space-time regions of variable length. The proposed cliques often span broader areas in space-time—at times the entire video —making them better equipped to capture an object's long term evolution in appearance and shape.

2.3.4 Supervised video segmentation

Until now, I have discussed video segmentation methods which are either completely unsupervised or require human guidance on the target video itself (the one that needs to be segmented). In this section, I will discuss methods which make use of human annotated training data to learn video segmentation models and can then be applied in an automatic fashion on the target video.

With the advent of deep learning based techniques, end-to-end learning from fully supervised training data has become one of the most successful paradigms for designing computer vision systems. Semantic segmentation networks for images have seen rapid advances in recent years and have achieved a lot of success on several benchmark datasets. State-of-the-art semantic segmentation techniques for *images* rely on fully convolutional deep learning architectures that are end-to-end trainable [23, 94, 102, 164].

Unfortunately, video segmentation has not seen such rapid progress.

We hypothesize that the lack of large-scale human segmented video segmentation benchmarks is a key bottleneck. Recent video segmentation benchmarks like Cityscapes [28] are valuable, but 1) it addresses category-specific segmentation, and 2) thus far methods competing on it process each frame independently, treating it like multiple image segmentation tasks.[2]

Unlike all these existing methods, this thesis proposes a two-stream deep segmentation network which is end-to-end trainable and is capable of accurately segmenting generic objects in video, whether or not they appear in training data (Chapter 8). This generalization is achieved by leveraging existing image classification and segmentation datasets to first build a generic appearance network. This network, when combined with large scale weakly labeled video datasets (only bounding boxes), opens a path towards training deep segmentation models that fuse spatial and temporal cues.

The proposed two-stream object segmentation network is not only generic but it also combines both appearance and motion in a unified framework. End to end deep learning for combining motion and appearance in videos has proven to be useful in several other computer vision tasks such as video classification [65, 101], action recognition [59, 128], object tracking [85, 95, 150] and even computation of optical flow [31]. While we take inspiration from these works, the two-stream network I propose is the first to present a unifying deep framework for segmenting objects in videos.

---

[2] https://www.cityscapes-dataset.com/benchmarks/

While the network is capable of segmenting objects in videos in a fully automatic fashion, it can also benefit from some human guidance at test time. This is similar to the semi-supervised setup we introduced previously. Since the human pinpoints the object of interest, existing semi-supervised methods [6, 36, 57, 98, 111, 118, 135, 141, 153] typically focus more on learning object appearance from the manual annotations. In contrast, I show that combining the generic objectness cues from the two-stream network with video specific appearance learned from manual annotations results in a even better performance for video object segmentation.

2.4 Roadmap

Having discussed the related work, I next describe my proposed approach to address these problems. In the next two chapters, I study the problem of interactively segmenting objects in images and videos. First, I discuss the proposed technique for actively making annotation choices in traditional interactive segmentation methods in Chapter 3. The novel point click based interactive segmentation algorithm will be discussed in Chapter 4. Next, the proposed active selection and segmentation propagation algorithm for weakly supervised image collections will be presented in Chapter 5, followed by an algorithm to predict compatibility for joint segmentation in Chapter 6. The next two chapters will discuss the proposed methods for video segmentation. This will include the ideas of supervoxel-based semi-supervised propagation (Chapter 7) and an end-to-end learning approach for generic pixel-level objectness in images and videos (Chapter 8). The remaining chapters will conclude the thesis and also discuss possible future directions.

Chapter 3

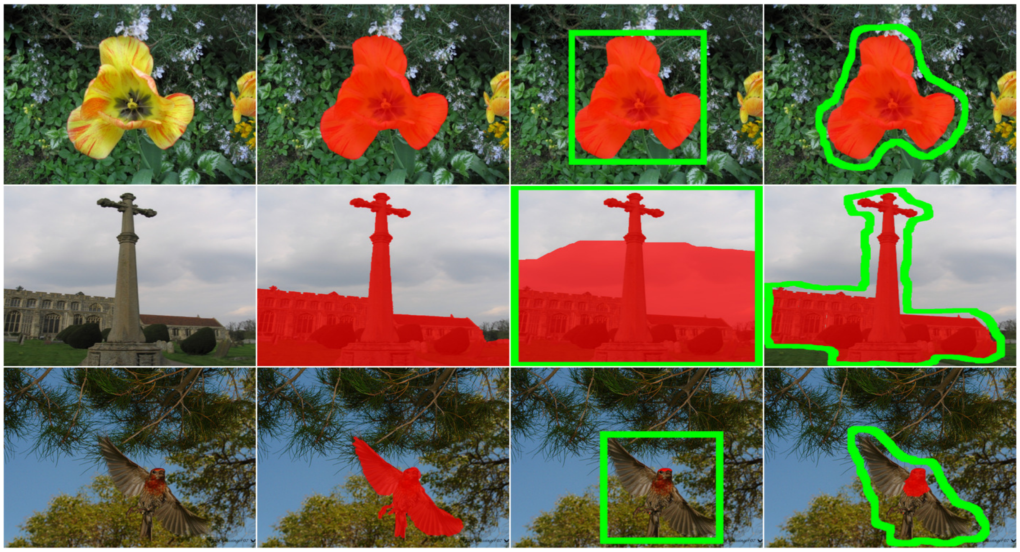
Interactive image segmentation with active human input

[1] Traditional interactive segmentation algorithms [13, 66, 100, 119] work by first requesting the user to indicate the foreground object with some mode of input. The pixels inside and outside the user-marked boundary are used to initialize the foreground and background appearance models, respectively. These appearance models are then used to assign likelihoods at each pixel of it being a foreground or background. These likelihoods are used to define energy functions which combine these likelihoods with smoothness priors defined over pixel neighborhoods. Minimizing these energy functions results in the final foreground/background segmentation (e.g., using graph cuts [13, 119]). Recent work builds on this basic idea by incorporating it into a co-segmentation problem [9], or applying topological priors [82] and shape constraints [45].

A common assumption in existing interactive segmentation pipelines is that the way humans guide the underlying segmentation model is fixed in advance [9, 13, 45, 66, 82, 100, 119]. However, simply fixing the input modality

---

[1] The work in this chapter was supervised by Dr. Kristen Grauman and originally published [52] in: Predicting Sufficient Annotation Strength for Interactive Foreground Segmentation. S. Jain and K. Grauman. In Proceedings of the International Conference on Computer Vision (ICCV), 2013, Sydney, Australia.



(a) Image (b) Ground Truth (c) Bounding Box (d) Sloppy Contour

Figure 3.1: Interactive segmentation results (shown in red) for three images using various annotation strengths (marked in green). Note how the most effective mode of input depends on the image content. The method presented in this chapter predicts the easiest input modality that will be sufficiently strong to successfully segment a given image. Best viewed in color.

leads to a suboptimal tradeoff in human and machine effort. The problem is that each mode of input requires a different degree of annotator effort. The more elaborate inputs take more manual effort, yet they leave less ambiguity to the system about which pixels are foreground. At the same time, depending on its content, an image might be better suited to be segmented by different modes of human input.

For example, Figure 3.1 shows (a) three images, (b) their ground truth foreground, and their interactive segmentation results (shown in red) using either (c) a bounding box or (d) a freehand outline as input (marked in green).

The flower (top row) is very distinct from its background and has a compact shape; a bounding box on that image would provide a tight foreground prior, and hence a very accurate segmentation with very quick user input. In contrast, the cross image (middle row) has a plain background but a complex shape, making a bounding box insufficient as a prior; the more elaborate freehand "sloppy contour" is necessary to account for its intricate shape. Meanwhile, the bird (bottom row) looks similar to the background, causing both the bounding box and sloppy contour to fail. In that case, a manually drawn tight polygon may be the best solution.

It is clear that the granularity at which the humans need to supervise the underlying algorithm is clearly a function of the image content. Simpler objects with distinct foregrounds and plain backgrounds require minimal amount of human guidance. On the other hand, more complex objects require more fine-grained guidance from the user. In this chapter, I outline my proposed algorithm to tailor the human input based on the image content, i.e., we want to request from the human annotator only sufficient supervision which can lead to a good segmentation for that image. As we will see, the proposed algorithm can operate in two different modes:

- **Single image mode:** Given a single image as input, the algorithm will ask the human user to provide the easiest (fastest) form of input that the system expects to be sufficiently strong to do the job.
- **Batch mode:** Given a batch of images as input together with a bud-

> get of time that the user is willing to spend guiding the system, the algorithm can optimize the *mix* of input types that will maximize total segmentation accuracy, subject to the budget. This allows, for example, the system to request a tight polygon on one very difficult image, sloppy contours on three moderately difficult ones, and bounding boxes on the remaining images.

To this end, I first define the annotation modes and interactive segmentation model my method targets (Sec. 3.1). Then, I define features indicative of image difficulty and learn how they relate to segmentation quality for each annotation mode (Sec. 3.2). Given a novel image, I forecast the relative success of each modality (Sec. 3.3). This allows my method to select the modality that is sufficient for an individual image. Finally, I propose a more involved optimization strategy for the case where a batch of images must be segmented in a given time budget (Sec. 3.4). The remaining sections in the chapter then present detailed experimental results and comparisons with other state-of-the-art methods.

3.1 Interactive segmentation model

I first discuss the input modalities and the segmentation model that my proposed method targets. My approach chooses from three annotation modalities, as depicted in Figure 3.2:

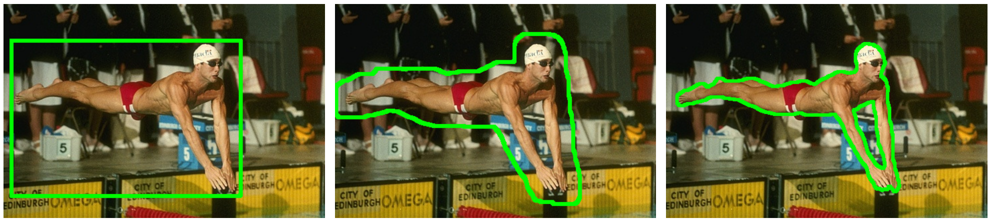

(a) Bounding box (b) Sloppy contour (c) Tight polygon

Figure 3.2: Possible modes of annotation

1. **Bounding box:** The annotator provides a tight bounding box around the foreground objects. This is typically the fastest input modality.
2. **Sloppy contour:** The annotator draws a rough contour surrounding the foreground. This gives a tighter boundary than a box (i.e., encompassing fewer background pixels) and offers cues about the object shape. It typically takes longer.
3. **Tight polygon:** The annotator draws a tight polygon along the foreground boundaries. Tight polygon is equated with perfect segmentation accuracy. This is the slowest modality.

All three are intuitive and well-used tools. My method extends naturally to handle other modalities where a user specifies foreground pixels (e.g., scribbles).

No matter the annotation mode, the pixels inside and outside the user-marked boundary are used to initialize the foreground and background models, respectively. Specifically, they are used to construct two Gaussian mixture models in RGB color space, G_{fg} and G_{bg}. Then standard graph-cut based interactive segmentation [13, 119] is applied with the mixture models as likelihood functions. Each image pixel is a node, and edges connect neighboring pixels. The objective is to assign a binary foreground/background label $y_p \in \{1, 0\}$ to each pixel p so as to minimize the total energy of all labels L:

$$E(L) = \sum_{p} A_p(y_p) \ + \sum_{p,q \in \mathcal{N}} S_{p,q}(y_p, y_q), \tag{3.1}$$

where $A_p(y_p) = -\log P(F_p|G_{y_p})$ is the unary likelihood term indicating the cost of assigning a pixel as foreground/background, and F_p denotes the RGB color for pixel p. The term $S_{p,q}(y_p, y_q) = \delta(y_p \neq y_q) \exp(-\beta \|F_p - F_q\|)$ is a standard smoothness prior that penalizes assigning different labels to neighboring pixels that are similar in appearance, where β is a scaling parameter and $\mathcal{N}$ denotes a 4-connected neighborhood.

I use the algorithm of [15] to minimize Eqn. 3.1, and use the GrabCut idea of iteratively refining the likelihood functions and the label estimates [119].

3.2 Learning segmentation difficulty per modality

Having defined the annotation choices and the basic engine for segmentations, I can now explain my algorithm's training phase. The main idea is to train a discriminative classifier that takes an image as input, and predicts whether a given annotation modality will be successful once passed to the interactive graph cuts solver above. In other words, one classifier will decide if an image looks "easy" or "difficult" to segment with a bounding box, another classifier will decide if it looks "easy" or "difficult" with a sloppy contour.

To compose the labeled training set, we require images with ground truth foreground masks. For each training example, we want to see how it would behave with each user input mode. For the bounding box case, we simply generate the bounding box that tightly fits the true foreground area. For the sloppy contour case, we dilate the true mask by 20 pixels to simulate a coarse human-drawn boundary.[2] After running graph cuts (optimizing Eqn. 3.1) for each one in turn, we obtain two *estimated foreground masks* per training image: fg_{box} and fg_{con}.

These masks are used to extract a series of features (defined next), which are then used to train two Support Vector Machine (SVM) based classifiers. Let O denote the normalized overlap between an estimated mask and the true foreground. Let $\bar{O}_{box}$ and $\bar{O}_{con}$ denote the median overlap among all training images for the two modes. The ground truth label on an image is positive ("easy", "successful") for an annotation modality x if $O > \bar{O}_x$. That is, the image is easy for that particular form of user input if its accuracy is better than at least half of the examples.[3]

Next I define features that reveal image difficulty. Graph cut segmentation performance is directly related to the degree of separation between the foreground and background regions. It tends to fail if the two are similar in

---

[2] In a user study, I find these masks are a good proxy; on average, they overlap with actual hand-drawn contours by 84%.

[3] While a regression model would also be a reasonable choice here, I found classification more effective in practice, likely because of the large spread in the overlap scores obtained through graph cuts segmentation.

appearance, or if the foreground object has a complex composition. Furthermore, the notion of separability is tied to the form of user input. For example, a bounding box input can fail even for an object that is very distinct from its background if it contains many background pixels. The features take these factors into account.

Let I_{FG} be an estimated foreground (as specified by either mask fg_{box} or fg_{con} in a training image), and let I_{BG} denote its complement. I define the following features:

Color separability: Since the segmentation model depends on foreground and background appearance, dissimilarity measures between them is computed and used as a feature. The χ^2 distance between the color histograms computed from I_{FG} and I_{BG} in RGB (16 bins per channel) and Lab (21 bins per channel) color space is recorded. The local color dissimilarity is also considered by computing the χ^2 distance between the RGB color histogram from I_{FG} and from a small 40-pixel region around I_{FG}. This captures how distinct the region is from its neighboring pixels. Finally, the KL-divergence between Gaussian mixture models estimated with I_{FG} and I_{BG} is also used as a feature.

Edge complexity: We expect edges to reflect the complexity of a foreground object. For this, a 5-bin edge orientation histogram from I_{FG} is recorded. It is done only for the foreground, as we do not want the annotation choice to be affected by background complexity. Next, as a measure of image detail, the sum of gradient magnitudes for I_{FG} and I_{BG} are computed, normalized by their areas. The ratio between foreground and background image detail is used as a feature.

Label uncertainty: The next feature directly captures how uncertain the segmentation result is. For this, the dynamic graph cuts approach proposed in [73] to compute the min-marginal energies associated with each pixel's graph cut label assignment is used. The energies are mapped to uncertainty by computing the change in min marginal energy when a pixel is constrained to take the non-optimal label, and record a 5-bin histogram of the uncertainty values within I_{FG}. Intuitively, an easy segmentation will have mostly labels with low uncertainty, and vice versa.

Boundary alignment and object coherence: We expect easy segments to align well with strong image boundaries. To estimate the extent of alignment, image is first divided into superpixels [37]. For every superpixel that lies on the boundary between I_{FG} and I_{BG}, the fraction of its area that lies inside I_{FG} is noted. Its average across all superpixels is used as a feature. Number of connected components in the resulting segmentation are also used as a measure of how coherent the object is.

Altogether, this leads to 17 features: four for color separability, six for edge complexity, five for label uncertainty, and two for boundary alignment and coherence. I stress that all features are object- and dataset-independent. This is important so that the algorithm can learn the abstract properties that reflect segmentation difficulty, as opposed to the specific appearance of previously seen objects that were difficult to segment.

3.3 Predicting difficulty on novel images

Given a novel image, they system must predict which of the annotation modes will be successful. To do so, it needs a coarse estimate of the foreground in order to compute the features above. I use a four step process. First, a salient object detector is applied that outputs a pixel-wise binary saliency map [93]. Second, the saliency map is refined with "superpixel smoothing", assigning the foreground label to each superpixel that overlaps a salient region by more than 50%. This yields a more coherent estimate aligned with strong image boundaries. Third, if we have multiple input images similar in appearance (i.e., the co-segmentation case), each superpixel is further reclassified using an SVM trained with superpixel instances originating in the current foreground-background masks. Finally, a bounding box and a sloppy contour (by dilation) are automatically generated, and then we run graph cuts to get the estimated masks for either modality. These estimates are used for I_{FG} (and their complements for I_{BG}) to compute the features defined above. While often an image has a primary foreground object of interest, my method (like any graph cuts formulation) can accommodate foregrounds consisting of multiple disconnected regions.

The foreground estimate in a test image need only give a rough placement of where the user might put the bounding box or sloppy contour. Indeed, the whole purpose of this work is to get the necessary guidance from a user.

Nonetheless, the estimates must be better than chance to ensure meaningful features. I find the saliency-based initializations[4] are a reasonable proxy (overlapping 47-71% on average for our datasets), though in no way replace the real human input that we will seek after applying my method.

Now the difficulty classifiers are applied to the test image. Recall that to properly balance effort and quality, the objective is to predict which mode is *sufficiently strong*. Always requesting tight polygons is sure to yield accurate results, but will waste human effort when the image content is "easy". Similarly, always requesting a bounding box is sure to be fast, but will produce lousy results when the image is too "hard". Therefore, if given a single image as input, the system uses a cascade to request the fastest annotation that is likely to succeed. That is, it shows the annotator a bounding box tool if the bounding box classifier predicts "easy". If not, it shows the sloppy contour tool if its classifier predicts "easy". If not, the system shows the user the tight polygon tool.

3.4 Annotation choices under budget constraints

In an alternative usage scenario, my system accepts a batch of images and a budget of annotation time as input. The objective is to select the optimal annotation tool *for each image* that will maximize total predicted accuracy, subject to the constraint that annotation cost must not exceed the budget.

---

[4] I also tried to use the saliency based masks during training, but found that training with ground-truth masks was more robust.

This is a very practical scenario. For example, today's data collection efforts often entail posting annotation jobs to a crowdsourcing service like Mechanical Turk; a researcher would like to state how much money (i.e., worker time) they are willing to spend, and get the best possible segmentations in return.

For a high budget, a good choice may be tight polygons on all of the hardest images, and sloppy contours on the rest. For a low budget, it might be bounding boxes on all but the most difficult cases, etc. Rather than hand code heuristics to capture such intuitions, I propose to automatically optimize the selection. Formulating the problem is possible since my approach explicitly accounts for the expected success/failure of a particular kind of user input for a given image.

Suppose we have n images to segment, and a budget of B, which could be specified in minutes or dollars. Let p_k^b and p_k^c denote the probability of successful interactive segmentation for image k with a bounding box or sloppy contour, as predicted by my model. The easy/difficult classifier outputs are mapped to probabilities of success using Platt's method. Let p_k^p denote the probability of success when using a tight polygon; by definition, $p_k^p = 1$. Let $\mathbf{x} = [x_1^b, x_1^c, x_1^p, \ldots, x_n^b, x_n^c, x_n^p]$ be an indicator vector with three entries for each image, reflecting the three possible annotation modalities one could apply to it. That is, $x_k^b = 1$ would signify that image k should be annotated with a bounding box. Let $\mathbf{c} = [c_1^b, c_1^c, c_1^p, \ldots, c_n^b, c_n^c, c_n^p]$ be a cost vector, where c_k^a denotes the cost associated with annotating image k with annotation type a, specified in the same units as B. That is, $c_k^b = 7$ means it will take 7 sec to draw a bounding box on image k.

I formulate the following objective to solve for the best batch of sufficiently strong annotations:

$$\begin{aligned} \mathbf{x}^* = \arg\max_{\mathbf{x}} \quad & \sum_{k=1}^{n} p_k^b x_k^b + p_k^c x_k^c + p_k^p x_k^p, \\ \text{s.t.} \quad & \mathbf{c}^T \mathbf{x} \leq B, \\ & x_k^b + x_k^c + x_k^p = 1, \ \forall k = 1, \dots, n, \\ & x_k^b, x_k^c, x_k^p \in \{0, 1\}, \ \forall k = 1, \dots, n. \end{aligned} \tag{3.2}$$

The objective says we want to choose the modality per image that will maximize the predicted accuracy. The first constraint enforces the budget, the second ensures we choose only one modality per image, and the third restricts the indicator entries to be binary. The objective is maximized using a linear programming (LP) based branch and bound method for solving integer programs, which finds the optimal integer solution by solving a series of successive LP-relaxation problems. It takes less than a minute to solve for about 500 images and 70 budget values.

While my approach supports image-specific annotation costs c_k, the biggest factor in cost is which annotation type is used. Therefore, c_k^b, c_k^c and c_k^p are each assumed to be constant for all images k, based on real user time data. One could optionally plug in fine-grained cost predictions per image when available, e.g., to reflect that high curvature contours are more expensive than smooth ones.

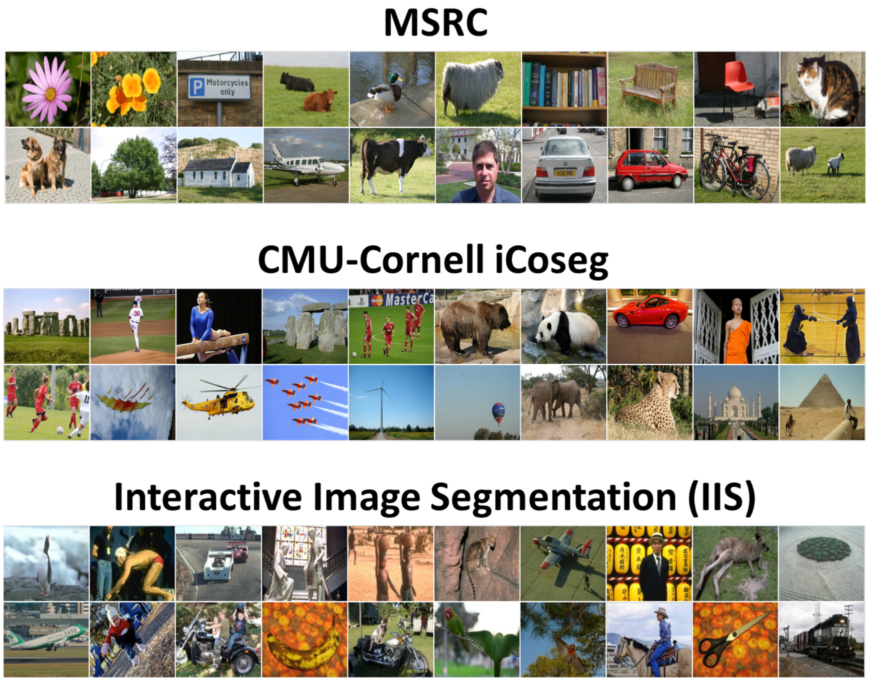


Figure 3.3: Overview of interactive image segmentation datasets. Best viewed in color.

3.5 Results

In this section, I present the results on different interactive image segmentation baselines and also compare with several state-of-the-art methods.

3.5.1 Datasets and baselines

Datasets: The proposed method is evaluated on three public datasets (see Figure 3.3) that provide pixel-level labels:

1. **Interactive Image Segmentation (IIS)** [45] consists of 151 unrelated images with complex shapes and appearance;

2. **MSRC** contains 591 images, and the multi-class annotations [97] were converted to foreground-background labels by treating the main object(s) (cow, flowers, etc.) as foreground. The same object class was never allowed to appear in both the training and test sets, to prevent my method from exploiting class-specific information.

3. **CMU-Cornell iCoseg** [9] contains 643 images divided into 38 groups with similar foreground appearance, allowing us to demonstrate my method in the optional co-segmentation setting.

Baselines: The proposed method is compared to the following baselines and other state-of-the-art methods:

1. **Otsu: [104]** finds the optimal grayscale threshold that minimizes the intra-class variance between foreground and background. To use it to estimate foreground-background separability, the *inter*-class variance (at the optimal threshold) is computed and normalized by total variance. Higher values indicate higher separability, and hence "easier" segmentation.

2. **Effort Prediction: [144]** predicts whether an image will be easy or hard for a human to segment, using features indicative of image complexity. I use the authors' public code. This is a state-of-the-art method for estimating image difficulty.

3. **Global Features:** I train two SVMs (one for bounding box, one for contours) to predict if an image is easy based on a 12-bin color histogram, color variance, and the separability score from [104]. This baseline illustrates the importance of the proposed features in capturing the estimated foreground's separation from background.

4. **GT-Input**: uses the ground-truth box/contour masks as input to my method, showing the impact of my features in the absence of errors in the saliency step.

5. **Random:** randomly assigns a confidence value to each modality in the budgeted annotation results.

Otsu and Effort Prediction use the same function for both boxes and contours, since they cannot reason about the different modalities. Note that methods for active interactive (co-)segmentation [9, 148] address a different problem, and are not comparable. In particular, they do not predict image difficulty, and they assume a human repeatedly gives feedback on multiple images with the same foreground.

All classifiers are linear SVMs, and the parameters are chosen by cross-validation. I quantify segmentation accuracy with the standard overlap score ($\frac{P \cap GT}{P \cup GT}$) between the predicted and ground truth masks P and GT.

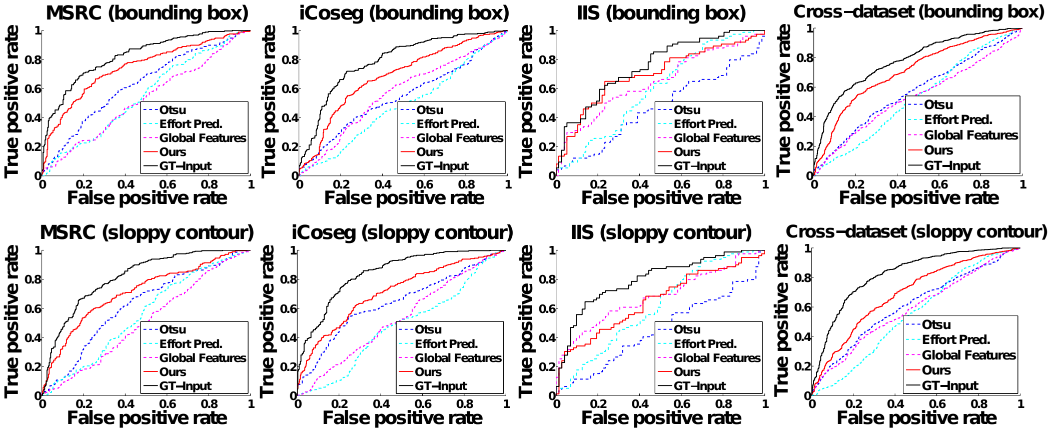


Figure 3.4: Difficulty prediction accuracy for each dataset (first three columns) and cross-dataset experiments (last column). The proposed method outperforms all baselines including Otsu [104], Global Features and Effort Prediction [144].

3.5.2 Predicting difficulty per modality

First we see how well all methods predict the success of each annotation modality. I test both in a *dataset-specific* and *cross-dataset* manner. For the former, I test in a leave-one-out (IIS, MSRC) or leave-one-group-out (iCoseg) fashion. For the latter, I test in a leave-one-dataset-out fashion. I use each method's confidence on the test images to compute ROC curves.

Figure 3.4 shows the results. My approach consistently performs well across all datasets, while none of the baselines has uniform performance (e.g., Otsu beats other baselines on MSRC, but fails badly on IIS). On MSRC and iCoseg, my approach significantly outperforms all the baselines, including the state-of-the-art Effort Prediction [144]. On IIS, it is again better for bounding boxes, but Global Features is competitive on sloppy contours. I attribute

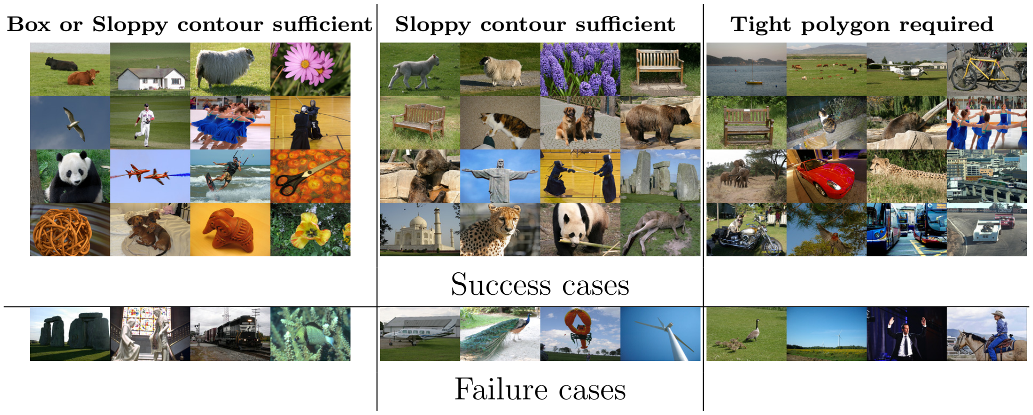


Figure 3.5: Qualitative results: **Left:** Example images which can be successfully segmented with both bounding box and sloppy contour annotations. **Middle:** Example images for which segmentation with bounding box input fails but sloppy contour is successful. **Right:** Example images for which both bounding box and sloppy contour fails. Best viewed in color.

this to the complex composition of certain images in IIS that makes saliency detection fail.

In the even more challenging cross-dataset setting (Fig. 3.4, right column), the advantage of my method remains steady. This is a key result. It shows that the proposed method is learning which generic cues indicate if a modality will succeed—not some idiosyncrasies of the particular objects or cameras used in the datasets. Whereas the Global Features and Effort Prediction [144] methods learn from the holistic image content, my method specifically learns how foreground-background separability influences graph cuts segmentation. Analyzing the linear SVM weights, I find label uncertainty, boundary alignment, and χ^2 color distance are the most useful features. The GT-Input result underscores the full power of the proposed features.

Figure 3.5 shows some typical success and failure cases. For the leftmost block of images, my method predicts a bounding box or contour would be sufficient. These images usually have uniform backgrounds, and distinct, compact foreground regions, which are easy to tightly capture with a box (e.g., flower, cows). For the center block, my method predicts a bounding box would fail, but a sloppy contour would be sufficient. These images usually have objects with complex shapes, for which even a tight box can overlap many background pixels (e.g., Christ the Redeemer, Taj Mahal). For the rightmost block, my method predicts neither a box or contour is sufficient. These images contain objects with intricate shape (e.g., bicycle) or high similarity to background (e.g., elephant, bird). Notably, the same object can look easy or difficult. For example, the skaters in the left block are close together and seem easy to annotate with a box, while the skaters in the right block are far apart and tight polygons are needed to extract their limbs. This emphasizes the object-independence of my method; its predictions truly depend on the complexity of the image.

Failures can occur if the salient region detection fails drastically (e.g., in the person image on right, the salient white shirt leads the method to think the image looks easy). It can also fail by overestimating the difficulty of images with low color separability (e.g., shadows in Stonehenge and white pixels by statues in left group), suggesting a more refined edge detector could help.

3.5.3 Annotation choices to meet a budget

Next I evaluate my idea for optimizing requests to meet a budget. I apply my method and the baselines to estimate the probability that each modality will succeed on each image. Then, for each method, the budget solution defined in Sec. 3.4 is used to decide which image should get which modality, such that total annotation time will not exceed the budget. For the cost of each modality in $\mathbf{c}$, I use the average time required by the 101 users in my user study: 7 sec for bounding box, 20 sec for sloppy contour, 54 sec for tight polygon. If the solution says to get a box or contour on an image, I apply graph cuts with the selected modality (Sec. 3.1). If the solution says to get a tight polygon, I simply use the dataset ground truth, since it was obtained with that tool. The final accuracy is the overlap in the estimated and ground truth foregrounds over all images.

Figure 3.6 plots the results as a function of budget size. The budget values range from the minimum possible (bounding boxes for all images) to the maximum possible (tight polygons for all images). The proposed method consistently selects the modalities that best use annotation resources: at almost every budget point, it achieves the highest accuracy.[5] This means that the method saves substantial human time. For example, in the cross-dataset result on 1,351 images, the best baseline needs 2.25 hours more annotation effort than my method does to obtain 90% average overlap.

---

[5] By definition, all methods yield the same solution for the two extremes, and hence the same accuracy.

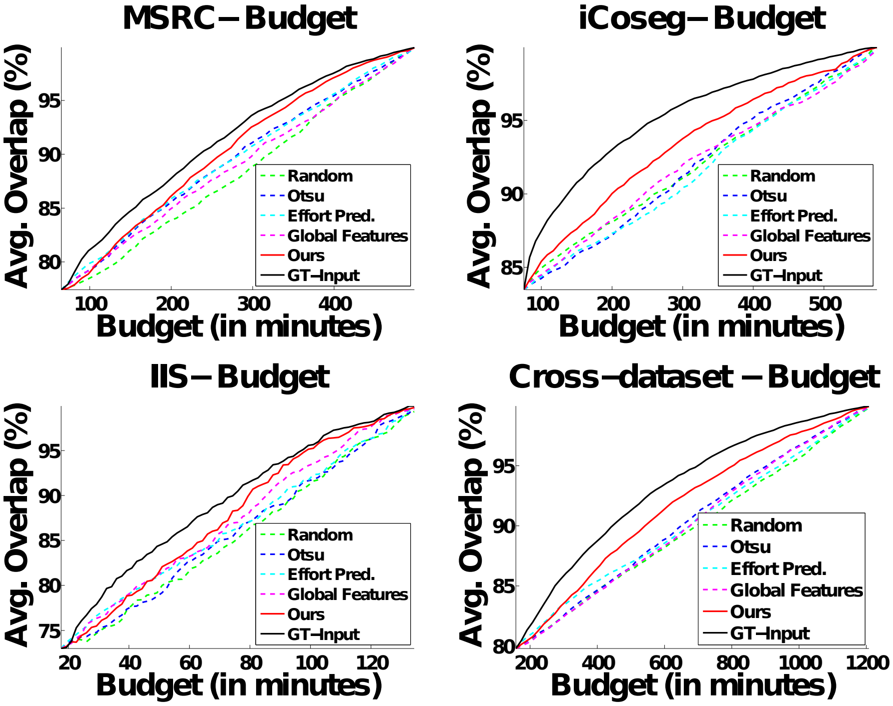


Figure 3.6: Choosing annotation modalities to meet a budget. The proposed method outperforms all baselines including Random Selection, Otsu [104], Global Features and Effort Prediction [144].

What choices does my method typically make? I find that as the budget increases, the bounding box requests decrease. The number of sloppy contour requests increases at first, then starts decreasing after a certain budget, making way for more images to be annotated with a tight polygon. For images where either a box or contour is likely to succeed, my method tends to prefer a box so that it can get a tight polygon for more images within the budget.

Object	**Avg. overlap (%)**		**Time saved (%)**
	All tight	**Ours**	
Flower	65.09	65.60	21.2 min (73%)
Car	60.34	60.29	3.9 min (15%)
Cow	72.90	66.53	9.2 min (68%)
Cat	51.79	46.56	13.7 min (23%)
Boat	51.08	50.77	1.4 min (10%)
Sheep	75.90	75.59	17.2 min (64%)

Table 3.1: Accuracy of a recognition system trained using our method and the baseline. It also shows the amount of annotator time which was saved in preparing the training images using my method.

3.5.4 Application to recognition

To further illustrate the practical impact of my approach, I next apply it to train a recognition system for the MSRC recognition challenge [97]. Suppose that we are given a set of images known to contain an object category of interest amidst a cluttered background. The goal is to learn a classifier that can differentiate object versus non-object regions. Rather than ask an annotator to give tight polygons on each training image—the default choice for strongly supervised recognition systems—I apply my cascaded modality selection. Meanwhile, the baseline approach simply gets a tight polygon on each of the images.

The resulting segmented images from either method are used to train a linear SVM classifier that can predict whether a new image contains the object or not. I evaluate using leave-one-out cross validation per class, and score accuracy by normalized overlap with ground truth. Dense SIFT features sampled on a regular grid with 30 pixels spacing are extracted, and clustered into 20 visual words. Each image is divided into regions using [37], and each region is represented with a histogram of visual words. Each region in a training image is assigned a label based on either the interactive segmentation result predicted to be sufficient (for my method) or the tight polygon ground truth only (for the baseline). I train SVMs with the histograms from the resulting foreground object regions in the training examples. At test time, each region in the image is classified as foreground or background to localize the object.

Table 3.1 shows the results. My approach substantially reduces the total annotation time required, yet its accuracy on novel images is still very competitive with the method that gets perfect tight polygons on all images.

3.5.5 User study

Finally, I conduct a user study with Amazon Mechanical Turk workers. I randomly select one third of the images from each dataset to make a diverse pool of 420 images. I present users with the necessary tools to do each modality, and time them as they work on each image. If an object has multiple foreground objects, they must annotate each one. I collect responses from 5 users for each annotation mode per image, then record the median time spent. In total, I obtain 2,100 responses per modality, from 101 unique users.

