NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

CO-SALIENCY BASED VISUAL OBJECT CO-SEGMENTATION AND CO-LOCALIZATION

KOTESWAR RAO JERRIPOTHULA

INTERDISCIPLINARY GRADUATE SCHOOL
RAPID-RICH OBJECT SEARCH (ROSE) LAB

2017

CO-SALIENCY BASED VISUAL OBJECT CO-SEGMENTATION AND CO-LOCALIZATION

KOTESWAR RAO JERRIPOTHULA

Interdisciplinary Graduate School
Rapid-Rich Object Search (ROSE) Lab

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2017

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

26/07/2017
. . . .
Date

KOTESWAR RAO JERRIPOTHULA
. . . .
Student Name

Abstract

Automatic foreground segmentation and localization in images or videos are very important and basic problems in computer vision. Due to lacking of sufficient information about the foreground object in a single image or a video, these tasks usually become very difficult. However, if a set of similar images (where foreground objects are of the same category) are provided for joint processing, the job becomes little easier because we can exploit the available commonness clue. Thus, by jointly processing similar images together we provide a kind of weak supervision to the system. Such a task of segmenting out the common foreground visual objects through joint processing of similar images is known as co-segmentation. Similarly, the task of localizing (proving bounding box to) the common foreground visual objects is known as co-localization. Co-segmentation and co-localization tasks have applications in image retrieval, image synthesis, datasets generation, object recognition, video surveillance, action recognition, etc. However, such joint processing brings in new challenges to handle: (i) variation in terms of poses, sub-categories, viewpoints, etc; (ii) complexity in design;(iii) difficulty in parameter setting due to increased number of variables; (iv) the speed; and (v) their futility in some cases compared to single processing. Many existing joint processing methods usually extend the single processing methods and succumb to complicatedly co-labelling the pixels or bounding box proposals. However, co-saliency idea to effectively carry out these tasks have not been well-explored, especially co-saliency generated by fusing raw saliency maps. Co-saliency basically means jointly processed saliency. In this thesis, we present four co-saliency based works: saliency fusion, saliency co-fusion, video co-localization, and object co-skeletonization.

In our saliency fusion idea, we propose to fuse the saliency maps of different images using dense correspondence technique. More importantly, this co-saliency estimation is guided by our proposed quality measurement which helps decide whether the saliency fusion improves

the quality of saliency map or not. This helps us to decide which is better for a particular case: joint or single processing. Idea is that high-quality saliency map should have well-separated foreground and background, also a concentrated foreground.

In our saliency co-fusion idea, to make the system more robust and to avoid heavy dependence on only a single saliency extraction method, we propose to apply multiple existing saliency extraction methods on each image to obtain diverse saliency maps and fuse them by exploiting the inter-image information, which we call saliency co-fusion. Note that while we fused saliency maps of different images in the above saliency fusion idea, we here fuse diverse saliency maps of the same image. It results in much cleaner co-saliency maps.

In our video co-localization idea, in contrast to previous joint frameworks that use bounding box proposals at every frame to attack the problem, we propose to leverage co-saliency activated tracklets to address the challenges of speed and variations. We develop co-saliency maps for few key frames (which we call as activators) only through inter-video commonness, intra-video commonness, and motion saliency. Again, the saliency fusion approach is employed. Object proposals of high objectness and co-saliency scores are then tracked across the short video intervals, between key frames, to build tracklets. The best tube for a video is obtained through tracklet selection from each of these intervals depending upon confidence and smoothness between adjacent tracklets.

Different from object co-segmentation and co-localization, we also explore a new joint processing idea called object co-skeletonization, which is defined as joint skeleton extraction of common objects in a set of semantically similar images. Noting that skeleton can provide good scribbles for segmentation, and skeletonization, in turn, needs good segmentation, we propose to couple co-skeletonization and co-segmentation tasks so that they are well informed of each other, and benefit each other synergistically. This coupled framework also greatly benefits from our co-saliency and fusion ideas.

Acknowledgments

When I first came to NTU, I hardly had any prior research experience except the FYP that I did during my undergraduate studies. I feel extremely grateful towards my supervisors, Prof. Jianfei Cai and Prof. Junsong Yuan, for giving me the opportunity to pursue my PhD under their expert guidance, which helped me a lot to develop gradually and progressively as a researcher. Their timely and precise inputs taught me many important lessons both in the research and my personal life which I will always relish and feel inspired by. I would also like to thank Dr. Lu Jiangbo for his energetic inputs, which inspired me further. I thank my teacher for giving me the right motivation behind why I should do my PhD. His golden words about PhD, “It’s 99% perspiration and only 1% inspiration, despite that what drives it is right motivation", sank into my heart and helped me overcome my struggles during the PhD. Lastly, I would like to thank my parents, mentors and friends for giving me all the encouragement and support that I needed for successfully completing this thesis.

Publications

This is a list of published/submitted works during my Ph.D. candidature.

1. **K. R. Jerripothula**, J. Cai, and J. Yuan, “Object Co-skeletonization with Co-segmentation " in Proceedings of the *IEEE International Conference on Computer Vision and Pattern Recognition* **(CVPR)**, 2017, Honolulu, United States of America.

2. **K. R. Jerripothula**, J. Cai, and J. Yuan, “CATS: Co-saliency Activated Tracklet Selection for Video Co-localization" in Proceedings of the *European Conference on Computer Vision* **(ECCV)**, Springer International Publishing, 2016, Amsterdam, The Netherlands.

3. **K. R. Jerripothula**, J. Cai, and J. Yuan, “Image Co-segmentation *via* Saliency Co-fusion" in *IEEE Transactions on Multimedia* **(T-MM)**, September, 2016.

4. **K. R. Jerripothula**, J. Cai, and J. Yuan, “Group Saliency Propagation for Large Scale and Quick Image Co-segmentation" in Proceedings of the *IEEE International Conference on Image Processing* (ICIP), 2015, Quebec City, Canada. (**Oral**)

5. **K. R. Jerripothula**, J. Cai, and J. Yuan, “QCCE: Quality Constrained Co-saliency Estimation for Common Object Detection" in Proceedings of the *IEEE International Conference on Visual Communication and Image Processing* (VCIP), 2015, Singapore. (**Oral**)

6. **K. R. Jerripothula**, J. Cai, F. Meng, and J. Yuan, “Automatic Image Co-segmentation using Geometric Mean Saliency" in Proceedings of the *IEEE International Conference on Image Processing* (ICIP), 2014, Paris, France. (**Top 10% Paper Award**)

7. **K. R. Jerripothula**, J. Cai, and J. Yuan, “Quality-guided Fusion-based Co-saliency Estimation for Image Co-segmentation and Co-localization", submitted to *IEEE Transactions on Multimedia* (T-MM) [submitted]

8. **K. R. Jerripothula**, J. Cai, and J. Yuan, “Efficient Video Object Co-localization with Co-saliency Activated Tracklets", submitted to *IEEE Transactions on Circuits and Systems for Video Technology* (T-CSVT) [submitted]

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In any image or a video (for e.g. see Figure 1.1), there is always some background or other that gets captured along with foreground which is usually unnecessary. In fact, many of the computer vision and multimedia applications (like recognition, skeletonization, action recognition, etc) require that the foreground is either segmented out or cropped out already. However, when we just have a single image or video, it's difficult to automatically accomplish this due to the lack of sufficient information about the foreground visual object. In recent years, the joint processing tasks like co-segmentation and co-localization have drawn significant interest among researchers, because commonness clue that exists in a set of similar images can compensate for such lack of information. The task of segmenting out the common foreground visual objects through joint processing of similar images is known as co-segmentation. Similarly, the task of localizing (providing bounding boxes) the common foreground visual objects is known as co-localization. With the extra information of commonness, joint processing certainly edges over individual processing.

However, the joint processing brings in new challenges. Even in the similar images, viz. having same semantic category objects, there is always a room for the existence of some variation or other in the terms of species, models, or viewpoints, etc, as shown in the Fig. 1.2. In such a scenario, the approach of joint-labeling of pixels or bounding box proposals becomes quite complicated, and also more numbers of variables/parameters get introduced. With the

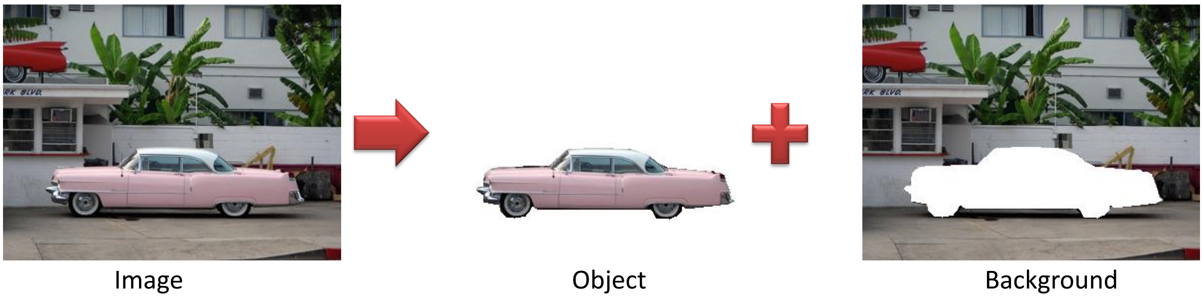


Figure 1.1: Whenever an object (car) is captured in an image, it is unavoidably captured along with its environment, i.e background.