Figure 3.7 (right) shows example user annotations. The most variance is seen among the sloppy contour inputs, since some users are more "sloppy"

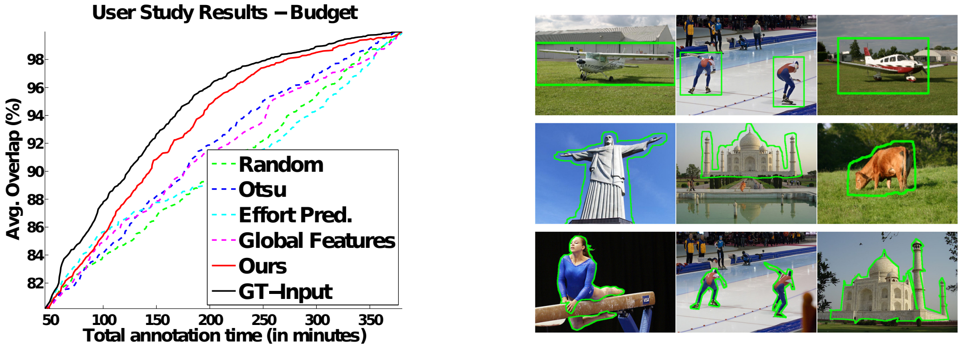


Figure 3.7: **Left:** Annotation choices under a budget with real user data. The proposed method outperforms all baselines including Random Selection, Otsu [104], Global Features and Effort Prediction [144]. **Right:** Example user annotations for bounding box (top), sloppy contour (middle), and tight polygon (bottom).

than others. Still, as expected, sloppy contours typically only improve interactive segmentation results (85.5% average overlap accuracy) compared to the faster bounding boxes (82.1% average overlap accuracy).

Figure 3.7 (left) shows the budgeted annotation results with real user data. The plot is like Figure 3.6, only here 1) The real users' boxes/contours are fed to the graph cuts engine, rather than simulate it from ground truth masks, and 2) The users' per-image annotation time is incurred at test time (on x-axis). Across all budgets, my method allocates effort more wisely, and it even narrows the gap with the GT-Input. This result confirms that even though the ultimate annotation time may vary not only per modality, but also per image, using a fixed cost per modality during *prediction* is sufficient to get good savings. Overall, this large-scale user study is promising evidence that by reasoning about the expected success of different annotation modalities, one can use valuable annotator effort much more efficiently.

3.6 Conclusion

In conclusion, this chapter proposed a method to predict sufficient annotation strength required for interactively segmenting an image. The proposed method can work on single images and can also jointly optimize the decision for a collection of images. Extensive experiments including a user study show that carefully tailoring human input based on the image content can lead to substantial savings in human annotation effort.

The method proposed in this chapter assumed a fixed underlying segmentation model i.e., graph cuts. However, several other forms of interactive segmentation algorithms (e.g., higher order potentials, geodesic distance transforms etc.) exist in the literature. While the overall idea of predicting sufficient annotation strength is fairly generic, how well the current training procedure generalizes to these other algorithms remains to be explored. Training separate models for different interactive segmentation algorithms will allow the system to learn the nuances of different algorithms and will make it more widely applicable.

Also, in its current form the method makes another key assumption—it only selects one modality per image; however its possible to relax this assumption by progressively increasing the granularity of the input for a particular image depending on the annotation budget and also the quality of the segmentation output generated by the currently chosen input modality.

Finally, it might be possible that a fully automatic segmentation algorithm itself might produce a high quality segmentation for a given image [46]. No human guidance may be required in such cases. The method currently does not handle this case; an extension which first predicts whether human interaction is required at all for a given image and requests sufficient input modality only if needed can lead to further reduction in annotation costs.

In summary, this chapter focused on optimizing the modality requested from a human annotator for a given input image using graph cuts based interactive segmentation algorithm. In the next chapter I will turn my attention to the formulation of the interactive segmentation engine itself. I will present a new formulation for interactive segmentation that requires only simple point clicks to accurately segment objects in images and videos.

Chapter 4

Interactive image and video segmentation with point clicks

[1]In the previous chapter, I demonstrated that its possible to optimize for the granularity of human input required to accurately segment an image using traditional interactive segmentation models. However there are two fundamental issues with the existing interactive segmentation algorithms:

1. Existing methods largely rely on the tried-and-true interaction modes used for image labeling; namely, the user draws a bounding box or an outline around the object of interest [13, 66, 100, 119]. These interactions are still very involved and require a substantial amount of user effort.
2. The output segmentations are very tightly coupled with the user interaction. It is essential to have a good number of pixels labeled via human input to learn sufficiently good appearance models for foreground and background regions. Only then we can expect to achieve good segmentation results [52].

---

[1] The work in this chapter was supervised by Dr. Kristen Grauman and originally published [58] in: Click Carving: Segmenting Objects in Video with Point Clicks. S. Jain and K. Grauman. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP), 2016, Austin, U.S.A.

Regardless of the exact input modality, the common assumption in traditional methods is to get the user's input *first*, and then generate a segmentation hypothesis thereafter [9, 13, 45, 66, 82, 100, 119]. The tight coupling between user input and segmentation output makes it difficult to incorporate a simpler form of human interaction i.e., point clicks within these models. Clicks, largely unexplored for interactive segmentation, are an attractive input modality due to their ease, speed, and intuitive nature (e.g., with a touch screen the user may simply point a finger) [10, 58, 114, 151]. However they carry very little information about the appearance of the object or background since only a single pixel is labeled via a click.

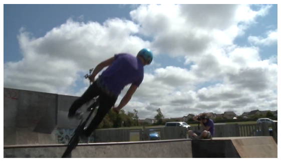
In this chapter, I propose a novel formulation of the interactive image segmentation problem called *Click Carving* which enables the use of simple point clicks to perform interactive segmentation. The key idea behind *Click Carving* is to reverse the standard flow of information that exists in traditional interactive segmentation methods. Instead of waiting for the human to give some input to generate any segmentation output, the *Click Carving* algorithm takes the lead by first generating thousands of plausible segmentations for a given image automatically. Among these thousands of segmentation outputs, at least a few should be accurate with high probability. The role of the user is to then efficiently pick out the best segmentation from this pool.

I will show that this indeed can be achieved through a voting algorithm, which collects user votes on object boundaries via point clicks. These votes are then used to re-rank the plausible hypotheses and user can pick the most accurate one once satisfied. I show that my proposed *Click Carving* algorithm results in large savings in annotation effort and often only requires a couple of clicks to obtain accurate segmentation. In experiments on six datasets we tested, only 2-4 clicks are typically required to accurately segment the object of interest. Note that the novel idea behind *Click Carving* is not so much about the "clicking" interface itself; rather it centers around the idea of simple point supervision as a sufficient cue to perform semi-automatic segmentation and the carving backend that efficiently discerns the most reliable proposals.

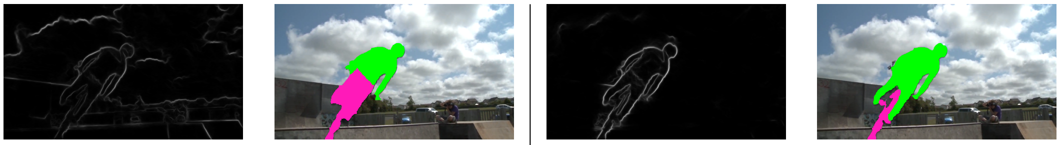
Aside from testing the approach with real users, I have also developed several simulated user clicking models in order to systematically analyze the relative merits of different clicking strategies. For example, is it more effective to click in the object center, or around its perimeter? How should multiple clicks be spaced? Is it advantageous to place clicks in reaction to where the system currently has the greatest errors? One interesting outcome of this study is that the behavior one might assume as a default—clicking in the object's interior [10, 151]—is much less effective than clicking on its boundaries. I show that boundary clicks are better able to discriminate between good and bad object proposal regions.

I also show that *Click Carving* can be effectively used to segment videos as well. This is achieved by first segmenting a video frame using *Click Carving* and then combining with my video segmentation propagation algorithm (Chapter 7). Existing methods also follow a similar process where the first frame is segmented by a human followed by a propagation step [6, 36, 57, 118, 135, 141]. In all these methods, the initial frame is either segmented manually or using traditional interactive methods [13, 66, 100, 119] which are much more expensive. In contrast, the proposed method is able to segment the initial frame using simple point clicks, resulting in a substantial savings in annotation costs for collecting spatio-temporal annotations for videos. Because of the ease with which this framework can assist even non-experts in making high quality annotations, it has great promise for scaling up image and video segmentation.

To this end, I first define the technique we use to automatically generate thousands of segmentation hypotheses (Sec. 4.1) also known as foreground region proposals. Then, I define the details of my proposed *Click Carving* interactive segmentation algorithm (Sec. 4.2). I then discuss the various clicking strategies including details of several simulated clicking algorithms which were used for detailed experimental evaluation (Sec. 4.3). In the case of a video frame, after segmenting it using *Click Carving*, the output needs to be propagated to all other frames of the video to obtain a complete video segmentation. For this, I make use of my supervoxel-propagation algorithm which I briefly refer to in Sec. 4.4. The complete discussion of the propagation algorithm is postponed till Chapter 7. The remaining sections in this chapter then present detailed experimental results and comparisons with other state-of-the-art methods.



(a) Video Frame



(b) Static Boundaries (c) Static Proposals (d) Motion Boundaries (e) Motion Proposals

Figure 4.1: Generation of object region proposals using both static and dynamic cues in a video frame. Best viewed in color.

4.1 Generating foreground region proposals

Existing interactive segmentation methods rely on human input (a bounding box, contour, or scribble) at the onset to generate results [9, 13, 45, 66, 82, 100, 119]. The key idea behind my *Click Carving* approach is to flip this process. Instead of the human annotator providing an input around the object of interest, the system generates many plausible segmentation mask hypotheses and the annotator efficiently navigates to the best ones with point clicks.

Specifically, I use state-of-the-art region proposal generation algorithms to generate 1000s of possible foreground segmentations for a given image or a video frame. As discussed in Chapter 2, region proposal methods aim to obtain high recall at the cost of low precision. Even though this guarantees that at least a few of these segmentations will be of good quality, it is difficult to filter out the best ones automatically with existing techniques.

To generate accurate region proposals, I use the multiscale combinatorial grouping (MCG) algorithm [5]. The original algorithm uses image boundaries to obtain a hierarchical segmentation, followed by a grouping procedure to obtain region-based foreground object proposals. In the case of static images, we use the exact algorithm from [5] which uses static image boundaries to generate foreground region proposals. However, when *Click Carving* is employed to segment a video frame, both static and motion boundaries are used to generate foreground region proposals. This is very useful for videos, where due to factors like motion blur etc., static image boundaries are not very reliable in many cases. On the other hand, optical flow provides a strong cue about the objects contours while the object is in motion. Hence also using motion boundaries [152] to generate per-frame motion region proposals using MCG is really helpful. The two sources are complementary in nature: for static objects, the per-frame region proposals obtained using static boundaries will be more accurate, and vice versa.

Figure 4.1 illustrates this with an example. Both the person and bike (Figure 4.1a) are in motion. As a result, the static boundaries are weaker (Figure 4.1b). Figure 4.1c shows the best static proposal for each object; the proposal quality for the bike is very poor. On the other hand, the motion boundaries (Figure 4.1d) are much stronger and result in accurate proposals for both the person and the bike (Figure 4.1e).

In summary, given a video frame, a set of foreground region proposals ($\mathcal{M}$) is generated for it by taking the union between the static region proposals

($\mathcal{M}_{static}$) and motion region proposals ($\mathcal{M}_{motion}$), i.e., $\mathcal{M} = \{\mathcal{M}_{static} \cup \mathcal{M}_{motion}\}$. For static images only static region proposals are computed i.e., $\mathcal{M} = \{\mathcal{M}_{static}\}$. On average, we obtain a total of about 2000 proposals per image or video frame, resulting in a very high overall recall. In what follows, I explain how *Click Carving* allows a user to efficiently navigate to the best proposal among these thousands of candidates.

4.2 Click Carving for discovering an object mask

The region proposal step yields a large set of segmentation hypotheses (1000s), out of which only a few are very accurate object segmentations. A naive approach that asks an annotator to manually scan through all proposals is both tedious and inefficient. I now explain how my Click Carving algorithm effectively and very quickly identifies the quality segmentations. I show that within a few clicks, it is possible to obtain a very high quality segmentation of the desired object of interest.

At a high level, my *Click Carving* algorithm converts the user clicks into votes cast for the underlying region proposals. The user initiates the algorithm by clicking somewhere on the boundary of the object of interest. This click casts a vote for all the proposals whose boundaries also (nearly) intersect with the user click. Using these votes, the underlying region proposals are re-ranked and the user is presented with the top-k proposals having the highest votes.

This process of clicking and re-ranking iterates. At any time, the user can choose any of the top-k as the final segmentation if he/she is satisfied, or he/she can continue to re-rank by clicking and casting more votes.

More specifically, each proposal is characterized, $\mathcal{M}_j \in \mathcal{M}$ with the following four components $(\mathcal{M}_j^m, \mathcal{M}_j^e, \mathcal{M}_j^s, \mathcal{M}_j^v)$:

- Segmentation mask $(\mathcal{M}_j^m)$: This quantity represents the actual region segmentation mask obtained from the MCG region proposal algorithm (static or dynamic).
- Contour mask $(\mathcal{M}_j^e)$: The algorithm requires the user to click on the object boundaries, which as I show later is much more discriminative than clicking on interior points and results in a much faster filtering of good segmentations. To infer the votes on the boundaries, the segmentation mask $\mathcal{M}_j^m$ is converted into a contour mask. This contour mask only contains the boundary pixels from $\mathcal{M}_j^m$. For error tolerance, the boundary mask is dilated by 5 pixels on either side. This reduces the sensitivity of the exact user click location, which need not coincide exactly with the mask boundary.
- Objectness score $(\mathcal{M}_j^s)$: The objectness score from the MCG algorithm [5] is used to break ties if multiple region proposals get the same number of votes. This score reflects the likelihood of a given region to be an accurate object segmentation.
- User votes $(\mathcal{M}_j^v)$: This quantity represents the total number of user votes received by a particular proposal at any given time. It is initialized to 0.

As a first step, the algorithm begins by computing a lookup table which allows us to efficiently account for the votes cast for each proposal by the user. Let n be the total number of pixels in a given image and m be the total number of region proposals generated for that image. The lookup table $\mathcal{T} \in \{0, 1\}^{n \times m}$ is defined and precomputed as follows:

$$\mathcal{T}(i, j) = \begin{cases} 1 & \text{if } \mathcal{M}_j^e(i) = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

where i denotes a particular pixel and j denotes a particular region proposal.

When the user clicks at a particular pixel location c, the weights for each of the region proposal are updated as follows:

$$\mathcal{M}_j^v = \mathcal{M}_j^v + \mathcal{T}(c, j). \tag{4.2}$$

The updated set of votes is used to re-rank all the region proposals. The proposals with equal votes are ranked in the order of their objectness scores. This interactive re-ranking procedure continues until the user is satisfied with any of the top-k proposals and chooses that as the final segmentation. In my implementation, k is set such that k copies of the image, one proposal on each, fit easily on one screen ($k = 9$).

Figure 4.2 illustrates the user interface and explains this process with two examples. I show the user interaction on the leftmost column. Red circles denote clicks. The "ContourMap" column shows the average contour map of the top-5 ranked proposals after the user click. Here the colors are a heat-map

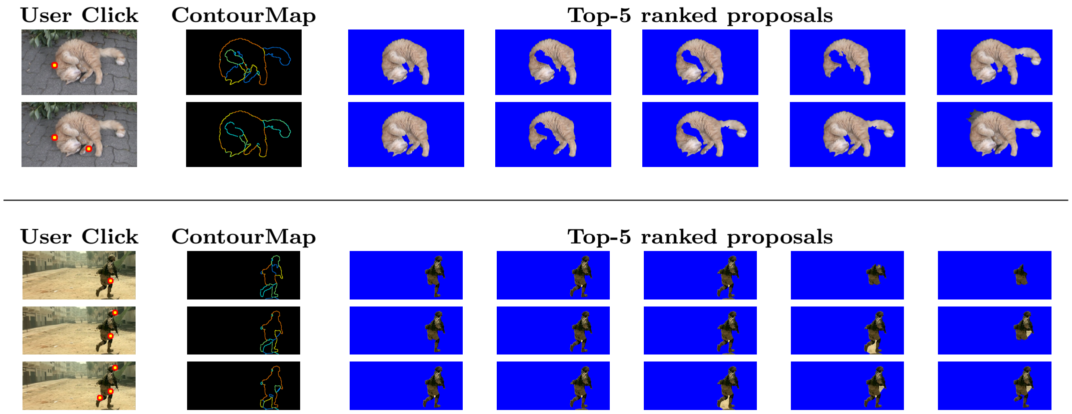


Figure 4.2: Click Carving based foreground segmentation. Best viewed in color. See text for details.

coding of the number of votes for a boundary fragment. Remaining columns show the top-5 ranked proposals.

The top two rows show an example "cat" image. The user places the first click on the left side of the object (top left image). We see that the resulting top ranked proposals (5 foreground images in top row) align well to the current user click, meaning they all contain a boundary near the click point. The average contour map of these top ranked proposals, informs the user about areas that have been carved well already (red lines) and which areas may need more attention (blue lines, or contours on the true object that remain uncolored). The user observes that most current top-k segmentations are missing the cat's right leg and decides to place the next click there (second row, leftmost image). The next ranking of the proposals brings up segmentations which cover the entire object accurately.

In the next example, I consider a frame from the "soldier" video in the

Segtrack-v2 dataset [84]. The user decides to place a click on the right side of the object (third row, leftmost image). This click itself retrieves a very good segmentation for the soldier. However, to explore further, the user continues by making more clicks. Each new constraint eliminates the bad proposals from the previous step, and after just three clicks, all the top-ranked proposals are of good quality. Please see the project page for video illustrations[2].

4.3 User clicking strategies

To quantitatively evaluate *Click Carving*, I employ both real human annotators and simulated users with different clicking strategies. I design a series of clicking strategies to simulate, each of which represents a hypothesis for how a user might efficiently convey which object boundaries remain missing in the top proposals. While real users are arguably the best way to judge final impact of my system (and so I include experiments that use them), the simulated user models are complementary. They allow us to run extensive trials and to see at scale which strategies are most effective. Simulated human users have also been studied in interactive segmentation for brush stroke placement [72].

The user models in our evaluation are categorized into three groups: human annotators, boundary clickers, and interior clickers.

---

[2] More details and videos can be found at: `http://vision.cs.utexas.edu/projects/clickcarving/`

1. **Human annotators:** I conduct a user study to analyze the performance of my method by recruiting three human annotators to work on each image. The three annotators included a computer vision student and 2 non-expert users. The human annotators were encouraged to click on object boundaries, while observing the current best segmentations. They were also given some time to familiarize themselves with the interface, before starting the actual experiments. They had a choice to stop by choosing one of the segmentations among the top ranked ones or continue clicking to explore further. A maximum budget of 10 clicks was used to limit the total annotation time, after which the annotation process stops and a final object mask selection had to be made. The target object was indicated to them before starting the experiment. In the case of multiple objects, each object was chosen as the target object in a sequential manner. I recorded the number of clicks, time spent, and the best object mask chosen by the user during each segmentation. The user corresponding to the median number of clicks is used for my quantitative evaluation. The total recorded time includes the time to both place the clicks and to select the best segmentation mask.

2. **Boundary clickers:** I design three simulated users which operate by clicking on object boundaries. To simulate these artificial users, I make use of the ground-truth segmentation mask of the target object. Equidistant points are sampled from the ground truth object contour to define object boundaries. Each simulated boundary clicker starts from the same

initial point. Principal component analysis (PCA) on the ground truth shape is used to find the axis of maximum shape variation. A ray from the centroid of the object mask is considered along the direction of this principal axis. The furthest point on the object boundary where this ray intersects is chosen as the starting point. The three boundary clickers that I design differ in how they make subsequent clicks from this starting point. They are:

(a) ***Uniform clicker:*** To obtain uniformly spaced clicks, the total number of boundary points is divided by the maximum click budget to obtain a fixed distance interval d. Starting from the initial point and walking along the boundary, a click is made every d points apart from the previous click location.

(b) ***Submod clicker:*** The uniform user has a high level of redundancy, since it clicks at locations which are still close to the previous clicks; hence the gain in information between two consecutive clicks might be small. Next I design a boundary clicker that tries to impact the maximum boundary region with each subsequent click. This is done by placing the click at a boundary point which is furthest away from its nearest user click among all boundary points. This resembles the sub-modular subset selection problem [76], where one tries to maximize the set coverage while choosing a subset. I employ a greedy algorithm to find the next best point.

(c) ***Active clicker:*** The previous two methods only looked at the ground truth segmentation to devise a click strategy, without taking into account the segmentation performance after each click is added. The active clicking strategy takes into account the current best segmentation among the top-k (vs. the ground truth) and uses that to make the next click decision. It is similar in design to the Submod user, except that it skips those boundary points which have already been labeled correctly by the top-ranked proposal. I find that this active simulated user comes the closest in mimicking the actual human annotators (see results for details).

3. **Interior clickers:** A novel insight of my method is the discriminative nature of boundary clicks. In contrast, default behavior and previous user models [10, 151] assumes a click in the interior of the object is well-suited. To examine this contrast empirically, my final simulated user clicks on interior object points. To simulate interior clicks, object pixel locations from the entire ground truth segmentation mask (up to the maximum click budget) are uniformly sampled and then clicks are placed sequentially on the object of interest.

4.4 Propagating the mask through the video

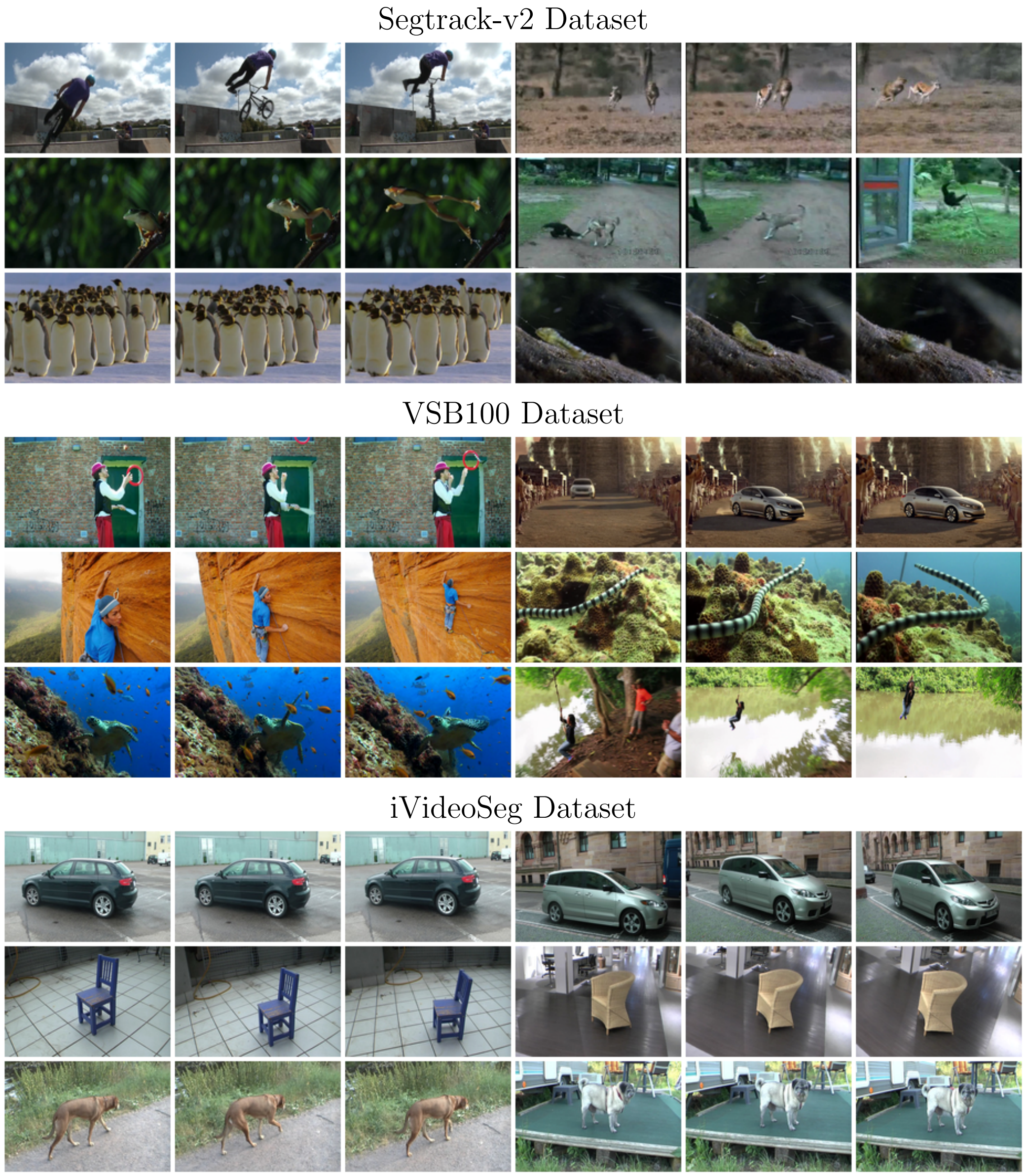
In the case of video, having discovered a good object mask using *Click Carving* in the initial frame, the next step is to propagate this segmentation to all other frames in the video. This gives us the complete video object segmentation. In my experiments, I use my proposed supervoxel based video propagation method to propagate the click-carved frame to the entire video volume. For the sake of brevity, I discuss the algorithmic details of my supervoxel-propagation approach only in Chapter 7. In this chapter, I just treat it as a black-box algorithm which allows us to propagate segmentation from the initial frame to all other frames in the video. Note that this “initial” segmentation can come from our *Click Carving* algorithm or can simply be drawn manually by a human labeler. The propagation algorithm is agnostic of how this “initial” segmentation was generated. It is only used to propagate the information.

4.5 Results

In this section, I provide detailed experiments and comparisons with state-of-the-art methods.

4.5.1 Datasets and metrics

I evaluate on six publicly available video and image datasets: Segtrack-v2 [84], VSB100 [39, 130], iVideoSeg [125], MSRC [97], CMU-Cornell iCoseg [9] and Interactive Image Segmentation (IIS) [45]. Out of these, the first three datasets are standard video datasets which are commonly used to evaluate video segmentation methods. The remaining datasets are the same as I used in the last chapter to evaluate interactive segmentation algorithms. Figure 3.3 and 4.3 show some visual examples from the datasets. For evaluating segmentation accuracy I use the standard intersection-over-union (IoU) overlap metric



Segtrack-v2 Dataset

VSB100 Dataset

iVideoSeg Dataset

Figure 4.3: Example video sequences from Segtrack-v2, VSB100 and iVideoSeg datasets. (best viewed in color).

between the predicted and ground-truth segmentations. A brief overview of the datasets:

- **SegTrack-v2** [84]: the most common benchmark to evaluate video object segmentation. It consists of 14 videos with a total of 24 objects and 976 frames. Challenges include appearance changes, large deformation, motion blur etc. Pixel-wise ground truth (GT) masks are provided for every object in all frames.
- **Berkeley Video Segmentation Benchmark (VSB100)** [39, 130]: consists of 100 HD sequences with multiple objects in each video. I use the "train" subset of this dataset in our experiments, for a total of 39 videos and 4397 frames. This is a very challenging dataset; interacting objects and small object sizes make it difficult to segment and propagate. I use the GT annotations of multiple foreground objects provided by [114] on every 20th frame.
- **iVideoSeg** [125]: This recent dataset consists of 24 videos from four different categories (car, chair, cat, dog). Some videos have viewpoint changes and others have large object motions. GT masks are available for 137 of all 11,882 frames.
- **Interactive Image Segmentation (IIS)** [45] consists of 151 unrelated images with complex shapes and appearance.

- **MSRC** contains 591 images, and the multi-class annotations [97] were converted to foreground-background labels by treating the main object(s) (cow, flowers, etc.) as foreground.
- **CMU-Cornell iCoseg** [9] contains 643 images divided into 38 groups with similar foreground appearance.

4.5.2 Click Carving for discovering an object mask

In this section, I test the accuracy/speed trade-off in terms of locating the best available proposal, and compare the simulated user models. Here, I present results on both video and image datasets. I first present the performance of *Click Carving* for interactively locating the best region proposal for the object of interest. For video datasets, I apply *Click Carving* on the first frame in all videos and average the results over the entire dataset. Evaluating on image datasets involves segmenting individual images using *Click Carving* and then averaging the score over the entire dataset.

In all experiments, I set the total click budget to be a maximum of 10 clicks per object. For simulated users, clicks are placed sequentially depending on its design, until a proposal which is within 5% overlap of the best proposal is ranked in the top-k or the click budget is exhausted. For the human user study, the user stops when they decide that they found a good segmentation within the top-k ranked proposals or have exhausted the click budget. For image datasets, 20% images from each dataset were randomly chosen for the human user study.

Table 4.1 shows the results for the video segmentation datasets and compares the performance with all simulated users. I compare both in terms of the number of clicks and time required[3] and also how close they get to the best proposal available in the pool of $\sim$2000 (BestProp). As expected, in all cases real users achieve the best segmentation performance and require far fewer clicks than all simulated users to achieve it. My simulated Active user, which takes into account the current state of the segmentation, comes closest to matching the human's performance. Also, I see clicking uniformly on the object boundaries requires more clicks on average than the Active and Submodular users, which try to impact the largest object area with each subsequent click. The Objectness baseline, which first ranks all the proposals using objectness scores and picks the best proposal among top-k (k=9), performs the worst. This shows that user interaction is key to picking good quality proposals among 1000s of candidates.

All users that operate by clicking on boundaries (Human, Uniform, Submod, and Active), come very close to choosing the best proposal in most cases. In contrast, clicking on the interior points requires substantially more clicks—often double the number. More importantly, the best segmentation it obtains is much worse in quality than the best possible segmentation. This makes the use of interior clicks impractical here even after accounting for the fact that they may be faster to provide than boundary clicks. This supports

---

[3] I use the average time per click from my human studies as an estimate for simulated boundary clickers. For interior clicks I use 2.4 seconds per click [10].

Segtrack-v2

	Objectness	Interior	Box-GC	Box-Prop	Uniform	Submod	Active	Human	BestProp
Clicks	0	6.29	2	2	4.46	3.83	3.34	**2.46**	-
Time (sec)	0	15.09	7	7	16.98	14.58	12.72	**9.37**	-
IoU	42.36	52.79	59.55	67.51	75.8	76.76	76.24	**78.77**	80.74

VSB100

	Objectness	Interior	Box-GC	Box-Prop	Uniform	Submod	Active	Human	BestProp
Clicks	0	7.05	2	2	5.34	5.28	5.23	**4.35**	-
Time (sec)	0	**16.92**	7	7	22.81	22.55	22.33	18.58	-
IoU	28.45	46.98	57.81	58.98	64.2	65.67	66.91	**69.63**	72.82

iVideoSeg

	Objectness	Interior	Box-GC	Box-Prop	Uniform	Submod	Active	Human	BestProp
Clicks	0	5.02	2	2	3.84	3.29	3.15	**2.84**	-
Time (sec)	0	12.05	7	7	15.20	13.02	12.47	**11.24**	-
IoU	50.69	72.54	65.43	68.04	77.57	77.84	**78.65**	78.24	81.34

Table 4.1: Click-carving proposal selection quality for real users (Human), the different user click models (Interior, Uniform, Submod, Active), Objectness, and Box baselines on video datasets. The results here show the segmentation score obtained for the first frame in every video using Click-carving. With an average of 2-4 clicks to carve the proposal boundaries, users attain IoU accuracies very close to the upper bound (BestProp). Objectness, Interior clicks, and the Box baselines are substantially weaker. IoU measures segmentation overlap with the ground truth; perfect overlap is 100. Best click based method is highlighted in bold.

my hypothesis that clicking on boundaries is much more discriminative in separating good proposals from the bad ones. Whereas a matching between an object proposal contour and a boundary click will rarely be accidental, several bad proposals may have the interior click point lie within them.

In fact, selecting the best proposal using an enclosing bounding box around the true object (Box-Prop, Table 4.1) is more effective than clicking on interior points. This is likely because a tight bounding box can eliminate a large number of proposals that extend outside its boundaries. On the other hand, an interior click cannot restrict the selected proposals to the ones which align well to the object boundaries. My method outperforms the bounding box selection by a large margin, showing the efficacy of my approach. My approach

MSRC									
	Objectness	Interior	Box-GC	Box-Prop	Uniform	Submod	Active	Human	BestProp
Clicks	0	3.62	2	2	3.35	2.74	2.86	**2.32**	-
Time (sec)	0	13.41	7	7	12.42	10.16	10.60	**8.6**	-
IoU	69.85	73.54	76.54	75.96	75.8	81.33	80.12	**81.57**	85.96

CMU-Cornell iCoseg									
	Objectness	Interior	Box-GC	Box-Prop	Uniform	Submod	Active	Human	BestProp
Clicks	0	4.25	2	2	3.76	2.98	3.24	**2.79**	-
Time (sec)	0	15.78	7	7	13.96	11.07	12.03	**10.36**	-
IoU	73.26	77.25	83.14	82.78	77.81	81.26	80.34	**82.13**	84.31

Interactive Image Segmentation (IIS)									
	Objectness	Interior	Box-GC	Box-Prop	Uniform	Submod	Active	Human	BestProp
Clicks	0	7.43	2	2	3.92	3.43	3.29	**3.12**	-
Time (sec)	0	29.98	7	7	15.81	13.84	13.27	**12.59**	-
IoU	68.11	65.63	72.28	74.69	70.21	74.46	76.18	**76.47**	78.68

Table 4.2: Click-carving proposal selection quality for real users (Human), the different user click models (Interior, Uniform, Submod, Active), Objectness, and Box baselines on interactive image segmentation datasets. With an average of 2-3 clicks to carve the proposal boundaries, users attain IoU accuracies very close to the upper bound (BestProp). Objectness, Interior clicks, and the Box baselines are substantially weaker. IoU measures segmentation overlap with the ground truth; perfect overlap is 100. Best click based method is highlighted in bold.

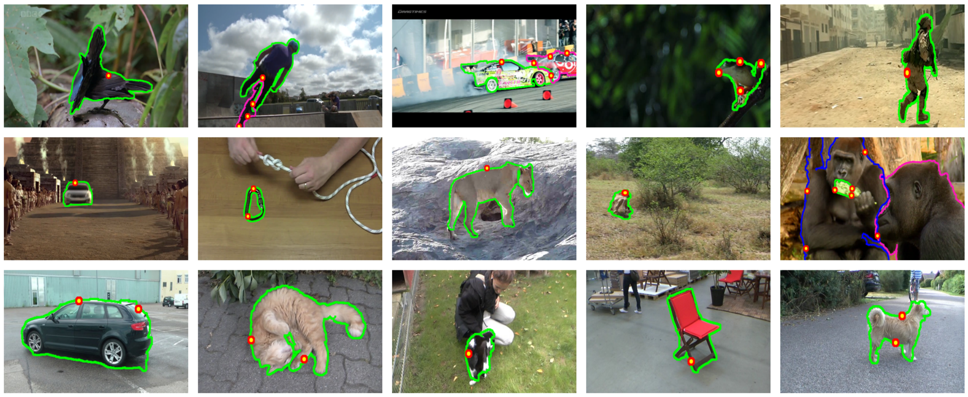
also significantly outperforms the standard GrabCut [119] interactive image segmentation method, initialized with a tight bounding box around the object (Box-GC, Table 4.1).

On Segtrack-v2 and iVideoSeg, *Click Carving* requires less than 3 clicks on average to obtain a high quality segmentation. For the most challenging dataset, VSB100, it obtains good results with an average of 4.35 clicks. This shows its potential to collect large amounts of segmentation data economically. The timing data reveals the efficiency and scalability of my method.

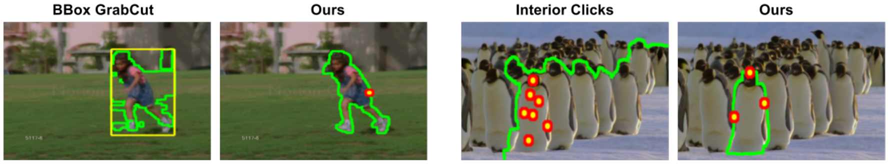
Table 4.2 shows the results for the three interactive image segmentation datasets and compares the performance with all simulated users and other relevant baselines. The trends here remain very similar to the ones observed for segmenting frames in the video datasets. These image datasets are relatively easier than the video datasets as can be seen by the upper bound scores from BestProp. Again the proposed *Click Carving* method only required 2-3 clicks on average to obtain high quality segmentation. It also significantly outperforms the standard GrabCut [119] interactive segmentation method in 2 out of 3 datasets. On the iCoseg dataset, GrabCut [119] is only slightly better. The foregrounds in iCoseg are very distinct from the backgrounds, which explain the strong performance of GrabCut [119]. On IIS dataset which is much harder, *Click Carving* outperforms it by more than 4% average overlap score.

Figure 4.4 (top) show qualitative results for *Click Carving.* In many cases (e.g., lions, soldier, cat), only a single click is sufficient to obtain a high quality segmentation. Several challenging instances like the cat (bottom row) and the lion (middle row), are segmented accurately with a single click. These objects would otherwise require a large amount of human interaction to obtain good segmentation (say using a GrabCut like approach). More clicks are typically needed when multiple objects are close-by or interacting with each other. Still, I observe that in many cases only a small number of clicks on each object results in good segmentations. For example, in the car video (top row), only 5 clicks are required to obtain final segmentations for both objects.

Figure 4.4 (bottom) highlights the key strengths of my method over two baselines. In the left example, I see that GrabCut [119] segmentation applied even with a very tight bounding box fails to segment the object. On the other hand, even with a single click, my proposed approach produces very accurate segmentation. The example on the right shows the importance of clicking on



Visual results for Click-Carving



Visual comparisons with baselines

Figure 4.4: **Top:** Qualitative results for *Click Carving*. The yellow-red dots show the clicks made by human annotators. The best selected segmentation boundaries are overlayed on the image (green). **Bottom:** Comparisons with baselines: The left example shows the segmentation I obtain with a single click as opposed to applying GrabCut segmentation with a tight bounding box. The example on the right shows the discriminative power of clicking on boundaries by comparing it with a baseline which clicks in the interior regions. Best viewed in color.

boundaries. Clicking on the interior fails to retrieve a good proposal, because several bad proposals also contain those interior clicks. Boundary clicks, which are highly discriminative, retrieve the best proposal quickly.

4.5.3 Click Carving for complete video segmentation

In this section I show how *Click Carving* results in large savings in annotation costs for full video segmentation. The previous section has already discussed the performance of *Click Carving* for efficiently segmenting a video frame. In this section I evaluate the segmentation performance after this initial click-carved frame is propagated to the entire video using my supervoxel-propagation algorithm. Here, I measure average segmentation overlap score over all the frames for all videos in a dataset. Hence, I compare with several state-of-the-art video segmentation methods. Here is a brief overview of all the methods I compare against:

Methods for comparison: I compare with several state-of-the art video segmentation methods [41, 43, 57, 84, 107, 125, 151, 153] and relevant baselines. Below I group them into six groups based on the amount of human annotation effort, i.e., the interaction time between the human and algorithm. In some cases, a human simply initializes the algorithm, while in others the human is in the loop always.

(1) Unsupervised: I use the state-of-the-art method of [107], which produces a single region segmentation result per video with zero human involvement.

(2) Multiple segmentation: Most existing unsupervised video segmentation methods produce multiple segmentations to achieve high recall. I consider both **a) Static object proposals (BestStaticProp):** where the best per frame region proposal (out of approx 2000 proposals per frame) computed using MCG algorithm, is chosen as the final segmentation for that frame **b) Spatio-temporal proposals** [43, 84]: These methods produce multiple spatio-temporal region tracks as segmentation hypotheses. To simulate a human picking the desired segmentation from the hypotheses, I use the dataset ground truth to select the most overlapping hypothesis. I use the duration of the video to estimate interaction time. This is a lower bound on cost, since the annotator has to at least watch the clip once to select the best segmentation. For the static proposals, I multiply the number of frames by 2.4 seconds, the time required to provide one click [10].

(3) Scribble-based: I consider two existing methods: **a) JOTS** [153]: the first frame is interactively segmented using scribbles and GrabCut. The segmentation result is than propagated to the entire video. I use the timing data from the detailed study by [99], who find it takes a human on average 66.43 seconds per image to obtain a good segmentation with scribbles. **b) iVideoSeg** [125]: This is a recently proposed state-of-the-art technique that uses scribbles to interactively label point trajectories. These labels are then used to segment the object of interest. I use the timing data kindly shared by the authors.

(4) Object outline propagation: the human outlines the object completely to initialize the propagation algorithm (typically in the first frame), which then propagates to the entire video. Here I again use my supervoxel based propagation algorithm to propagate the human drawn outline to the entire video. Timing data from [52, 89] indicate it typically takes 54-79 seconds to manually outline an object; I use the more optimistic 54 seconds for this baseline.

(5) Bounding box: Rather than segment the object, the annotator draws a tight bounding box around it. The baseline **BBox-VidGC** uses that box to obtain a segmentation for the video as follows. A Gaussian Mixture Model (GMM) based appearance model is learned for foreground and background pixels according to the box, then applied in a standard spatio-temporal MRF defined over pixels. The unaries are derived from the learnt GMM model and contrast-sensitive spatial and temporal potentials are used for smoothness.

(6) 1-Click based: I also consider baselines which perform video segmentation with a single user click. **a) TouchCut** [151] the only prior work using clicks for video segmentation. **b) Click-VidGC:** This is similar to BBox-VidGC except that I take a small region around the click to learn the foreground model. The background model is learnt from a small area around image boundaries. **c) Click-STProp:** To propagate the impact of a user click to the entire video volume, I use the spatio-temporal proposals from [103]. I do this by selecting all proposals which enclose the click inside them. Foreground and background appearance models are learnt using the selected proposals and refined using a spatio-temporal MRF. I again use the timing data from [10], which reports that a human takes about 2.4 seconds to place a single click on the object of interest.

Next I discuss the results for video segmentation, where we propagate

Unsup.	**Multiple Segmentations**			**Scribbles**	**Outline**	**Bounding Box**	**Click Based**		
[107]	**[43]**	**[84]**	**BestStaticProp**	**[153]**	**[57]**	**BBox-VidGC**	**Click-VidGC**	**Click-STProp**	**Ours**
Avg. Accuracy									
35.24	51.89	65.92	78.48	71.91	67.86	23.04	16.81	46.18	63.65
Annot. Effort									
-	336.6 tracks	60 tracks	120k proposals	1 frame	1 frame	2 clicks	1 click	1 click	2.46 clicks
Annot. Time (sec)									
0	673.2	120	142.5	66.43	54	7	2.4	2.4	9.37

Table 4.3: Video segmentation accuracy (IoU) on all 14 videos from Segtrack-v2. The last column shows results with real human users. The bottom two rows summarize the amount of human annotation effort required to obtain the corresponding segmentation performance, for all methods. My approach leads to an excellent trade-off between video segmentation accuracy and human annotation effort.

the results of *Click Carving* to the remaining frames in the video.

Video segmentation propagation on Segtrack-v2: Table 4.3 shows the results on Segtrack-v2. I compare using the standard intersection-over-union (IoU) metric with a total of 9 methods which use varying amounts of human supervision. The unsupervised algorithm [107] that uses no human input results in the lowest accuracy. Among the approaches which produce multiple segmentations, BestStaticProp and [84] have the best accuracy. This is expected because these methods are designed for having high recall, but it requires much more effort to sift through the multiple hypotheses to pick the best one. For example, it is prohibitively expensive to go through 2000 segmentations for each frame to get to the accuracy level of BestStaticProp. The method of [84] produces much fewer segmentations, but still requires 12x more time than my method to achieve comparable performance.

The scribble based method [153] achieves the best overall accuracy on this dataset, but is 6 times more expensive than my method. An interesting comparison is between my proposed *Click Carving* method and the "Outline" baseline which also uses the same supervoxel propagation algorithm but is initialized from a human-labeled object outline. My method which is initialized from slightly imperfect—but much quicker to obtain—click-carved object boundaries achieves comparable performance. This shows that we do not need very accurate human-drawn object boundaries to obtain good segmentation performance. Using computer generated segmentations coupled with my *Click Carving* interactive selection algorithm is sufficient to obtain high performance.

Moving on to the methods that require less human supervision, i.e., bounding boxes and clicks, we see that *Click Carving* continues to hold advantages. In particular, BBox-VidGC and Click-VidGC result in poor performance, indicating that more nuanced propagation methods are needed than just relying on appearance-based segmentation alone. Click-STProp, which obtains a spatial prior by propagating the impact of a single click to the entire video volume, results in much better performance than solely appearance based methods. However, my method, which first translates clicks into accurate per-frame segmentation before propagating them, yields a 17% gain (37% relative gain).

All these trends show that my method offers an excellent trade-off between segmentation performance and annotation time. Figure 4.5 (left), visually depicts this trade-off. All methods which result in better segmentation

	Unsup.	**Outline**	**Bounding Box**	**Click Based**		
	[107]	**[57]**	**BBox-VidGC**	**Click-VidGC**	**Click-STProp**	**Ours**
Avg. Accuracy	17.79	61.43	14.74	11.14	26.76	56.15
Annot. Effort	-	1 frame	2 clicks	1 click	1 click	4.35 clicks
Annot. Time (sec)	0	54	7	2.4	2.4	18.58

Table 4.4: Video segmentation accuracy (IoU) on all 39 videos in VSB100; format as in Table 4.3. My approach provides an excellent trade-off between video segmentation accuracy and human annotation effort.

accuracy than ours need substantially more human effort. Even then the gap in the performance in relatively small. On the flip side, the methods which require less annotation effort than us also result in a significant degradation in segmentation performance.

Video segmentation propagation on VSB100: Next, I test on VSB100. This is an even more challenging dataset and very few existing methods have reported foreground propagation results on it. Since this dataset includes several videos that contain multiple interacting objects in challenging conditions, *Click Carving* tends to require more clicks (4.35 on average). My method again outperforms all baselines which require less human effort and results in comparable performance with [57], but at a much lower cost. Figure 4.5 (right) again reflects this trend.

Video segmentation propagation on iVideoSeg: I also compare our method on the recently proposed iVideoSeg dataset [125]. I compare with three methods [41, 43, 125] out of which [125] is the current state-of-the-art method for interactive foreground segmentation in videos. I use the timing information provided by the authors [125]. I compare the performance of our method on

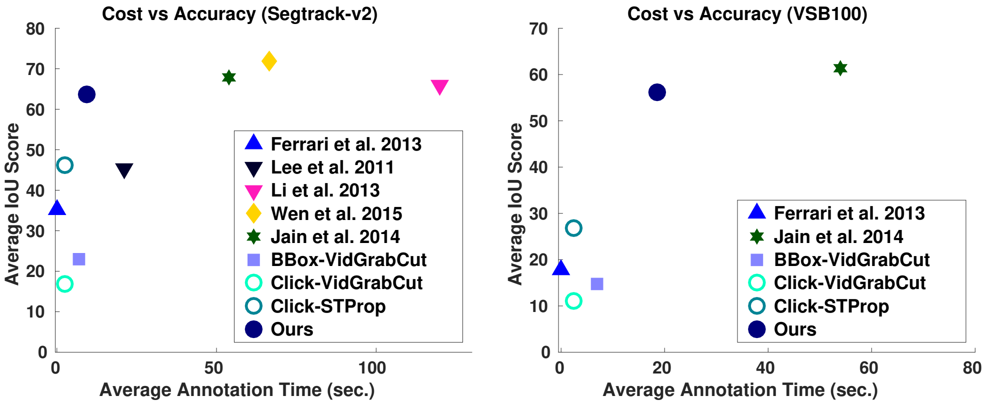


Figure 4.5: Cost vs accuracy on Segtrack (left) and VSB100 (right). The Click Carving based video propagation results in similar accuracy as state-of-the-art methods, but it does so with much less human effort. The plots show a comparison between Click Carving and the unsupervised method of Ferrari et al. [107], spatio-temporal object proposal methods from Lee et al. [81], Li et al. [84], semi-supervised propagation methods from Wen et al. [153], Jain et al. [57] and other relevant baselines. Click Carving offers an excellent trade-off between cost and accuracy. Best viewed in color.

all 24 videos in the dataset (300-1000 frames per video) using the real user annotation times. The methods of [41, 43, 125] run for multiple iterations i.e., a human provides annotation on several frames, observes the results and repeats until he/she is satisfied. This requires a human to evaluate the current video segmentation result and decide if more annotation is required. The authors provided timing and accuracy data for 4-5 iterations on each video.

In contrast my method does one-shot selection instead of iterative refinement. My method pre-selects the frames on which to request human annotation (every 100th frame in this case). For each selected frame, we ask a human annotator to use the *Click Carving* method to find the best region

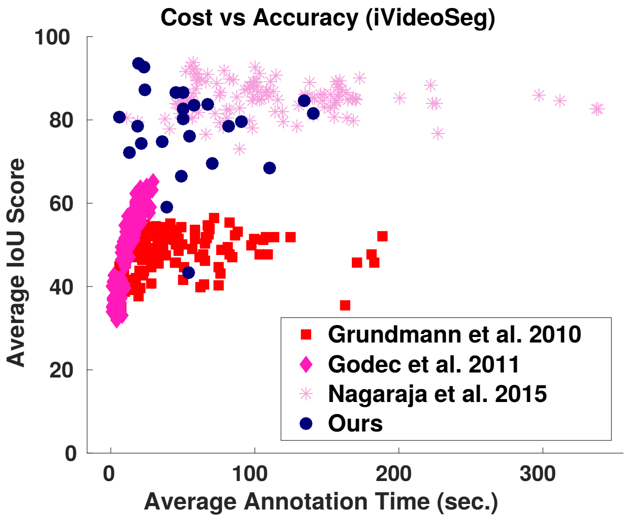


Figure 4.6: Cost vs accuracy on iVideoSeg dataset. The Click Carving based video propagation results in similar accuracy as state-of-the-art methods, but it does so with much less human effort. The plots show a comparison between Click Carving and the unsupervised segmentation method from Grundmann et al. [43], semi-supervised propagation method from Godec et al. [41] and interactive video segmentation method from Nagaraja et al. [125]. Click Carving offers an excellent trade-off between cost and accuracy. Best viewed in color.

proposal while recording their timing. The total time for the video is simply the sum of time taken for each selected frame. The video segmentation propagation is re-initialized whenever a new labeled frame is available.

Figure 4.6 shows the results. For all methods, each data point on the plot shows time vs. accuracy for a particular video at a particular iteration. My method outperforms both [41, 43] by a considerable margin. When compared with [125], my method achieves similar segmentation accuracy but with less than half the total annotation time. On average over all 24 videos, [125] takes 110.05 seconds to achieve an IoU score of 80.04. In comparison my method only takes 54.35 seconds to reach an IoU score of 77.68.

Comparison with TouchCut: To my knowledge TouchCut [151] is the only prior work which utilizes clicks for video segmentation. In that work, the user places a click somewhere on the object, then a level-sets technique transforms the click to an object contour. This transformed contour is then propagated to the remaining frames. Very few experimental results about video segmentation are discussed in the paper, and code is not available. Therefore, I am only able to compare with TouchCut on the three Segtrack videos reported in their paper. Table 4.5 shows the result. When initialized with a single click, my method outperforms TouchCut in two out of three videos. With one more click, it performs better in all 3 videos.

	TouchCut	**Ours (1-click)**	**Ours (2-clicks)**
birdfall2	248	213	187
girl	1691	2213	1541
parachute	228	225	198

Table 4.5: Comparison with TouchCut [151] in terms of pixel error (lower is better).

Qualitative results on video segmentation propagation: Figure 4.7 - 4.9 show some qualitative results for video segmentation propagation on the three datasets that we used in our experiments. The left-most image in each row shows the best region proposal chosen by a human annotator using *Click Carving*. Subsequent images show the results of segmentation propagation, when initialized from this selected proposal.

4.6 Conclusion

In this chapter, I presented a novel interactive image segmentation technique, *Click Carving*, using which only a few clicks are required to obtain accurate object segmentations in images and videos. My method strikes an excellent balance between accuracy and human effort resulting in large savings. Because of the ease of use even for non-experts, my method offers great promise for scaling up image and video segmentation which can be beneficial for several research communities.

In the future, several extensions can be incorporated in the current method to improve it even further. Firstly, in its current form the proposed method makes two key assumptions: 1) The overall segmentation quality is upper bounded by the quality of the underlying region proposals and 2) the user has a choice of only picking one among the top ranked hypotheses as the final segmentation. Clearly, both these assumptions restrict the overall quality of the segmentation that the current method can generate. A natural extension would be to merge multiple slightly imperfect proposals selected by the user into a single and more accurate segmentation. The user can then have the option to manually edit this segmentation to further improve the quality.

Secondly, the user currently only places clicks on objects. This is a restrictive assumption especially in cases where multiple other objects are overlapping and occluding the target object. In such cases, allowing the user to also indicate the background regions through "negative clicks" can possibly eliminate large number of irrelevant regions very efficiently. This straightforward extension can further reduce the total amount of annotation time required for segmenting objects using *Click Carving*.

Thirdly, in its current form the system treats per-frame segmentation and video propagation as separate tasks. These different tasks can be unified together by incorporating space-time segmentation proposals directly in the algorithm for complete video segmentation. In that case, user clicks will directly re-rank complete space-time segmentations of objects in video instead of the current two step process.

Finally in case of video segmentation, the system currently assumes that the user will evaluate the segmentation output and will re-initialize the propagation by another per-frame segmentation done using *Click Carving*, whenever it starts to fail. An active variant of the current system which takes into account the annotation budget and also the quality of propagation to automatically predict when a new *Click Carving* based re-initialization is needed can be very useful.

Together the last two chapters have focused on handling the segmentation of an individual image or video. In the next chapter I will expand the scope to consider jointly segmenting a collection of related images at once.

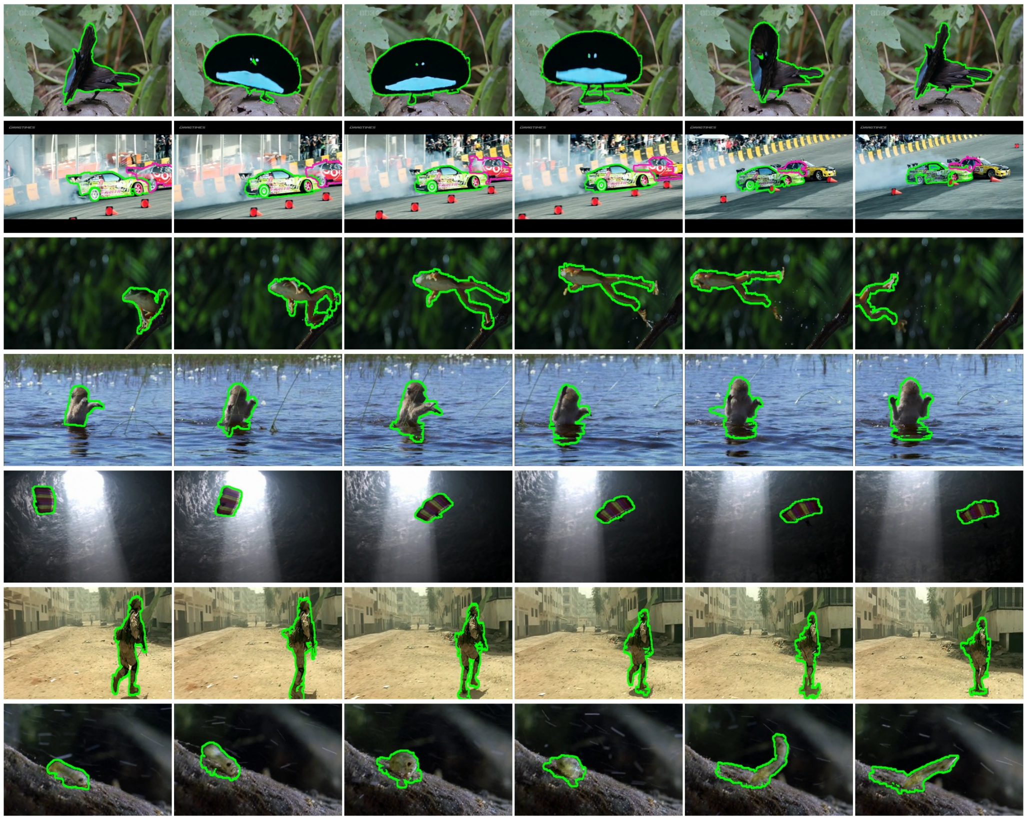


Figure 4.7: Qualitative results for video segmentation on Segtrack-v2 dataset: The results using my supervoxel based segmentation propagation method initialized from the segmentation in the left-most image. This initialization is obtained using our Click Carving method with static and motion-based proposals. Best viewed in color.

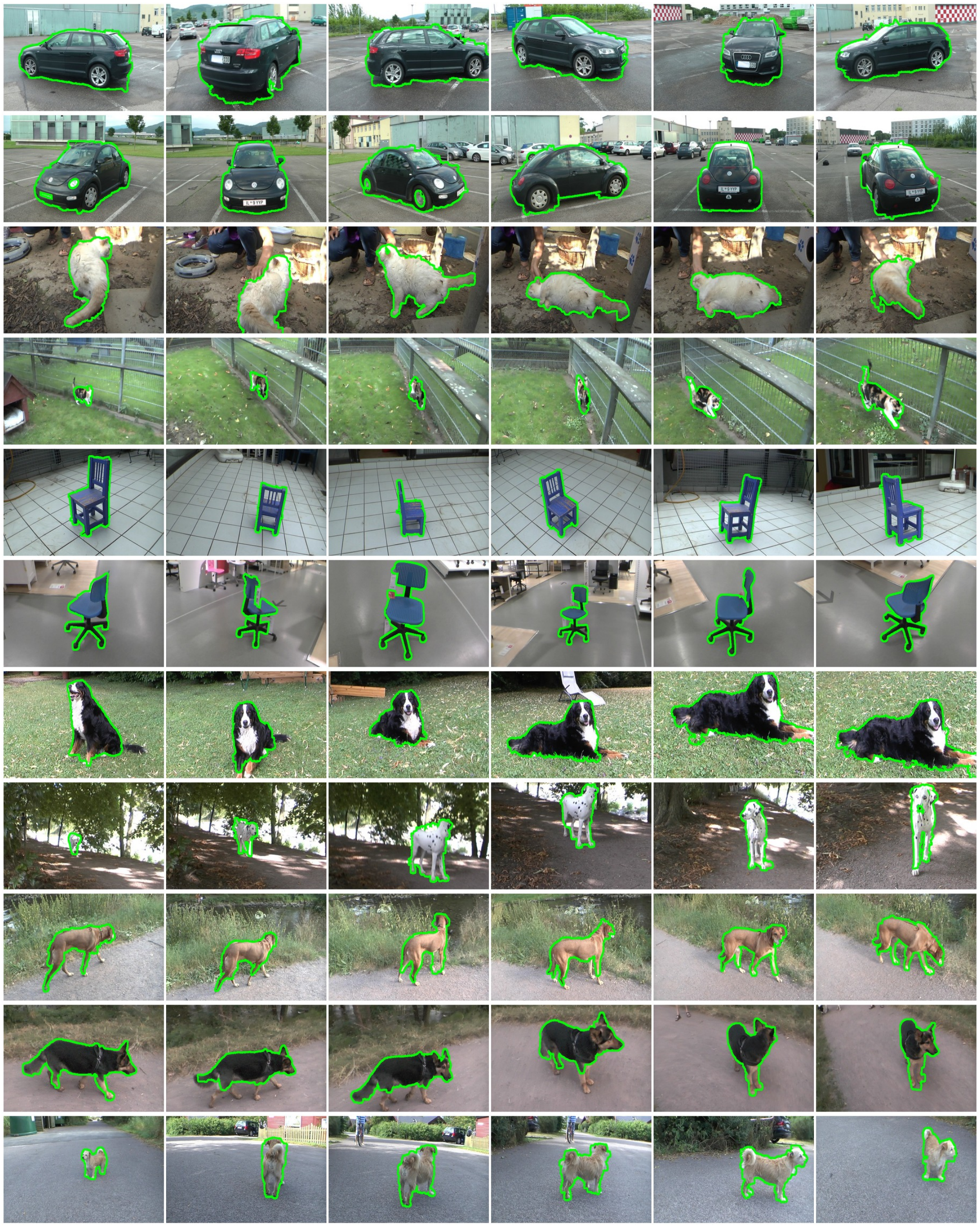


Figure 4.8: Qualitative results for video segmentation on iVideoSeg dataset: The results using my supervoxel based segmentation propagation method initialized from the segmentation in the left-most image. This initialization is obtained using our Click Carving method with static and motion-based proposals. Best viewed in color.

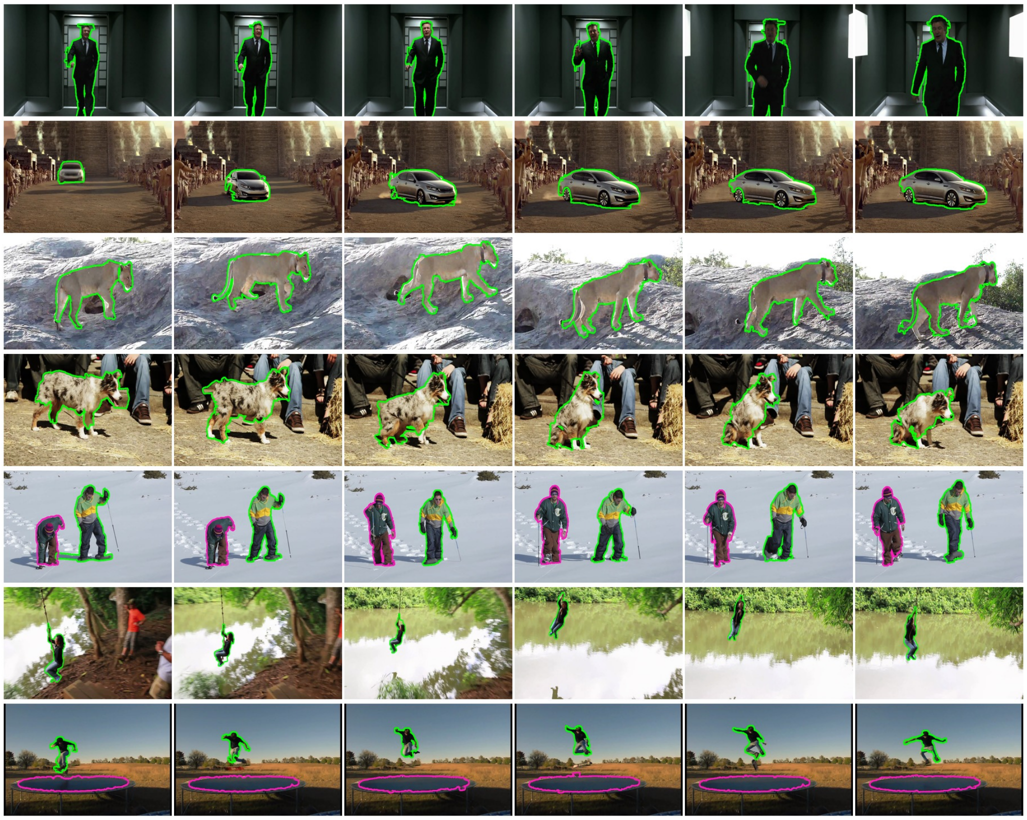


Figure 4.9: Qualitative results for video segmentation on VSB100 dataset: The results using my supervoxel based segmentation propagation method initialized from the segmentation in the left-most image. This initialization is obtained using our Click Carving method with static and motion-based proposals. Best viewed in color.

Chapter 5

Active segmentation propagation in image collections

[1]In the previous chapters, the graph cuts based interactive segmentation [13, 119] or the *Click Carving* algorithm [58] was applied individually on each image. For segmenting each and every image, the user needs to provide individual guidance through several possible modes of human interactions that I previously described. Even if we have to apply these algorithms on a collection of images, each image needs to be individually segmented by a human annotator. The underlying assumption thus far is that an image collection comprises of a set of totally unrelated images, hence each image needs to be segmented individually.

However, there has been a lot of recent interest in segmenting a pool of images known to contain the same object category (e.g. a collection of "airplane" images) [2, 30, 44, 64, 121, 122, 124, 131, 140]. These collections are readily available on Internet and are easy to obtain through a simple keyword search. However, collecting spatial annotations which delineate the boundaries

---

[1] The work in this chapter was supervised by Dr. Kristen Grauman and originally published [54] in: Active Image Segmentation Propagation. S. Jain and K. Grauman. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2016, Las Vegas, U.S.A.

of the common object in all images still remains challenging. It is natural to think that this additional information i.e., *all images contain objects from the same category* should be beneficial when segmenting images from such collections.

In this chapter, I explore the weakly supervised segmentation problem. It refers to the problem of segmenting a collection of images all of which belong to the same object category. I show that since the images here belong to the same category, the repeated patterns between them can be exploited for segmentation [2,30,44,64,121,124,131,140] and also while making annotation choices [122]. I show that this can be done by jointly segmenting all images in the collection by defining a joint segmentation graph over all images. This process mutually benefits individual images because information about the object will propagate from an image to its neighbors. I also show that instances from this image collection can be actively selected for human annotation by accounting for their overall utility for the entire collection. This allows us to inject actively chosen human annotations directly in the joint segmentation graph, which will guide the segmentations of other unsegmented instances.

More specifically, the proposed approach for joint segmentation of an image collection operates by alternating between these two components in a stage-wise manner:

1. **Segmentation propagation:** To propagate human-drawn segmentations from some subset of images to all unsegmented images, a joint

graph between object-like regions from all pairs of images is constructed. An energy minimization procedure on this joint graph is used for efficient propagation.

2. **Active selection:** To select a subset of images most suitable for effective segmentation propagation to all unsegmented images, a second joint graph between all image pairs is defined using global image similarity features. The active selection process favors choosing images that are *uncertain*—poorly explained by any images labeled so far, as well as *influential*—similar to many unlabeled images, making their foreground mask transferrable—and mutually *diverse*—so as to avoid redundant human effort.

Stagewise propagation is a key element in the proposed method's design, which permits both human-annotated *and* automatically annotated images to influence the system's view of what most needs human attention next. This characteristic of making stagewise active annotation choices separates it from other propagation based methods which are passive in nature (i.e., they try to best use a predefined set of labeled images) [44] or else selects them in a one-shot manner without reacting to the impact of previously annotated examples [122]. The proposed method in this chapter is also fundamentally different from the active learning methods for recognition which aim to train a model that will make accurate category label predictions on unseen test images (e.g., [127, 139, 142]). As such, they are tightly coupled to a particular classifier and iteratively refine it. In contrast, the goal here is to generate accurate foreground estimates for all images in the collection, which makes it a transductive setting.

Experiments demonstrate that the proposed method outperforms several alternative active baselines and methods which do passive labeling [44]. Applying this method to 1 million ImageNet images, the results show substantial savings in human annotation effort (upto 40% reduction in the amount of data annotated), thus emphasizing the value of intelligently focusing human effort for foreground extraction. As a special case, the proposed method is capable of running in a fully automatic manner too (i.e., without any human annotation), where it produces state-of-the-art foreground segmentation accuracy when compared to a variety of recent methods. Overall, the proposed method strikes an excellent balance between human annotation effort and accuracy. Depending on the amount of annotation budget available, the system can automatically adapt itself by requesting only the most useful instances for human annotation, thus resulting in much better segmentation performance than other methods within the prescribed budget constraints.

In the following, I first describe the regions and descriptors I use to construct the image graph (Sec. 5.1). Then I define my joint segmentation procedure to simultaneously solve for all foreground masks, given foreground annotations on only a subset of the images (Sec. 5.2). I then introduce my active procedure for identifying the set of images that should be annotated next (Sec. 5.3). Figure 5.1 visually illustrates all the steps. The remaining sections

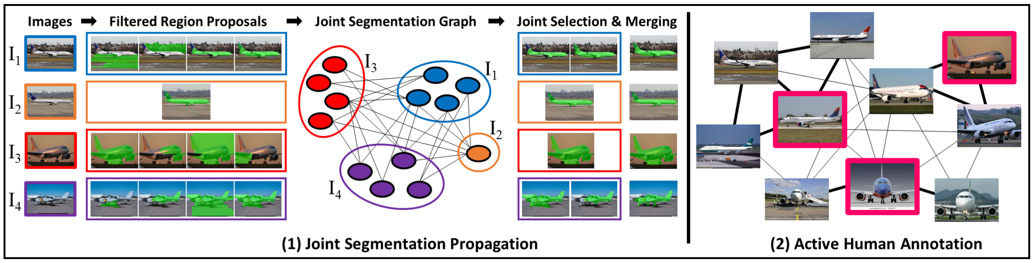

Images
Filtered Region Proposals
Joint Segmentation Graph
Joint Selection & Merging
I1
I2
I3
I4
I3
I1
I2
I4
(1) Joint Segmentation Propagation
(2) Active Human Annotation

Figure 5.1: **(1) Joint segmentation propagation:** Given a set of images $\{I_1, I_2, I_3, I_4\}$ with I_2 already segmented by a human, the goal is to generate foreground segmentations for the remaining images. First a set of filtered region proposals is generated for each image. Next, a joint segmentation graph over these region proposals (edges = region similarity) is defined. An energy function defined over this graph is minimized to obtain a set of good proposals for each image, which are then fused to obtain the final segmentation. **(2) Active human annotation:** My active selection method works over a joint graph defined over all images in the collection (**darker edges** = high similarity). These pairwise similarities allow us to identify influential images (most useful for others) and also help in enforcing diversity in selection (to avoid redundancy). Uncertainty (not depicted here) is also accounted for by predicting the quality of the current segmentation. Example selections by my method are shown in pink. Best viewed in color.

in this chapter then present detailed experimental results and comparisons with other state-of-the-art methods.

Problem setup: Let $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ be a collection of weakly supervised images, all of which contain instances of the same object category. My goal is to jointly segment these images, yielding a foreground object mask $\mathcal{M} = \{M_1, M_2, \ldots, M_N\}$ for each one.

5.1 Region proposals and descriptors

The segmentation graph in my method is defined over *region proposals*. Region proposals are "object-like" segments that are prioritized among all bottom-up regions as those being most likely to agree with true object boundaries [5, 22]. I assume that at least *some* of them capture the foreground object well—and possibly more than one per image. Thus, the goal of my joint segmentation procedure is to identify the subset of region proposals that are good, and fuse them to obtain the final segmentation (see Sec. 5.2 for details). Apart from being more efficient than traditional pixel-based graphs (e.g., [121]), I show that a region-based representation lets us define strong pairwise consistency potentials based on regions matched across images.

Existing region proposal methods typically produce $\sim$ 500-2000 regions per image, a large sample that may include redundant candidates and background objects. To refine the set of proposals, I develop the following filtering steps. First a set of generic object proposals are generated. Also a saliency map for the image is computed using [61]. Next two ranked lists of these proposals are obtained using saliency and objectness scores [22], respectively. Only the union of the top 30% from each list is retained. Then, the reduced set is clustered into r clusters. To capture shape and spatial alignment, respectively, the regions' HOG similarity and spatial overlap (IoU metric) are used, and the clustering is done using k-medoids. The r cluster centers (typically r=10) form the final set of proposals for each image. I found that this careful filtering was much more accurate than constraining the number of region proposals using the objectness scores directly. For example, on the MIT dataset my filtering step results in a mean average best score (MABO) of 72.2 with only 10 proposals. In contrast, simply retaining the top 10 proposals using scores from [22] results in a MABO of 64.95. The clustering step selects diverse proposals, leading to higher recall with fewer proposals.

Let $\mathcal{R} = \{R_{ij}\}$ denote the set of all region proposals in all N images, where R_{ij} denotes the j-th region for image I_i. My joint segmentation approach, to be defined next, relies on both image and region-level features. For each image I_i, a global appearance descriptor denoted I_i^{c} is extracted. For each region R_{ij}, two features are extracted: a saliency rating R_{ij}^{s}, and a region appearance descriptor R_{ij}^{c}.[2]

5.2 Semi-automatic joint foreground segmentation

I define a Markov Random Field (MRF) joint segmentation graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$ based on the filtered region proposals across all images in the collection. Each region $R_{ij} \in \mathcal{R}$ forms a node and the edges $\mathcal{E}$ connect pairs of regions. During segmentation, the edges will encourage consistent labels for similar regions, while the nodes will encourage foreground labels for salient regions that are consistent with well-segmented exemplars. A sparse set of edges $\mathcal{E}$ are kept by only connecting regions whose similarity exceeds a threshold τ. No edges connect regions in the same image.

Let $\mathcal{Y} = \{Y_{ij}\}$ be a set of binary region labels, where:

$$Y_{ij} = \begin{cases} 1 & \text{if proposal } R_{ij} \text{ is a good segmentation for } I_i \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

---

[2] One could choose from a variety of features; I employ off-the-shelf CNN-based descriptors and saliency metrics (see Sec. 5.4 for details).

Let $\mathcal{S} \subseteq \mathcal{I}$ denote the current subset of images labeled with foreground masks by human annotators. (I explain in Sec. 5.3 how the composition of this set is iteratively and actively defined.) Once an image I_s has been labeled, meaning it first appears in $\mathcal{S}$, the graph is adjusted accordingly. First, all nodes R_{sj} are replaced by the single mask region given by the human annotator, denoted $\bar{R}_s$, and its label is clamped to $Y_s = 1$. Then, the edge set $\mathcal{E}$ is modified appropriately, such that in image s, only the mask $\bar{R}_s$ has edges to similar regions in unlabeled images.[3] These updates inject the human-labeled regions into the segmentation pipeline, allowing us to propagate the valuable information through the pairwise terms (defined below).

There are several ways to use the human-labeled masks to guide the joint segmentation. One could use them to train a foreground appearance model (e.g., as in iCoseg [9]). However, this is most effective only in the stricter co-segmentation setting where the same exact foreground object instance repeats across images. An alternative could be to directly transfer the segmentation from labeled images to unlabeled images, e.g., using dense matching [90, 161]. However, due to variations in scale and shape of foreground objects, global alignment is difficult in many cases.

Instead, my approach relies on strong matches discovered between foreground regions in human-labeled images and region proposals in unlabeled im-

---

[3] For simplicity of notation, below I continue to use R_{ij} for all regions unless strictly required; it should be understood that $\forall I_i \in \mathcal{S}$ there is only one proposal, instead of r proposals.

ages. The intuition is that a good region proposal (i.e., one close to the actual foreground object segment) will strongly match a human-labeled ground truth region. On the contrary, a bad proposal will have weaker matches.

I define the following energy function $E(\mathcal{Y})$ for jointly segmenting the image collection $\mathcal{I}$:

$$E(\mathcal{Y}) = \sum_{R_{ij}} - \log \Phi(Y_{ij}) \;+ \sum_{R_{ij}, R'_{ij} \in \mathcal{E}} \Psi(Y_{ij}, Y'_{ij}). \tag{5.2}$$

The unary term is defined as

$$\Phi(Y_{ij}) = \begin{cases} Y_{ij} & \text{if } i \in \mathcal{S} \\ \alpha^{\mathrm{s}}\, \Phi^{\mathrm{s}}(Y_{ij}) \; + \alpha^{\mathrm{m}} \Phi^{\mathrm{m}}(Y_{ij}) & \text{if } i \in \mathcal{I} \backslash \mathcal{S}. \end{cases}$$

This unary prefers to label as foreground those regions that are (1) *salient* and/or (2) form a good *match* with some previously labeled foreground mask. The variables α^{s} and α^{m} weight the influence of the saliency and matching terms, respectively. The *saliency* term is defined using the saliency region feature (R^{s}_{ij}) as:

$$\Phi^{\mathrm{s}}(Y_{ij}) = Y_{ij} R^{\mathrm{s}}_{ij} + (1 - Y_{ij})(1 - R^{\mathrm{s}}_{ij}), \tag{5.3}$$

so that we favor assigning $Y_{ij} = 1$ if R_{ij} is very salient.