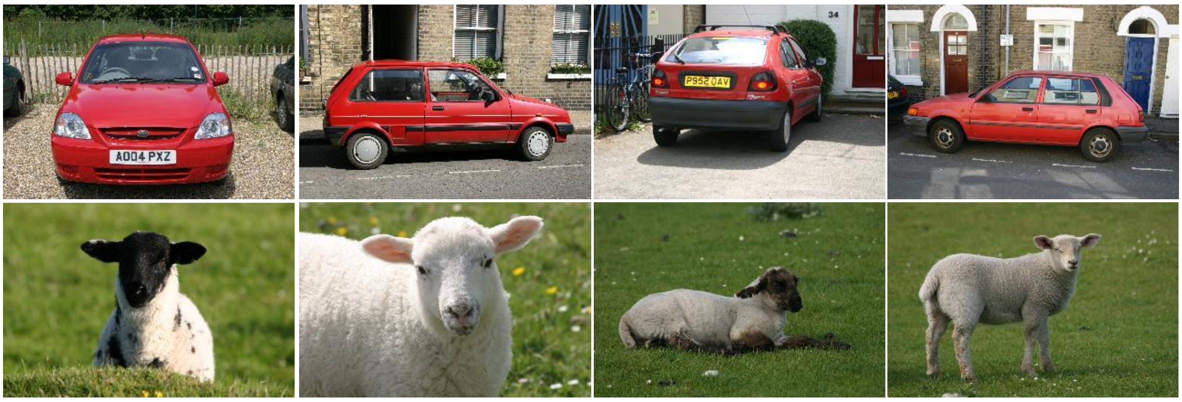


Figure 1.2: Even in the images of same semantic objects, there exists variation.

increased number of parameters, setting them appropriately adds another challenge, especially when facing large and diverse datasets. Also, joint processing of images need not perform better than individual processing always. This certainly raises up the challenge of appropriate selection between joint processing and individual processing on the case basis.

The concept of co-segmentation was first introduced in [77], which used histogram matching to simultaneously segment out the common object from a pair of images. Since then, many co-segmentation algorithms have been proposed in the literature, ranging from early image pair co-segmentation [30][70], multiple image co-segmentation [39][43][48][65], interactive image co-segmentation [7][19][8] to the recent multiple objects co-segmentation [69][56][59][42], multiple group co-segmentation [66], noisy image set co-segmentation [78], large-scale co-segmentation [28][34] and shape alignment targeted co-segmentation [20]. Variety of models have been proposed and improved upon like MRF models [70][13], discriminative clustering models [39][40], diffusion models [43][42], proposal-based models [97][11], etc, trying to solve the problem the problem of co-segmentation. Trying to relax the problem of co-labeling the pixels (in co-segmentation) to co-labelling the bounding box proposals, the problem of co-localization was first introduced by [90], which used saliency prior, similarity and discriminability for bounding boxes to develop a constrained quadratic program based formulation and solved it. Later on, these joint processing tasks have been extended to video domain as well, e.g., [41][45]. Similar to these ideas, idea of co-saliency in terms of jointly developing saliency maps have also been proposed. While some methods develop co-saliency from the scratch like in [25], some methods modify the raw saliency map with the help of repeatedness constraint like in [13].

1.2 Motivation and Contributions

Although effective, existing methods usually become complicated due to co-labeling approach and require parameter tuning, and thus either denying their application for large scale datasets, or some kind of supervised learning is undertaken to learn these parameters. Also, none of these methods accounts for the issue over automatic selection between joint-processing and single processing.

Contrary to the existing methods, we take much simpler approach: develop co-saliency maps by fusing raw saliency maps and carry out single processing. Although the idea of using co-saliency is not new in the context of joint processing, they have not been so well explored for solving co-segmentation and co-localization, especially by fusing raw saliency maps. Let's have a look at car images in Fig. 1.3, note how common object (car) pixels are salient in some and non-salient in others, backgrounds are non-salient in some and salient in others. Taking a specific example, pixels of car's side-door are salient in all but not in the last row. However, we can comfortably assume that (i) common object pixels are generally salient if not in every given image, and (ii) background pixels are generally non-salient if not in every given image. Such an assumption encourages us to link the saliency values of corresponding pixels and develop general saliency values. In fact, these general saliency values can be used to correct the original saliency maps. These co-saliency maps (or general saliency maps) can be readily used to segment or localize the objects because joint processing has already taken place. And

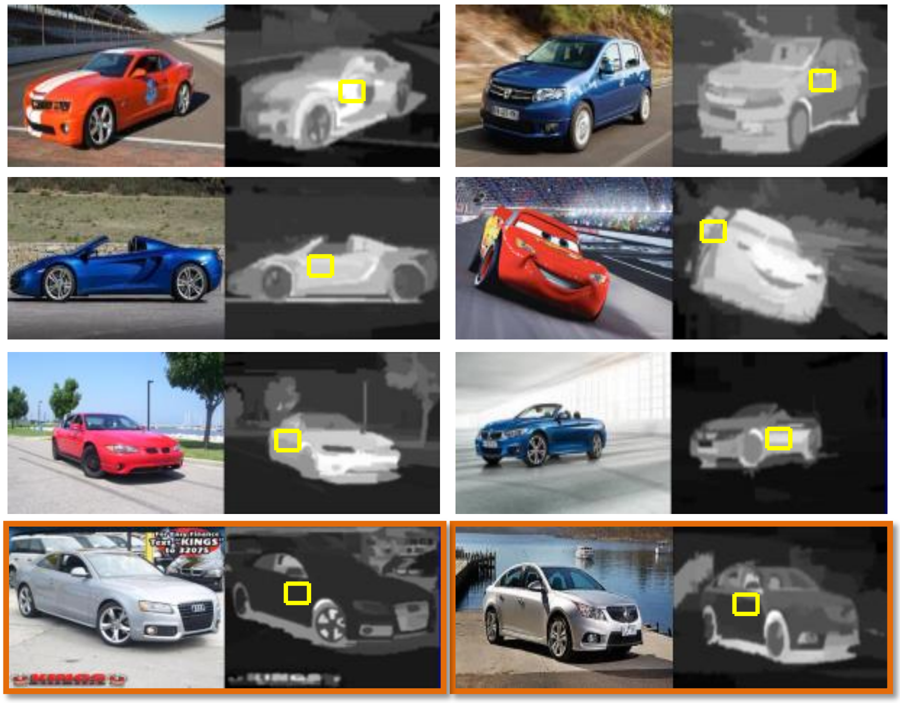


Figure 1.3: Generally, common object parts, say side door, are salient. We can exploit this generality to enhance the saliency of objects in the images where it is not salient such as images in the bottom row.

to perform the joint processing involved here, we do not need to process too many images together, just a few neighboring images are sufficient. This pretty much takes care of variation issues. Moreover, fusion process can be kept quite simple, which reduces the extra burden of parameters.

This thesis presents four types of saliency fusion based joint processing techniques. Initially, a basic idea, fusion of saliency maps across different images was developed. Although good enough for segmentation and localization of common objects, the resultant jointly processed co-saliency maps depends upon a particular saliency extraction process only, i.e. all the aspects of saliency may not be covered. Also, these maps are not so clean and require some regularization causing the loss of details. Therefore, we develop another idea of saliency fusion called saliency co-fusion, where multiple saliency maps (covering multiple fronts) of the same image are fused while reciprocating with other similar images. Although, it does cover several fronts of saliency and provide much cleaner co-saliency maps, it is at the cost of requiring to generate multiple saliency maps, and thus require more pre-computation and memory space. Then, there are also motion saliency maps in addition to the visual saliency maps which can act as object priors, therefore, we extend the idea of our saliency fusion-based joint processing to videos, where we perform joint processing only for few key frames, and then leverage tracklets to extend the localization to other frames. In this manner, we perform video co-localization in an efficient manner. We also introduce a new joint processing task called co-skeletonization, which we couple with co-segmentation in our fusion-based joint processing framework for their mutual benefits. The detailed contributions made by each of these ideas are as discussed below:

- **Saliency Fusion:** First, a saliency fusion based co-saliency approach is proposed where the warped saliency maps of other images are fused with the image's original saliency map. Second, because joint processing need not be helpful always, some quality metrics are designed to compare the co-saliency maps with the saliency maps to monitor if quality improves by such joint processing. Idea is that if it does, co-saliency map can be used, otherwise, the saliency map itself is used. The third contribution is integrating the two ideas and carrying out the joint processing.

- **Saliency Co-fusion:** First, a saliency co-fusion based co-saliency approach is proposed where fusion takes place between multiple saliency maps of the same image but with the joint processing goals (highlighting the common object regions and suppressing the backgrounds), which is called as saliency co-fusion. Second, an optimization framework is proposed to carry out such saliency co-fusion. Third, smoother and pleasing resultant maps are obtained compared to the ones obtained by saliency fusion across the images.

- **Video Co-localization:** First, three different types of co-saliency maps (inter-video, intra-video and motion) are proposed; these are developed only for few key frames at regular intervals. Second, we propose to leverage tracklets, which are reliable for short durations, to avoid unnecessary joint processing of so many frames. The third contribution is the way co-saliency is used to activate and select the optimal tracklets for performing video co-localization eventually.

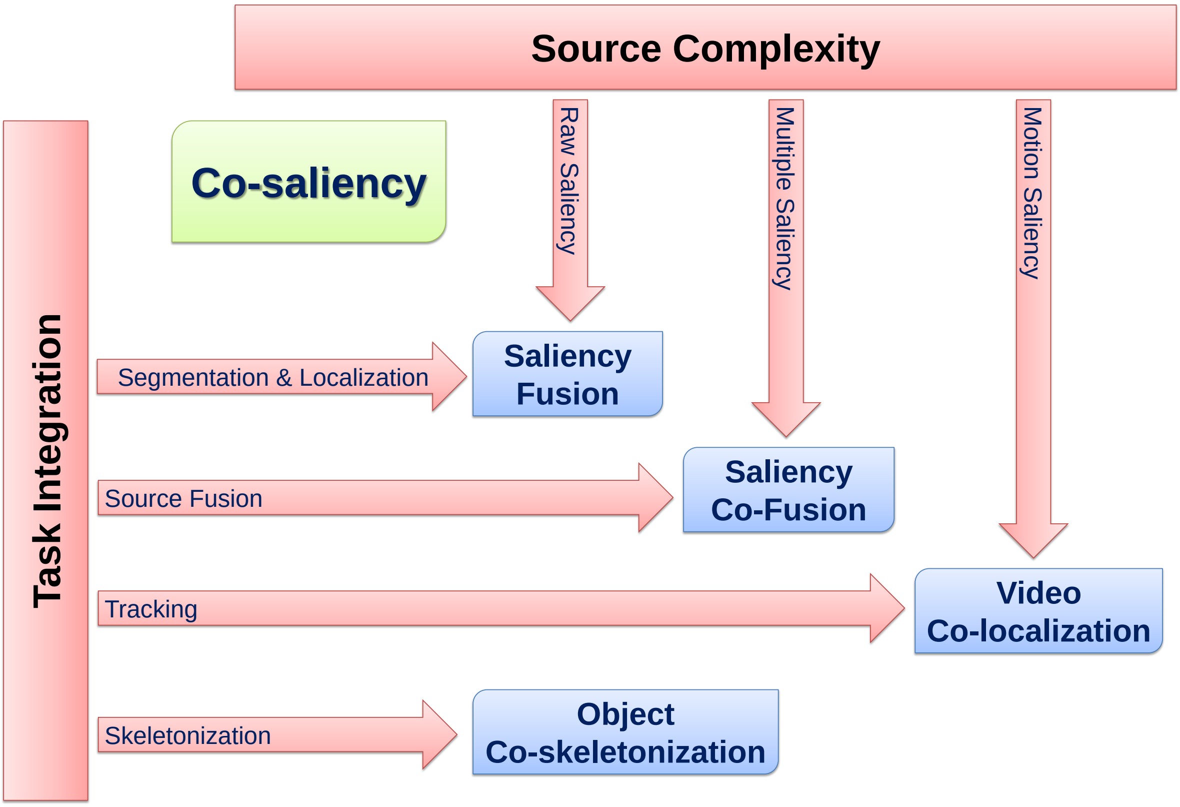


Figure 1.4: Our four works are related by the central theme of Co-saliency and how the source complexity (for co-saliency generation) and the task integration (for co-saliency enhancement/better performance) are varied.