The *match* component of the unary term encodes that a region proposal with a good ground truth region match is likely foreground. In particular, matches for a region are identified by considering its "local neighborhood" of images in the graph. For each unlabeled image I_i, its p nearest neighbors from the labeled set $\mathcal{S}$ are retrieved using the image-level features I^{c}_i. Denote that set $\mathcal{N}(I_i, \mathcal{S})$. Then, for each region proposal R_{ij}, the best matching ground truth foreground region is found among these p neighbors, and the matching score is used in the unary term:

$$\Phi^{\mathrm{m}}(Y_{ij}) = Y_{ij}R^{\mathrm{m}}_{ij} + (1 - Y_{ij})(1 - R^{\mathrm{m}}_{ij}), \text{where} \tag{5.4}$$

$$R^{\mathrm{m}}_{ij} = \max_{p \in \mathcal{N}(\mathcal{I}_i, \mathcal{S})} sim(R^{\mathrm{c}}_{ij}, \bar{R}^{\mathrm{c}}_{p}), \tag{5.5}$$

and sim is the cosine similarity, and $\bar{R}_p$ denotes the p-th ground truth region.

The pairwise term in Eq (5.2) encourages similar-looking regions to take the same label:

$$\Psi(Y_{ij}, Y'_{ij}) = \delta(Y_{ij} \neq Y'_{ij}) \; sim(R^{\mathrm{c}}_{ij}, {R'}^{\mathrm{c}}_{ij}). \tag{5.6}$$

This term enforces consistency in my joint selection of good region proposals, since a penalty proportional to region similarity is incurred if the two regions receive different labels.

The minimum energy solution $\mathcal{Y}^* = \arg\min_{\mathcal{Y}} E(\mathcal{Y})$ yields a set of good region proposals for each image in the collection. Note that there is no constraint that only one proposal should be selected per image. It is purposely allowed to select *multiple* good regions per image, for two reasons. First, an image can naturally have multiple good region proposals (e.g., covering different object parts). As we will see next, my fusion step can take these multiple partial proposals to obtain a single accurate segmentation. Second, it allows us to efficiently and exactly minimize my energy function using graph-cuts [14]. I found that this works much better in practice than approximate inference techniques. A complete round of propagation for $N = 1,400$ images takes just **1 minute** on a single CPU (excluding feature extraction). In contrast, the state-of-the-art propagation method of [122] would take 225 hours to propagate labels (excluding both feature extraction and SIFT-Flow).

To obtain the final segmentation mask M_i, the chosen good region proposals Y_i^* are fused. The selected regions are used as a rough prior for the object's spatial extent, and then that's used to build an image-specific foreground appearance model. Specifically, for each chosen proposal in I_i, the p nearest human-labeled masks are retrieved. Those masks are transferred to I_i (a simple resizing and transfer, similar to [68, 79] is used), next the transferred masks of all proposals are averaged, and mean thresholded to obtain a spatial prior. Next, as an appearance prior a Gaussian Mixture Model (GMM) over RGB color values for all pixels in the spatial prior is learned. Finally, the combined appearance and spatial prior are used to define an image-specific MRF, which is minimized using graph cuts to obtain M_i.

In summary, my semi-supervised segmentation propagation algorithm is designed to be accurate (through careful filtering of regions and use of sparse actively chosen human annotations) and efficient (by avoiding expensive dense matching steps [121] and by using an efficient graph cuts energy minimization framework instead of costly approximate inference techniques as in [30]).

5.3 Active selection for propagation

I now describe my stagewise algorithm to actively select images for annotation. The active selection procedure takes as input the image collection $\mathcal{I}$, an annotation budget k specifying the number of images to get labeled per stage, and the number of total annotation stages T. In each stage t, annotations for the actively chosen batch $\mathcal{S}_t$ are collected, $\mathcal{S}$ is augmented with that newly labeled data ($\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_t$) next, and then the segmentations are propagated as described above. The output after T rounds is the resulting propagated masks $\mathcal{M}$ on all images. Note that throughout the stages, each unlabeled mask is continually refined, and its intermediate results affect subsequent stages' active selections.

My active selection algorithm accounts for three criteria—influence, diversity, and uncertainty. The former two criteria account for relationships between images that are relevant to propagation, while the latter accounts for the inherent difficulty of individual images.

An image *influential* for propagation is similar to many other images in the collection. Intuitively, labeling such a "hub" image can directly improve the mask quality of the related images, particularly given my match-based unaries and localized image neighborhoods (Eq (5.5) and Eq (5.6)). The influence of a candidate batch $\mathcal{S}_t$ is measured as:

$$\textsc{Influence}(\mathcal{S}_t) = \frac{1}{|\mathcal{S}'_t|} \sum_{I_i \in \mathcal{S}_t} \sum_{I_j \in \mathcal{S}'_t} sim(I_i^{\mathrm{c}}, I_j^{\mathrm{c}}), \tag{5.7}$$

where $\mathcal{S}'_t$ denotes all unlabeled images not in the candidate batch $\mathcal{S}_t$ and *sim* is the cosine similarity.

A batch of images that are *diverse* ensures broad coverage over the entire collection. Selecting images which are influential but also very similar would not lead to a large information gain. Hence, A penalty for selecting mutually similar images is also added:

$$\textsc{Diversity}(\mathcal{S}_t) = -\frac{1}{|\mathcal{S}_t|} \sum_{I_i \in \mathcal{S}_t} \sum_{I_j \in \mathcal{S}_t} sim(I_i^{\mathrm{c}}, I_j^{\mathrm{c}}). \tag{5.8}$$

An image that is *uncertain*—inherently difficult to segment automatically—is also a good candidate for human supervision. The uncertainty of a batch is quantified as:

$$\textsc{Uncertainty}(\mathcal{S}_t) = \frac{1}{|\mathcal{S}_t|} \sum_{I_i \in \mathcal{S}_t} D(M_i), \tag{5.9}$$

where $D(\cdot)$ is a learned predictor of image difficulty. This prediction function is trained to infer when an image is badly segmented. Taking inspiration from prior work [22, 52, 117], I devise a set of descriptors suggestive of segmentation quality, and train a regression function using images for which we know each region's overlap with the true foreground. Given a region, the predictor returns its expected normalized overlap with the ground truth.

Specifically, a random forest regressor is trained using 1,385 images from the MSRC [97], iCoseg [9], and IIS [45] datasets. The regression target is the overlap score with ground truth. To generate training samples, CPMC [22] region proposals are sampled whose overlap falls in the top and bottom 5% of all proposals. I use the following features as indicators of segmentation quality:

- **Boundary alignment:** Similar to the cue used in Chapter 3, alignment of a segmentation region with superpixel boundaries is used as a cue to measure segmentation quality. This is done by measuring how much each superpixel that lies on the boundary between foreground and background region maximally straddles inside or outside.

- **Object coherence:** Number of connected components in the image segmentation region is used as a measure of object coherence.

- **Color separability:** A good segmentation is likely to have a difference in appearance with the background. Color histograms in RGB space (16 bins per channel) for both the region proposal and background are computed and χ^2 distance between them is used as a feature.

- **Region compactness:** Good segmentations are more likely to be compact in nature. Hence I use the following region features to capture that:

    1. **Extent:** area ratio between the region and a tight bounding box surrounding it.
    2. **Solidity:** area ratio between region and it's convex hull
    3. **Size:** good segmentations are not usually abnormally large or small, hence we use area ratio between region and the complete image as another feature.

Algorithm 1 Active Selection Algorithm

```
procedure ActiveSelection
    Input: $\mathcal{I}$, $\mathcal{I}^u = \mathcal{I}$, $\mathcal{I}^l = \phi$;
    Define: $\mathcal{F}(\mathcal{S}) = \textsc{Influence}(\mathcal{S}) + \textsc{Diversity}(\mathcal{S})$, $\mathcal{S} \subseteq \mathcal{I}$;
    for each stage $t = 1,2, ..., T$ do
        Candidate set: $\mathcal{I}^u_t = \phi$;
        for $i = 1,2, ..., K$ do
            $s^* = \arg\max_{s \in \mathcal{I}^u \setminus \mathcal{I}^u_t} D(M_s^{t-1})$ ; $\mathcal{I}^u_t = \mathcal{I}^u_t \cup s^*$;
        end for
        $\mathcal{S}_t = \phi, \mathcal{S}_t^{'} = \mathcal{I}^u_t$;
        for $i = 1,2, ..., k$ do
            $s^* = \arg\max_{s \in \mathcal{S}_t^{'}} \mathcal{F}(\mathcal{S}_t \cup s) - \mathcal{F}(\mathcal{S}_t)$;
            $\mathcal{S}_t = \mathcal{S}_t \cup s^*$ ; $\mathcal{S}_t^{'} = \mathcal{S}_t^{'} \setminus s^*$;
        end for
        $\mathcal{I}^l = \mathcal{I}^l \cup \mathcal{S}_t$; $\mathcal{I}^u = \mathcal{I}^u \setminus \mathcal{S}_t$;
    end for
end procedure
```

We would like to identify the set maximizing all three criteria simultaneously. This is a combinatorial problem over all subsets $\mathcal{S}_t \subseteq \mathcal{I}$ and impractical to solve optimally. I instead employ a greedy approach to account for all factors. First, the $K > k$ most uncertain unlabeled images are extracted, as judged using the predictor $D(M_i)$ applied to the current mask estimated at the end of the previous stage. From among that pool, a subset $\mathcal{S}_t$, accounting for both influence and diversity is selected. Starting with an empty set, an image is iteratively added one at a time until the budget k is reached. The selected image is the one giving the maximal marginal increase for $\textsc{Influence}(\mathcal{S}_t) + \textsc{Diversity}(\mathcal{S}_t)$. See Algorithm 1 for complete pseudocode.

My greedy algorithm is inspired by the maximization procedure typically used for monotone submodular functions, which offers theoretical guarantees [75]. Due to the diversity penalty, my objective is non-monotonic, hence known approximation guarantees do not apply; nonetheless, it works well in practice. It is also fast: for a pool of 1,400 unlabeled images, my active selection requires just seconds.

5.4 Results

In this section, I provide detailed experiments and comparisons with state-of-the-art methods.

5.4.1 Datasets and baselines

Datasets: I evaluate the proposed active segmentation propagation algorithm on two benchmark datasets:

- **ImageNet:** I conduct a large-scale evaluation of my approach using ImageNet [123] (∼1M images, 3,624 classes). I follow the setup of [131], and consider all images with bounding box annotations available.[4] Figure 5.2 shows some visual examples from the dataset.
- **MIT Object Discovery:** This challenging dataset consists of Airplanes, Cars and Horses [121]. Its intra-class appearance variation is

---

[4] Since ImageNet lacks segmentation ground truth for all images, (1) I evaluate my masks against the bounding boxes, using a tight bounding box around the predicted segmentation and (2) when my method requests a human-drawn segmentation, it gets the region proposal with maximum overlap with the ground-truth bounding box.

Figure 5.2: Examples from ImageNet dataset. (best viewed in color).

much greater than that of older co-segmentation datasets (MSRC [97] or iCoseg [9]). Figure 5.3 shows some visual examples from the dataset.

Baselines: Apart from an ablated version of my method (i.e., w/o uncertainty), I compare with these baselines:

- **Passive:** This is a simple passive baseline where at every stage, k images are randomly picked from the unlabeled set to be labeled by humans.
- **PageRank Selection [122]:** This is the only active propagation method in the literature, making it critical for comparison. It uses PageRank importance ranking and clustering to pick k good images at each stage.
- **Semantic Propagation [44]:** An existing propagation method that promotes propagation between semantically related classes. It seeds the propagation with labeled images from existing datasets.
- **State-of-the art weakly supervised methods:** I compare the special case of my method (only weak supervision) with several existing approaches [25, 63, 64, 67, 121, 131]. Other weakly supervised methods [106,

108, 112] for semantic segmentation consider multi-label data, and so are not directly comparable.

Evaluation metrics: I use: (1) **Jaccard Score:** Standard intersection-over-union (IoU) metric between predicted and ground truth segmentation masks (for MIT) and between bounding boxes (for ImageNet), and (2) **Cor-Loc Score:** Percentage of images correctly localized according to PASCAL criterion (i.e IoU > 0.5) used in [131]. For MIT I use the segmentation masks (Seg-CorLoc) and for ImageNet I use bounding boxes (BBox-CorLoc) since it lacks ground truth masks.

Implementation details: Region proposals for MIT are generated using CPMC [22] and for ImageNet using MCG [5] (due to efficiency). For global appearance I_i^{c}, 4096-dim Convolutional Neural Network (CNN) features [77] are extracted using Caffe [60]. I chose CNN features because of their state-of-the-art performance in image recognition. For saliency R_{ij}^{s}, the region's pixel-level saliency values from [61] are averaged. For region appearance R_{ij}^{c}, a CNN feature for the region's tight bounding box is extracted. I set: $\tau = 0.7, p = 5, \alpha^{\mathrm{s}} = \alpha^{\mathrm{m}} = 0.5, \#$ rounds $T = 20, k = (\#$ images$/T), K = 4 * k$. All parameters were set after manual inspection of few images, then fixed for all experiments. In all experiments human annotation is simulated using ground truth data. The run-time is dominated by the cost of computing pairwise similarities between region proposals, $O((Nr)^2)$ for N images and r region proposals per image.

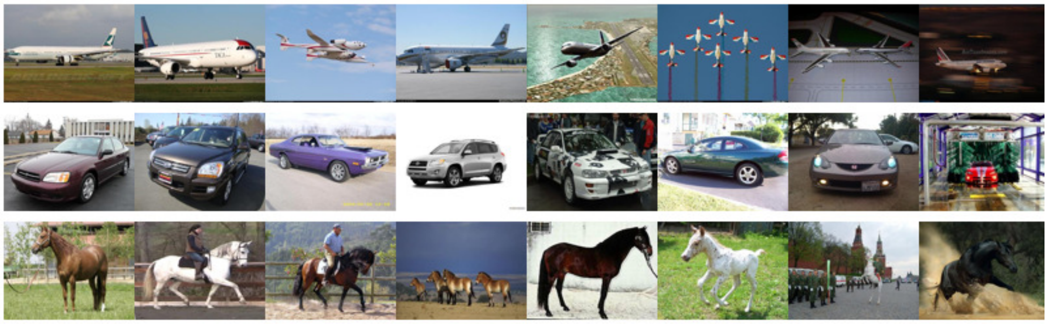


Figure 5.3: Example active annotation choices for the 3 image collections (Airplane, Car, Horse) in the MIT dataset during the first stage with $k = 10$. The algorithm selects influential and diverse images (e.g., prototypical shapes) with some relatively difficult/unusual ones (best viewed in color).

5.4.2 Active segmentation propagation

First I present results for active selection. In this setting annotators are iteratively requested to provide true segmentations for a subset of images. These labeled images are then used to improve the joint segmentation of other unlabeled images in the collection.

Figure 5.3 shows qualitative examples of annotation choices made by my active selection algorithm. The impact of all the components is quite visible in the choices. Several influential and diverse images which provide good coverage over the collection are chosen, along with some relatively difficult and unusual ones.

Figures 5.4 and 5.5 show the quantitative results. On the extreme left, we have the performance of the purely weakly supervised setting (no human input) and on the extreme right, annotators provide ground-truth segmentations for all images in the collection. In between we see the trade-off between

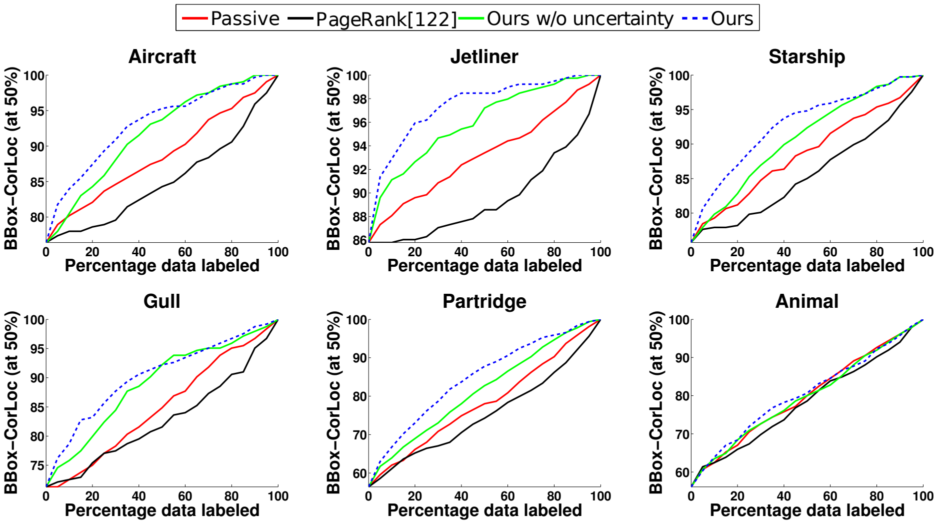


Figure 5.4: Active propagation for varying amounts of human annotation on a subset of the 3,624 ImageNet total synsets which were tested. Since only bounding box ground truth is available, I show bounding-box localization (BBox-CorLoc) accuracy. Last plot (Animal) shows a failure case. Best viewed in color.

actively allocating human effort versus other baselines. Since this is a transductive setting where the goal is to generate segmentations for all images, I plot average results over all the images in the collection (whether human or computer segmented). This scoring protocol has an additional advantage of averaging over the same number of images after each round of annotation, making trends on the x-axis easy to interpret.

For all metrics and datasets, the proposed approach outperforms all baselines. While all methods naturally improve with more labeled data, the slope of my improvement curve is substantially sharper using minimal human effort—sometimes dramatically so (e.g., Jetliner on ImageNet or Airplane on

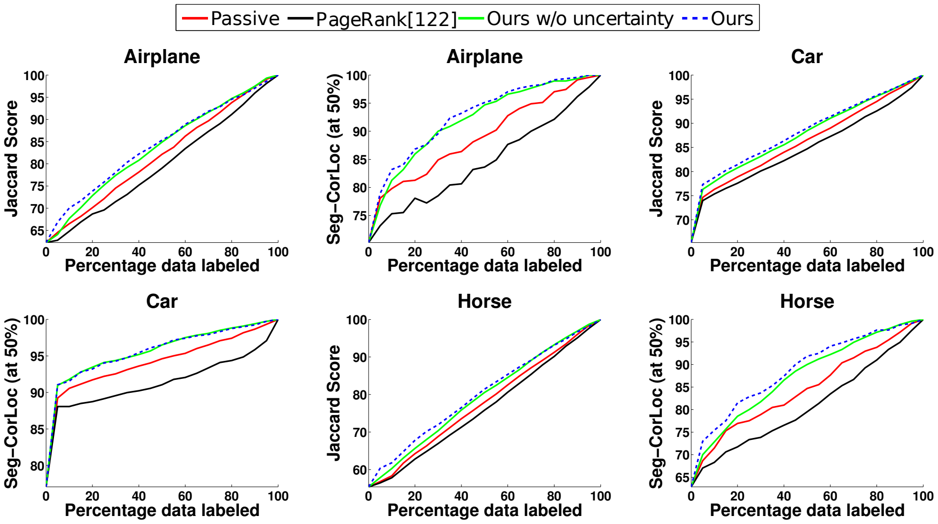


Figure 5.5: Active propagation results for varying amounts of human annotation for MIT Object Discovery dataset. I show both segmentation overlap (Jaccard) and segmentation localization (Seg-CorLoc) accuracy for each of the three classes. Best viewed in color.

MIT). It is important to note that all methods are using identical CNN features and the same propagation algorithm, hence my gains exactly show the impact of making wiser annotation choices.

Surprisingly, the Passive baseline outperforms the active PageRank method employed in [122]. I believe this is because PageRank emphasizes the influence property more, and, despite its clustering component, fails to select sufficiently diverse examples[5] (in [122] no comparison with a passive baseline is shown). On the other hand, my method takes into account influence, diversity, and uncertainty to choose good candidates for annotation.

---

[5] Restricting my proposed method to use "influence" alone also performs worse than passive and comparable to [122].

This leads to better annotation choices and in turn better propagation. Omitting uncertainty from my approach decreases accuracy, showing the value of this segmentation-specific active selection component.

While all methods fare better on the “easier” task of localization (vs estimating pixel-perfect masks), my gains are actually substantially higher for localization (as measured by Seg-CorLoc and BBox-CorLoc). In addition, for both datasets, my gains are much higher for larger collections (> 100 images). Larger collections exhibit both greater redundancy as well as several modes within the data. My method successfully exploits these patterns while making annotation choices. For example, for MIT “Airplanes”, the system correctly localizes 90% of the images with only 30% of the data labeled by annotators. In contrast, the Passive and active PageRank baselines require significantly more annotations (55% and 70%, respectively) to achieve the same accuracy.

Figure 5.4 also shows an interesting failure case for the ImageNet “Animal” class. Upon inspection, I found that it contains images from several different animal types with very little structural similarity; in this case, my active annotation method did not fare any better than the baselines.

I stress that, to my knowledge, Rubinstein et al.[122] represents the only prior attempt to incorporate active selection with segmentation propagation. Before any inference, that method seeds a dense-flow graph with images chosen with a PageRank sampling. My stage-wise method takes a very different strategy, iteratively self-inspecting its own estimates and redirecting human attention accordingly. As seen in Figure 5.4 and 5.5, my approach significantly outperforms the one-shot PageRank approach [122] in all experiments, and my propagation method is orders of magnitude faster (cf. Sec 5.2).

I also compare with the other state of the art segmentation propagation approach from Guillaumin et al. [44]. In their method, segmentations are propagated from a fixed set of seed images. The propagation in that case makes use of a semantic hierarchy and the propagation takes place between semantically related categories. For a fair comparison, all images which are common between my experimental setup and that of [44] are considered. This gives us a total of 99,020 images across 352 ImageNet classes. From the data provided by the authors, it was found that the ground-truth bounding boxes for 67,029 of those images were used to seed the propagation in [44]. For the same amount of labeled data my active segmentation propagation approach achieves a Jaccard score of 65% as opposed to 62.63% by [44]. More importantly, reducing the supervision budget for my method, it achieves the same accuracy as this (passive) state of the art propagation method [44] when using 26% *less* human-annotated data. This large savings in human effort shows the clear value of actively determining where human guidance is most needed.

5.4.3 Weakly supervised foreground segmentation

Next I test my method in a purely weakly supervised setting against several existing methods. In this special case, weak supervision (i.e., all images have an object from the same category) is the only information available. No additional human annotation is requested. This corresponds to setting $\mathcal{S} = \emptyset$,

Methods	**MIT dataset (subset)**			**MIT dataset (full)**		
	Airplane	**Car**	**Horse**	**Airplane**	**Car**	**Horse**
# Images	82	89	93	470	1208	810
Joulin et al. [63]	15.36	37.15	30.16	n/a	n/a	n/a
Joulin et al. [64]	11.72	35.15	29.53	n/a	n/a	n/a
Kim et al. [67]	7.9	0.04	6.43	n/a	n/a	n/a
Rubinstein et al. [121]	55.81	64.42	51.65	55.62	63.35	53.88
Chen et al. [25]	54.62	**69.2**	44.46	60.87	62.74	**60.23**
Ours	**58.65**	66.47	**53.57**	**62.27**	**65.3**	55.41

Table 5.1: Comparison with state-of-the-art methods on MIT dataset for weakly supervised joint foreground segmentation (Metric: Jaccard score).

$\alpha^{\mathrm{s}} = 1$ and $\alpha^{\mathrm{m}} = 0$.

Table 5.1 compares my approach to several existing methods [25, 63, 64, 67, 121] on the MIT (subset from [121] and full) dataset. My approach outperforms all existing methods in 4 out of 6 cases and has consistently good accuracy in all cases. This is really encouraging because my joint segmentation model is simpler and more efficient than existing methods (e.g [121] uses dense matching, [25] uses negative training data to train detectors). The key strengths of my propagation design lie in carefully selecting region proposals that have good coverage over the objects and are not redundant (without this performance drops by 8% on average), combined with the region-based matching potentials. Jointly selecting good region proposals then helps in discovering similar pattern configurations over the entire collection. The method of [25] possibly benefits from stronger discriminative exemplar-appearance models for the Horse class in MIT (full).

Table 5.2 shows results on ImageNet. The “Top obj” baseline is the

ImageNet dataset	
# Classes	**# Images**
3,624	939,516

Methods	**BBox-CorLoc**
Top obj. box [3]	37.42
Tang et al. [131]	53.20
Ours	**57.64**

Table 5.2: Comparison with state-of-the-art methods on ImageNet for weakly supervised joint foreground segmentation (Metric: Avg. BBox-CorLoc).

result of taking the top Objectness window [3], as reported in [131]. My method outperforms the state of the art [131] by a considerable margin, which again highlights the strengths of my joint segmentation graph. With nearly 1 million images, a performance gain of 4.44% means that the system correctly localizes 41,715 more images than [131].

Figure 5.6 shows qualitative results. My method is able to segment objects well in spite of large intra-class variations. Because of the joint segmentation graph, my method can successfully segment some challenging instances where the object is not easily separable from the background but matches well with similar regions in easier images.

5.5 Conclusion

In this chapter, I proposed an approach for active segmentation propagation in weakly supervised image collections. The proposed approach can actively request human annotations which are most useful for the entire collection as a whole. Having a subset of images actively labeled by human annotators, the proposed approach can then propagate the human labeled segmentations to the unsegmented images in the collection. Experimental results

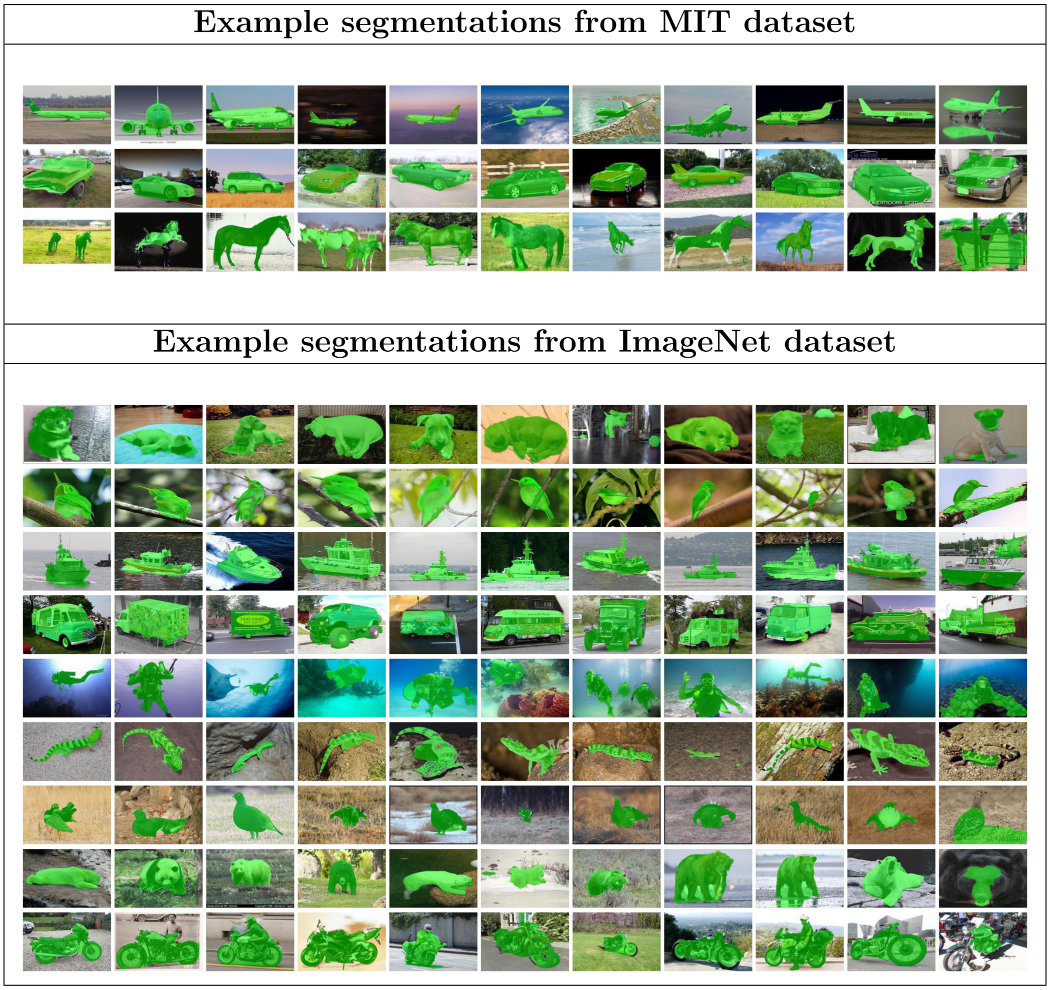

Example segmentations from MIT dataset
Example segmentations from ImageNet dataset

Figure 5.6: Qualitative results for weakly supervised joint segmentation. The segmentation result is highlighted with a green overlay over the image. The last column in each row shows a failure case. Failures occur when there is ambiguity in the object of interest (e.g., airplane's shadow, top row in MIT dataset) or parts of the object are more salient than the complete object (e.g., dog's face, top row in ImageNet dataset). Best viewed in color.

show that even without any human annotation, the proposed method outperforms several state-of-the-art methods for weakly supervised segmentation.

The active selection algorithm significantly outperforms the baseline methods, and makes better annotation decisions leading to better segmentation propagation. Overall, the proposed method results in an excellent trade-off between cost and accuracy and can adapt itself depending on the amount of human annotation budget available (including zero budget – where it can operate in a purely automatic manner).

In the future, the proposed approach can be enhanced by extending the ideas of active selection and segmentation propagation to more heterogeneous multi-class image collections. While the proposed method's design does not preclude it from being applied on heterogeneous multi-class collections, the current set of experiments always assumed that the image collection contains images from a single category. It will be interesting to relax this assumption and study the interactions between instances from different categories and whether they can still benefit from joint segmentation and active selection.

In the current experimental setup, the same amount of human annotation effort is assumed for every image. However in practice it is not true; As we saw in Chapter 3, different images may contain objects with different complexities and the annotation time will vary accordingly. Moreover, for each image which is selected for human annotation, segmentation is drawn from scratch by the human annotator. This is clearly sub-optimal because at every step in the overall process, each image has a current segmentation hypothesis. This can serve as an initial segmentation for the human annotator who can simply edit this segmentation instead of drawing from scratch. This process can possibly reduce the total amount of annotation time required for that particular image.

Finally, the current method works under the assumption that simple nearest neighbors in image feature space is a sufficiently strong metric to construct the joint segmentation and active selection graphs. However in complex images, a global similarity metric might not be sufficient to capture fine-grained intra-class variations. Discovering better ways to construct these graphs, which possibly capture finer nuances of segmentation propagation, is an interesting research direction to explore. In the next chapter, I will present my proposed algorithm for predicting compatibility between images for joint segmentation, which is a first step towards exploring this idea.

Chapter 6

Predicting compatibility for joint segmentation of image pairs

[1] In the previous chapter, I presented an active segmentation propagation algorithm which allows us to jointly segment a weakly supervised image collection. A key component in my pipeline was an algorithm to actively select images which will be most valuable for human segmentation. There I used image features extracted from the entire image to measure the compatibility of segmentation while constructing the joint graph over all images in the collection. In particular, image-nodes were only linked if they were very similar in this feature space. However global features may not completely capture the structural similarity between image pairs, which is naturally a key driver for a successful segmentation propagation. More specifically, global features may not be able to capture scale or viewpoint variations, or diversity in an object category's visual appearance. This has an unwanted effect of pairing *incompatible* image pairs for joint segmentation.

However, the majority of the existing methods for joint segmentation

---

[1] The work in this chapter was supervised by Dr. Kristen Grauman and originally published [53] in: Which Image Pairs Will Cosegment Well? Predicting Partners for Cosegmentation. S. Jain and K. Grauman. In Proceedings of the Asian Conference on Computer Vision (ACCV), 2014, Singapore.

simply assume that all images are mutually amenable to a joint segmentation [2, 21, 63, 64, 67, 133, 140, 154]. This assumption was only true in very early works in this domain which focused on jointly segmenting input images showing the very same object against distinct backgrounds [120]. However, in modern image collections with large intra-class variations in appearance and viewpoints, possibly containing noisily labeled instances [80, 121], this no longer holds true. In fact, for this very reason, recent studies report the discouraging outcome that, on some datasets, standard single-image segmentation actually exceeds its joint segmentation counterpart—despite the latter's presumed advantage of having access to a batch of weakly labeled data [121, 140].

In this chapter, I reconsider this assumption and demonstrate that *not all images are mutually valuable for joint segmentation*. Pairing an image with a right partner can lead to an improved segmentation performance. As a first step in this direction, I study this problem in a limited setting where only a single pair of images will be segmented in a joint manner. The problem of jointly segmenting only a pair of images is commonly referred to as "cosegmentation" in the literature, which is the terminology I use in this chapter. For this I also develop a new joint segmentation algorithm which is more suitable for single image pairs. Note that this is a simplified setting in comparison with the previous chapter, where I developed an algorithm which can jointly segment an entire collection of images simultaneously. However this limited setting allows us to carefully study the "compatibility" aspect of the problem. Incorporating the techniques described in this chapter into my joint active segmentation propagation approach is a promising future direction.

As input, my method takes a **"query"** image I^q and a pool of candidate partner images $\mathcal{P} = \{I^1, \ldots, I^N\}$. Among those N candidates, my method selects the best partner image for I^q, that is, the image that when paired with I^q for cosegmentation is expected to produce the most accurate result. Then, as output, my method returns the result of cosegmenting I^q with its selected partner, namely, a foreground mask for I^q. In the following, I refer to a candidate partner image as a **"source"** image, denoted $I^s \in \mathcal{P}$.

For predicting the best partner for a **"query"** image, I introduce a learning approach that uses a paired description of the **"source"** and **"query"** images to predict their degree of joint segmentation success. The paired description captures not only to what extent the images seem to agree in appearance, but also the uncertainty resulting from their shared foreground model. I formulate the task in a learning-to-rank objective, where successful pairs are constrained to rank higher than those that will likely segment poorly together.

Same as the previous chapter, I study the weakly supervised setting, where images in $\mathcal{P}$ contain the same object category as I^q. This forces my method to perform fine-grained analysis to select among all the possibly relevant partners. Even with weak supervision, not all images are satisfactory cosegmentation partners, since they contain objects exhibiting complex appearance and viewpoint variations, as discussed above. Experiments on two challenging datasets show that there is great potential in focusing joint segmentation only on those images where it is most valuable.

In the following, I first define a basic single-image segmentation algorithm (Sec. 6.1). I then expand that basic engine to handle cosegmentation of a pair of images (Sec. 6.2). I then introduce my ranking approach to predict the compatibility of two images for cosegmentation (Sec. 6.3). The remaining sections in this chapter then present detailed experimental results and comparisons with other state-of-the-art methods.

6.1 Single-image segmentation engine

I first describe an approach to perform *single-image* segmentation. In addition to serving as a baseline for the cosegmentation methods, I also use the output of the single-image segmentation when cosegmentation compatibility is predicted (cf. Sec. 6.3). The method below produces good foreground initializations, though alternative single-image methods could also be plugged into my framework.

Given an image I^i, the goal is to estimate a label matrix L^i of the same dimensions, where $L^i(p) = y_p^i$ denotes the binary label for the pixel p, and $y_p^i \in \{0, 1\}$. The label 0 denotes background (bg) and 1 denotes foreground (fg). I use a standard Markov Random Field (MRF) approach, where each pixel p is a node connected to its spatial neighbors.

I define the MRF's unary potentials using saliency and a foreground color model, as follows. Since this is a single-image segmentation, there is no external knowledge about where the foreground is. Thus, we rely on a generic saliency metric to estimate the plausible foreground region, then boostrap an approximate foreground color model from those pixels. Specifically, for image I^i, first its pixel-wise saliency map S^i is computed using a state-of-the-art algorithm [61]. Next, that real-valued map is thresholded by its average, yielding an initial estimate for the foreground mask. Then, the pixels inside (outside) that mask are used to learn a Gaussian mixture model (GMM) for the foreground (background) in RGB space. Let G^i_{fg} and G^i_{bg} denote those two mixture models.

The single-image MRF energy function uses these color models and the saliency map:

$$E_{sing}(L^i) = \sum_p A^i_p(y^i_p) \ + \sum_p X^i_p(y^i_p) \ + \sum_{p,p' \in \mathcal{N}} T^i_{p,p'}\big(y^i_p, y^i_{p'}), \tag{6.1}$$

where A^i_p and X^i_p are unary terms, $T^i_{p,p'}$ is a pairwise term, and $\mathcal{N}$ consists of all 4-connected neighborhoods. The *appearance likelihood* term is defined as:

$$A^i_p(y^i_p) = -\log P(F^i(p)|G^i_{y^i_p}), \tag{6.2}$$

where $F^i(p)$ denotes the RGB color for pixel p in image I^i. This term reflects the cost of assigning a pixel as fg (bg) according to the GMM models. The *saliency prior* unary term is defined as:

$$X^i_p(y^i_p = 1) = -\log P(S^i(p)), \tag{6.3}$$

where $S^i(p)$ denotes the saliency value for pixel p. This term reflects the cost of assigning a pixel as fg, where more salient pixels are assumed more likely to be foreground. For the background label, we have the corresponding term,

$X_p^i(y_p^i = 0) = -\log(1 - P(S^i(p)))$. Finally, the pairwise term,

$$T_{p,p'}^i(y_p^i, y_{p'}^i) = \delta(y_p^i \neq y_{p'}^i) \exp(-\beta \|F^i(p) - F^i(p')\|), \tag{6.4}$$

is a standard smoothness prior that penalizes assigning different labels to neighboring pixels that are similar in color, where β is a scaling parameter.

I employ graph cuts to efficiently minimize Eqn. 6.1 and apply five rounds of iterative refinement (as in GrabCut [119]), alternating between learning the likelihood functions and obtaining the label estimates. The result is a label matrix $L_{sing}^{i*} = \arg\min_{L^i} E_{sing}(L^i)$.

6.2 Paired-image cosegmentation engine

Next I define the cosegmentation engine I use in my implementation, which expands on the single-image approach above. During training, my method targets a given cosegmentation algorithm, as I show in the next section. Any existing cosegmentation algorithm could be plugged in; the role of my method is to improve its results by focusing on the most compatible image partners.

Given a query and source image pair, I^q and $I^s \in \mathcal{P}$, an energy function over their joint labeling is defined. This model is initialized using GMM appearance models learned from $L_{sing}^{q^*}$ and $L_{sing}^{s^*}$, the single-image results for the two inputs obtained by optimizing Eqn. (6.1). Specifically, the foreground (background) pixels from both label masks are pooled to learn the joint GMM G_{fg}^{qs} (G_{bg}^{qs}) in RGB space. Here and below, the superscript qs denotes a joint term that is a function of both the query and source images.

Let L^{qs} be shorthand for the two label matrices output by the cosegmentation, $L^{qs} = (L^q, L^s)$. My joint energy function takes the following form:

$$E_{coseg}(L^{qs}) = E_{sing}(L^q) \ + E_{sing}(L^s) \ + \Theta^{qs}_{app}(L^{qs}) \ + \Theta^{qs}_{match}(L^{qs}), \tag{6.5}$$

where the first two terms refer to the single-image energy for either output, as defined in Eqn. (6.1), and Θ^{qs}_{app} and Θ^{qs}_{match} capture the energy of a joint label assignment based on appearance and matching terms, respectively (and will be defined next). Note that even though the energy function contains terms for individual label matrices, they are optimized *jointly* to minimize Eqn. (6.5).

The *joint appearance likelihood* term is defined as

$$\Theta^{qs}_{app}(L^{qs}) = \sum_{p \in I^q} A^{qs}_p(y^q_p) \ + \sum_{r \in I^s} A^{qs}_r(y^s_r), \tag{6.6}$$

and it captures the extent to which the two output masks deviate from the expected foreground/background appearance discovered with saliency. As before, each A^{qs}_p term is defined as the negative log likelihood over the GMM probabilities; however, here it uses the joint GMM appearance models G^{qs}_{fg} and G^{qs}_{bg} obtained by pooling pixels from the two images' initial foreground estimates.

The *matching likelihood* term $\Theta^{qs}_{match}(L^{qs})$ leverages a dense pixel-level correspondence to establish pairwise links between the two input images. Let $\mathcal{F}_{qs}(p)$ denote the 2D flow vector from pixel p in image I^q to its match in image I^s. I introduce an edge in the cosegmentation MRF connecting each pixel $p \in I^q$ to its matching pixel $r \in I^s$, where $r = p + \mathcal{F}_{qs}(p)$. Using these correspondences, the matching likelihood is a contrast-sensitive smoothness potential over linked (matched) pixels in the two images:

$$\Theta^{qs}_{match}(L^{qs}) = \sum_{p \in I^q, r \in I^s} \delta(y^q_p \neq y^s_r) \exp(-\beta \| D^q(p) - D^s(r) \|), \tag{6.7}$$

where $D^i(p)$ is a local image descriptor computed at pixel p (I use dense SIFT [90]), and β is a scaling constant. This energy term encourages similar-looking *matched* pixels between the query and source to take the same fg/bg label.

The matching in Eqn. (6.7) helps cosegmentation's robustness. I compute $\mathcal{F}_{qs}$ using the Deformable Spatial Pyramid (DSP) matching algorithm [69], an efficient method that regularizes match consistency across a pyramid of spatial regions and permits cross-scale matches. By linking $p \in I^q$ to $r \in I^s$—rather than naively linking $p \in I^q$ to $p \in I^s$—I gain robustness to the translation and scale of the foreground object in the two input images. This is valuable when the inputs do share a similar-looking object, but its global placement or size varies. Notably, this flexibility is lacking in a strictly image-based global comparison approach (like GIST [134] and the scale-sensitive SIFT Flow as used in [121]). It thus enables mutual discovery of the object between the two images.

To optimize Eqn. (6.5), I again employ graph cuts with iterative updates. This yields the cosegmented output image pair:

$$(L^{q^*}_{coseg}, L^{s^*}_{coseg}) = \arg\min_{L^{qs}} E_{coseg}(L^{qs}). \tag{6.8}$$

Note that the Markov Random Field (MRF) models defined in this chapter are similar to the other MRF models which were defined in the previous chapters at a high level. In all cases, the models were primarily designed to capture the affinity of pixels in images to be foreground or background. The details differ in individual cases. For example, in this Chapter and in Chapter 3, the MRFs were defined over pixels which is more suitable for segmenting individual images or a single pair. In Chapter 5 it was defined over regions because it allowed us to scale the model on a large number of images.

6.3 Learning cosegmentation compatibility to predict partners

Having defined the underlying single-image and paired-image segmentation algorithms, I can now present my approach to predict which partner image is best suited for cosegmentation with a novel query image. There are two main components:

1. Extracting features that are suggestive of cosegmentation success.
2. Training a ranking function to prioritize successful partners.

We are given a training set $\mathcal{T} = \{(T^1, L^1), \ldots, (I^M, L^M)\}$ of M images labeled with their ground truth foreground masks, where T^i denotes an image and L^i denotes its mask. This set is not only disjoint from the candidate partner set $\mathcal{P}$ defined above, it also does *not* contain images of the same object category as what appears in $\mathcal{P}$ or the eventual novel queries. This is important, since it means my approach is required to learn generic cues indicative of cosegmentation compatibility, as opposed to object-specific cues. While object-specific cues are presumably easier to exploit, it may be impractical to train a model for every new object class of interest. Instead, all learning is done on data and classes disjoint from the weakly supervised image set $\mathcal{P}$.

Training a ranker for cosegmentation compatibility First, the cosegmentation algorithm (Sec. 6.2) is applied to every pair of images in $\mathcal{T}$. Each image in the training set acts as a "query" in turn, while the remaining images act as its candidate source images. Let (T_q^i, T_s^j) denote one such query-source pair comprised of training images T^i and T^j. For each pairing, the cosegmentation quality that results for T_q^i is recorded, that is, the intersection-over-union overlap score between the ground truth L^i and the cosegmentation estimate L_{coseg}^{i*} that results from optimizing Eqn. (6.5) with T^i as the query and T^j as the source. After computing these scores for all training pairs $(i, j) \in \{1, \ldots, M\}$, we have a set of training tuples $\langle T^i, T^j, o_{ij} \rangle$, where o_{ij} denotes the overlap score for pair i, j. The scores will vary across pairs depending on their compatibility.

Next, a ranked list of source images is generated for each training example. These M-length ranked lists are used to train a ranking function. As input, the learned ranking function f takes features computed on an image pair $\phi(I^q, I^s)$ (to be defined below), and it returns as output a score predicting their cosegmentation compatibility. For simplicity a linear ranking function is trained:

$$f(\phi(I^q, I^s)) = \boldsymbol{w}^T \phi(I^q, I^s), \tag{6.9}$$

where $\boldsymbol{w}$ is a vector of the same dimensionality as the feature space. To learn $\boldsymbol{w}$ from the training tuples, we want to constrain it to return higher scores for more compatible pairs. Let $\mathcal{O}$ be the set of *pairs* of all training tuples $\{(i,j),(i,k)\}$ for which $o_{ij} > o_{ik}$, for all $i = 1, \ldots, M$. Using the SVM Rank formulation of [62], I seek the projection of the data that preserves these training set orders, with a regularizer that favors a large margin between nearest-projected pairs:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\boldsymbol{w}||_2^2 + C \sum \xi_{ijk}^2 \\ \text{s.t.} \quad & \boldsymbol{w}^T \phi(T^i, T^j) \geq \boldsymbol{w}^T \phi(T^i, T^k) + 1 - \xi_{ijk} \\ & \forall (i,j,k) \in \mathcal{O}, \end{aligned} \tag{6.10}$$

where the constant C balances the regularizer and constraints. In other words, the model should score a training pair with greater overlap higher than one with lower overlap.[2]

Defining features indicative of compatibility Next I define the features $\phi(I^q, I^s)$. Their purpose is to expose the images' compatibility for cosegmentation. I define features of two types: 1) *source image features* meant to capture

---

[2] Alternatively, one could use regression. However, ranking has the advantage of giving us more control over which training tuples are enforced, and it places emphasis only on the relative scores (not absolute values), which is what I care about for deciding which partner is best.

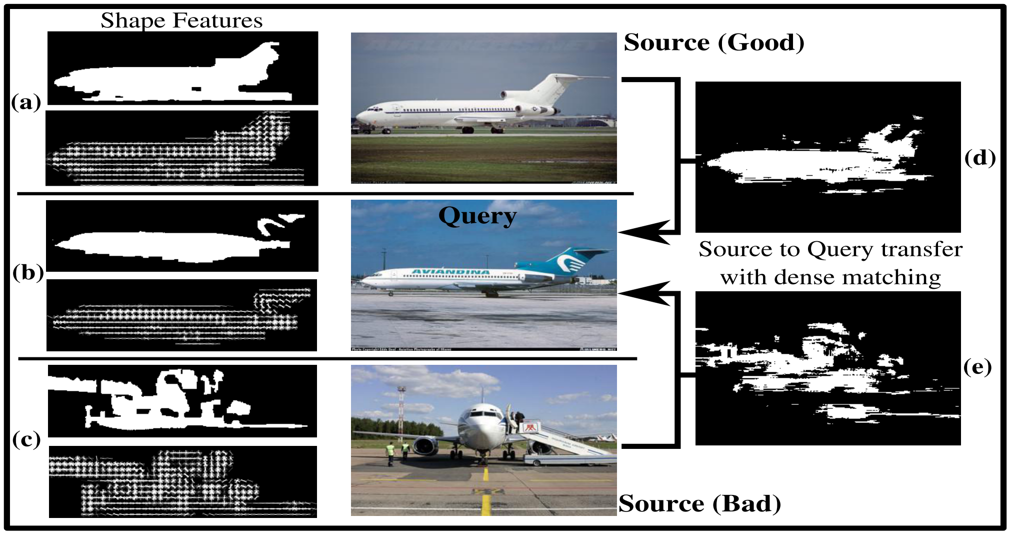


Figure 6.1: Feature illustration. **Center:** an example query and two candidate source images. **(a-c)**: Cropped single-image segmentation masks (top) and corresponding HOGs (bottom). These features are good indicators of foreground shape similarity, as we can see by comparing the query (b) to its good and bad source partners (a) and (c), respectively. **(d-e)**: Results of mask transfer with dense matching from the source image to the query image. The success of this transfer clearly depends on the compatibility between the query and source (i.e., it succeeds in (d) but fails in (e)).

the quality of the source in general, and 2) *inter-image features* meant to capture the likelihood of success in coupling a particular source and query. The former makes use of the single-image segmentation mask $L^{s^*}_{sing}$ from Sec. 6.1; the latter makes use of the cosegmentation estimates $L^{q^*}_{coseg}$ and $L^{s^*}_{coseg}$ from Sec. 6.2.

Source image features Ideally, we would like to cosegment with a source image that is easy to segment on its own, since then it has better ability to guide the foreground (when the query is compatible). Thus, my three source features aim to expose the predicted quality of the source's single-image segmentation:

- *Foreground-background separability*: $L_{sing}^{s^*}$ is first used to compute separate color histograms for the (estimated) foreground and background regions. The χ^2 distance between the two histograms is used as a feature. More distinctive foregrounds will yield higher χ^2 distances.
- *Graph cuts uncertainty*: Dynamic graph cuts [73] are used to measure each pixel's graph cut uncertainty. These uncertainties are binned from the foreground pixels of $L_{sing}^{s^*}$ into 5 bins and this distribution is used as a feature. It captures how uncertain the single image segmentation is.
- *Number of connected components*: The number of connected components in $L_{sing}^{s^*}$ is used as a measure of how coherent the source's single-image segmentation is.

Inter-image features To detect good partner candidates, the quality of the source image alone is insufficient; I also want to look explicitly at the compatibility of the particular input pair. Thus, my three inter-image features aim to reveal the predicted success of the pair's cosegmentation:

- *Foreground similarity*: The foreground similarity between the source and query is computed using their estimated foregrounds from single-image segmentation. Specifically, two χ^2 distances are recorded: one between their color histograms, and one between their SIFT bag-of-words histograms. By excluding background from this feature, we leave open the possibility to discover compatible partners with varying backgrounds.

- *Shape similarity*: The cropped foreground region from $L^{s^*}_{sing}$ is resized to the size of the cropped foreground region from $L^{q^*}_{sing}$. To gauge shape similarity, both the overlap between those masks as well as the L_2 distance on the HOG features computed on the original images at those masked positions (see Figure 6.1 (a-c)) are recorded.
- *Dense matching quality*: $L^{s^*}_{sing}$ is warped to the query using the dense matching flow field $\mathcal{F}_{qs}$ from DSP [69]. To capture the matching quality, the overlap score between the transferred source mask and $L^{q^*}_{sing}$ (see Figure 6.1 (d-e)) is recorded. Here the saliency-driven foreground masks and dense matching serve as two independent signals of alignment. If the two images permit an accurate dense match that agrees with the saliency-based foreground, there is evidence that they are closely related. This compatibility cue offers some tolerance to foreground translation and scale variation in the two inputs.
- *GIST similarity*: To capture global layout similarity of the image pair, the L_2 distance between their GIST [134] descriptors is recorded.

Altogether, we have seven and six feature dimensions for the source and inter-image features, respectively. These are concatenated to form a 13-dimensional $\phi(I^q, I^s)$ feature. These descriptors are used in training (Eqn. (6.10)).

Analyzing the learned weights, we find that the dense matching quality, shape similarity, GIST similarity, and foreground-background separability are the most useful features for this task.

Predicting the partner for a novel image At test time, we are given a novel image I^q and the partner candidate set $\mathcal{P}$. The algorithm operates by computing its descriptor $\phi(I^q, I^s)$ for every $I^s \in \mathcal{P}$, applying the learned ranking function, and selecting as its partner the one that maximizes the predicted cosegmentation compatibility:

$$I^{p^*} = \arg\max_{I^s \in \mathcal{P}} f(\phi(I^q, I^s)). \tag{6.11}$$

Finally, the foreground segmentation for I^q that results from cosegmenting the pair (I^q, I^{p^*}) using the algorithm in Sec. 6.2 is returned as the output.

6.4 Results

In this section, I provide a detailed description of the experiments that were conducted to evaluate the proposed method. In all cases, I assume a weakly supervised setting, where we cosegment only image pairs which belong to the same object class.

6.4.1 Datasets and baselines

Datasets: I evaluate my approach on two challenging publicly available datasets. The first is **MIT Object Discovery** (MIT), a dataset recently introduced for evaluating object foreground discovery through cosegmentation [121].[3] It consists of Internet images of objects from three classes: Airplane, Car, and Horse. The images within a class contain significant appearance and viewpoint variation. I use the 100-image per class subset designated by the authors to enable comparisons with multiple other existing methods. The second dataset is the **Caltech-28**, a subset of 28 of the Caltech-101[4] classes designated by [2] for study in weakly supervised joint segmentation. The 30 images per class originate from Internet search and cover an array of different objects.

Methods compared: I compare to results reported by a number of state-of-the-art cosegmentation techniques, namely [63, 64, 67, 121] on MIT and [2, 21, 70, 119] on Caltech-28. In addition, I implement several baseline techniques:

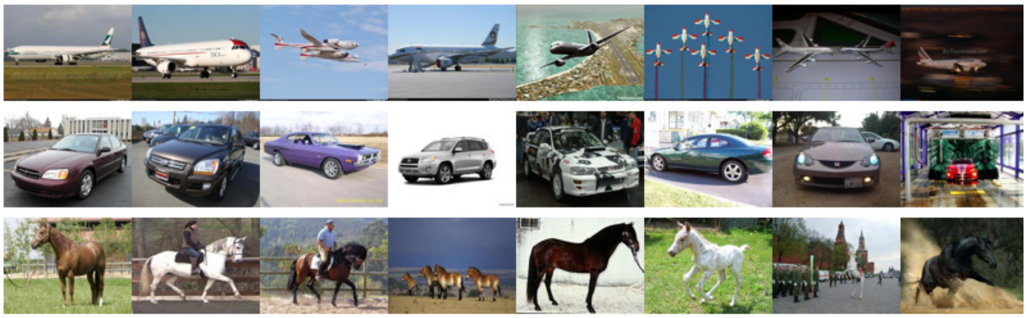
- **Single-Seg:** the saliency-based single-image approach defined in Sec. 6.1. This baseline reveals to what extent a query benefits at all from cosegmentation.
- **Rand-Coseg:** the cosegmentation approach defined in Sec. 6.2 applied with a random image *from the same object category* as the partner source image, averaged over 20 trials. This baseline helps illustrate the need to actively choose a cosegmentation partner among a weakly labeled dataset.

---

[3] `http://people.csail.mit.edu/mrub/ObjectDiscovery/`

[4] `http://www.vision.caltech.edu/ImageDatasets/Caltech101/`

MIT Object Discovery Dataset



Caltech-28 Dataset

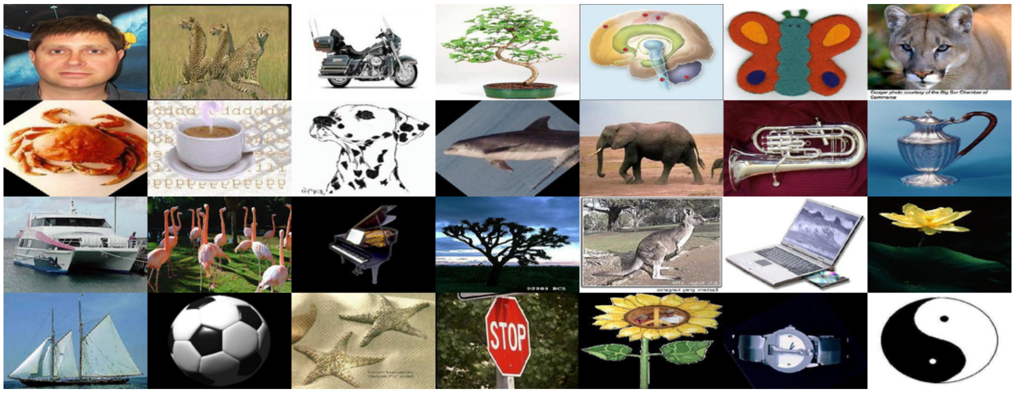


Figure 6.2: Examples from MIT Object Discovery and Caltech-28 datasets. (best viewed in color).

- **GIST-Coseg:** the same cosegmentation approach is applied using the source image that looks most similar to the query, in terms of GIST descriptors. This baseline highlights how image similarity alone—used in existing work [80, 121]—can be insufficient to determine good partners for cosegmentation.

- **Ours-Best k:** I apply my method, but instead of choosing the single maximally ranked image for cosegmentation, I refer to ground truth to pick the best partner from among the $k = 5$ source images my method ranks most highly.

- **Upper bound:** the upper bound for cosegmentation accuracy. I use ground truth to select the partner leading to the maximum overlap score for each query. This reveals the best accuracy any method could possibly attain for the cosegmentation partner selection problem.

All baselines reference the exact same candidate set $\mathcal{P}$ as my method. My method's training set $\mathcal{T}$ is always disjoint from $\mathcal{P}$, and furthermore $\mathcal{P}$ and $\mathcal{T}$ never overlap in object class. For example, when applying my method to Cars in the MIT data, I train it using only images of Airplanes and Horses. To quantify segmentation accuracy, I use the standard intersection-over-union **overlap** accuracy score (Jaccard index), unless otherwise noted.

Implementation details: The color model GMMs consist of 5 mixture components. The scale parameters β are set automatically as the inverse of the mean of all individual distances. I use 50 visual words for the SIFT bag-of-words used in the inter-image foreground similarity, and 11 bins per color channel in all color histograms. The approximate run time per pair is between 10-12 seconds, which is dominated by the SIFT extraction step.

6.4.2 Results on MIT Object Discovery dataset

Table 6.1 shows my results against the baselines on all three classes in the MIT dataset. I observe several things from this result. First, the large

	Single-Seg	Rand-Coseg	GIST-Coseg	Ours	Ours-Best k	Upper bound
Airplane	39.14	42.22	42.34	**45.81**	46.26	57.39
Car	46.76	52.47	50.95	**53.63**	54.31	61.81
Horse	49.82	51.69	**52.73**	50.18	52.86	63.52

Table 6.1: Overlap accuracy on the MIT Object Discovery dataset.

gap between Single-Seg and the Upper bound underscores the fact that cosegmentation can indeed exceed the accuracy of single-image segmentation on challenging images—*if* suitable partners are used. Despite the images' diversity within a single class, the shared appearance in the optimally chosen partner is beneficial. Second, I see that my approach outperforms the baselines in nearly every case. This supports my key claim: it is valuable to actively choose an appropriate cosegmentation partner by learning the cues for success/failure. In two of three classes the method outperforms the GIST-Coseg baseline, showing that off-the-shelf image similarity is inferior to my learning approach for this problem. The Horse class is an exception, where it underperforms than the GIST-Coseg baseline. This is likely due to weak saliency priors in some of the more cluttered Horse images. Third, the fact that the Rand-Coseg approach does as well as it does (in fact, nearly as good as the GIST-Coseg method for Airplanes) indicates that many images of the same class offer *some* degree of help with cosegmentation. Hence, my method's gain is due to its fine-grained analysis of the candidate source images. Finally, the bump in accuracy it achieves if considering the k top-ranked source images (Ours-Best k) indicates that future refinements of my method should consider ways to exploit the ranked partners beyond the top-ranked example.

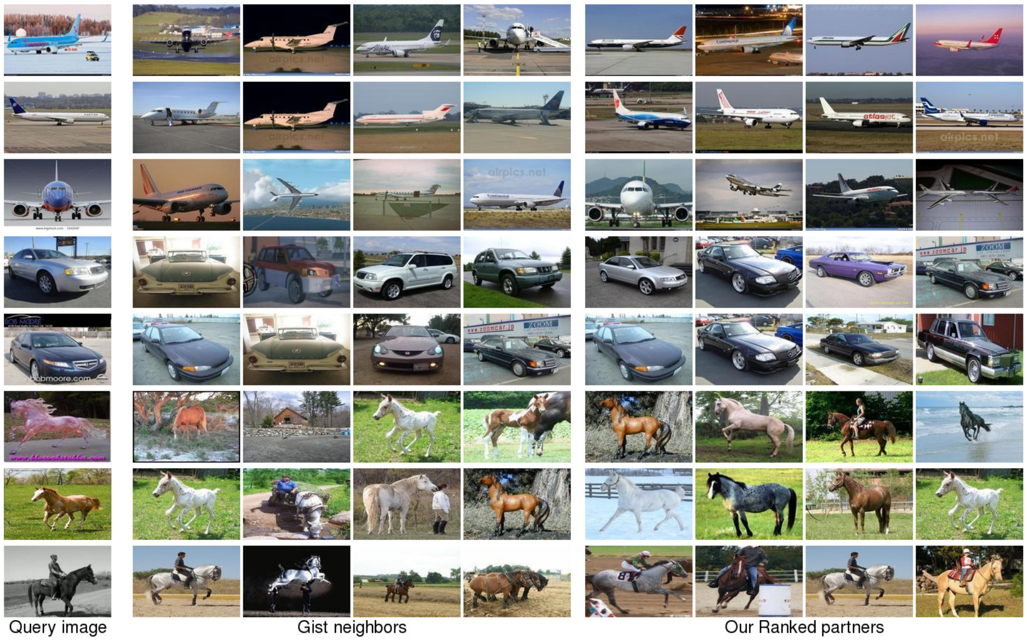


Figure 6.3: Examples of the four top-ranked neighbors for a novel query, using either the GIST nearest neighbors (center block) or my learned ranking function (right block). Best viewed in color. While both methods can identify similar-looking source images among their top-ranked set, my method identifies partners that are more closely aligned in viewpoint or appearance and thus amenable to cosegmentation.

	Joulin et al. [63]	Joulin et al. [64]	Kim et al. [67]	Ours	Rub. et al. [121]
Airplane	15.26	11.72	7.9	45.81	**55.81**
Car	37.15	35.15	0.04	53.63	**64.42**
Horse	30.16	29.53	6.43	50.18	**51.65**

Table 6.2: Comparison to state-of-the-art cosegmentation methods on the MIT Object Discovery dataset, in terms of average overlap.

Figure 6.3 shows examples of the top-ranked partner images produced by the GIST-Coseg baseline and my approach, for a variety of query images in the MIT dataset. My method's learning strategy pays off: it focuses on source images that have more fine-grained compatability with the query image. The GIST neighbors are globally similar, but can be too distinct in viewpoint or appearance to assist in cosegmenting the query. In contrast, the partner source images retrieved by my ranking algorithm are better equipped to share a foreground model due to their viewpoint, appearance, and/or individual saliency.

Table 6.2 compares the result to several state-of-the-art cosegmentation methods.[5] My method outperforms several existing methods by a large margin, except the method of Rubinstein et al. [121] and the joint segmentation propagation algorithm which I proposed in the previous chapter. The disadvantage in this case may be due to the fact that both Rubinstein et al. [121] and active segmentation propagation algorithm operates over a joint graph of all images in the class at once, whereas here only pairs of images are considered for cosegmentation. This suggests a promising future direction to extend my

---

[5] These are the overlap accuracies reported in [121], where the authors applied the public source code to generate results for [63, 64, 67].

		Single-Seg	Rand-Coseg	GIST-Coseg	Ours	Ours-Best k	Upper bound
Best	brain	73.31	72.43	72.54	**75.73**	76.09	76.22
	ferry	54.99	55.87	55.23	**57.64**	57.71	58.02
	dalmatian	39.58	39.13	38.15	**40.23**	40.94	41.59
	ewer	63.87	62.58	63.87	**65.86**	66.18	66.53
	joshua tree	53.04	54.05	54.45	**56.21**	57.12	57.52
	cougar face	58.19	57.39	56.51	**58.25**	58.53	59.05
	sunflower	70.48	70.10	69.77	**71.29**	72.07	73.48
	motorbike	**57.38**	55.86	55.79	57.21	58.12	58.59
	euphonium	57.72	57.25	58.32	**59.45**	60.27	60.28
	kangaroo	59.79	59.26	59.13	**60.24**	60.57	61.81
Worst	lotus	76.71	75.98	**78.38**	77.59	79.51	80.16
	grand piano	67.21	67.28	**67.93**	66.58	67.01	68.33
	crab	61.86	62.25	**62.11**	61.23	62.3	62.46
	watch	55.00	56.4	**57.72**	56.11	56.16	58.30

Table 6.3: Accuracy on the Caltech-28 dataset, in terms of average overlap. I show the 10 best and 4 worst performing classes.

algorithm, e.g., by using my compatibility predictions as weights within the complete joint segmentation graph from the previous chapter.

6.4.3 Results on Caltech-28 dataset

Table 6.3 shows the results for the Caltech-28 dataset, in the same format as Table 6.1 above. I show a representative set of the top 10 cases where the method most outperforms GIST-Coseg and the bottom four cases where the method most underperforms GIST-Coseg.

The analysis is fairly similar to my MIT dataset results. There is good support for actively selecting a cosegmentation partner: my method outperforms the Rand-Coseg and GIST-Coseg baselines in most cases. Overall, the proposed method outperforms GIST-Coseg in 23 of the 28 classes, and Single-Seg in 20 of the 28 classes. My method is also quite close to the Upper bound on this dataset, only 1.5 points away on average.

Method	Average Precision
Spatial Topic Model-Coseg [21]	67
Single-Seg	82.71
GrabCut-Coseg (see [2])	81.5
ClassCut-Coseg [2]	83.6
BPLR-Coseg [70]	85.6
Ours	**85.81**

Table 6.4: Comparison to state-of-the-art cosegmentation algorithms on the Caltech-28 dataset.

However, for the Caltech data, the gap between Single-Seg and the Upper bound—while still noticeably wider than the gap between my method and the Upper bound—is also narrowed considerably compared to the MIT data. This indicates that the Caltech images have greater regularity within a class and/or more salient foregrounds (both of which are true upon visual inspection). In fact, Single-Seg can even outperform the cosegmentation methods in some cases (e.g., see motorbike). This finding agrees with previous reports in [121, 140]; while one hopes to see gains from the "more supervised" cosegmentation task, single-image segmentation can be competitive either when the intra-class variation is too high or the foreground is particularly salient.

Finally, I compare my method to state-of-the-art cosegmentation methods using their published numbers on the Caltech-28. Table 6.4 shows the results, in terms of average precision (the metric reported in the prior work). My method is more accurate than all the previous results. Notably, all the prior cosegmentation results ([2, 21, 70] and the multi-image GrabCut [119] extension defined in [2]) indiscriminately use all the input images for joint segmentation, whereas my method selects the single most effective partner per query. This result is more evidence for the advantage of doing so.

6.5 Conclusion

Cosegmentation injects valuable implicit top-down information for segmentation, based on commonalities between related input images. Rather than assume that useful partners for cosegmentation will be known in advance, I proposed an algorithm to predict which pairs will work well together. My results on two challenging datasets are encouraging evidence that it is worthwhile to actively focus cosegmentation on relevant pairs.

While so far this study was limited to only studying this problem in the context of image pairs, I believe that measuring compatibility between image pairs for mutual segmentation transfer has much wider applicability. Extending the algorithm from pairs to the weakly-suervised multi-image joint segmentation scenario and also possibly to the fully unsupervised setting is a very promising future direction.

Having discussed my proposed methods for segmenting images in various settings, in the next chapter I will describe my approach for semi-supervised segmentation propagation in videos.

Chapter 7

Supervoxel consistent foreground propagation in video

[1] Whereas the previous chapters deal largely with segmenting images interactively, the remainder of the dissertation looks closely at segmenting objects from *video*. Different from the algorithms for segmenting images, a video segmentation algorithm can directly benefit from the temporal continuity in video data. While segmentation propagation in an image collection had to rely on similarity scores between images which are inherently noisy, the temporal prior in video allows for direct constraints on how the propagation should proceed (e.g., through connections in time).

In this chapter, I introduce a semi-supervised approach for video segmentation propagation using supervoxel higher order potentials. The proposed semi-supervised video segmentation propagation algorithm takes a video clip as input and some labeled frames in which an annotator has outlined the foreground object of interest. The output is a space-time segmentation with foreground (fg) or background (bg) labels to every pixel in every frame. This

---

[1] The work in this chapter was supervised by Dr. Kristen Grauman and originally published [57] in: Supervoxel-Consistent Foreground Propagation in Video. S. Jain and K. Grauman. In Proceedings of the European Conference on Computer Vision (ECCV), 2014, Zurich, Switzerland.

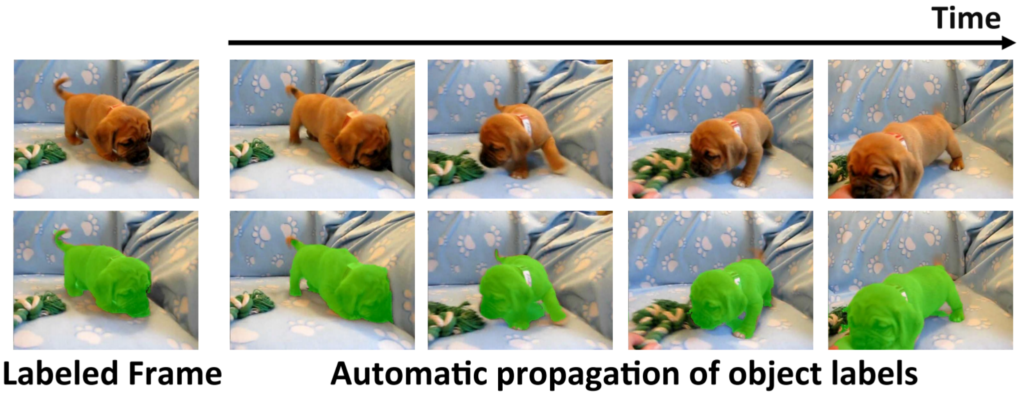


Figure 7.1: Automatic propagation of foreground segmentation in videos from a single/multiple labeled frame(s). Here we see human drawn segmentation on a single frame being propagated to all the other frames in the video using my supervoxel based propagation (Chapter 7) algorithm [57]. Best viewed in color.

is done by defining a space-time graph and energy function that respect the "big picture" of how objects move and evolve throughout the clip (see Figure 7.1).

The propagation paradigm has several advantages. First, it removes ambiguity about what object is of interest, which, despite impressive advances [81, 84, 96, 160], remains an inherent pitfall for purely unsupervised methods for video segmentation [26, 40, 43, 81, 84, 96, 157, 158, 160]. Accordingly, the propagation setting can accommodate a broader class of videos, e.g., those in which the object does not move much, or shares appearance with the background. Second, propagation from just few human-labeled frames can be substantially less burdensome than human-in-the-loop systems that require constant user interaction [7, 36, 87, 116, 145, 149], making it a promising tool

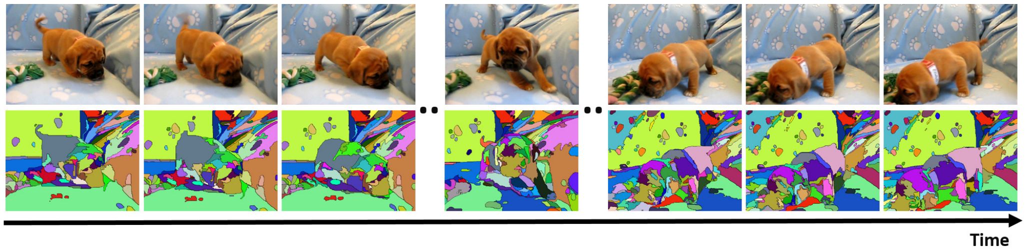


Figure 7.2: Example supervoxels, using [43]. Unique colors are unique supervoxels, and repeated colors in adjacent frames refer to the same supervoxel. Notice that a number of larger supervoxels remain steady in early frames, then some split/merge as the dog's pose changes, then a revised set again stabilizes for the latter chunk of frames. Best viewed in color.

for gathering object tubes at a large scale.

Key to my idea is the use of *supervoxels* (see Figure 7.2). Supervoxels are space-time regions computed with a bottom-up unsupervised video segmentation algorithm [43, 157, 158]. They typically oversegment—meaning that objects may be parcelled into many supervoxels—but the object boundaries remain visible among the supervoxel boundaries. They vary in shape and size, and will typically be larger and longer for content more uniform in its color or motion. Though a given object part's supervoxel is unlikely to remain stable through the entire video, it will often persist for a series of frames. The proposed approach exploits this partial stability of the supervoxels but also guards against their noisy imperfections. As discussed in Chapter 2, existing methods for segmentation propagation [6, 36, 118, 135, 141] only account for short range interactions through noisy optical flow based connections between adjacent frames. In contrast, the proposed supervoxel based method is able to enforce long range temporal consistency and is more robust to flow errors.

In the proposed propagation method supervoxels are leveraged in two ways. First, each supervoxel is projected into each of its child frames to obtain spatial superpixel nodes. These nodes have sufficient spatial extent to compute rich visual features. Plus, compared to standard superpixel nodes computed independently per frame [6, 26, 36, 40, 116, 118, 135], they benefit from the broader perspective provided by the hierarchical space-time segment that generates the supervoxels. For example, optical flow similarity of voxels on the dog's textured collar (Figure 7.2) may preserve it as one node, whereas per-frame segments may break it into many. Secondly, supervoxels are leveraged as a higher-order potential. Augmenting the usual unary and pairwise terms, a soft label consistency constraint is enforced among nodes originating from the same supervoxel. Again, this provides broader context to the propagation engine.

The proposed approach is validated on three challenging datasets, SegTrack [135], YouTube Objects [115], and Weizmann [42], and compared to state-of-the-art propagation methods. It outperforms existing techniques overall, with particular advantage when foreground and background look similar, inter-frame motion is high, or the target changes shape between frames.

As stated earlier, the proposed supervoxel based propagation technique also effectively combines with my *Click Carving* based interactive segmentation algorithm (Chapter 4). Instead of manually annotating the initial frame, we can use *Click Carving* to segment that frame interactively, which can then be propagated using the supervoxel propagation method. This results in a

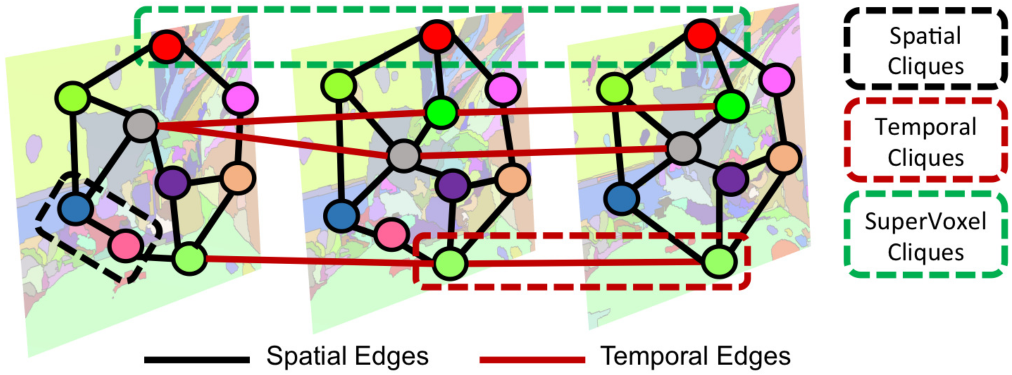


Figure 7.3: Proposed spatio-temporal graph. Nodes are superpixels (projected from supervoxels) in every frame. Spatial edges exist if the superpixels have boundary overlap (black); temporal edges are computed using optical flow (red). Higher order cliques are defined by supervoxel membership (dotted green). For legibility, only a small subset of nodes and connections are depicted. Best viewed in color.

substantial savings in human annotation cost for video segmentation.