- **Co-skeletonization:** First, the joint processing task of co-skeletonization is introduced where skeleton masks (derived from saliency maps initially) are fused to develop joint skeleton priors. Second, the way we couple it with the co-segmentation task. This is done not only because skeleton requires shape information but also because they can help each other synergistically. While good skeletons can give good seeds for segmentation, good segmentation can yield good skeleton in return. Third, the skeleton pruning process is improved. A skeleton needs to be simple in spite of not-so-clean segmentations, which we are likely to get due to the joint processing of several images. Fourth, a new dataset named CO-SKEL dataset is developed for co-skeletonization benchmarking.

In summary, the central theme of proposed methods is co-saliency. Our four works are based on how the source complexity (for co-saliency generation) and the task integration is varied (for co-saliency enhancement/better performance) as shown in Fig. 1.4. In the saliency fusion work, co-saliency source is basically the raw saliency and it is integrated with other joint processing tasks such as co-segmentation and co-localization. The saliency co-fusion idea benefits from multiple sources while undertaking the task of fusing saliency information from multiple sources. The video co-localization idea takes the advantage of available motion saliency and tracking technique. Object co-skeletonization idea is similar to the saliency fusion idea, but it is integrated with an extra task of skeletonization.

1.3 Thesis Structure

This thesis contains seven chapters and the rest of the thesis is organized as follows. The structure of this thesis is shown in Figure 1.5 along with the details of our main chapters. In Chapter 2, we investigate previous works related to co-saliency, co-segmentation, co-localization and co-skeletonization. In Chapter 3, the idea of quality-guided saliency fusion is proposed. In Chapter 4, the idea of saliency co-fusion is presented. In Chapter 5, our extension of saliency fusion idea to videos is presented where co-saliency priors are generated at regular intervals and are used to activate and select tracklets for eventual video co-localization. In Chapter 6, a new joint processing task called co-skeletonization is proposed, and it is discussed how it is coupled with the co-segmentation task for their mutual benefits. In Chapter 7, this thesis is concluded and some pointers to future works are given.

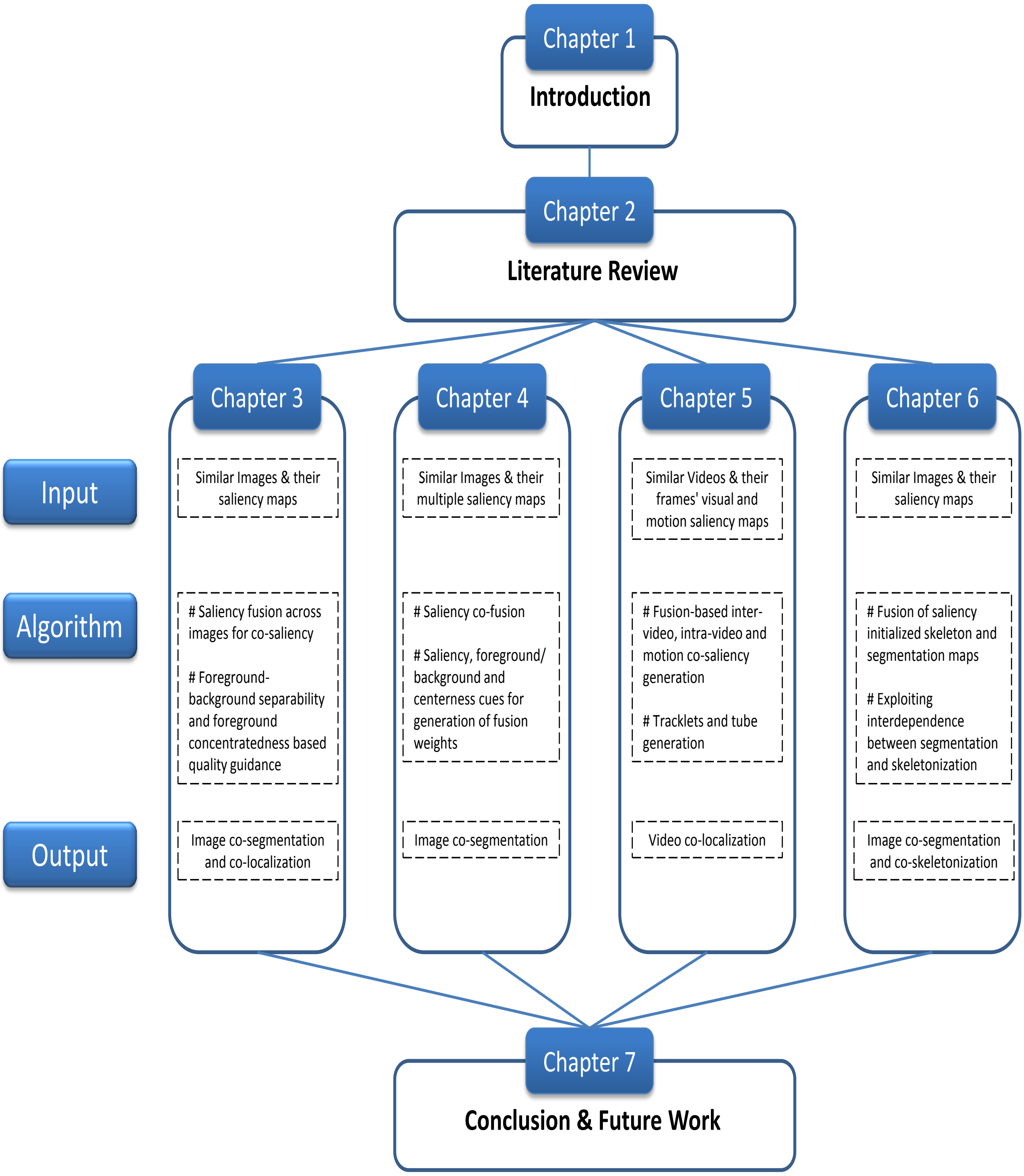

Chapter 1
Introduction
Chapter 2
Literature Review
Chapter 3
Chapter 4
Chapter 5
Chapter 6
Input
Algorithm
Output
Similar Images & their saliency maps
Similar Images & their multiple saliency maps
Similar Videos & their frames' visual and motion saliency maps
Similar Images & their saliency maps
# Saliency fusion across images for co-saliency
# Foreground-background separability and foreground concentratedness based quality guidance
# Saliency co-fusion
# Saliency, foreground/ background and centerness cues for generation of fusion weights
# Fusion-based inter-video, intra-video and motion co-saliency generation
# Tracklets and tube generation
# Fusion of saliency initialized skeleton and segmentation maps
# Exploiting interdependence between segmentation and skeletonization
Image co-segmentation and co-localization
Image co-segmentation
Video co-localization
Image co-segmentation and co-skeletonization
Chapter 7
Conclusion & Future Work

Figure 1.5: The structure of this thesis

Chapter 2

Literature Review

In this chapter, we conduct the literature survey of four topics related to our proposed methods, namely co-saliency (jointly estimating saliency in a given set of images), co-segmentation (jointly segmenting the common objects in a given set of images), co-localization (jointly localizing the common objects in a given set of images), co-skeletonization (jointly skeletonizing the common objects in a given set of images).

2.1 Co-saliency

Co-saliency typically refers to the common saliency existing in a set of images containing similar objects. The term co-saliency was first coined in [32] in the sense of what is unique in a set of similar images and the concept was later linked to extracting common saliency, which is very useful for many practical applications [14][50]. The visual saliency phenomenon basically relates to something that is distinct and attracts human eyes. Similar to visual saliency, the idea behind visual co-saliency is that when we see similar objects across images repeatedly, they naturally attract our eyes. This phenomenon is called repeatedness, and when this phenomenon is incorporated into usual saliency definition, it becomes co-saliency. In fact, the co-saliency object priors developed in [13] is defined as the following:

$$\text{Co-saliency} = \text{Saliency} \times \text{ Repeatedness} \tag{2.1}$$

where repeatedness was calculated using SIFT distances and sigmoid function, and the resultant co-saliency priors were then used for efficient co-segmentation by [13]. The approach is transforming the existing saliency maps into co-saliency maps. Then, there are other methods like [25], where co-saliency is generated from the scratch by jointly considering multiple images. It develops three cluster-based cues: contrast cue (visual feature uniqueness), spatial cue (central-bias) and corresponding cue (repeatedness) using the color and texture features, and multiply them eventually. Although the method is simple, general, efficient and effective, its application is limited to the images of the same object or very similar objects captured at different viewpoints or instances, because of its reliance on color and texture features, which could be misleading in handling image-sets with huge intra-class variation. Then, there are deep learning approaches such as [107], which introduces deep intra-group semantic information and wide cross-group heterogeneousness information for co-saliency detection. In this way, it can capture the concept-level properties of the co-salient objects and suppress the common backgrounds in the image group. A systematic review of all the main co-saliency methods developed so far is available in [106]. The main difference between existing co-saliency models and ours is the idea of fusion. While existing method either try to modify the existing saliency maps or create new ones from scratch, we try to fuse the existing saliency maps.