In the following, I describe the three main stages of our approach: 1) a spatio-temporal graph is constructed from the video sequence using optical flow and supervoxel segmentation (Section 7.1); 2) a Markov Random Field is defined over this graph with suitable unary potentials, pairwise potentials, and higher order potentials (Section 7.2); and 3) the energy of this MRF is minimized by iteratively updating the likelihood functions using label estimates (Section 7.3). Section 7.4 then presents detailed experimental results and comparisons with other state-of-the-art methods.

7.1 Space-time MRF graph structure

I first formally define the proposed spatio-temporal Markov Random Field (MRF) graph structure G consisting of nodes $\mathcal{X}$ and edges $\mathcal{E}$. Let $\mathcal{X} = \{X_t\}_{t=1}^{T}$ be the set of superpixels[2] over the entire video volume, where T refers to the number of frames in the video. X_t is a subset of $\mathcal{X}$ and contains superpixels belonging only to the t-th frame. Therefore each X_t is a collection of superpixel nodes $\{x_t^i\}_{i=1}^{K_t}$, where K_t is the number of superpixels in the t-th frame.

A random variable $y_t^i \in \{+1, -1\}$ is associated with every node to represent the label it may take, which can be either object (+1) or background (-1). My goal is to obtain a labeling $\mathcal{Y} = \{Y_t\}_{t=1}^{T}$ over the entire video. Here, $Y_t = \{y_t^i\}_{i=1}^{K_t}$ represents the labels of superpixels belonging only to the t-th frame. Below, (t, i) indexes a superpixel node at position i and time t.

An edge set $\mathcal{E} = \{\mathcal{E}_s, \mathcal{E}_t\}$ is defined for the video. $\mathcal{E}_s$ is the set of spatial edges between superpixel nodes. A spatial edge exists between a pair of superpixel nodes (x_t^i, x_t^j) in a given frame if their boundaries overlap (black lines in Figure 7.3). $\mathcal{E}_t$ is the set of temporal edges. A temporal edge exists between a pair of superpixels (x_t^i, x_{t+1}^j) in adjacent frames if any pixel from x_t^i tracks into x_{t+1}^j using optical flow (red lines in Figure 7.3). I use the algorithm of [19] to compute dense flow between consecutive frames. Let $[(t, i), (t', j)]$ index an edge between two nodes. For spatial edges, $t' = t$; for temporal edges,

---

[2] Throughout, I use “superpixel” to refer to a supervoxel projection into the frame.

$t' = t + 1$.

Finally $\mathcal{S}$ is used to denote the set of supervoxels. Each element $v \in \mathcal{S}$ represents a higher order clique (one is shown with a green dashed box in Figure 7.3) over all the superpixel nodes which are a part of that supervoxel. Let y_v denote the set of labels assigned to the superpixel nodes belonging to the supervoxel v.

For each superpixel node x_t^i, I compute two image features using all its pixels: 1) an RGB color histogram with 33 bins (11 bins per channel), and 2) a histogram of optical flow, which bins the flow orientations into 9 uniform bins. The two descriptors are concatenated and the visual dissimilarity between two superpixels $\mathcal{D}(x_t^i, x_{t'}^j)$ is computed as the Euclidean distance in this feature space.

7.2 Energy function with supervoxel label consistency

Having defined the graph structure, I can now explain the proposed segmentation pipeline. I define an energy function over $G = (\mathcal{X}, \mathcal{E})$ that enforces long range temporal coherence through higher order potentials derived from supervoxels $\mathcal{S}$:

$$E(\mathcal{Y}) = \underbrace{\sum_{(t,i)\in\mathcal{X}} \Phi_t^i(y_t^i)}_{\textit{Unary potential}} + \underbrace{\sum_{\substack{[(t,i),(t',j)]\in\mathcal{E}\\ t'\in\{t,t+1\}}} \Phi_{t,t'}^{i,j}\big(y_t^i, y_{t'}^j\big)}_{\textit{Pairwise potential}} + \underbrace{\sum_{v\in\mathcal{S}} \Phi_v\big(y_v\big)}_{\textit{Higher order potential}} \quad . \tag{7.1}$$

The goal is to obtain the video's optimal object segmentation by minimizing Eqn. 7.1: $\mathcal{Y}^* = \arg\min_{\mathcal{Y}} E(\mathcal{Y})$. The unary potential accounts for the cost of assigning each node the object or background label, as determined by appearance models and spatial priors learned from the labeled frame. The pairwise potential promotes smooth segmentations by penalizing neighboring nodes taking different labels. The higher order potential, key to my approach, ensures long term consistency in the segmentation. It can offset the errors introduced by weak or incorrect temporal connections in the adjacent frames.

Next I give the details for each of the potential functions.

7.2.1 Unary potential

The unary potential in Eqn. 7.1 has two components, an appearance model and a spatial prior:

$$\Phi_t^i(y_t^i) = \underbrace{\lambda_{app} \, A_t^i(y_t^i)}_{Appearance\ prior} \quad + \underbrace{\lambda_{loc} \, L_t^i(y_t^i)}_{Spatial\ prior}, \tag{7.2}$$

where λ_{app} and λ_{loc} are scalar weights reflecting the two components' influence.

To obtain the appearance prior $A_t^i(y_t^i)$, the human-labeled frame is used to learn Gaussian mixture models (GMM) to distinguish object vs. background. Specifically, all the pixels inside and outside the supplied object mask are used to construct the foreground G_{+1} and background G_{-1} GMM distributions, respectively, based on RGB values. To compute the likelihood that a superpixel x_t^i is object or background, the mean likelihood over all pixels within the superpixel is used:

$$A_t^i(y_t^i) = -\log \frac{1}{|x_t^i|} \sum_{p \in x_t^i} P(F_p | G_{y_t^i}), \tag{7.3}$$

where F_p is the RGB color value for pixel p and $|x_t^i|$ is the pixel count within the superpixel node x_t^i.

The spatial prior $L_t^i(y_t^i)$ penalizes label assignments that deviate from an approximate expected spatial location for the object:

$$L_t^i(y_t^i) = -\log P(y_t^i|(t,i)), \tag{7.4}$$

where (t,i) denotes the location of a superpixel node. To compute this prior, we start with the human-labeled object mask in the first frame and propagate that region to subsequent frames using both optical flow and supervoxels.[3] In particular, we define:

$$P(y_{t+1}^k|(t+1,k)) = \sum_{(i,t)\in\mathcal{B}_k} \psi\left(x_{t+1}^k, x_t^i\right)\ \delta\left(P(y_t^i|(t,i)) > \tau\right), \tag{7.5}$$

where $\mathcal{B}_k$ is the set of superpixel nodes tracked backwards from x_{t+1}^k using optical flow, and δ denotes the delta function. The δ term ensures that transfer happens only from the most confident superpixels, as determined in the prior frame of propagation. In particular, the contribution of any x_t^i with confidence lower than $\tau = 0.5$ is ignored.

The term $\psi(x_{t+1}^k, x_t^i)$ in Eqn. 7.5 estimates the likelihood of a successful label transfer from frame t to frame $t+1$ at the site x^k. If, via the flow, we find the transfer takes place between superpixels belonging to the same supervoxels, then the transfer is predicted to succeed to the extent the corresponding

---

[3] If a frame other than the first is chosen for labeling, the system propagates from that frame out in both directions. See Sec. 7.4.4 for extension handling multiple labeled frames.

superpixels overlap in pixel area, $\rho = \frac{|x_t^i|}{|x_{t+1}^k|}$. Otherwise, that overlap is further scaled by the superpixels' feature distance:

$$\psi(x_{t+1}^k, x_t^i) = \begin{cases} \rho & \text{if } (x_{t+1}^k, x_t^i) \in v \text{ (same supervoxel)} \\ \rho \exp\left(-\beta_u \mathcal{D}(x_{t+1}^k, x_t^i)\right) & \text{otherwise,} \end{cases}$$

where β_u is a scaling constant for visual dissimilarity.

7.2.2 Pairwise potential

In order to ensure that the output segmentation is smooth in both space and time, standard pairwise terms for both spatial and temporal edges are used:

$$\Phi_{t,t'}^{i,j}(y_t^i, y_{t'}^j) = \delta(y_t^i \neq y_{t'}^j) \exp\left(-\beta_p \mathcal{D}(x_t^i, x_{t'}^j)\right), \tag{7.6}$$

where β_p is a scaling parameter for visual dissimilarity. The penalty for adjacent nodes having different labels is contrast-sensitive, meaning that they are modulated by the visual feature distance $\mathcal{D}(x_t^i, x_{t'}^j)$ between the neighboring nodes. For temporal edges, this potential is further weighted by ρ, the pixel overlap between the two nodes computed above with optical flow. Both types of edges encourage output segmentations that are consistent between nearby frames.

7.2.3 Higher order potential

Finally, I define the supervoxel label consistency potential, which is crucial to my method. While the temporal smoothness potential helps enforce segmentation coherence in time, it suffers from certain limitations. Temporal edges are largely based on optical flow, hence they can only connect nodes in adjacent frames. This inhibits long-term coherence in the segmentation. In addition, the edges themselves can be noisy due to errors in flow.

Therefore, I propose to use higher order potentials derived from the supervoxel structure. As discussed above, the supervoxels group spatio-temporal regions which are similar in color and flow. Using the method of [43], this grouping is a result of long-term analysis of regions, and thus can overcome some of the errors introduced from optical flow tracking. For instance, in the datasets I use below, supervoxels can be up to 400 frames long and occupy up to 70% of the frame. At the same time, the supervoxels themselves are not perfect—otherwise the system would be done! Thus, I use them to define a soft preference for label consistency among superpixel nodes within the same supervoxel.

The Robust P^n model [71] is adopted to define these potentials. It consists of a higher order potential defined over supervoxel cliques:

$$\Phi_v(y_v) = \begin{cases} N(y_v)\frac{1}{Q}\gamma_{\max}(v) & \text{if } N(y_v) \leq Q \\ \gamma_{\max}(v) & \text{otherwise,} \end{cases} \tag{7.7}$$

where y_v denotes the labels of all the superpixel nodes within the supervoxel $v \in \mathcal{S}$, and $N(y_v)$ is the number of nodes within the supervoxel v that do not take the dominant label. That is, $N(y_v) = \min(|y_v = -1|, |y_v = +1|)$. Following [71], Q is a truncation parameter that controls how rigidly we want to enforce the consistency within the supervoxels. Intuitively, the more confident we are that the supervoxels are strictly an oversegmentation, the higher Q

should be.

The penalty $\gamma_{\max}(v)$ is a function of the supervoxel's size and color diversity, reflecting that those supervoxels that are inherently less uniform should incur lesser penalty for label inconsistencies. Specifically, $\gamma_{\max}(v) = |y_v| \exp(-\beta_h \sigma_v)$, where σ_v is the total RGB variance in supervoxel v.

7.3 Energy minimization and parameters

The energy function defined in Eqn. 7.1 can be efficiently minimized using the α-expansion algorithm [71]. The optimal labeling corresponding to the minimum energy yields my initial fg-bg estimate. That output is iterative refined by re-estimating the appearance model—using only the most confident samples based on the current unary potentials—then solving the energy function again. The method iterates three times to obtain the final output.

The only three parameters that must be set are λ_{app} and λ_{loc}, the weights in the appearance potential, and the truncation parameter Q. I determined reasonable values ($\lambda_{app} = 100$, $\lambda_{loc} = 40, Q = 0.2\ |y_v|$) by visual inspection of a couple outputs, then fixed them for all videos and datasets. (This is minimal effort for a user of the system. It could also be done with cross-validation, when sufficient pixel-level ground truth is available for training.) The remaining parameters β_u, β_p, and β_h, which scale the visual dissimilarity for the unary, pairwise, and higher order potentials, respectively, are all set automatically as the inverse of the mean of all individual distance terms.

7.4 Results

In this section, I provide detailed experiments and comparisons with state-of-the-art methods.

7.4.1 Datasets and baselines

Datasets and metrics: I evaluate on three publicly available video segmentation datasets: SegTrack [135], YouTube-Objects [115], and Weizmann [42]. Figure 7.4 shows some visual examples from each dataset. For SegTrack and YouTube, the true object region in the first frame is supplied to all methods. I use standard evaluation metrics: average pixel label error and intersection-over-union overlap.

Methods compared: I compare to five state-of-the-art methods: four for semi-supervised foreground label propagation [27, 36, 135, 141], plus the state-of-the-art higher order potential method of [26]. Note that unsupervised multiple-hypothesis methods [81, 84, 96, 160] are not comparable in this semi-supervised single-hypothesis setting. I also test the following baselines:

- **SVX-MRF:** an MRF comprised of supervoxel nodes. The unary potentials are initialized through the labeled frame, and the smoothness terms are defined using spatio-temporal adjacency between supervoxels. It highlights the importance of the design choices in the proposed graph structure.

- **SVX-Prop:** a simple propagation scheme using supervoxels. Starting

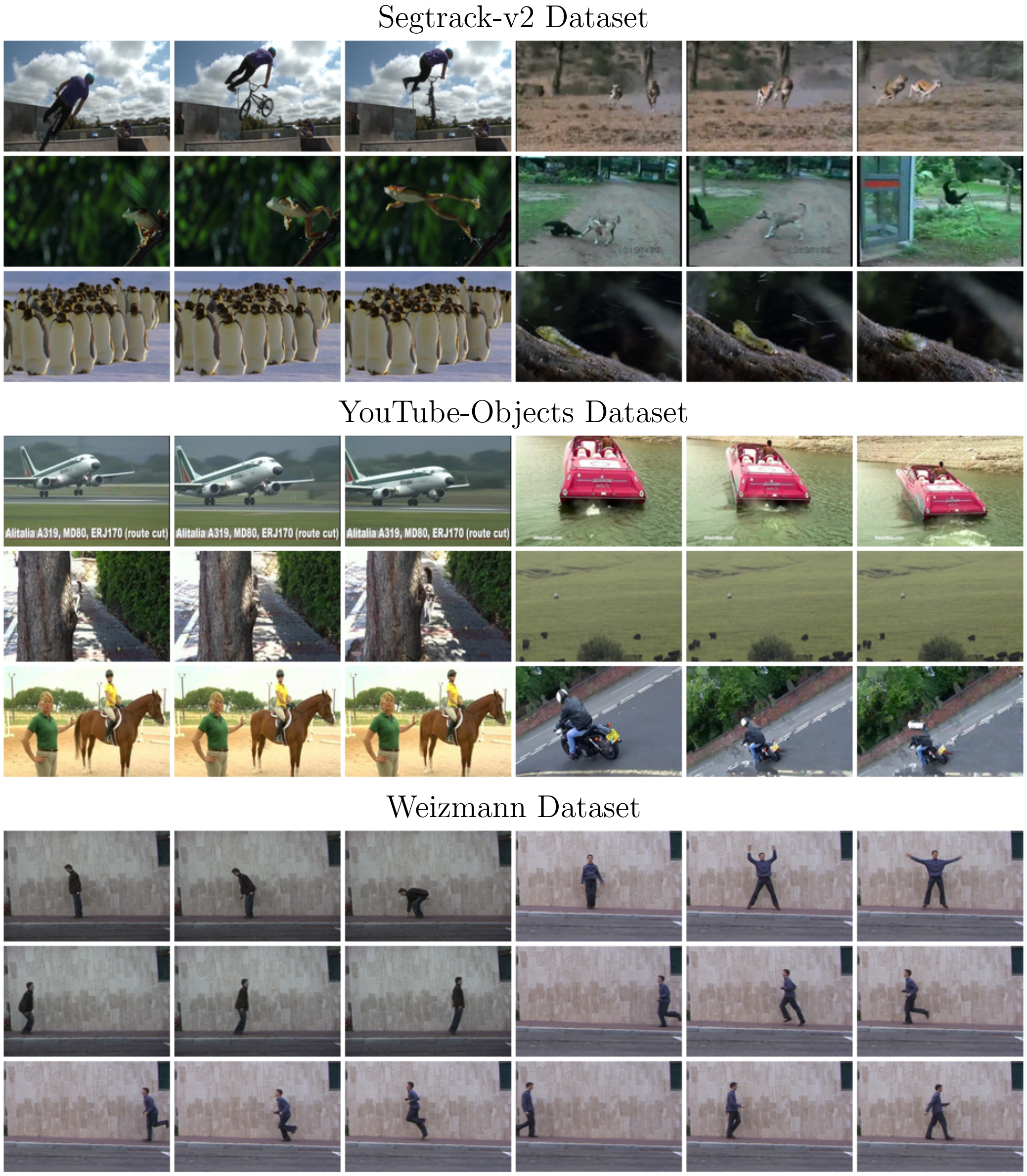

Segtrack-v2 Dataset
YouTube-Objects Dataset
Alitalia A319, MD80, ERJ170 (route cut)
Alitalia A319, MD80, ERJ170 (route cut)
Alitalia A319, MD80, ERJ170 (route cut)
Weizmann Dataset

Figure 7.4: Example video sequences from Segtrack-v2, YouTube-Objects and Weizmann datasets. (best viewed in color).

from the labeled frame, the propagation of foreground labels progresses through temporally linked (using optical flow) supervoxels. It illustrates that it's non-trivial to directly extract foreground from supervoxels.

- **PF-MRF:** the existing algorithm of [141], which uses a pixel-flow (PF) MRF for propagation. Note that the authors also propose a method to actively select frames for labeling, which I do not employ here.
- **Ours w/o HOP:** a simplified version of my method that lacks higher order potentials (Eqn. 7.7), to isolate the impact of supervoxel label consistency.

7.4.2 Results on SegTrack dataset

SegTrack [135] was designed to evaluate object segmentation in videos. It consists of six videos, 21-71 frames each, with various challenges like color overlap in objects, large inter-frame motion, and shape changes. Pixel-level ground truth is provided, and the standard metric is the average number of mislabeled pixels over all frames, per video. The creators also provide difficulty ratings per video with respect to appearance, shape, and motion.

Table 7.1 shows the results, compared to all existing propagation results in the literature. The proposed method outperforms the state-of-the-art in 4 of the 6 videos. Especially notable are the substantial gains on the challenging "monkeydog" and "birdfall" sequences. Figure 7.5 (top row) shows examples from "monkeydog" (challenging w.r.t shape & motion [135]). My method

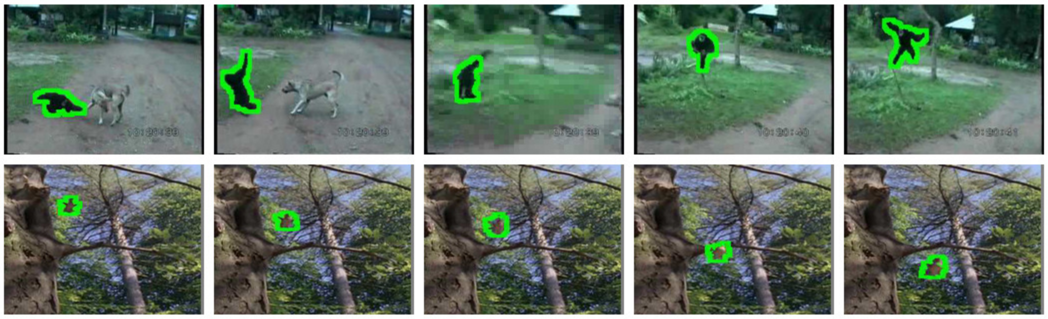


Figure 7.5: Example results on SegTrack. Best viewed in color.

	Ours	PF-MRF [141]	Fathi[36]	Tsai[135]	Chockalingam[27]
birdfall	**189**	405	342	252	454
cheetah	1170	1288	**711**	1142	1217
girl	2883	8575	**1206**	1304	1755
monkeydog	**333**	1225	598	563	683
parachute	**228**	1042	251	235	502
penguin	**443**	482	1367	1705	6627

Table 7.1: Average pixel errors for all existing propagation methods on SegTrack.

successfully propagates the foreground, despite considerable motion and deformation. Figure 7.5 (bottom row) is from “birdfall” (challenging w.r.t motion & appearance [135]). My method propagates the foreground well in spite of significant fg/bg appearance overlap.

The weaker performance on “cheetah” and “girl” is due to undersegmentation in the supervoxels, which hurts the quality of my supervoxel cliques and the projected superpixels. In particular, “cheetah” is low resolution and foreground/background appearance strongly overlap, making it more difficult for [43] (or any supervoxel algorithm) to oversegment. This suggests a hierarchical approach that considers fine to coarse supervoxels could be beneficial,

	Ours	Ours w/o HOP	SVX-MRF	SVX-Prop
birdfall	**189**	246	299	453
cheetah	**1170**	1287	1202	1832
girl	**2883**	3286	3950	5402
monkeydog	**333**	389	737	1283
parachute	**228**	258	420	1480
penguin	**443**	497	491	541

Table 7.2: Average pixel errors (lower is better) for other baselines on SegTrack.

which I leave as future work.

PF-MRF [141], which propagates based on flow links, suffers in several videos due to errors and drift in optical flow. This highlights the advantages of the broader scale nodes formed from supervoxels: the supervoxel based graph is not only more efficient (it requires 2-3 minutes per video, while PF-MRF requires 8-10 minutes), but it also is robust to flow errors. The prior superpixel graph methods [36, 135] use larger nodes, but only consider temporal links between adjacent frames. Thus, the gains here confirm that long-range label consistency constraints are important for successful propagation.

Table 7.2 compares my method to the other baselines on SegTrack. SVX-Prop performs poorly, showing that tracking supervoxels alone is insufficient. SVX-MRF performs better but still is much worse than my method, which shows that it's best to enforce supervoxel constraints in a soft manner. The higher order potentials (HOP) help my method in all cases (compare cols 1 and 2 in Table 7.2). To do a deeper analysis of the impact of HOPs, I consider the sequences rated as difficult in terms of motion and shape by [135], "monkeydog" and "birdfall". On their top 10% most difficult frames, the relative gain of HOPs is substantially higher. On “birdfall” HOPs yield a 40% gain on the most difficult frames (as opposed to 23% over all frames). On “monkeydog” the gain is 18% (compared to 13% on all frames).

7.4.3 Results on YouTube-Objects dataset

Next I evaluate on the YouTube-Objects [115]. I use the subset defined by [132], who provide segmentation ground truth. However, that ground truth is approximate—and even biased in our favor—since annotators marked supervoxels computed with [43], not individual pixels. Hence, I collected fine-grained pixel-level masks of the foreground object in every 10-th frame for each video using Amazon Mechanical Turk[4]. In all, this yields 126 web videos with 10 object classes and more than 20,000 frames. To my knowledge, these experiments are the first time such a large-scale evaluation is being done for the task of foreground label propagation; prior work has limited its validation to the smaller SegTrack.

Table 7.3 shows the results in terms of overlap accuracy. My method outperforms all the baselines in 8 out of 10 classes, with gains up to 8 points over the best competing baseline. Note that each row corresponds to multiple videos for the named class; my method is best on average for over 100 sequences.

On YouTube, PF-MRF [141] again suffers from optical flow errors,

---

[4] Available at: `http://vision.cs.utexas.edu/projects/videoseg/`

obj (#vid)	Ours	Ours w/o HOP	SVX-MRF	SVX-Prop	PF-MRF [141]
aeroplne (6)	**86.27**	79.86	77.36	51.43	84.9
bird (6)	**81.04**	78.43	70.29	55.23	76.3
boat (15)	**68.59**	60.12	52.26	48.70	62.44
car (7)	**69.36**	64.42	65.82	50.53	61.35
cat (16)	**58.89**	50.36	52.9	36.25	52.61
cow (20)	**68.56**	65.65	64.66	51.43	58.97
dog (27)	**61.78**	54.17	53.57	39.10	57.22
horse (14)	**53.96**	50.76	47.91	28.92	43.85
mbike (10)	60.87	58.31	45.23	42.23	**62.6**
train (5)	66.33	62.43	47.26	55.33	**72.32**

Table 7.3: Average accuracy per class on YouTube-Objects (higher is better). Numbers in parens denote the number of videos for that class.

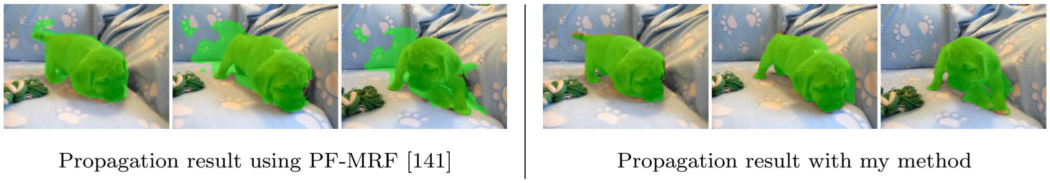

Propagation result using PF-MRF [141]
Propagation result with my method

Figure 7.6: The supervoxel based propagation method resolves dragging errors common in flow-based MRFs.

which introduce a “dragging effect”. For example, Figure 7.6 shows the PF-MRF pixel flow drags as the dog moves on the sofa (left), accumulating errors. In contrast, my method propagates the foreground and background more cleanly (right). The SVX-MRF baseline is on average 10 points worse than ours, and only 25 seconds faster.

Comparing the first two columns in Table 7.3, we see that supervoxel HOPs have the most impact on “boat”, “dog”, and “cat” videos. They tend to have substantial camera and object motion. Thus, often, the temporal links based on optical flow are unreliable. In contrast, the supervoxels, which depend on not only motion but also object appearance, are more robust. For

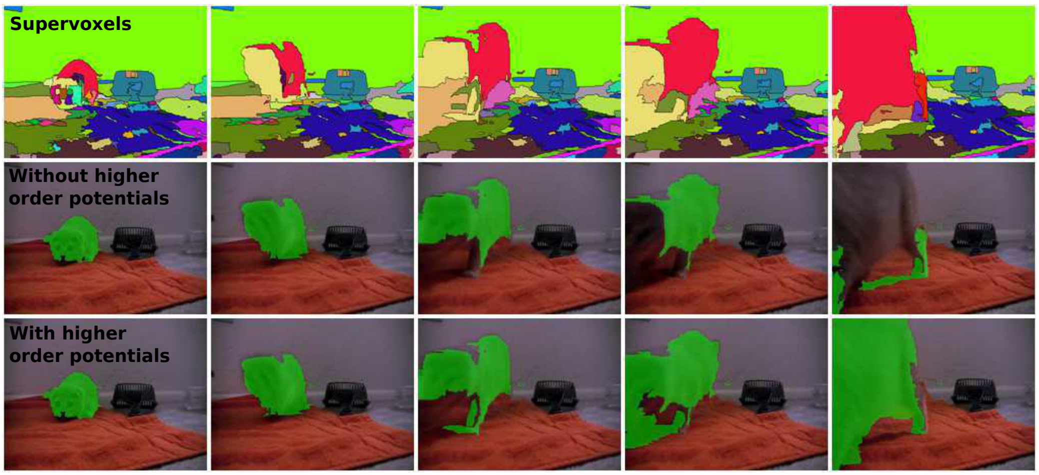
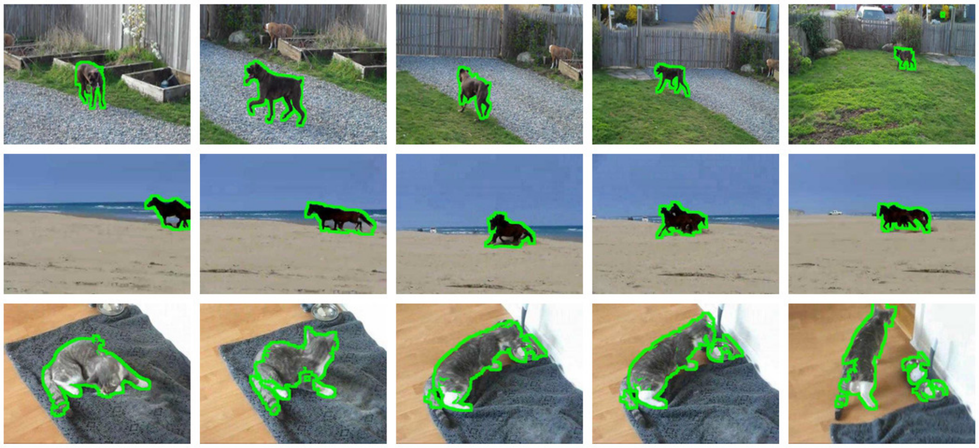


Figure 7.7: Label propagation with and without HOPs (frames 31, 39, 42, 43, 51).

example, Figure 7.7 shows a challenging case where the cat suddenly jumps forward. Without the HOP, optical flow connections alone are insufficient to track the object (middle row). However, the supervoxels are still persistent (top row), and so the HOP propagates the object properly (bottom row).

Figure 7.8 shows more qualitative results. My method performs well even in the cases where there is significant object or camera motions. The cat (third row) also shows its robustness to foreground-background appearance overlap. In the failure case (last row), it initially tracks the cat well, but later incorrectly merges the foreground and ladder due to supervoxel undersegmentations.

Success cases



Failure case

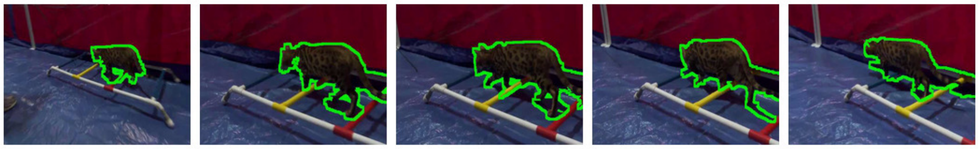


Figure 7.8: Qualitative results highlighting the performance under fast motion, shape changes, and complex appearance. The first image in each row shows the human-labeled first frame of the video. See text for details.

7.4.4 Results on Weizmann dataset

Lastly, I use the Weizmann dataset [42] to compare to [26], which uses higher order spatial cliques and short temporal cliques found with flow. The dataset consists of 90 videos, from 10 activities with 9 actors each.

Figure 7.9 shows the results in terms of foreground precision and recall, following [26]. Whereas my method outputs a single fg-bg estimate (2 segments), the method of [26] outputs an oversegmentation with about 25 segments per video. Thus, the authors use the ground truth on each frame

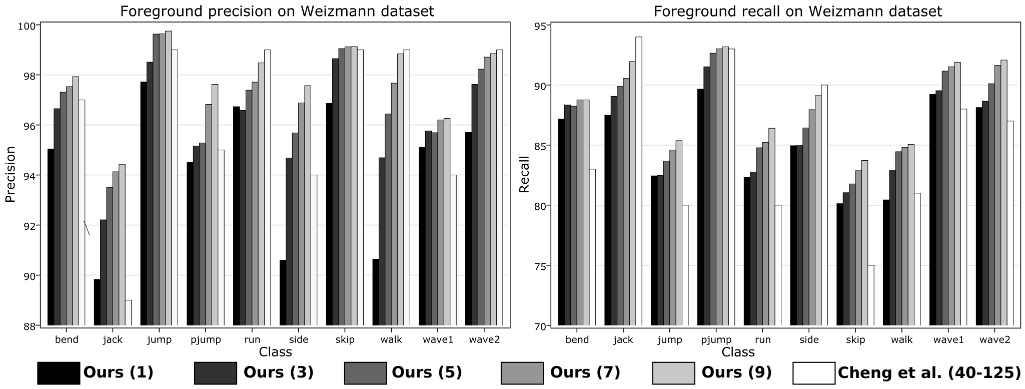


Figure 7.9: Foreground precision (left) and recall (right) on Weizmann. Legend shows number of labeled frames used per result (1 to 9 for my method, 40-125 for [26]).

to map their outputs to fg and bg labels, based on majority overlap; this is equivalent to obtaining on the order of 25 manual clicks per frame to label the output. In contrast, my propagation method uses just 1 labeled frame to generate a complete fg-bg segmentation. Therefore, I show the results for increasing numbers of labeled frames, spread uniformly through the sequence. This requires a multi-frame extension of my method—namely, it takes the appearance model G_{y_t} from the labeled frame nearest to t, and re-initialize the spatial prior $L_t^i(y_t^i)$ at every labeled frame.

With just 5 labeled frames (compared to the 40-125 labeled frames used in [26]), the results are better in nearly all cases. Even with a single labeled frame, the performance is competitive. This result gives strong support for the proposed formulation of a long-range HOP via supervoxels. Essentially, the method of [26] achieves a good oversegmentation, whereas my method achieves accurate object tubes with long range persistence.

7.5 Conclusion

In conclusion, this chapter introduced a new semi-supervised approach to propagate object regions in a video. The proposed method is capable of enforcing long-term temporal consistencies in the output segmentation using a supervoxel higher order potential. Extensive results on the SegTrack, YouTube-Objects and Weizmann datasets show that the proposed approach outperforms many state-of-the-art methods and several important baselines while propagating from a single/few labeled frames.

In the future, there can be several possible extensions to address some weaknesses of the current propagation engine. Firstly, in its current form the method uses supervoxels from a particular level in the hierarchical segmentation of the video. The current choice of this parameter is heuristic in nature i.e., it selects an intermediate level of the hierarchy. However, using supervoxels from a fixed level for all videos is sub-optimal. These supervoxels can be too fine or too coarse at that level depending on the individual video content or quality. It will be useful to consider a coarse-to-fine approach which can define higher order potentials to integrate information from the entire hierarchy. Moreover, the performance also suffers due to the over-segmentation errors which the supervoxels introduce. A coarse-to-fine approach can potentially remedy that as well.

Secondly, in its current form the propagation engine requires the complete video to be available for the propagation to take place. This can potentially be problematic for longer videos where propagating information across very large temporal intervals could be difficult. A straightforward extension, which propagates information in a streaming fashion, i.e., by processing only a subset of frames at a time and conditioning future propagation on the previously propagated frames, could be useful in such cases.

In addition, the current method always assumes that the propagation happens from the first frame or uniformly sampled frames. These “keyframes” from which the propagation takes place can instead be adaptively selected depending on the content of the video [141]. For example, more frames can be selected for human annotation from parts of the video undergoing large deformations instead of the more static parts where things do not change much and propagation can take place smoothly.

Finally, the current propagation method completely relies on the manual annotation to obtain complete video segmentation. It will be interesting to integrate this information with additional priors independent from the human annotation which can possibly capture where the object lies in the video. As a first step towards this, in the next chapter I will introduce the idea of a *generic pixel-level objectness* in images and videos and show that combining it with the supervoxel based propagation results in an even better segmentation performance.

Chapter 8

Pixel objectness in images and videos

[1]In the previous chapter, I proposed a segmentation propagation algorithm for videos, which takes a manually segmented frame in a video and propagates it to the entire video volume. Typically in a video segmentation propagation algorithm, one relies heavily on the manual segmentations provided by the human annotators to drive the underlying segmentation model [6, 36, 57, 118, 135, 141]. These are typically used to capture the appearance of the object of interest by learning strong appearance models from the manually segmented frames. However, as objects move and deform away from the manually segmented frame, the learned appearance model gets weaker and it becomes necessary to request further human guidance on future frames. This paradigm in a sense relies purely on the human guidance for assigning likelihoods to every pixel about being an object or background. Motion information from video is also typically restricted to creating temporal and higher-order constraints for propagating information [6, 36, 118, 135, 141].

---

[1] The work in this chapter was supervised by Dr. Kristen Grauman and originally published [55] in: FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. S. Jain and B. Xiong and K. Grauman. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2017, Hawaii, U.S.A. S. Jain contributed on the problem formulation and the design of network architecture.

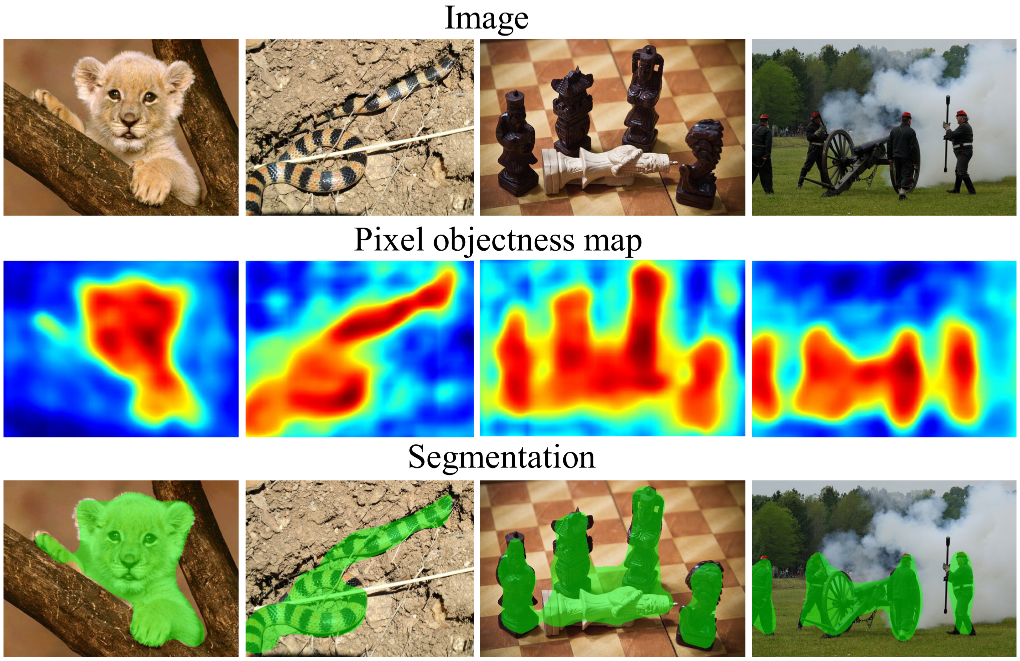


Figure 8.1: The goal is to predict an objectness map for each pixel (2nd row) and a single foreground segmentation (3rd row). Left to right: The proposed method can accurately handle objects with occlusion, thin objects with similar colors to background, man-made objects, and multiple objects. It is class-independent, meaning it is *not* trained to detect the particular objects in the images and videos.