2.2 Co-segmentation

The co-saliency topic discussed above outputs only a prior, meaning a continuous map, but co-segmentation is supposed to give binary or multi-label masks. Many co-segmentation algorithms have been proposed in the literature. Early approaches [77][30][70] focused on segmenting just a pair of images containing one common object. It was later extended to deal with multiple images containing one common object with more effective or more efficient models enforcing inter-image consistency [96][39][43][79][71][105]. However, there are also some algorithms that have been designed for segmenting multiple common foregrounds from a given image set [69][56][59][42], where the best performers make use of supervised information. Then, there are some interactive co-segmentation approaches [7][19][8] as well. However, the existing works can be roughly classified into the following based on the frameworks they employ:

2.2.1 MRF Models

Markov Random Field (MRF) model is the most widely used model in image co-segmentation where the foreground similarity constraint is added to traditional MRF segmentation model for a single image. Usually, a new global term is added to the energy function which accounts for foreground similarity. Therefore, the total energy E_m becomes

$E_m = E_u + E_p + E_g$

where E_u and E_p are unary and pairwise terms for single image MRF Segmentation model and the new global term E_g accounts for foreground similarity. The co-segmentation performance depends a lot upon this new global term, therefore it has been designed in several ways to increase the accuracy and simplify energy minimization. For instance, [77] initially matched the appearance histograms by using L1-norm for foreground similarity and Trust Region Graph Cuts (TRGC) for energy minimization, then [70] used L2-norm for foreground similarity and pseudo-boolean optimization for energy minimization, later [30] proposed reward based global term resulting in simple minimization through max flow algorithm, and then [13] proposed another histogram based global term to overcome difficulty in energy minimization. These models generally become complicated and would need the same objects to be present in the images due to the foreground similarity constraint, and therefore, will not be able to handle well when intra-class variations or pose variations are present. However, by accounting for saliency and matching in [78] and [13], this problem was tackled, but parameter tuning becomes essential for effective results using such methods.

2.2.2 Discriminative Clustering Models

“Discriminative clustering was first introduced by [101] and relies explicitly on supervised classification techniques such as the support vector machine (SVM) to perform unsupervised clustering: it aims at assigning labels to the data so that if an SVM were run with these labels, the resulting classifier would separate the data with high margin."[39]. In [39], this discriminative clustering was employed for maximizing the separability between foreground and background, and normalized laplacian was used for spatial consistency to tackle the co-segmentation problem. The optimization problem is solved using efficient low-rank optimization after convex relaxation. In [40], this model was extended to perform multi-class co-segmentation, i.e. same multiple objects frequently appear in a set of images, and an image may contain more than one such objects. These models take all the images together for joint processing and get troubled by the existing intra-class variations of objects, as it gets revealed in [78] when applied on challenging MSRC and iCoseg datasets.

2.2.3 Heat Diffusion Models

In [43], an anisotropic heat diffusion model was used for co-segmentation where heat diffusion represents similarity constraint, and the framework could segment images into multiple regions and handle large scale image group. The algorithm can be summarized as “Given a system under heat diffusion and finite K heat sources, where should one place all the sources in order to maximize the temperature of the system?"[43]. In the segmentation context, it translates to finding the K segment centers that can maximize the segmentation confidence of every pixel in the image. This model was extended by alternating foreground estimation and region assignments to be able to segment multiple foregrounds in [42]. These models also have similar problems as the previous ones.

2.2.4 Proposal based Models

Most of the above methods were mostly seen in the perspective of matching between images. Although they handle object perspective inherently, what is matching may not be the object at all when seen from the perspective of the single image. As mentioned earlier, there may be lots of background stuff like the sky, grass, etc, which might also be matching. So, [97] proposed to introduce the term “objectness" explicitly that exploits the object-like segmentations proposals from [11] and used more features that are important for single image segmentation. They learn to measure similarity across images to choose the correct object like segmentation proposal by employing random forest regressor helped by few groundtruth examples. A similar approach of selection of segmentation proposals was addressed in [68] and was solved using shortest path algorithm. This thesis derives some inspiration from these models, i.e. in terms of using the idea of selecting the object-like proposals and weed out unnecessary ones.

2.2.5 Segmentation transfer or Interactive Models

In contrast to unsupervised co-segmentation models described above, there are some supervised co-segmentation models as well where some of the images might be already having ground truth masks or human interaction is integrated into the process. Specifically, for large scale foreground extraction, where we can't keep adjusting the parameters as the above methods would require, [44] showed how they could segment half-million images using the transfer of human annotated segmentation masks. So far, this method is state-of-the-art in applying co-segmentation on Imagenet [21] dataset. Previously as well, there has been such attempts like in [43] but the evaluation was very limited as they used bounding boxes for evaluation whereas [44] uses proper segmentation masks for subsets of images. [12] is another co-segmentation method that is highly scalable to perform foreground extraction in large-scale datasets where they improve upon the results obtained by GrabCut [76] by finding the optimal hyperplane that can separate foregrounds and backgrounds in the feature space. A few interactive co-segmentation approaches [7][19][8] have also been proposed, where users can give scribbles for one or a small number of the images. Thus, the extracted prior information is then used to influence the segmentation of the entire image set. [7] proposes an automatic recommendation system as well which recommends where the user should scribble next. Recently, [23] proposed an active segmentation propagation approach where they could actively determine which are the images that need human annotation at any stage, and then propagate the foreground estimates to unlabelled images. They prioritize those images which are uncertain and influential while selected ones being mutually diverse. The difference with other interactive methods is in prioritizing the images for human annotations compared to choosing any random one.

2.3 Co-localization

Image co-localization is also similar to co-segmentation in terms of the idea, i.e. using multiple images with output as a bounding box around the object instead of object segment. This has been introduced by [90] along with the way to handle noisy datasets, where it is able to avoid assigning the bounding box if the image does not contain the common object. The performance of this method has been further improved in [41]. However, it's a joint framework and optimizes over all the images, whereas we explore inter-image information via co-saliency first and then perform co-saliency based localization on individual images. Another work [17] proposes a generic co-localization where objects across the images need not be common. Slightly different from the co-localization, there are some bounding-box propagation algorithms [27][95] where some images already have bounding boxes and they are utilized to localize objects in unannotated images, it is like a supervised scenario.

Video Co-localization is a task of jointly localizing the shared object in a set of videos. The recent work of [41] and [75] proposed a joint framework to locate common objects across videos. In [41], it used quadratic programming framework to co-select bounding box proposals in all the frames in all the videos together. While in [75], it formed candidate tubes and co-selected tubes across the videos to locate the shared object. Handling inter-video, intra-video variations and temporal consistency simultaneously often become a difficult task for such joint frameworks. This is especially so when extremes such as bounding box in a frame or candidate tube for entire video is chosen as processing unit. Recently, [45] proposed an approach of developing foreground confidence for bounding boxes and selecting bounding boxes while maintaining the temporal consistency. The presence of noisy bounding box proposals mandates taking an iterative approach in [45]. All these methods [41][45][75] assumed the object is present in all the frames in all the videos, but [98] overcame such an assumption through providing few labels of relevant frames and irrelevant frames to effectively guide the object discovery.

2.4 Co-skeletonization

To the best of our knowledge, there has been hardly any prior work on co-skeletonization, but there are some works on shape clustering [85] where skeletons have been jointly processed to group different shapes, which might be useful for good neighborhood retrieval in our context. Also, there has been a good amount of research on skeletonization, which is for a single image. The research on skeletonization can be divided into three parts. First, there are some algorithms [18, 80, 82] which can perform skeletonization if the segmentation of an object is provided. Generally, these algorithms are quite sensitive to the distortions of the given shape.

However, it has been improved greatly in the recent methods such as [84]. Second, there are also some traditional image processing methods [54, 104, 108] which can generate skeletons by exploiting gradient intensity maps. They generate skeletons even for background stuff like sky, sea, etc. Thus, they usually need some kind of prior to suppress such skeletons. Third, there are also some supervised learning based methods which require some groundtruth skeletons for learning. It includes both traditional machine learning based methods [83, 91] and the recent deep learning based methods [86, 100]. The performance of the traditional machine learning based methods has not been satisfactory due to its limited feature learning capability. On the other hand, the recent deep learning based methods have made great progress in the skeletonization process as reported in [86], but at the cost of requiring complex training process.

Chapter 3

Quilty-guided Fusion-based Co-saliency Estimation for Image Co-segmentation and Co-localization

Foreground segmentation or localization is a very useful and important step for many vision and multimedia applications such as recognition and streaming, since it separates the object of interest from the background and thus facilitates more efficient subsequent processing or understanding. When dealing with only a single image, visual saliency has been a common cue used for highlighting the foreground. However, single-image saliency has obtained limited success when facing images with cluttered backgrounds, or images where the foreground has similar attributes as the background, which cause object of interest to be less salient. Recognizing the limitations of individual processing, in recent years various joint processing works such as co-saliency [14][50][13][25][48], co-segmentation [97][78][35][65], co-localization [90][37], knowledge transfer [28][27][95] have been proposed, and have been demonstrated quite effective in extracting foregrounds in a batch mode. The basic idea of all these works is to exploit the commonness across a set of images that contain some common object, which gives inter-image prior information about the common object, a clear advantage that certainly lacks in the individual processing.

Despite such an advantage, the existing joint processing algorithms also bring in new challenges. 1) As shown in [97][78], joint processing of images might not perform better than individual processing. The recently proposed video co-localization work [41] also cannot perform better than the individually processing [74]. This certainly raises up the question: *Given a set of images for foreground segmentation or localization, should we process them jointly or individually?* 2) Due to the way of co-labeling pixels [78] or co-selection of bounding boxes [90] in a set of images, most of the existing high-performance joint processing algorithms are usually complicated with large numbers of variables, which unavoidably have the scalability issue. 3) For effective co-segmentation or co-localization, the existing joint processing algorithms usually require tuning quite a few parameters, which further increases the complexity, especially when facing large and diverse datasets.

To address the above challenges, in this chapter we propose a co-saliency framework, where we explore inter-image information via co-saliency and then perform co-saliency based segmentation or localization on individual images. In this way, we avoid the scalability issue of directly performing co-labelling or co-localization on multiple images simultaneously. At the heart of proposed co-saliency framework are two key components: saliency quality measurement and fusion based co-saliency.

Quality measurement: In the first component, we propose a metric to measure and compare the quality of each individual saliency map with that of its corresponding co-saliency map, so as to answer the first challenge, i.e. joint processing or not. Our quality metric is developed based on two empirical observations: 1) a better saliency map should have a better separation between the foreground and the background; 2) a better saliency map should have a better foreground concentration, i.e. preferring the foreground to be a concentrated saliency region. Figure 3.1 gives two examples, where the top example (cartoon) highlights the object region better in the individually processed saliency map [16] compared to the jointly processed co-saliency map [49], while in the bottom example (human), the co-saliency map looks better than the individually processed saliency map. For both examples in Figure 3.1, our proposed metric generates the appropriate quality scores at the bottom-right of each saliency map. Note that there has been a work [60] comparing different saliency maps of an image using different saliency detection methods in a supervised manner, while here we compare an individual saliency map with its co-saliency map and we do it in a completely unsupervised way.

Fusion based co-saliency: The second component of our framework is to deal with the issue that for each image, how to fuse its own saliency map with the saliency information

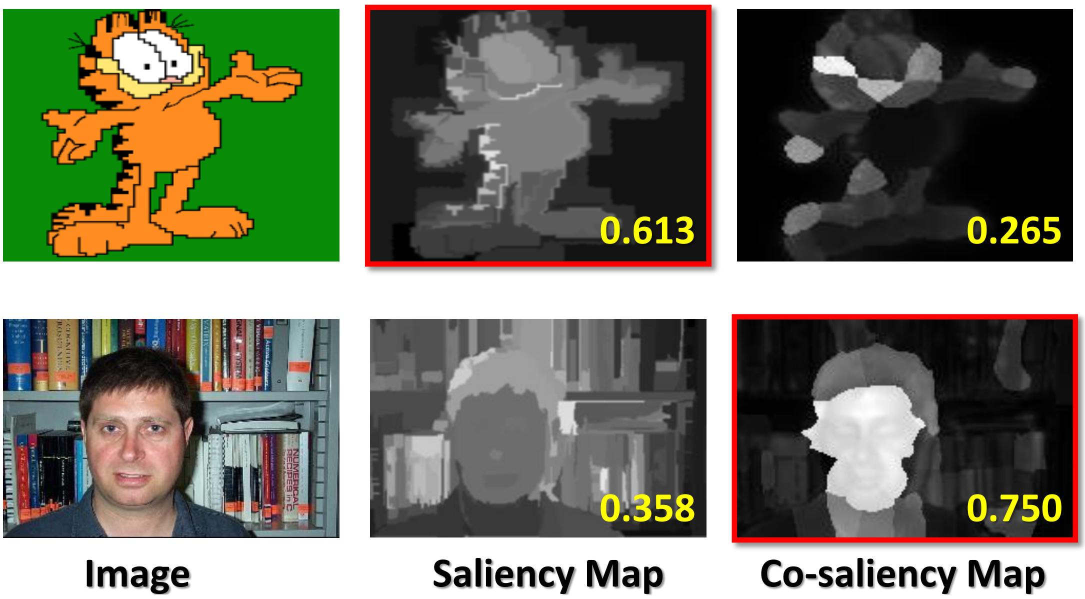


Figure 3.1: The desired saliency map that highlights the object could either be the original saliency map itself (e.g. in the case of cartoon with clear background) or the jointly processed co-saliency map (in the case of human with complicated background). Our quality measurement gives appropriate scores (shown at the bottom-right of each saliency map) to be able to select the right one (red-bordered).

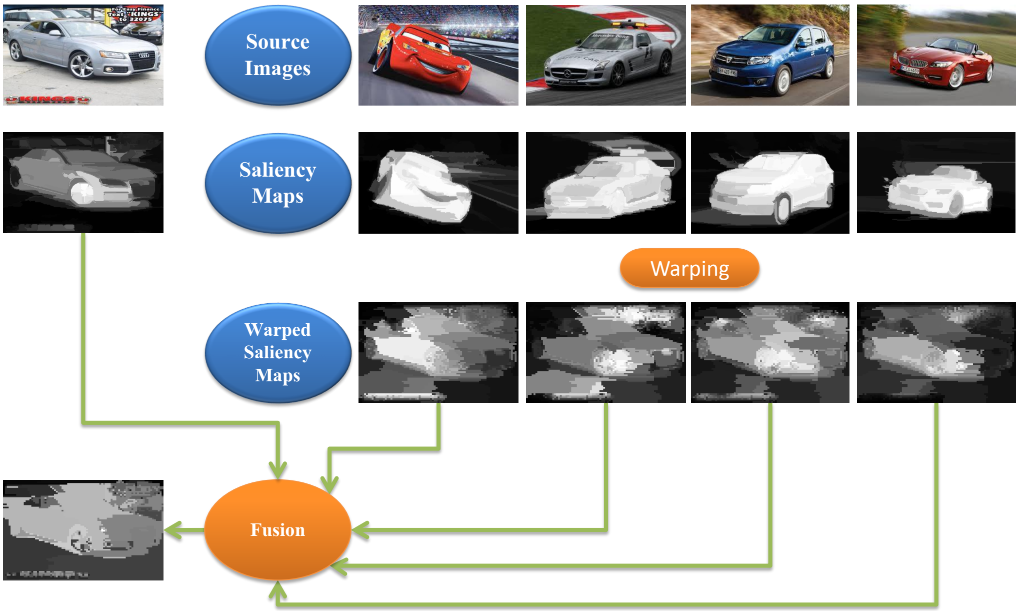


Figure 3.2: An example to show that through the joint processing of co-saliency via warping and fusion, the images on the right which have salient common objects (car) could render help to the first image where the common object (car) has only weak saliency.

from other images so as to boost up the common object saliency while suppressing the background saliency. Our basic idea is to make use of the existing techniques on dense correspondences [55] to align individual object pixels for saliency fusion. Figure 3.2 illustrates the proposed joint process for generating co-saliency map. In particular, for one image, the individual saliency maps of its neighbors are warped to align with its own saliency map and then all the aligned saliency maps are fused together. The underlying assumption here is that the common object or its parts are salient in general, if not in every image.

In the proposed co-saliency framework, images iteratively interact with one another to generate fused saliency maps, which can update the respective saliency maps only if they are of higher quality.

This chapter makes the following major contributions: 1) designing a metric for saliency quality measurement to compare the individually processed priors with those obtained by joint processing; 2) developing a simple saliency fusion based co-saliency estimation method for overcoming the complexity and the parameter setting challenges; 3) achieving good results comparable to state-of-the-arts in the applications of foreground segmentation and localization on several benchmark datasets including the large-scale dataset, ImageNet. Also, this framework can be easily modified to cope with other scenarios such as when there are some groudtruth segmentation / localization maps available or reducing the complexity when dealing with large scale datasets.

3.1 Proposed Method

In this section, we first discuss our objective and the proposed solution, second the quality measurement system, third how images interact, fourth more efficient way of interacting, and finally applications.

3.1.1 Objective and Proposed Solution

Let $\mathbf{I} = \{I_1, I_2, \cdots, I_m\}$ be an image-set containing m similar images. Denote set of their corresponding saliency maps as $\mathbf{S} = \{S_1, S_2, \cdots, S_m\}$. Functions $\phi(\cdot)$ and $\psi(\cdot)$ denote quality functions for the separation measure (between foreground and background) and the concentration measure (of foreground), respectively. Perceiving the output scores of these functions like probabilities, we define the total quality of any saliency map S_i as the product of these two measures, i.e. $\phi(S_i)\psi(S_i)$, following the multiplication rule of probability to account for both the measures simultaneously. Addition gives more a sense of 'either of them is fine', therefore multiplication is preferred. In the pursuit of common object discovery through interaction, we assume that higher quality saliency maps are better. Therefore, we define our objective as

$$\begin{aligned} \mathbf{S}^* &= \arg\max_{\mathbf{S}} \sum_{i=1}^{m} \phi(S_i)\psi(S_i) \\ s.t.\ S_i &\in \{S_i^k | k = 1, \cdots, K\}, \end{aligned} \tag{3.1}$$

where we want to achieve saliency map set $\mathbf{S}^*$ such that the total quality of comprising saliency maps is maximum, and S_i can be any saliency map of an image ranging from the original saliency map to the saliency map obtained after K interactions, where K is set as 5 by default. So, if S_i^* is the highest quality saliency map in set $\{S_i^k | k = 1, \cdots, K\}$, $\mathbf{S}^* = \{S_1^*, \cdots, S_m^*\}$, i.e. corresponding set of highest quality saliency maps of image-set $\mathbf{I}$. During the interaction process, saliency maps from other similar images fuse together with the saliency map of each image to develop its fused saliency map. Denote $\mathbf{F} = \{F_1, F_2, \cdots, F_m\}$ as the set of such fused saliency maps resulted by such interaction of similar images.

We propose the following approach to achieve our objective. After the interaction, if the quality of saliency map improves by the fusion process, then only corresponding fused saliency maps can update the current saliency map. Otherwise, current saliency map is considered as the final one. In this manner, total saliency quality of set increases progressively. Different images may obtain their final saliency maps at different iterations. In order to track each image and avoid further fusion for them after obtaining their final saliency maps, we define corresponding break variable ρ_i (set as 0 initially), which gets triggered to 1 at such an occurrence for image I_i. Once ρ_i gets triggered to 1, it cannot be changed. Figure 3.3 depicts the flowchart for this. However, in the supervised scenario, saliency maps of images having annotations are replaced with the annotations, and their ρ_i is triggered right in the beginning.