However there are some inherent appearance and motion properties of objects in images and videos which clearly separate them from the background. Gestalt principles of grouping also suggest that these properties are fairly generalizable across object categories. It is quite natural to think that modeling these generic appearance and motion properties can help in generating a strong prior for each pixel as being an "object" or "background". I refer to these priors as "pixel objectness" for the remaining discussion.

In this chapter, I show that indeed low-level appearance and motion signals contain rich information about a pixel being on an "object" or background, which can be modeled in a data-driven manner (see Figure 8.1 for some examples). More specifically, I explore whether it is possible to learn a model that quantifies how likely a pixel belongs to an object of *any* class, and should be high even for objects unseen during training. The models that I develop in this chapter are truly "generic" in nature that generalize to thousands of object categories. Moreover, once trained they will be able to do it without any human guidance or input during test time.

Generic objectness signals from both appearance and motion are complex. For example, an object typically exhibits several intra-class variations including scale changes and complex shapes which a generic objectness model needs to capture. For motion, a single object may display multiple motions simultaneously, background and camera motion can intermingle, and even small-magnitude motions could be informative. This makes it almost impossible to hand-design rules which can generalize to thousands of object categories. It is essential to have learnable models for objectness which can exploit large volumes of training data to learn these rich signals.

Hence, in this chapter I introduce a two-stream deep network with end-to-end trainable appearance and motion streams which capture objectness cues present in each signal. The appearance stream requires an RGB image as input, while the motion stream requires optical flow from a video frame as input. The individual appearance and motion streams can be trained respectively to produce pixel-level objectness scores using appearance and motion information. Naturally for images we rely only on the appearance stream to generate pixel objectness maps from the RGB input. In the case of videos, a fusion module towards the end combines these individual appearance and motion streams in a unified manner to generate the final per-pixel objectness map for each frame. These objectness maps can then be thresholded to obtain binary "object" versus "background" segmentations for a given image or a video. This also gives us a fully automatic algorithm to perform object segmentation in images and videos.

Note that my proposed two-stream deep network is not restricted to segmenting objects that stand-out as is the case with automatic salient object segmentation methods [61, 86, 88, 93, 110, 162, 163]. It also produces only a single segmentation hypothesis as opposed to object proposal methods [5, 22, 33, 51, 74, 113, 137, 165] which generate thousands of segmentation outputs making it difficult to automatically select a single best segmentation. Also in contrast with fully automatic video segmentation methods [35, 81, 107, 160] which *strongly rely on motion alone* to seed the segmentation process and thus can fail in segmenting static objects, my proposed method uses both appearance and motion in a unified manner to segment all objects in videos (static or moving).

A standard way to train such a deep network would be to simply take large scale image and video segmentation datasets with per-pixel segmentations from thousands of object categories. However no such datasets exist till date which makes it very challenging to train such a network. In the following sections I show that this can be achieved by decoupling the individual streams and first independently learning a generic "appearance" network using large-scale image classification datasets (1000 classes) combined with per-pixel image segmentation data from a small number of object categories (20 classes). This "appearance" network, as we demonstrate, generalizes for segmenting thousands of object categories. It is further combined with weakly annotated video datasets to obtain high quality segmentations which are then used to train the motion stream. Finally, the fusion module which combines these individual streams is trained with only a small amount of boundary annotations from videos.

The appearance stream generalizes well and accurately segments foreground objects in images and video frames using appearance alone. For videos, the results show the reward of learning from both signals in a unified framework: a true synergy, with substantially stronger results than what we can obtain from either one alone. It significantly advances the state of the art for fully automatic video object segmentation on multiple challenging datasets. I also show that this generic pixel-level objectness can be combined with the video-specific appearance signals learned from a manually segmented frame (e.g., as obtained with *Click Carving* from Chapter 4) and together they result in an even stronger segmentation performance.

In the following, first I discuss my proposed appearance stream to segment generic objects from images and individual frames using appearance alone (Sec. 8.1). Then I describe the procedure to bootstrap the training of the motion stream from the outputs of the appearance stream and weakly labeled videos (Sec. 8.2). Next, I describe the fusion step that combines the two streams together to perform fully automatic video object segmentation (Sec. 8.3). I also present a semi-supervised extension, which augments this trained network with some manual annotations on the test video and further improves the performance (Sec. 8.4). Finally, I discuss the experimental results and compare with several state-of-the-art methods to demonstrate both its generalizability across object categories and also superior performance for video segmentation.

8.1 Appearance stream

The proposed appearance stream takes either an RGB image or video frame as input and directly outputs a generic pixel-level objectness map using appearance alone. A good generic appearance model should 1) predict a pixel-level map that aligns well with object boundaries, and 2) generalize so it can assign high probability to pixels of unseen object categories.

Challenges in dense foreground-labeled training data: Potentially, one way to address both challenges would be to rely on a large annotated image dataset that contains a large number of diverse object categories with pixel-level foreground annotations. However no such datasets currently exists. Existing datasets contain boundary-level annotations for merely dozens of categories (20 in PASCAL [34], 80 in COCO [89]), and/or for only a tiny fraction of all dataset images (0.03% of ImageNet's 14M images have such masks). To naively train a *generic* foreground object segmentation system, one might expect to need foreground labels for many more representative categories than what's available today.

Mixing explicit and implicit representations of objectness: This challenge motivates us to consider a different means of supervision to learn this generic foreground appearance stream. My idea is to train this stream to predict pixel level objectness using a mix of *explicit* boundary-level annotations and *implicit* image-level object category annotations. From the former, the system will obtain direct information about image cues indicative of generic foreground object boundaries. From the latter, it will learn object-like features across a wide spectrum of object types—but *without* being told where those objects' boundaries are.

To this end, this appearance stream is initialized using a powerful generic image representation learned from millions of images labeled by their object category, but lacking any foreground annotations. Then, this stream is further fine-tuned directly to produce dense binary segmentation maps, using relatively few images with pixel-level annotations originating from a small number of object categories. Note that at this point the appearance stream is completely decoupled from the motion stream and is being individually trained.

Since the pretrained network is trained to recognize thousands of objects, I hypothesize that its image representation has a strong notion of ob-

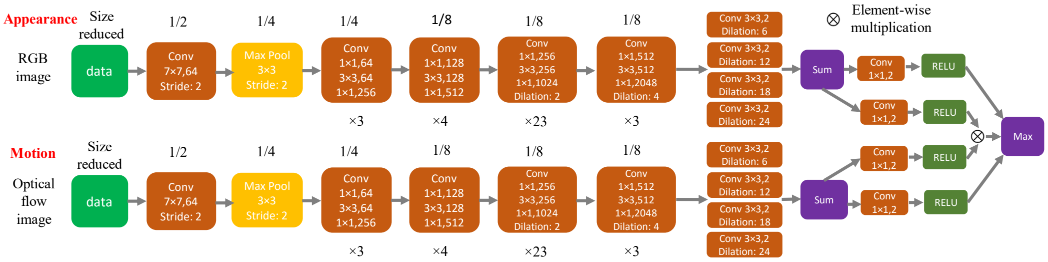


Figure 8.2: Network structure for the two-stream model with separate appearance and motion streams followed by a fusion module to combine them in a unified manner. Each convolutional layer except the first 7× 7 convolutional layer and the fusion blocks is a residual block [47], adapted from ResNet-101. The reduction in resolution is shown at top of each box and the number of stacked convolutional layers in the bottom of each box.

jectness built inside it, even though it never observes *any* segmentation annotations. Meanwhile, by subsequently training with explicit dense foreground labels, we can steer the appearance stream to fine-grained cues about boundaries that the standard object classification networks have no need to capture. This way, even if the appearance stream is trained with a limited number of object categories having pixel-level annotations, I expect it to learn generic representations helpful to predict pixel level objectness.

In particular, I adapt the image classification model ResNet-101 [47] and re-purpose it for doing segmentation. It is initialized with weights pre-trained on ImageNet, which provides a representation equipped to perform image-level classification for some 1,000 object categories. This appearance stream is then trained to perform well on the dense foreground pixel labeling task using a modestly sized semantic segmentation dataset. As we will see in the results, the learned appearance stream possesses a strong notion of objectness, making it possible to identify foreground regions of more than 3,000 object categories despite seeing ground truth masks for only 20 during training.

To re-purpose the classification network for doing segmentation, the last two groups of convolution layers are replaced with atrous convolution layers (also known as dilated convolution) to increase feature resolution. This results in only an 8× reduction in the output resolution instead of a 32× reduction in the output resolution in the original ResNet model. In order to improve the model's ability to handle both large and small objects, the classification layer of ResNet-101 is replaced with four parallel atrous convolutional layers with different sampling rates to explicitly account for object scale. Then the predictions from all four parallel atrous convolutional layers are fused by summing all the outputs. The loss is the sum of cross-entropy terms over each pixel position in the output layer, where ground truth masks consist of only two labels—object foreground or background. The model is trained using the Caffe implementation of [24]. This stream takes an image or a video frame of arbitrary size and produces an objectness map of the same size. See Figure 8.2 (top stream).

8.2 Motion stream

The appearance stream described in the previous section is capable of segmenting generic objects in images and videos using appearance information alone. However, in the case of videos, motion also plays an important and often complementary role to the appearance. Hence, for segmenting objects in video, I propose to develop a parallel motion stream which takes as input optical flow data encoded as an RGB image and outputs per-pixel objectness score using motion alone. These are then combined together using a fusion module for the final pixel-objectness output for a video frame (see Figure 8.2).

The direct parallel to the appearance stream discussed above would entail training the motion stream to map optical flow maps to video frame foreground maps. However, an important practical catch to that solution is training data availability. While ground truth foreground image segmentations are at least modestly available, datasets for video object segmentation masks are small-scale in deep learning terms, and primarily support evaluation. For example, Segtrack-v2 [84], one of the most commonly used benchmark datasets for video segmentation, contains only 14 videos with 1066 labeled frames. DAVIS [109] contains only 50 sequences with 3455 labeled frames. None contain enough labeled frames to train a deep neural network. Semantic video segmentation datasets like CamVid [18] or Cityscapes [28] are somewhat larger, yet limited in object diversity due to a focus on street scenes and vehicles. A good training source for our task would have ample frames with human-drawn segmentations on a wide variety of foreground objects, and would show a good mix of static and moving objects. No such large-scale dataset exists and creating one is non-trivial.

I propose a solution that leverages readily available *image* segmentation annotations together with *weakly annotated video* data to train my model.

In brief, the appearance stream trained in the previous section is allowed to hypothesize likely foreground regions in frames of a large video dataset annotated only by bounding boxes. Since the appearance alone need not produce perfect segmentations in video, I devise a series of filtering stages by which the system zeros in on high quality estimates of the true foreground. These instances bootstrap pre-training of the optical flow stream, then the two streams are joined to learn the best combination from minimal human labeled training videos.

More specifically, given a video dataset with bounding boxes labeled for each object,[2] the category labels are first ignored and the boxes alone are mapped to each frame. Then, the appearance stream, thus far trained only from images labeled by their foreground masks is applied to compute a binary segmentation for each frame. Next the box and segmentation are deconflicted in each training frame. First, the binary segmentation is refined by setting all the pixels outside the bounding box(es) as background. Second, for each bounding box, its checked whether the smallest rectangle that encloses all the foreground pixels overlaps with the bounding box by at least 75%. Otherwise the segmentation is discarded. Third, regions where the box contains more than 95% pixels labeled as foreground are discarded, based on the prior that good segmentations are rarely a rectangle, and thus probably the true foreground spills out beyond the box. Finally, segments where object

---

[2] We rely on ImageNet Video data, which contains 3862 videos and 30 diverse objects. See Section 8.5.

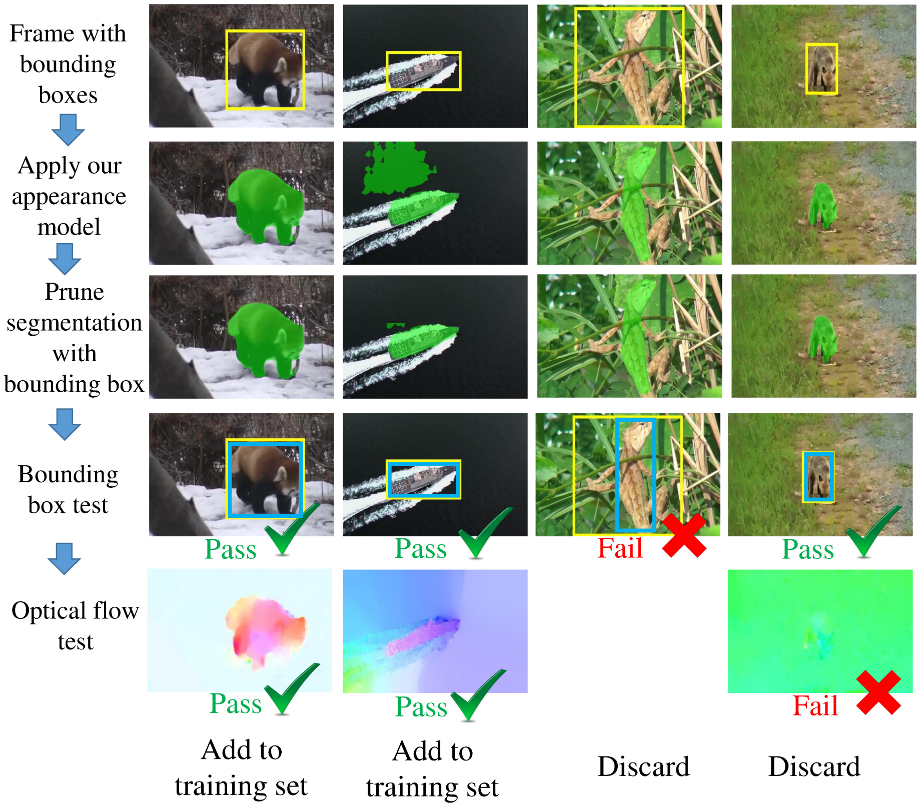


Figure 8.3: Procedures to generate (pseudo)-ground truth segmentations. The appearance model is first applied to obtain initial segmentations (second row, with object segment in green), followed by a pruning step which sets pixels outside bounding boxes as background (third row). Next, the bounding box test (fourth row, yellow bounding box is ground truth and blue bounding box is the smallest bounding box enclosing the foreground segment) and optical flow test (fifth row) are applied to determine whether the segmentation should be added to the motion stream's training set or discarded. Best viewed in color.

and background lack distinct optical flow are eliminated, so that the motion model can learn from the desired cues. Specifically, the frame's optical flow is computed using [91] and converted to an RGB flow image [8]. If the 2-norm between a) the average value within the bounding box and b) the average value in a box whose height and width are twice the original size exceeds 30, the frame and filtered segmentation are added to the training set.[3] See Figure 8.3 for a visual illustration of these steps.

To recap, bootstrapping from the preliminary appearance model, followed by bounding box pruning, bounding box tests, and the optical flow test, I can generate accurate per-pixel foreground masks for thousands of diverse moving objects—for which no such datasets exist to date. Note that by eliminating training samples with these filters, I aim to reduce label noise for training. However, at test time my system will be evaluated on standard benchmarks for which each frame is manually annotated (see Sec. 8.5).

With this data, I now turn to training the motion stream. Analogous to the strong generic appearance model, we also want to train a strong generic motion model that can segment foreground objects purely based on motion. The exact same network architecture as the appearance model (see Figure 8.2) is used here. The motion model takes only optical flow as the input and is trained with automatically generated pixel level ground truth segmentations. In particular, the raw optical flow is converted to a 3-channel (RGB) color-coded optical flow image [8]. This color-coded optical flow image is used as the input to the motion network. Again the network is initialized with pre-trained weights from ImageNet classification [123]. Representing optical flow using

---

[3] threshold chosen by initial visual inspection

RGB flow images allows us to leverage the strong pre-trained initializations as well as maintain symmetry in the appearance and motion arms of the network.

An alternative solution might forgo handing the system optical flow, and instead input two raw consecutive RGB frames. However, doing so would likely demand more training instances in order to discover the necessary cues. Another alternative would directly train the joint model that combines both motion and appearance, whereas we first "pre-train" each stream to make it discover convolutional features that rely on appearance or motion alone, followed by a fusion layer (below). My design choices are rooted in avoiding bias in training the model. Since the (pseudo) ground truth comes from the initial appearance network, either supplying two consecutive RGB frames or training jointly from the onset is liable to bias the network to exploit appearance at the expense of motion. By feeding the motion model with only optical flow, it ensures that the motion stream learns to segment objects from motion.

8.3 Fusion model

The final processing in my pipeline for segmenting videos joins the outputs of the appearance and motion streams, and aims to leverage a whole that is greater than the sum of its parts. I now describe how to train the joint model using both streams.

An object segmentation prediction is reliable if 1) either the appearance model or the motion model predicts the object segmentation with very strong confidence 2) both the appearance model and the motion model predict the segmentation. This motivates the network structure of the joint model.

I implement the idea by creating three indepedent parallel branches: 1) A 1×1 convolution layer followed by a RELU is applied to the output of the appearance model 2) A 1×1 convolution layer followed by a RELU is applied to the output of the motion model 3) The structure of first and second branches is replicated and an element-wise multiplication is applied on their outputs. The element-wise multiplication ensures the third branch outputs confident predictions of object segmentation if and only if both appearance model and motion model have strong predictions. Finally a layer that takes the element-wise maximum is applied to obtain the final prediction. See Figure 8.2.

As discussed above, we do not fuse the two streams in an early stage of the networks because we want them both to have strong independent predictions. Another advantage of this approach is that it only introduces six additional parameters in each 1×1 convolution layer, for a total of 24 trainable parameters. The fusion model can then be trained with very limited annotated video data, without overfitting.

8.4 Semi-supervised extension

The joint model discussed in the previous section allows for an automatic segmentation of objects in images and videos. However, on its own it cannot disambiguate between multiple objects present in the image or a video frame. The model is designed to only assign objectness scores to individual pixels. Moreover if the underlying appearance and motion signals are weak, it will be difficult for the model to output accurate objectness scores, which naturally will have an impact on the final segmentation. Hence, in this section I propose a semi-supervised extension to my two-stream deep segmentation model which combines the generic pixel objectness scores from the deep network with the video specific information learned from a manually segmented frame. This is done by incorporating the pixel objectness scores as additional unary terms in the supervoxel based video propagation technique discussed in Chapter 7. This effectively combines generic pixel objectness priors with video specific information from sparse human annotations and results in an improved performance for segmenting objects in videos.

8.5 Results

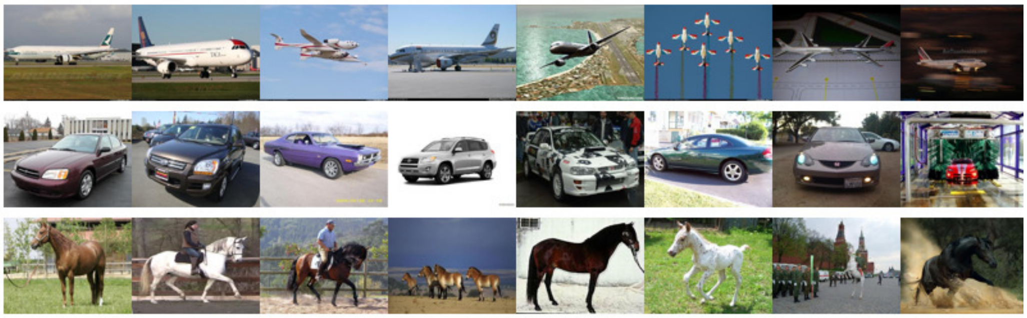
In this section I present experimental results and compare with other state-of-the-art methods. I discuss the results in two main parts. In the first part, I provide a detailed analysis on the generalization ability of the proposed appearance stream for segmenting objects using appearance alone. This is done through a detailed comparison on several image segmentation and localization datasets. This is extremely important, because all other components in my proposed method strongly rely on the outputs from the appearance stream. In the second part, I discuss the video segmentation performance of the complete joint model including the results from the semi-supervised extension. I first briefly describe the implementation details and then move on to presenting the results.

Implementation details: To train the appearance stream I rely on the PASCAL VOC 2012 segmentation (20 categories) dataset [34] and use a total of 10,582 training images with binary object versus background masks. As weak bounding box video annotations, I use the ImageNet-Video dataset [123]. This dataset comes with a total of 3,862 training videos from 30 object categories with 866,870 labeled object bounding boxes from over a million frames. Post refinement using my ground truth generation procedure (see Sec. 8.2), we are left with 84,929 frames with good pixel segmentations which are then used to train my motion model. For training the joint model a subset of held-out videos from the dataset is used. Each stream is trained for a total of 20,000 iterations, using "poly" learning rate policy (power = 0.9) with momentum (0.9) and weight decay (0.0005). No post-processing is applied on the segmentations obtained from the networks.

8.5.1 Generalization of appearance stream

In this section, I study the generalization ability of the proposed appearance stream by conducting large-scale experiments on several image datasets which have ground-truth for object segmentation (pixel-level object masks) or object localization (bounding boxes around the objects). Together these datasets cover objects from more than 3000 categories and thus provide strong evidence about the generalization of the proposed model. Please note that all the results in this section are obtained by simply thresholding (at 0.5) the

MIT Object Discovery Dataset



ImageNet Dataset

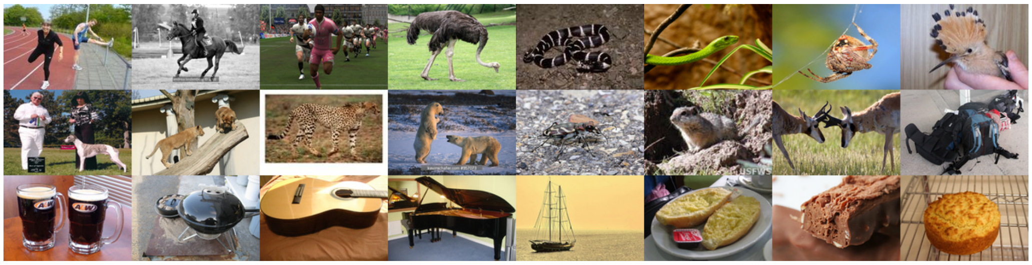


Figure 8.4: Examples from MIT Object Discovery and ImageNet datasets. (best viewed in color).

pixel objectness scores from the appearance stream alone. No motion information is available, hence motion stream and fusion module are not used here for anything.

8.5.1.1 Datasets, baselines and metrics

Datasets: I use three challenging datasets (Figure 8.4):

- **MIT Object Discovery:** This challenging dataset consists of Airplanes, Cars, and Horses [121]. It is most commonly used to evaluate weakly supervised segmentation methods. Note that this was also used in Chapter 5 and 6 for experiments. The images were primarily collected using internet search and the dataset comes with per-pixel ground truth segmentation masks.

- **ImageNet-Localization:** I conduct a large-scale evaluation of my approach using ImageNet [123] ($\sim$1M images with bounding boxes, 3,624 classes). The diversity of this dataset lets us test the generalization abilities of my method.
- **ImageNet-Segmentation:** This dataset contains 4,276 images from 445 ImageNet classes with pixel-wise ground truth from [44]. I use this dataset to evaluate segmentation performance on a large number of object classes.

Baselines: I compare to the following state-of-the-art methods:

- **Saliency Detection:** I compare to four salient object detection methods [61, 86, 162, 163], selected for their efficiency and state-of-the-art performance. All these methods are designed to produce a complete segmentation of the prominent object (versus localized fixation maps) and output continuous saliency maps, which are then thresholded by per image mean to obtain the segmentation.[4]
- **Object Proposals:** I also compare with state-of-the-art region proposal algorithms, multiscale combinatorial grouping (MCG) [5] and Deep-

---

[4] This thresholding strategy was chosen because it gave the best results.

Mask [113]. These methods output a ranked list of generic object segmentation proposals. The top ranked proposal in each image is taken as the final foreground segmentation for evaluation. I also compare with SalObj [88] which uses saliency to merge multiple object proposals from MCG into a single foreground.

- **Weakly supervised joint-segmentation methods:** These approaches rely on an additional weak supervision which comes in the form of prior knowledge that all images in a given collection share a common object category [25, 54, 63, 64, 67, 121, 131]. Note that my method lacks this additional supervision on test images.

Evaluation metrics: Depending on the dataset, I use: (1) **Jaccard Score:** Standard intersection-over-union (IoU) metric between predicted and ground truth segmentation masks and (2) **BBox-CorLoc Score:** Percentage of objects correctly localized with a bounding box according to PASCAL criterion (i.e IoU > 0.5) used in [30, 131].

For MIT and ImageNet-Segmentation, I use the segmentation masks and evaluate using the Jaccard score. For ImageNet-Localization I evaluate with the BBox-CorLoc metric, following the setup from [54, 131], which entails putting a tight bounding box around my method's output.

8.5.1.2 MIT Object Discovery dataset

First I present results on the MIT dataset [121]. I do separate evaluation on the complete dataset and also a subset defined in [121]. I compare my method with 13 existing state-of-the-art methods including saliency detection [61, 86, 162, 163], object proposal generation [5, 113] plus merging [88] and joint-segmentation [25, 54, 63, 64, 67, 121]. I compare with author-reported results for the joint-segmentation baselines, and use software provided by the authors for the saliency and object proposal baselines.

Table 8.1 shows the results. The proposed method outperforms several state-of-the-art saliency and object proposal methods—including recent deep learning techniques [86, 113, 163] in three out of six cases, and is competitive with the best performing method in the others.

The gains over the joint segmentation methods are arguably even more impressive because my proposed appearance stream simply segments a single image at a time—no weak supervision!—and still substantially outperforms all weakly supervised joint segmentation techniques. I stress that in addition to the weak supervision in form of segmenting common object, the previous best performing method [54] also makes use of a pre-trained deep network; we use strictly less total supervision than [54] yet still perform better. Furthermore, most joint segmentation methods involve expensive steps such as dense correspondences [121] or region matching [54] which can take up to hours even for a modest collection of 100 images. In contrast, my method directly outputs the final segmentation in a single forward pass over the deep network and takes

Methods	**MIT dataset (subset)**			**MIT dataset (full)**		
	Airplane	**Car**	**Horse**	**Airplane**	**Car**	**Horse**
# Images	82	89	93	470	1208	810
Joint Segmentation						
Joulin et al. [63]	15.36	37.15	30.16	n/a	n/a	n/a
Joulin et al. [64]	11.72	35.15	29.53	n/a	n/a	n/a
Kim et al. [67]	7.9	0.04	6.43	n/a	n/a	n/a
Rubinstein et al. [121]	55.81	64.42	51.65	55.62	63.35	53.88
Chen et al. [25]	54.62	69.2	44.46	60.87	62.74	60.23
Jain et al. [54]	58.65	66.47	53.57	62.27	65.3	55.41
Saliency						
Jiang et al. [61]	37.22	55.22	47.02	41.52	54.34	49.67
Zhang et al. [162]	51.84	46.61	39.52	54.09	47.38	44.12
DeepMC [163]	41.75	59.16	39.34	42.84	58.13	41.85
DeepSaliency [86]	69.11	83.48	57.61	**69.11**	83.48	**67.26**
Object Proposals						
MCG [5]	32.02	54.21	37.85	35.32	52.98	40.44
DeepMask [113]	**71.81**	67.01	58.80	68.89	65.4	62.61
SalObj [88]	53.91	58.03	47.42	55.31	55.83	49.13
Ours	66.59	**85.45**	**61.12**	67.34	**85.12**	65.10

Table 8.1: Comparison with state-of-the-art methods on MIT Object Discovery dataset. My method outperforms several state-of-the-art methods for saliency detection, object proposal generation, and joint segmentation. (Metric: Jaccard score).

only 0.6 seconds per image for complete processing.

8.5.1.3 ImageNet-Localization dataset

Next I present the segmentation results on ImageNet-Localization dataset. This involves testing the proposed appearance stream on about 1 million images from 3,624 object categories. This also lets us test how generalizable it is to unseen categories, i.e., those for which the method sees no foreground examples during training.

Table 8.2 shows the results. When doing the evaluation over all categories, I compare my method with five methods which report results on this

ImageNet-Localization dataset		
All	**# Classes**	**# Images**
	3,624	939,516
Non-PASCAL	**# Classes**	**# Images**
	3,149	810,219

Methods	**BBox-CorLoc**	
	All	Non-Pascal
Top-Objectness (Alexe) [3]	37.42	n/a
Tang et al. [131]	53.20	n/a
Jain et al. [54]	57.64	n/a
Saliency [61]	41.28	39.35
Top-Objectness (MCG) [5]	42.23	41.15
Ours	**62.45**	**60.36**

Table 8.2: Comparison with state-of-the-art methods on ImageNet-Localization dataset. My proposed appearance stream outperforms several state-of-the-art methods and also generalizes very well to unseen object categories. (Metric: BBox-CorLoc).

dataset [3, 54, 131] or are scalable enough to be run at this large scale [5, 61]. My method significantly improves the state-of-the-art. The saliency and object proposal methods [3, 5, 61] result in much poorer segmentations. My method also significantly outperforms the joint segmentation approaches [54, 131], which are the current best performing methods on this dataset. In terms of the actual number of images, the gains translate into correctly segmenting 42,900 more images than [54] (which, like us, leverages ImageNet features) and 83,800 more images than [131]. This reflects the overall magnitude of our gains over state-of-the-art baselines.

Does my learned segmentation model only recognize foreground objects that it has seen during training, or can it generalize to unseen object categories? Intuitively, ImageNet has such a large number of diverse categories that this gain in performance would not have been possible if my method was only over-fitting to the 20 seen PASCAL object categories. To empirically verify this intuition, I next exclude those ImageNet categories which are directly related to the PASCAL objects, by matching the two datasets' synsets.

ImageNet-Segmentation dataset	
Jiang et al. [61]	43.16
Zhang et al. [162]	45.07
DeepMC [163]	40.23
DeepSaliency [86]	61.12
MCG [5]	39.97
DeepMask [113]	58.69
SalObj [88]	41.35
Guillaumin et al. [44]	57.3
Ours	**64.22**

Table 8.3: Comparison with state-of-the-art methods on ImageNet-Segmentation dataset. The proposed appearance stream outperforms all state-of-the-art methods showing that it produces high-quality object boundaries (Metric: Jaccard score).

This results in a total of 3,149 categories which are exclusive to ImageNet ("Non-PASCAL"). See Table 8.2 for the data statistics.

We see only a very marginal drop in performance; my method still significantly outperforms both the saliency and object proposal baselines. This is an important result, because during training the segmentation model *never saw any dense object masks for images in these categories.* Bootstrapping from the pretrained weights of the Resnet-classification network, the appearance stream is able to learn a transformation between its prior belief on what looks like an object to complete dense foreground segmentations.

8.5.1.4 ImageNet-Segmentation dataset

Finally, I measure the pixel-wise segmentation quality on a large scale. For this I use the ground truth masks provided by [44] for 4,276 images from 445 ImageNet categories. For this dataset the current best results are due to the segmentation propagation approach of [44]. We found that DeepSaliency [86] and DeepMask [113] further improve it. Note that like my method, DeepSaliency [86] also trains with PASCAL [34]. DeepMask [113] is trained with a much larger COCO [89] dataset. My proposed appearance stream outperforms all methods, significantly improving the state-of-the-art (see Table 8.3). This shows that the appearance stream not only generalizes to thousands of object categories but also produces high quality object segmentations.

8.5.1.5 Qualitative results

Figure 8.5 shows qualitative results for the ImageNet dataset for both PASCAL and Non-PASCAL categories. The appearance stream accurately segments foreground objects from both sets. The examples from the Non-PASCAL categories highlight its strong generalization capabilities. It is able to segment objects across all scales and appearance variations, including multiple objects within an image. The bottom few examples show its remarkable ability to segment even man-made objects, which are especially distinct from the kind of objects in PASCAL dataset. The bottom row shows some failure cases. It has more difficulty in segmenting scene-centric images. It is understandable because in most scene-centric images, the entire scene is of primary importance and it is more difficult to clearly identify foreground objects.

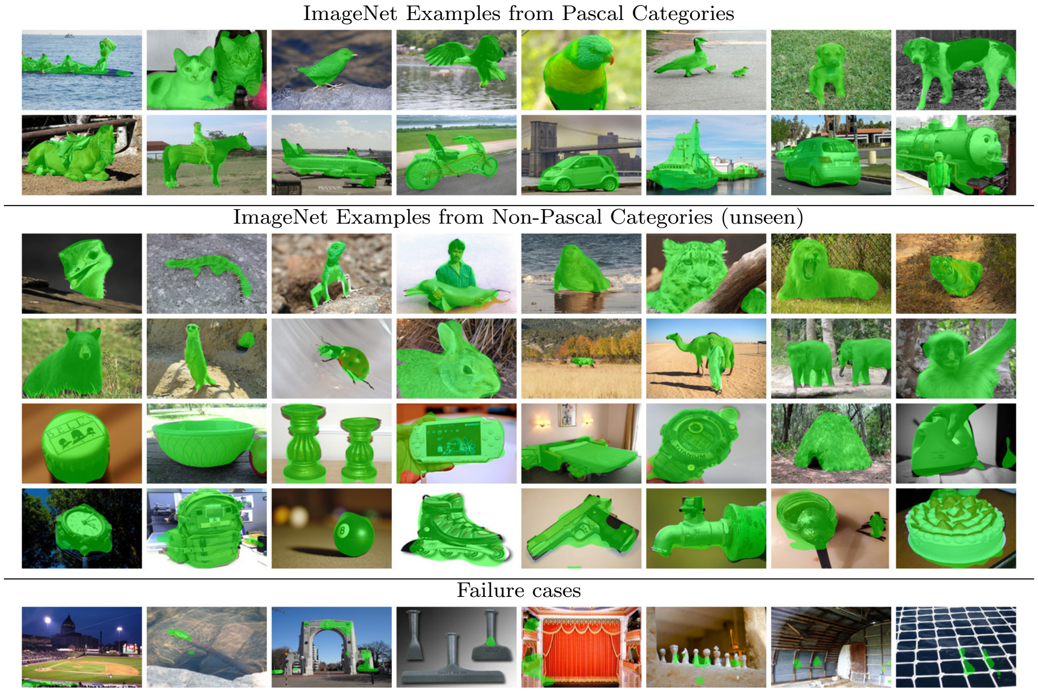

ImageNet Examples from Pascal Categories
ImageNet Examples from Non-Pascal Categories (unseen)
Failure cases

Figure 8.5: Qualitative results: I show qualitative results on images belonging to PASCAL (top) and Non-PASCAL (middle) categories. The segmentation model generalizes remarkably well even to those categories which were unseen in any foreground mask during training (middle rows). Typical failure cases (bottom) involve scene-centric images where it is not easy to clearly identify foreground objects (best viewed in color).

8.5.2 Video Segmentation Results

Having demonstrated the good performance and generalization ability of my proposed appearance stream for segmenting foreground objects, I now discuss the segmentation results for video datasets. This involves computing the pixel objectness from the complete joint model and then thresholding to obtain the foreground segmentation for a frame in a video.

8.5.2.1 Datasets, baselines and metrics

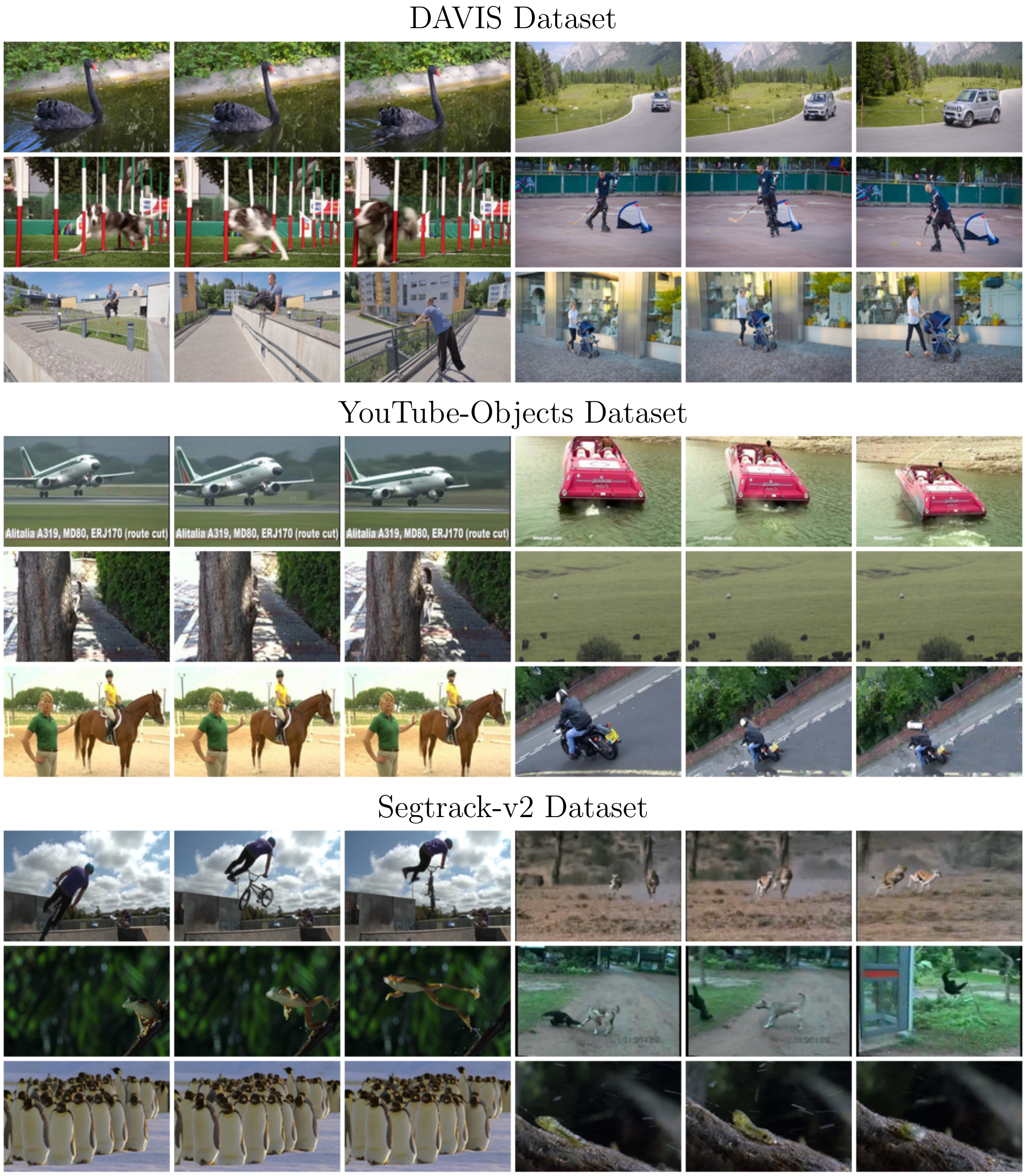


Figure 8.6: Example video sequences from DAVIS, YouTube-Objects and Segtrack-v2 datasets. (best viewed in color).