Therefore, saliency map S_i^k, fused saliency map F_i^k at k^{th} iteration, and ρ_i help in deter-

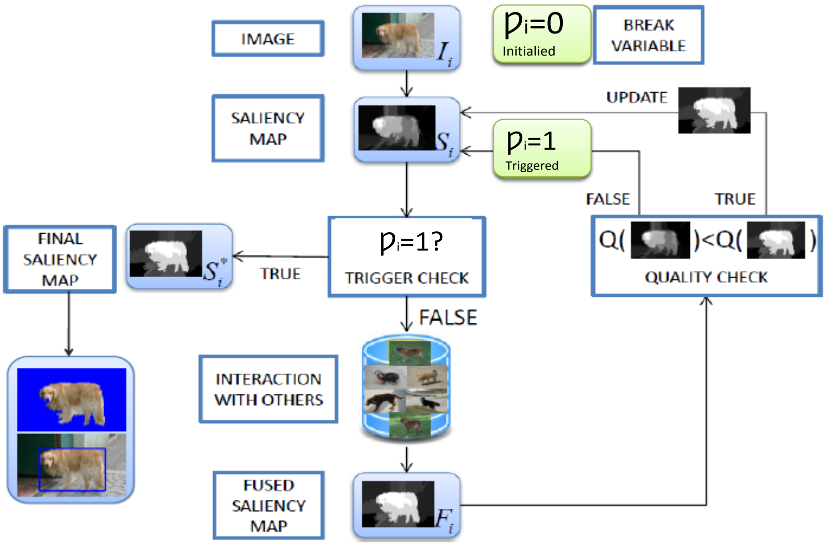


Figure 3.3: Flowchart of the proposed method (for an image I_i): Saliency map S_i is iteratively updated by the fused saliency map F_i as long as the fused saliency map F_i is of higher quality. Drop in the quality (Q) triggers the break variable ρ_i to stop any further updates for the image.

mining S_i^{k+1} (saliency map at next iteration) in the following manner:

$$S_i^{k+1} = \begin{cases} S_i^k, & \text{if } \rho_i = 1; \\ F_i^k, & \text{if } \rho_i = 0 \text{ and } \phi(F_i^k)\psi(F_i^k) > \phi(S_i^k)\psi(S_i^k); \\ S_i^k, & \text{if } \rho_i = 0 \text{ and } \phi(F_i^k)\psi(F_i^k) < \phi(S_i^k)\psi(S_i^k), \end{cases} \tag{3.2}$$

where the first case denotes that an image has already achieved its final saliency map and there is no need for update. The second case denotes that image has not yet achieved its final saliency map, and since the quality has improved by fusion, fused saliency map updates the current saliency map. The third case denotes that although image has not yet achieved its high quality saliency map, but since quality has decreased by fusion, there is no need for update and current saliency map is taken as final one. And it is the third case that triggers ρ_i.

Since we ensure that no way a lower quality fused saliency map can update the current saliency map, total saliency quality of the set $\mathbf{S}$ therefore can only get higher after any given iteration, and algorithm eventually stops when either $\forall \rho_i = 1$ or $k = K$. At this point, we have our $\mathbf{S}^*$.

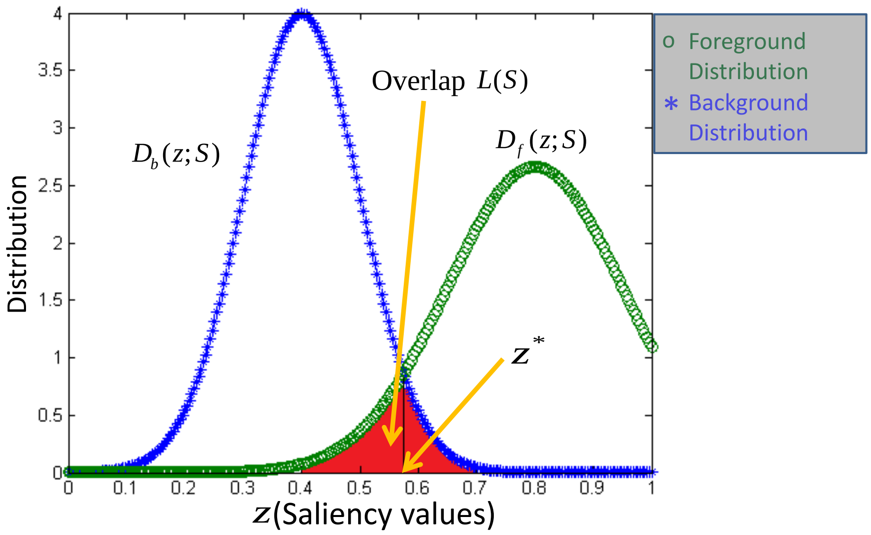


Figure 3.4: Separation measure (ϕ) of quality of saliency map is measured using overlap of estimated likelihood distributions of the two classes: Foreground and Background

3.1.2 Quality Measurement System

In this section, we propose two measures for determining the quality of any given saliency map S: (i) separation measure, which measures separation between foreground and background; and (ii) concentration measure, which measures how concentrated foreground pixels are. In order to assign likelihoods (foreground or background), we apply Otsu's threshold on S.

3.1.2.1 Separation Measure (ϕ)

A high quality saliency map should have well-separated foreground and background likelihoods. Assuming distributions of these likelihoods to be of Gaussian in nature, we attempt to measure separation between the two. Let $\mu_f(S)$, $\mu_b(S)$, $\sigma_f(S)$, and $\sigma_b(S)$ denote foreground mean, background mean, foreground standard deviation, and background standard deviation, respectively, computed based on the two likelihood distributions. Lets denote $D_f(z;S)$ and $D_b(z;S)$ as foreground and background Gaussian distributions, respectively, where z takes saliency value ranging between 0 and 1. Specifically,

$$D_f(z;S) = \frac{\mathrm{e}^{-(\frac{z-\mu_f(S)}{\sigma_f(S)})^2}}{\sigma_f(S)\sqrt{2\pi}} \text{ and } D_b(z;S) = \frac{\mathrm{e}^{-(\frac{z-\mu_b(S)}{\sigma_b(S)})^2}}{\sigma_b(S)\sqrt{2\pi}}, \tag{3.3}$$

as plotted in the Figure 3.4 for an example. It is clear that the less the two distributions overlap with each other, the better the saliency map is, i.e., the foreground and background are more likely to be separable. In order to calculate such overlap, it is needed to figure out the intersecting point z^* (see Figure 3.4). It can be obtained by equating the two functions, i.e. $D_f(z;S) = D_b(z;S)$, which finally leads to

$$\begin{aligned} & z^2\Big(\frac{1}{\sigma_b^2} - \frac{1}{\sigma_f^2}\Big) - 2z\Big(\frac{\mu_b}{\sigma_b^2} - \frac{\mu_f}{\sigma_f^2}\Big) \\ & + \frac{\mu_b^2}{\sigma_b^2} - \frac{\mu_f^2}{\sigma_f^2} + 2\log\big(\frac{\sigma_b}{\sigma_f}\big) = 0. \end{aligned} \tag{3.4}$$

Note that we have omitted expressing "(S)" along with the means and variances for clarity. When we solve the above quadratic equation, we get

$$\begin{aligned} z^* = \frac{\mu_b\sigma_f^2 - \mu_f\sigma_b^2}{\sigma_f^2 - \sigma_b^2} \pm \frac{\sigma_f\sigma_b}{\sigma_f^2 - \sigma_b^2} \times \Bigg(& \big(\mu_f - \mu_b\big)^2 - \\ & 2\Big(\sigma_f^2 - \sigma_b^2\Big)\Big(\log\big(\sigma_b\big) - \log\big(\sigma_f\big)\Big)\Bigg)^{\frac{1}{2}}. \end{aligned} \tag{3.5}$$

Having obtained z^*, overlap $L(S)$ can now be computed as

$$L(S) = \int_{z=0}^{z=z^*} D_f(z;S)\,\mathrm{d}z + \int_{z=z^*}^{z=1} D_b(z;S)\,\mathrm{d}z\,. \tag{3.6}$$

And finally, separation measure ϕ for saliency map S is calculated as

$$\phi(S) = \frac{1}{1 + \log_{10}\big(1 + \gamma\,L(S)\big)}. \tag{3.7}$$

where γ is set as number of bins used for representing the two distributions. In Figure 3.5, we show a set of images with their saliency maps and separation measures. It can be seen that

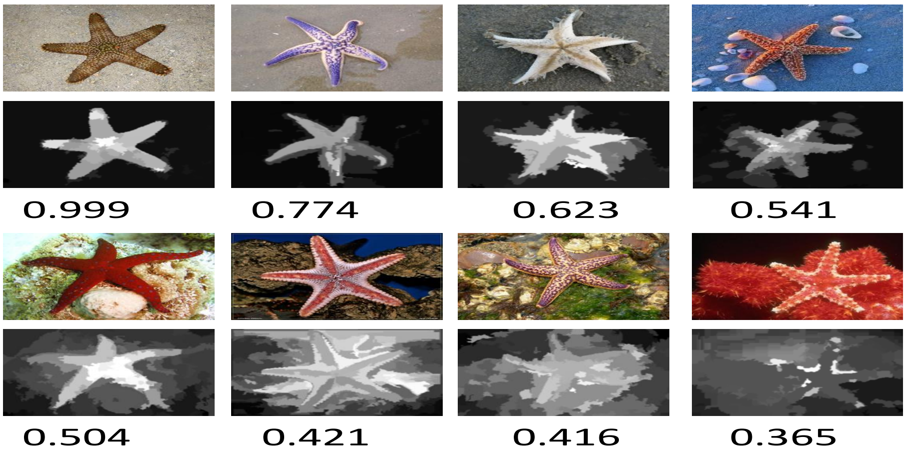


Figure 3.5: Sample Images with their saliency maps and separation measures (ϕ) of quality. Saliency maps with low scores fail to highlight the starfish effectively.

saliency maps become unfit to highlight starfish as separation measure decreases from top-left to the bottom-right.

3.1.2.2 Concentration measure (ψ)

A high-quality saliency map should also have concentrated foreground pixels. Often they get distributed into multiple object components spatially. Ideally, there should be one largest object component and other components (if any) will disperse from that component. Bigger the contribution of this largest component to the foreground, higher will be the concentration of foreground. At the same time, lesser the dispersion of foreground into several object components, higher will be the foreground concentration again. Let $\mathbf{O}(S) = \{O_1(S), O_2(S), \cdots, O_{|\mathbf{O}(S)|}(S)\}$ denote set of these object components. Contribution $C_u(S)$ of $O_u(S)$ towards the total foreground is measured as

$$C_u(S) = \frac{\big[O_u(S)\big]}{\sum\limits_{u=1}^{|\mathbf{O}(S)|} \big[O_u(S)\big]}\,, \tag{3.8}$$

where $\big[\,\cdot\,\big]$ denotes area of $O_u(S)$ and $|\cdot|$ denotes cardinality. Essentially, it is the fraction of the total foreground area covered by the object component. Now, if $u^* = \arg\max\limits_{u} C_u(S)$, then

Table 3.1: Illustration of our concentration measure ψ by varying the $\mathbf{O}(S)$. Note: Numeric values here mean areas of the comprising object components and total foreground area is 100. It can be seen that it decreases as largest component's contribution decreases and number of object components increase.

$\mathbf{O}(S)$	$\psi(S)$
$\{100\}$	1
$\{90, 10\}$	0.95
$\{90, 4, 3, 2, 1\}$	0.92
$\{80, 10, 10\}$	0.867
$\{50, 50\}$	0.75
$\{50, 30, 20\}$	0.667
$\{25, 25, 25, 25\}$	0.438

concentration measure ψ for saliency map S is calculated by

$$\psi(S) = C_{u^*}(S) + \big(1 - C_{u^*}(S)\big)\frac{1}{|\mathbf{O}(S)|}, \tag{3.9}$$

where first term measures contribution made by the largest component and second term measures lowness of dispersion in the foreground by calculating the reciprocal of the total number of components. These two terms are adaptively balanced by the sum of remaining contributions, i.e. $1 - C_{u^*}(S)$. This ensures that concentration measure always lies between $C_{u^*}(S)$ and 1. In Table 3.1, we illustrate how a saliency map S having 100 pixels with foreground likelihood measures while varying its object components set $\mathbf{O}(S)$. It can be observed how concentration measure ψ decreases from top to bottom as the largest component's contribution decreases. And it also decreases with the increasing dispersion of the foreground. For instance, the third set shows lower concentration value than the second one because of higher dispersion, although both have same largest object component's contribution.

3.1.3 Interaction

Images interact with each other hoping for saliency quality improvement. Our interaction process consists of 3 steps: grouping, saliency warping, and saliency fusion as shown in Figure 3.6.

3.1.3.1 Grouping

Considering intra-class variation that can exist in terms of the viewpoint, size, color, location etc, of the common objects, we divide images into a number of image groups so that images within the same group have somewhat similar appearances. Specifically, weighted GIST descriptor [72] (weighted by saliency map following [78]) is used to represent each image. We use k-means clustering for this grouping. Let there be N clusters. Denote Z_n as set of images in the n^{th} cluster, where $n \in \{1, \cdots, N\}$. In general, 10 images per group are good enough for our approach, and we set N accordingly. This grouping can also assist in feature selection for the matching purpose. Shape features such as SIFT may be reliable for feature matching in general, but it is not the same with color feature due to the possible intra-class variation in terms of color. But when it is the same object instance across images, color plays a very vital role, such as in the iCoseg dataset. Therefore, in order to adaptively detect such a case, for any sub-group, we calculate a metric δ that measures the color histogram variance (across images in group) averaged over histogram bins using

$$\delta(Z_n) = \frac{1}{N_b} \sum_{j=1}^{N_b} \sqrt{\frac{1}{|Z_n - 1|} \sum_{I_i \in Z_n} \left(H_{I_i}^{S_i}(j) - \hat{H}_n(j) \right)^2}, \tag{3.10}$$

where $H_{I_i}^{S_i}$ denotes normalized color histogram (with $N_b = 512$ bins indexed by j) of only the salient pixels in I_i. $\hat{H}_n$ denotes average of such histograms in Z_n. Higher the δ, more the color feature becomes unreliable for interaction. We consider only salient pixels because we assume that common objects pixels are generally salient, and this gives some information about the common object. So, we concatenate color feature to the dense SIFT feature of images in the sub-group Z_n, only if $\delta(Z_n) < \epsilon$ (See Section 3.3 for discussion on setting ϵ).

3.1.3.2 Saliency Warping

Warping [55] basically is a process of aligning one image w.r.t. another image by establishing dense correspondence. The idea behind saliency warping is that by alignment of corresponding pixels in other images to the pixels of an image, saliency information across corresponding pixels can be shared to estimate a suitable saliency value for the pixel. Following [78], masked Dense correspondence [55] (masked by Otsu thresholded saliency map) is used to find the

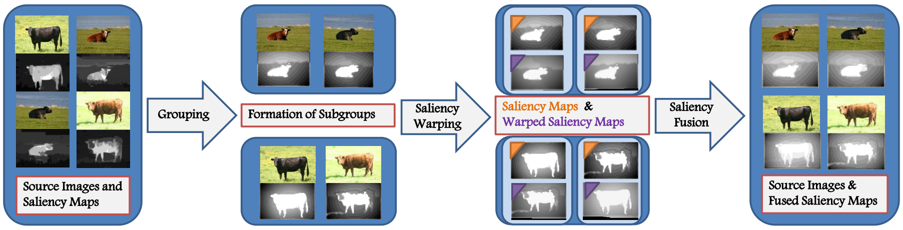


Figure 3.6: Interaction process includes three steps: grouping, saliency warping, and saliency fusion.

corresponding pixels. The difference is that feature used in our approach may also include the color feature in addition to the SIFT feature, depending upon the $\delta(Z_n)$ value.

Particularly, if w_{ij} denotes flow field, warped saliency map W_{ij} of I_j for I_i is formed by $W_{ij}(p) = S_j(p{+}w_{ij}(p))$. In this manner, warped saliency maps of other images in the group are formed for every image in the group. These warped saliency maps are considered as candidate saliency maps comprising of candidate saliency values for each pixel in an image. Let $\mathbf{W}_i^k$ be set of all the candidate saliency maps for image $I_i \in Z_n$ at k^{th} iteration including its own saliency map, and thus it is defined as

$$\mathbf{W}_i^k = \begin{cases} \{S_i^k, W_{ij}^k | I_j \in Z_n \backslash I_i\}, \text{ if } \rho_i = 0; \\ \{S_i^k\}, \text{ else}, \end{cases} \tag{3.11}$$

where the set consists of warped saliency maps in addition to the saliency map if break variable is not yet triggered. Hence, break variables become crucial in avoiding the costly warping processes when they are not required.

3.1.3.3 Saliency Fusion

Now that we have collected candidate saliency maps for I_i in the set $\mathbf{W}_i^k$, we can fuse them in any number of ways, such as average, geometric mean or median etc. Also, we can make use of the quality scores as weights to improve the chances of fused saliency map to have better quality. Let $\mathbf{Q}_n^k = \{\phi(S_j^k)\psi(S_j^k)|I_j \in Z_n\}$ be set of quality scores of saliency maps of group Z_n at k^{th} iteration. Let the fusion function be denoted as $\mathcal{F}$ and we define the fused saliency

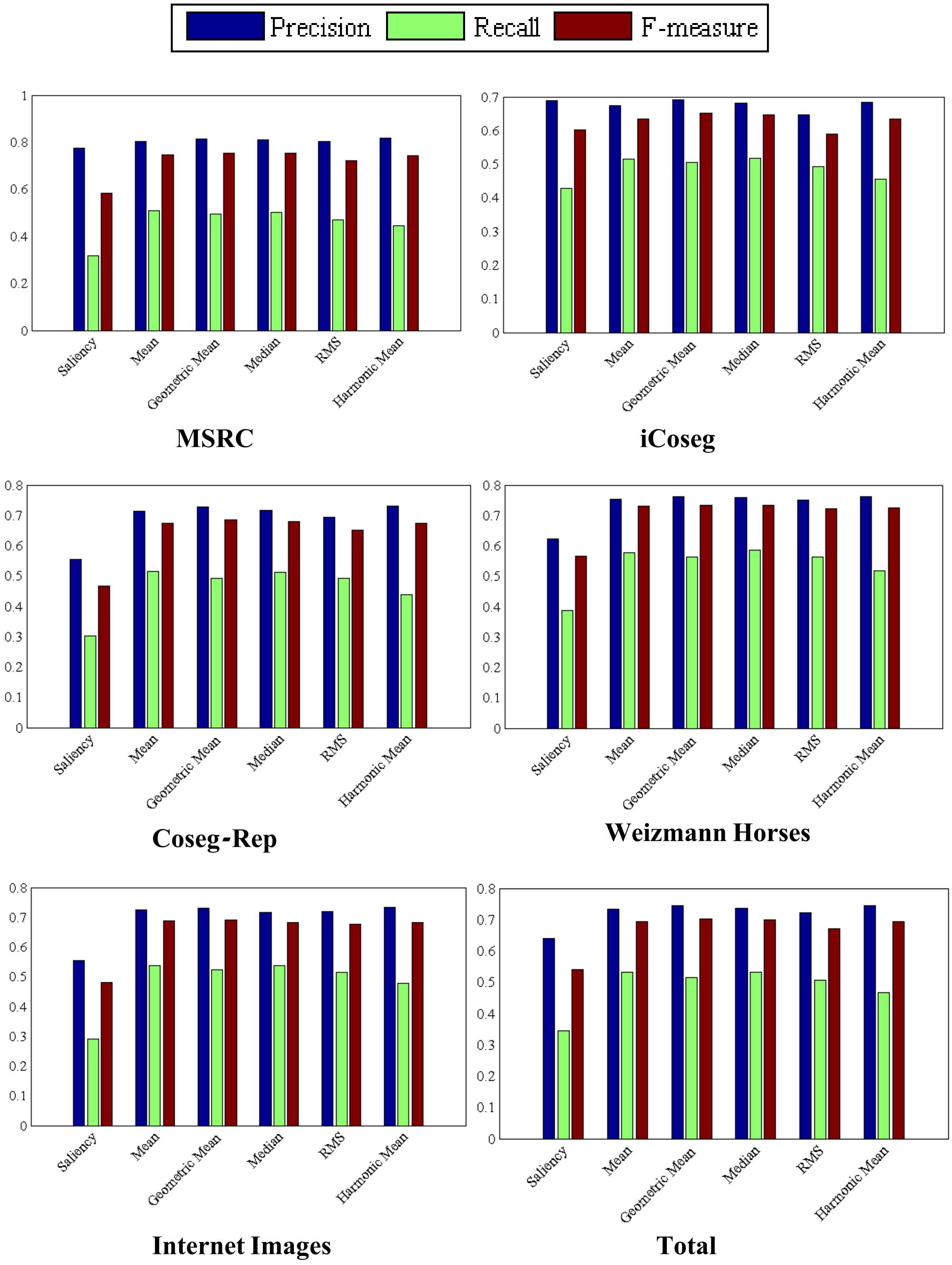


Figure 3.7: Scores obtained using original saliency maps and using our fused saliency maps through various fusion functions on various datasets. It can be seen that fusion improves the performance, and geometric mean and median fusion functions are best.

map of $I_i \in Z_n$ at k^{th} iteration as

$$F_i^k = \mathcal{F}(\mathbf{W}_i^k, \mathbf{Q}_n^k) \tag{3.12}$$

where the fusion function takes two inputs, i.e. a set of candidate saliency maps of the image and set of quality scores of its group.