Datasets: I evaluate the method on three challenging video object segmentation datasets: DAVIS [109], YouTube-Objects [57, 115, 132] and Segtrack-v2 [84]. Figure 8.6 shows some visual examples from the datasets. To measure accuracy the standard Jaccard score is used, which computes the intersection over union overlap (IoU) between the predicted and ground truth object segmentations. The three datasets are:

- **DAVIS [109]:** the latest and most challenging video object segmentation benchmark consisting of 50 high quality video sequences of diverse object categories with $3,455$ densely annotated, pixel-accurate frames. The videos are unconstrained in nature and contain challenges such as occlusions, motion blur, and appearance changes. While the videos contain both static and moving objects, only the prominent moving objects were annotated in the ground-truth.
- **YouTube-Objects [57, 115, 132]:** consists of challenging Web videos from 10 object categories and is commonly used for evaluating video object segmentation. I use the subset defined in [132] and the ground truth provided by [57] for evaluation.
- **SegTrack-v2 [84]:** one of the most common benchmarks for video object segmentation consisting of 14 videos with a total of $1,066$ frames with pixel-level annotations. For videos with multiple objects with individual ground-truth segmentations, I treat them as a single foreground for evaluation.

Baselines: I compare with several state-of-the-art methods for each dataset as reported in the literature. Here I group them together based on whether they can operate in a fully automatic fashion (automatic) or require a human in the loop (semi-supervised) to do the segmentation:

- **Automatic methods:** Automatic video segmentation methods do not require any human involvement to segment new videos. Depending on the dataset, I compare with the following top-performing state of the art methods: FST [107], KEY [81], NLC [35] and COSEG [136]. All use some form of unsupervised motion or objectness cues to identify foreground objects followed by post-processing to obtain spatio-temporal object segmentations.
- **Semi-supervised methods:** Semi-supervised methods bring a human in the loop. They have some knowledge about the object of interest which is exploited to obtain the segmentation (e.g., a manually annotated first frame). I compare with the following state-of-the-art methods: HVS [43], HBT [41], FCP [111], IVID [125], HOP [57], and BVS [98]. The methods require different amounts of human annotation time to operate, e.g. HOP, BVS, and FCP make use of manual complete object segmentation in the first frame to seed the method; HBT requests a bounding box around the object of interest in the first frame; HVS, IVID require a human to constantly guide the algorithm whenever it starts to fail. Please note that HOP refers to my own supervoxel-based propagation method from the previous chapter.

Note that the automatic variant of my method requires human annotated data only during training. At test time it operates in a fully automatic fashion. Thus, given a new video, in that case my method requires equal effort as the automatic methods, and less effort than the semi-supervised methods. In the semi-supervised extension, where the outputs from the proposed joint segmentation model and my supervoxel based propagation method (HOP) from the previous chapter are combined, it requires the same effort as other semi-supervised methods. Apart from these comparisons, I also examine some natural baselines and ablated versions of my complete method:

- **Flow-thresholding (Flow-Th):** To examine the effectiveness of motion alone in segmenting objects, I adaptively threshold the optical flow in each frame using the flow magnitude. Specifically, I compute the mean and standard deviation from the L2 norm of optical flow magnitude and use "mean+unit std." as the adaptive threshold.

- **Flow-saliency (Flow-Sal):** Optical flow magnitudes can have large variances, hence I also try a variant which normalizes the flow by applying a saliency detection method based on [61] to the flow image itself. This is again followed by an average thresholding to obtain the segmentation.

- **Appearance model (Ours-A):** To quantify the role of appearance in segmenting objects, I obtain segmentations using only the appearance stream of my model.

- **Motion model (Ours-M):** To quantify the role of motion, I obtain segmentations using only the motion stream of my model. Note that this stream only sees the optical flow image and has no information about the object's appearance.

- **Joint model (Ours-Joint):** My complete joint model that learns to combine both motion and appearance together to obtain the final object segmentation.

- **Semi-supervised joint model (Ours-Joint-HOP):** My complete joint model combined with the semi-supervised supervoxel-based propagation algorithm from the previous chapter.

Quality of training data: To ascertain that the quality of training data, automatically generated for training my motion stream is good, it is first compared it with a small amount of human annotated ground truth. A set of 100 frames that passed both the bounding box and optical flow tests was randomly selected. We collected human-drawn segmentations for these 100 frames on Amazon Mechanical Turk. The crowd workers were first presented a frame with a bounding box labeled for each object, and then asked to draw the detailed segmentation for all objects within the bounding boxes. Each frame was labeled by three crowd workers and the final segmentation is obtained by majority vote on each pixel. The results indicate that my strategy to gather pseudo-ground truth is effective. On the 100 labeled frames, Jaccard overlap with the human-drawn ground truth is 77.8 (and 70.2 before pruning with bounding boxes).

I now present the quantitative comparisons of my method with several state-of-the-art methods and baselines, for each of the three datasets in turn.

8.5.2.2 DAVIS dataset

Table 8.4 shows the results, with some of the best performing methods on this dataset taken from the benchmark results [109]. My method outperforms all existing video segmentation methods on this dataset and significantly advances state-of-the-art. My method is significantly better than simple flow baselines. This supports my claim that even though motion contains a strong signal about foreground objects in videos, it is not straightforward to simply threshold optical flow and obtain those segmentations. A data-driven approach that learns to identify motion patterns indicative of objects as opposed to backgrounds or camera motion is required.

The fully automatic appearance and motion variants of my method themselves result in a very good performance. The performance of the motion variant is particularly impressive, knowing that it has no information about object's appearance and purely relies on the flow signal. When combined together, the fully automatic joint model results in a significant improvement, with an absolute gain of up to 11% over individual streams. This joint model when further combined with the supervoxel-based semi-supervised propaga-

DAVIS (50 videos)		
Methods	Human in loop?	Avg. IoU
Flow-Th	No	42.95
Flow-Sal	No	30.22
FST [107]	No	57.5
KEY [81]	No	56.9
NLC [35]	No	64.1
HVS [43]	Yes	59.6
HOP [57]	Yes	61.12
FCP [111]	Yes	63.1
BVS [98]	Yes	66.5
Ours-A	No	64.69
Ours-M	No	60.18
Ours-Joint	No	**71.51**
Ours-Joint-HOP	Yes	**74.68**

Table 8.4: Video object segmentation results on DAVIS dataset. I show the average accuracy over all 50 videos. The fully automatic variant of my method itself outperforms several state-of-the art methods, including the ones which actually require human supervision during segmentation. The best performing methods grouped by whether they require human-in-the-loop or not during segmentation are highlighted in bold. Metric: Jaccard score, higher is better.

tion algorithm (Ours-Joint-HOP), results in the overall best performance. This highlights the strengths of incorporating human guidance when there is ambiguity or when the underlying appearance and motion signals might be weak.

The proposed method is also significantly better than fully automatic methods, which typically rely on motion alone to identify foreground objects. This illustrates the benefits of a unified combination of both motion and appearance. Most surprisingly, the fully automatic variant of my method significantly outperforms even the existing state-of-the-art semi supervised tech-

YouTube-Objects dataset (126 videos)											
Methods	Flow-Th	Flow-Sal	FST [107]	COSEG [136]	HBT [41]	HOP [57]	IVID [125]	Ours-A	Ours-M	Ours-Joint	Ours-Joint-HOP
Human in loop?	No	No	No	No	Yes	Yes	Yes	No	No	No	Yes
airplane (6)	18	33	71	69	74	86	**89**	**83**	59	82	80
bird (6)	32	34	71	**76**	56	81	**82**	61	64	64	72
boat (15)	4	23	43	54	58	69	**74**	**73**	40	72	**74**
car (7)	22	49	65	70	34	69	71	75	61	**75**	**77**
cat (16)	20	32	52	67	31	59	68	68	49	**68**	**70**
cow (20)	17	29	45	49	42	69	**79**	**70**	39	68	73
dog (27)	18	25	65	48	37	62	70	69	55	**69**	**74**
horse (14)	12	24	54	56	44	54	**68**	**63**	40	60	67
mbike (10)	13	17	44	40	49	61	62	62	43	**63**	**65**
train (5)	18	24	30	53	39	66	**78**	**63**	43	62	64
Avg. IoU	17	29	54	58	46	68	**74**	**69**	49	68	72

Table 8.5: Video object segmentation results on YouTube-Objects dataset. I show the average performance for each of the 10 categories from the dataset. The final row shows an average over all the videos. The fully automatic variant of my method outperforms several state-of-the art methods, including the ones which actually require human supervision during segmentation. The semi-supervised variant outperforms the best performing semi-supervised method IVID in half the categories. However, note that IVID requires a human in the loop always to correct mistakes hence is much more expensive. The best performing methods grouped by whether they require human-in-the-loop or not during segmentation are highlighted in bold. Metric: Jaccard score, higher is better.

niques, which require substantial human annotation on every video they process. All those existing methods rely only on the human guidance to guide the segmentation process. The superior performance of my semi-supervised variant which utilizes both the human guidance and generic pixel objectness priors demonstrates the effectiveness of combining them together instead of relying on one or the other.

8.5.2.3 YouTube-Objects dataset

Table 8.5 shows a similarly strong result on the YouTube-Objects dataset. This dataset shares categories with the PASCAL segmentation benchmark used to train my appearance stream. Accordingly, I observe that the appearance stream itself results in the best performance among the fully automatic variants of my method. Moreover, this dataset has a mix of static and moving objects which explains the relatively weaker performance of my motion model alone. The combined joint model works similarly well as appearance alone. Again, augmenting the joint model with a human segmented frame results in the overall best performance.

Overall, my method again outperforms the flow baselines and all the automatic methods by a significant margin (see Table 8.5). The publicly available code for NLC [35] runs successfully only on 9% of the YouTube dataset (1725 frames); on those, its Jaccard score is 43.64%. The proposed model outperforms it by a significant margin of 28% on these frames. Even among human-in-the-loop methods, it outperforms all methods except IVID [125]. However I would like to point out that IVID [125] requires a human in the loop consistently to track the segmentation performance and correct whatever mistakes the algorithm makes. This can take up to minutes of human annotation time for each video. In contrast, even the fully automatic variants in my proposed method perform very competitively and the semi-supervised variant (Ours-Joint-HOP) which only receives a one-shot guidance (i.e., a single manually segmented frame) outperforms IVID in 5 out of 10 categories.

Segtrack-v2 (14 videos)		
Methods	Human in loop?	Avg. IoU
Flow-Th	No	37.77
Flow-Sal	No	27.04
FST [107]	No	53.5
KEY [81]	No	57.3
NLC [35]	No	**80***
HBT [41]	Yes	41.3
HVS [43]	Yes	50.8
HOP [57]	Yes	60.54
Ours-A	No	56.88
Ours-M	No	53.04
Ours-Joint	No	61.40
Ours-Joint-HOP	Yes	**65.36**

Table 8.6: Video object segmentation results on Segtrack-v2. I show the average accuracy over all 14 videos. For NLC results are averaged over 12 of the 14 videos as reported in their paper [35]. The proposed method outperforms all other methods except NLC which is exceptionally strong on this dataset. The best performing methods grouped by whether they require human-in-the-loop or not during segmentation are highlighted in bold. Metric: Jaccard score, higher is better.

8.5.2.4 Segtrack-v2 dataset

In Table 8.6, my method outperforms all semi-supervised and automatic baselines except NLC [35] on Segtrack. While my approach significantly outperforms NLC [35] on the DAVIS and YouTube-Objects datasets, NLC is exceptionally strong on this dataset. The relatively weaker performance of my proposed method could be due to the low quality and resolution of the Segtrack-v2 videos, making it hard for my network based model to process them. Nonetheless, the joint model still provides a significant boost over both the appearance and motion streams, showing that it again realizes the synergy of motion and appearance in a useful way. Moreover, the semi-supervised variant again results in an overall best performance amongst all the proposed variants.

8.5.2.5 Qualitative evaluation

Figure 8.7 shows qualitative results of my method. The top half shows visual comparisons between different components of my method including the appearance, motion, and joint models. I also show the optical flow image that was used as an input to the motion stream. These images help reveal the complexity of learned motion signals. In the bear example, the flow is most salient only on the bear's head, still my motion stream alone is able to segment the bear completely. The boat, car, and sail example shows that even when the flow is noisy—including strong flow on the background—my motion model is able to learn about object shapes and successfully suppresses the background regions. The rhino and train examples show cases where the appearance model fails to segment accurately but when combined with the motion stream, the joint model produces accurate segmentations.

The bottom half of Figure 8.7 shows visual comparisons between my method and state-of-the-art automatic [35, 107] and semi-supervised [98, 111] methods. The automatic methods have a very weak notion about object's appearance; hence they completely miss parts of objects [35] or cannot disambiguate the objects from background [107]. Semi-supervised methods [98, 111], which rely heavily on the initial human-segmented frame to learn about object's appearance, start to fail as time elapses and the object's appearance changes considerably. In contrast, my method successfully learns to combine generic cues about object motion and appearance, segmenting much more accurately across all frames even in very challenging videos[5].

8.6 Conclusion

In this chapter I introduced the notion of a generic pixel-level objectness in images and videos. This was realized through a novel two-stream deep network with parallel appearance and motion streams. Each stream individually captured the notion of pixel objectness through appearance and motion cues respectively. The fusion module which then combined these two streams together in a unified manner achieved a deep synergy between the motion and appearance information.

The proposed appearance stream generalizes to thousands of object categories and also allowed us to train the complete network for video segmentation. Results on the video segmentation benchmarks show sizeable improvements over several state-of-the-art methods. Throughout the chapter, the proposed method also addresses several practical challenges and shows that it is possible to train generic pixel objectness models without the availability of large scale image and video datasets with boundary annotations.

---

[5] Additional results and videos available at: `http://vision.cs.utexas.edu/projects/fusionseg/`

Finally, I also demonstrated that combining human guidance with the generic pixel-level objectness results in further improvements for segmenting objects in videos.

Building on the strengths of the current pixel objectness model there are several natural extensions which are possible. Firstly, the current model uses late fusion to combine motion and appearance together. This particular design choice is primarily governed by the lack of sufficient training data. Given enough training data, exploring other architectures involving early fusion of the appearance and motion streams can be potentially useful in finding even better ways of fusing these complementary sources of information.

Secondly, pixel objectness in videos currently relies on information extracted from a single frame (for appearance) or adjacent frames (for motion). Incorporating longer range information either through the use of 3D convolutions or recurrent models can further improve the way in which the model learns about an object's motion and dynamics.

Another key weakness of the current pixel-level objectness method is that it is not instance aware. Right now it treats all objects as a single foreground. Going from this single foreground prior to an instance-aware prior will be really useful for downstream applications (for e.g., in visual search, scene understanding etc.)

Finally, taking inspiration from the effectiveness of combining generic pixel-level objectness with supervoxel based propagation, in the future it will interesting to explore ideas which combine my Click Carving algorithm (Chapter 4) and the generic-pixel level objectness. The objectness prior can be incorporated in the ranking process which can possibly lead to more speedups.

Overall, in the previous chapters I explored different aspects of human-machine collaboration for segmenting foreground objects in images and videos. I presented novel algorithms developed in this thesis for interactively segmenting objects in individual images, jointly segmenting objects in weakly supervised image collections, and also for segmenting objects in videos. In the next chapter, I will discuss some possible directions for future work.

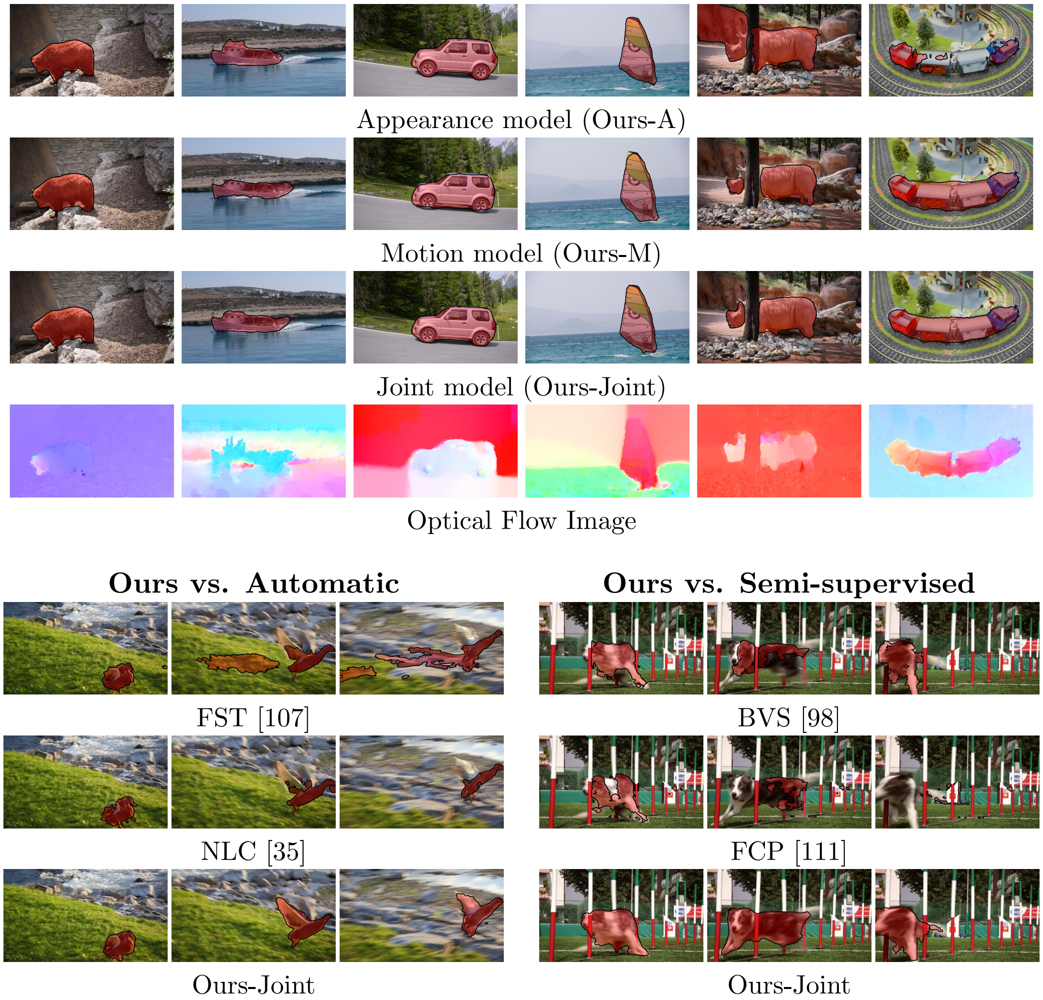


Figure 8.7: Qualitative results: The top half shows examples from my appearance, motion, and joint models along with the flow image which was used as an input to the motion network. The bottom rows show visual comparisons of the joint model with existing automatic and semi-supervised baselines (best viewed in color and see text for the discussion).

Chapter 9

Future Work

In the previous chapters, I developed methods which explored different aspects for human-machine collaboration for foreground segmentation in images and videos. There are several interesting avenues for future research which include some specific ideas which can directly extend the work presented in this thesis and some broader themes for more long-term research goals.

First, it would be very interesting to incorporate the active selection ideas from Chapter 5 in the context of videos. Currently, the segmentation propagation in Chapter 7 is done only from a fixed set of video frames (for example, the first frame). However, it is natural to think that the frames from which propagation happens can be actively chosen such that when labeled by human annotators the propagation will be more likely to succeed. For example choosing frames where objects are undergoing large motions or occlusions may be more important for the propagation to succeed than choosing frames where the object is mostly static. This can be further enhanced by a stage-wise algorithm, which propagates from an initial set of actively chosen frames, automatically identifies when propagation engine starts to fail, and requests more annotations accordingly.

Second, the idea of predicting compatibility for co-segmentation of image pairs which was developed in Chapter 6 can naturally be incorporated in the joint segmentation and active selection methods which were discussed in Chapter 5. Currently the image neighborhoods in the joint segmentation and active selection graphs in Chapter 5 only rely on similarity between image features. A more data-driven approach that can adapt itself to the strengths and weaknesses of a particular joint segmentation and selection algorithm can potentially lead to an improved segmentation propagation. Moreover the idea of joint segmentation was restricted to image collections. It will be interesting to explore these ideas in the context of segmenting a collection of weakly-supervised videos.

Third, the generic pixel-level objectness for images and videos which was developed in (Chapter 8) can provide a strong prior for foreground objects in several other problems. For example, it can potentially be used to improve the performance of image search engines by focusing on foreground regions while performing query to target matching. Content-aware resizing algorithms can also be enhanced by explicitly penalizing for removing the foreground content. This generic pixel-objectness prior can also be used to enhance interactive segmentation algorithms, where the human guidance can be augmented with this prior while generating the segmentation output.

Fourth, the idea of *Click Carving* can be further adapted for the task of segmenting objects in video. In the current form, the user segments an object in an initial frame and then this manually segmented frame is propagated to the entire video to obtain the segmentation. However the key idea behind Click Carving (to pre-generate thousands of segmentation hypotheses) can directly be expanded to videos instead of this two step process. This will require us to generate thousands of space-time segmentation proposals instead of per frame proposals which we currently have. The user can than directly select a space-time proposal using the Click Carving idea to do video segmentation.

In the long term, I believe that human-machine collaboration can be a very effective approach for solving challenging computer vision and machine learning problems. While in this thesis the primary focus was on the problem of image and video segmentation, the broad idea of actively engaging human annotators can be applied to several other domains such as in robotics and natural language processing. Another interesting research direction is to explore alternative means of engaging human annotators. In all modern crowdsourcing platforms, monetary benefit is still the key driver for human annotators while the tasks remain mundane. Designing gamified interfaces or providing additional value to the users while they guide the system can potentially allow us to further scale these algorithms for real world applications.

Chapter 10

Conclusion

In this thesis, I presented novel algorithms for segmenting foreground objects in images and videos. The key idea in this thesis was to bring the complementary strengths of humans and machines together to solve this problem more efficiently and effectively. The resulting algorithms can actively reason about the modes of user interaction through which humans can guide the system, can identify where the human guidance is most needed, and are also capable of propagating human guidance to other unguided instances whenever possible. Together it results in human-machine collaborative systems which lead to large savings in human annotation costs while achieving high levels of performance.

Towards this goal, I first studied the problem of interactively segmenting objects in images and videos. First, I proposed a method to predict the input modality which is sufficiently strong for segmenting objects in images using traditional interactive segmentation methods. This demonstrated the utility in actively reasoning about the extent to which a human needs to guide a segmentation system. Next, I developed a novel interactive segmentation algorithm which is capable of segmenting objects in images and videos using simple point clicks. In contrast with existing modalities of human interaction used in the current algorithms, this requires only a fraction of human effort and often outperforms alternative and more expensive methods significantly.

Having developed novel algorithms for interactive segmentation of a single image, I next studied the problem of jointly segmenting objects in weakly supervised image collections. For this, I developed a novel segmentation propagation and active selection algorithm that can actively select images for human annotation which, once labeled, will be most useful for jointly segmenting the entire collection. I showed that this stage-wise approach results in a significant reduction in the amount of human annotation required to obtain good quality segmentations for the entire collection. In this context, I also introduced the idea of predicting compatibility between image partners for joint segmentation and demonstrated that segmenting compatible images together results in an improved segmentation performance.

Finally, turning from images to videos, I studied the problem of semi-supervised video propagation and designed a supervoxel-based propagation algorithm which can exploit long-range connection in videos to accurately propagate information. Results show that the supervoxel-based propagation algorithm outperforms several state-of-the-art segmentation algorithms and is much more efficient in practice. I also introduced the idea of a generic pixel-level objectness in images and videos, which was implemented using an end-to-end trainable deep neural network. Pixel objectness itself allowed us to obtain high quality image and video segmentation results. Moreover when combined with the human guidance in the supervoxel propagation algorithm, together they resulted in a state-of-the-art video segmentation algorithm.

Throughout, I addressed key issues that arise both from the perspective of designing novel segmentation algorithms and also for efficiently utilizing the human guidance that is available on demand. Extensive experiments on challenging datasets and detailed comparisons with state-of-the-art methods and relevant baselines validated the effectiveness of the proposed methods.

Overall, this thesis helps realize the potential of human-machine collaboration for foreground segmentation in images and videos. The proposed methods result in significant savings in human annotation costs and thus have the potential for enabling large-scale collection of image and video segmentation data across several domains in an economical manner. Moreover, they can potentially make a significant impact in improving the solutions for several important real world problems such as image and video search, image synthesis, and post-production video editing. Finally, these methods can be a key component in higher-level computer vision systems for activity and scene understanding, where accurately segmenting foreground objects is extremely important.

Bibliography

[1] Ejaz Ahmed, Scott Cohen, and Brian Price. Semantic object selection. In *CVPR*, June 2014.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. In *ECCV*, 2010.

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an Object? In *CVPR*, 2010.

[4] O. Aodha, N. Campbell, J. Kautz, and G. Brostow. Hierarchical subquery evaluation for active learning on a graph. In *CVPR*, 2014.

[5] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[6] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *CVPR*, 2010.

[7] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: Robust video object cutout using localized classifiers. In *SIGGRAPH*, 2009.

[8] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[9] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive Co-segmentation with Intelligent Scribble Guidance. In *CVPR*, 2010.

[10] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

[11] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. *CVPR*, 2015.

[12] A. Borji, M. M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, Dec 2015.

[13] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *CVPR*, 2001.

[14] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, September 2004.

[15] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 2001.

[16] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, 2009.

[17] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *ICML*, pages 59–66. AAAI Press, 2003.

[18] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009.

[19] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI*, 2011.

[20] Thomas Brox and Jitendra Malik. Object Segmentation by Long Term Analysis of Point Trajectories. In *ECCV*, 2010.

[21] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classication of objects and scenes. In *ICCV*, 2007.

[22] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012.

[23] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[24] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[25] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Enriching Visual Knowledge Bases via Object Discovery and Segmentation. In *CVPR*, 2014.

[26] H.-T. Cheng and N. Ahuja. Exploiting nonlocal spatiotemporal structure for video segmentation. In *CVPR*, 2012.

[27] Prakash Chockalingam, S. Nalin Pradeep, and Stan Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009.

[28] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[30] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International Journal of Computer Vision*, 100(3):275–293, September 2012.

[31] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, December 2015.

[32] E. Elhamifar, G. Sapiro, A. Yan, and S. Sastry. A convex optimization framework for active learning. In *ICCV*, 2013.

[33] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

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

[36] A. Fathi, M. Balcan, X. Ren, and J. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, 2011.

[37] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, September 2004.

[38] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *CVPR*, 2015.

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

[40] Fabio Galasso, Roberto Cipolla, and Bernt Schiele. Video segmentation with superpixels. In *ACCV*, 2012.

[41] Martin Godec, Peter M. Roth, and Horst Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011.

[42] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *TPAMI*, 29(12):2247–2253, 2007.

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

[44] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. ImageNet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110(3):328–348, 2014.

[45] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010.

[46] Danna Gurari, Suyog Jain, Margrit Betke, and Kristen Grauman. Pull the plug? predicting if computers or humans should segment images. In *CVPR*, 2016.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[48] D. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[49] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Semi-supervised SVM Batch Mode Active Learning with Applications to Ima ge Retrieval. *ACM Transactions on Information Systems*, 1(1), 2009.

[50] Derek Hoiem, Martial Hebert, and Andrew Stein. Learning to find object boundaries using motion cues. *ICCV*, 2007.

[51] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *TPAMI*, 2015.

[52] S. Jain and K. Grauman. Predicting sufficent annotation strength for interactive foreground segmentation. In *ICCV*, 2013.

[53] S. Jain and K. Grauman. Which image pairs will cosegment well? predicting partners for cosegmentation. In *ACCV*, 2014.

[54] S. Jain and K. Grauman. Active image segmentation propagation. In *CVPR*, 2016.

[55] Suyog Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017.

[56] Suyog Jain, Bo Xiong, and Kristen Grauman. Pixel objectness. *arXiv preprint arXiv:1701.05349*, 2017.

[57] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[58] Suyog Dutt Jain and Kristen Grauman. Click carving: Segmenting objects in video with point clicks. In *HCOMP*, October 2016.

[59] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 2013.

[60] 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.

[61] Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. Saliency detection via absorbing markov chain. In *ICCV*, 2013.

[62] T. Joachims. Optimizing search engines with clickthrough data. In *KDD*, 2002.

[63] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[64] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[65] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[66] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, pages 321–331, 1988.

[67] G.H. Kim, E.P. Xing, L. Fei Fei, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011.

[68] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV*, 2012.

[69] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.

[70] Jaechul Kim and Kristen Grauman. Boundary preserving dense local regions. In *CVPR*, 2011.

[71] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[72] Pushmeet Kohli, Hannes Nickisch, Carsten Rother, and Christoph Rhemann. User-centric learning and evaluation of interactive segmentation systems. *IJCV*, 100(3):261–274, December 2012.

[73] Pushmeet Kohli and Philip H. S. Torr. Measuring uncertainty in graph cut solutions. *CVIU*, 112(1):30–38, 2008.

[74] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *ECCV*, 2014.

[75] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems (to appear)*. Cambridge University Press, February 2014.

[76] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *National Conference on Artificial Intelligence (AAAI), Nectar track*, July 2007.

[77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[78] Srinivas S. Kruthiventi, Kumar Ayush, and R. Venkatesh Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *CoRR*, abs/1510.02927, 2015.

[79] Daniel Kuettel and Vittorio Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012.

[80] Y. J. Lee and K. Grauman. Collect-Cut: Segmentation with top-down cues discovered in multi-object images. In *CVPR*, 2010.

[81] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[82] Victor S. Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[83] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011.

[84] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video Segmentation by Tracking Many Figure-Ground Segments. In *ICCV*, 2013.

[85] Hanxi Li, Yi Li, and Fatih Porikli. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In *BMVC*, 2014.

[86] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. DeepSaliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 25(8), Aug 2016.

[87] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, 2005.

[88] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. The secrets of salient object segmentation. *CVPR*, 2014.

[89] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[90] C. Liu, J. Yuen, and A. Torralba. Sift flow: dense correspondence across different scenes and its applications. *TPAMI*, 2011.

[91] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis.* PhD thesis, Citeseer, 2009.

[92] Nian Liu, Junwei Han, Dingwen Zhang, Shifeng Wen, and Tianming Liu. Predicting eye fixations using convolutional neural networks. In *CVPR*, 2015.

[93] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *TPAMI*, 33(2):353–367, February 2011.

[94] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, November 2015.

[95] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015.

[96] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.

[97] Tomasz Malisiewicz and Alexei A. Efros. Spatial support for objects via multiple segmentations. In *BMVC*, 2007.

[98] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, pages 743–751, 2016.

[99] Kevin McGuinness and Noel E. OConnor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434 – 444, 2010. Interactive Imaging and Vision.

[100] E. Mortensen and W. Barrett. Intelligent scissors for image composition. In *SIGGRAPH*, 1995.

[101] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[102] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[103] Dan Oneata, Jerome Revaud, Jakob Verbeek, and Cordelia Schmid. Spatio-temporal object detection proposals. In *ECCV*, Sep 2014.

[104] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Trans on Sys, Man and Cybernetics*, 9(1):62–66, January 1979.

[105] Junting Pan, Kevin McGuinness, Elisa Sayrol, Noel O'Connor, and Xavier Giro-i Nieto. Shallow and deep convolutional networks for saliency prediction. In *CVPR*, 2016.

[106] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015.

[107] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[108] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR Workshop*, 2015.

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

[110] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[111] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, December 2015.

[112] Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[113] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollr. Learning to segment object candidates. In *NIPS*, 2015.

[114] J. Pont-Tuset, M.A. Farré, and A. Smolic. Semi-automatic video object segmentation by advanced manipulation of segmentation hierarchies. In *CBMI*, 2015.

[115] Alessandro Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[116] Brian L. Price, Bryan S. Morse, and Scott Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009.

[117] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[118] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007.

[119] C. Rother, V. Kolmogorov, and A. Blake. Grabcut -interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[120] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *CVPR*, 2006.

[121] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[122] Michael Rubinstein, Ce Liu, and William T. Freeman. Annotation propagation in large image databases via dense image correspondence. In *ECCV*, 2012.

[123] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[124] J. Serrat, A. Lopez, N. Paragios, and J. C. Rubio. Unsupervised co-segmentation through region matching. In *CVPR*, 2012.

[125] Naveen Shankar Nagaraja, Frank R. Schmidt, and Thomas Brox. Video segmentation with just a few strokes. In *ICCV*, December 2015.

[126] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[127] B. Siddiquie and A. Gupta. Beyond Active Noun Tagging: Modeling Contextual Interactions for Multi-Cl ass Active Learning. In *CVPR*, 2010.

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

[129] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[130] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, Washington, DC, USA, 2011.

[131] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014.

[132] Kevin Tang, Rahul Sukthankar, Jay Yagnik, and Li Fei-Fei. Discriminative segment annotation in weakly labeled video. In *CVPR*, 2013.

[133] S. Todorovic and N. Ahuja. Unsupervised category modeling, recognition, and segmentation in images. *TPAMI*, 30(12):2158–2174, 2008.

[134] Antonio Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 53(2):169–191, July 2003.

[135] David Tsai, Matthew Flagg, and James Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, 2010.

[136] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *ECCV*, 2016.

[137] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[138] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.

[139] Alexander Vezhnevets, Joachim M Buhmann, and Vittorio Ferrari. Active learning for semantic segmentation with expected change. In *CVPR*, 2012.

[140] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[141] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012.

[142] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *CVPR*, 2010.

[143] Sudheendra Vijayanarasimhan and Kristen Grauman. Multi-level active prediction of useful image annotations for recognition. In *NIPS*, 2009.

[144] Sudheendra Vijayanarasimhan and Kristen Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009.

[145] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. In *NIPS*, 2011.

[146] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV*, 2010.

[147] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011.

[148] D. Wang, C. Yan, S. Shan, and X. Chen. Active learning for interactive segmentation with expected confidence change. In *ACCV*, 2012.

[149] Jue Wang, Pravin Bhat, Alex Colburn, Maneesh Agrawala, and Michael F. Cohen. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, 2005.

[150] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *ICCV*, December 2015.

[151] Tinghuai Wang, Bo Han, and John Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding*, 120:14 – 30, 2014.

[152] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Learning to Detect Motion Boundaries. In *CVPR 2015*, Boston, United States, June 2015.

[153] Longyin Wen, Dawei Du, Zhen Lei, Stan Z. Li, and Ming-Hsuan Yang. Jots: Joint online tracking and segmentation. In *CVPR*, June 2015.

[154] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 2005.

[155] Zhengyang Wu, Fuxin Li, Rahul Sukthankar, and James M. Rehg. Robust video segment proposals with painless occlusion handling. In *CVPR*, June 2015.

[156] Fanyi Xiao and Yong Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016.

[157] C. Xu and J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012.

[158] Chenliang Xu, Caiming Xiong, and Jason J Corso. Streaming Hierarchical Video Segmentation. In *ECCV*, 2012.

[159] Gang Yu and Junsong Yuan. Fast action proposals for human action detection and search. In *CVPR*, June 2015.

[160] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.

[161] Honghui Zhang, Jianxiong Xiao, and Long Quan. Supervised label transfer for semantic segmentation of street scenes. In *ECCV*, 2010.

[162] Jianming Zhang and Stan Sclaroff. Saliency detection: a boolean map approach. In *ICCV*, 2013.

[163] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context learning. In *CVPR*, 2015.

[164] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[165] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

Vita

Suyog Dutt Jain was born in India on 02 May 1985, the son of Sunil Dutt Jain and Sangeeta Jain. He received the Bachelor of Engineering Degree from Manipal University, Manipal in 2008. Following this he joined University of Texas at Austin in Fall 2009 and received his Masters in Computer Science Degree in August 2011. Subsequently, he joined the Computer Vision Group headed by Prof. Kristen Grauman in September 2012 as a Graduate Research Assistant while pursuing his doctoral studies at University of Texas at Austin.

Permanent address: suyogjain@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.