In order to show the importance of fusing warped saliency maps of other similar images, and to choose an appropriate fusion function, we compare Otsu thresholded saliency maps with groundtruth masks on various co-segmentation datasets. We report overall precision, recall and f-measure results in the Figure 3.7. It can be seen that fusing saliency maps greatly improves the performance over using original saliency maps on all the datasets. There are very little differences among performances between fusion strategies used here. However, though this empirical experiment, we notice that mean, geometric mean and median functions perform the best. We choose weighted median in our experiments for fusion finally, because it is robust to outliers. The way median filtering is used for removing the salt and pepper noise in images inspires us to adopt median filter for application on the corresponding pixels across images (quite different from neighboring pixels in an image).

We use regularization to make saliency scores consistent within a superpixel region. Specifically, [94] is adopted to generate superpixels, and each pixel's saliency score is replaced with average saliency score of its superpixel.

3.1.4 Improving Efficiency

We have seen that above approach of interaction involves aligning each image w.r.t. each and every image in a group, but so many of computationally expensive alignments while considering large datasets is certainly undesirable. In order to overcome this, we modify our approach slightly. Assume that dense correspondence is precise, which means that same sets of corresponding pixels will get together every time we try to collect them for different images in the group. In that case, collecting candidate saliency values at each pixel for each image becomes repetitive. Instead, an efficient way would be to collect candidate saliency values for one time only and propagate the fused result. Therefore, for every group, we choose a key image, say $\mathcal{I}_n$ in Z_n, for which alone we obtain corresponding pixels and calculate the fused saliency map

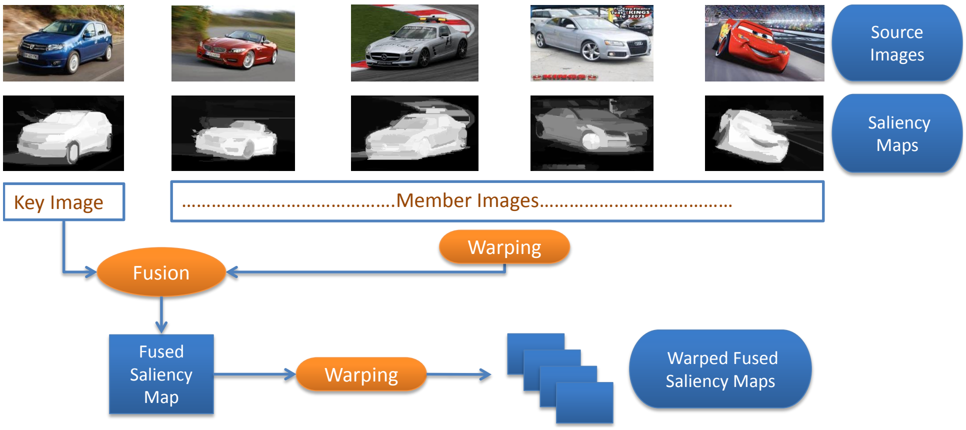

Source Images
Saliency Maps
Key Image
....Member Images..
Warping
Fusion
Fused Saliency Map
Warping
Warped Fused Saliency Maps

Figure 3.8: We improve efficiency of our interaction process by collecting candidate saliency maps only for the key image and then aligning back the fused saliency map to other member images.

first. Then this fused saliency map is aligned back to different group members to form candidate saliency maps for the group members (as shown in Figure 3.8). Let $\mathcal{W}_{in}^k$ denote warped fused saliency map of $\mathcal{I}_n$ to image I_i at k^{th} iteration. $\mathbf{W}_i^k$ can now be redefined in the following way:

$$\mathbf{W}_i^k = \begin{cases} \{S_i^k, W_{ij}^k | I_j \in Z_n \backslash I_i\}, \text{ if } I_i = \mathcal{I}_n \text{ and } \rho_i = 0; \\ \{S_i^k, \mathcal{W}_{in}^k\}, \text{ if } I_i \neq \mathcal{I}_n \text{ and } \rho_i = 0; \\ \{S_i^k\}, \text{ if } \rho_i = 1, \end{cases} \tag{3.13}$$

where the first case is for key image, second case is for member images, and third case is when break variable is already triggered. And as far as fusion is concerned, it is performed in the following way:

$$F_i^k = \begin{cases} \mathcal{F}\left(\mathbf{W}_i^k, \mathbf{Q}_n^k\right), \text{ if } I_i = \mathcal{I}_n; \\ \mathcal{F}\left(\mathbf{W}_i^k, \{\phi(S_i^k)\psi(S_i^k), \sum\limits_{I_j \in Z_n} \phi(S_j^k)\psi(S_j^k)\}\right), \text{ else}, \end{cases} \tag{3.14}$$

where fusion for the key image remains the same as earlier. But for member images, since it is a fusion between a saliency map and a warped fused saliency map of key image, we give high weight as much as the sum of all the quality scores in the group to the warped fused saliency map, because it is highly reliable for having been fused over several saliency maps.

We make an analysis how this modification affects the time-complexity. In the original method, in a group consisting of x images, every time we calculate fused saliency map for an image, saliency maps of all other $(x-1)$ images need to warp to this image, which requires computing the costly dense correspondences for $(x-1)$ times. Thus, the interaction of all x images will need computing dense correspondences for $x(x-1)$ times. This suggests $O(x^2)$ complexity, which is time-consuming and it is undesirable while dealing with large-scale datasets. But after the modification, we need to compute dense correspondence for $(x-1)$ times for the key image alone to generate its fused saliency map first, and then warp it back to member images, requiring computing dense correspondences for another $(x-1)$ times. Thus, in total it turns out to be only $2(x-1)$, suggesting $O(x)$ complexity. Thus, this modification resulted in reducing the time-complexity of proposed method from quadratic to linear.

3.1.5 Applications

The obtained final high-quality saliency maps have applications in object-level segmentation and localization.

Segmentation: Based on the final saliency map S_i^*, we obtain the final object mask using GrabCut algorithm [76], in which foreground ($\mathcal{FG}_i$) and background ($\mathcal{BG}_i$) seed locations are determined by

$$p \in \begin{cases} \mathcal{FG}_i, & \text{if } S_i^*(p) > \tau; \\ \mathcal{BG}_i, & \text{if } S_i^*(p) < \upsilon_i, \end{cases} \tag{3.15}$$

where pixel p will be considered as background seed location if its final saliency value is less than υ_i (Otsu's threshold [73] value of S_i^*). Similarly, pixel p will be considered as foreground seed location if its saliency value is greater than τ, which we call foreground threshold parameter. By default, we set τ as 0.75.

Localization: For localization, we first threshold final saliency map S_i^* with some threshold, say τ (same as in segmentation application above), and identify sparsely located spatial group $\big($same as the object components such as $O_u(S_i^*)\big)$ of white pixels (having saliency values greater than τ) as the candidate objects. Out of them, we only choose dominant objects that make at least half the contribution made by the largest object, i.e. $C_u(S_i^*) \geqslant 0.5 \times C_{u^*}(S_i^*)$. Such a criterion allows localization of multiple dominant objects if they are of somewhat similar size. By this, we also ensure that insignificant objects in the image that are present (may be due to complex backgrounds) are not considered for localization. Also, since these dominant objects may not be having similar edges as ones in the image, we identify nearest edge locations to the pixels in the concerned dominant object and adjust the bounding box to extreme edge locations in the four directions.

3.2 Experiments

We conduct extensive experiments to evaluate our method in terms of the applications discussed in the previous section. In this section, we first provide details of different datasets and evaluation metrics used, then we proceed with the evaluation.

3.2.1 Datasets

Several public co-segmentation and co-localization datasets are already available on which we can evaluate our final saliency maps.

In literature, most popularly used datasets for the co-segmentation evaluation are MSRC [88] and iCoseg [7] datasets. We also evaluate on recently developed Coseg-Rep [20] and Internet images [78] datasets. MSRC dataset contains only 14 categories with 419 images in total. iCoseg dataset contains 38 categories with 643 images in total. For the fair comparison with the existing methods [78][24], like them, we also use subset of the dataset which includes 30 categories and a total of 530 images. Coseg-Rep dataset contains 23 categories and 572 images in total. Also, Internet images dataset released by [78] contains 3 categories: Airplane, Car, and Horse, with 4347, 6381 and 4542 images, respectively. All these datasets are not so big and suitable for our original interaction approach. For evaluating our efficient interaction approach for eventual segmentation in the large-scale scenario, ImageNet [22] setup of 0.5 million images in [28] is used.

Recently, for co-localization evaluation, [90] used tight bounding boxes across the groundtruth segmentation masks of Internet images dataset as the ground truth bounding boxes. Following them, we also evaluate our method on the same setting. For the large-scale localization evaluation, we use ImageNet again in both unsupervised and supervised setups as suggested by previous methods, [90] and [95], respectively. Since our method can work in both supervised and unsupervised scenarios, for distinguishing between the unsupervised results and supervised results, suffixes (U) and (S) are used, respectively. As per the setup in [90], there are 1 million images for which bounding boxes are available in ImageNet, and they are spread over 3627 classes. In the supervised setup, [95] divides images with available ground truth bounding boxes into source sets (or training set) and target sets (or test set). For images that belong to source set, we replace saliency maps with ground-truth bounding boxes, and the task now is to obtain bounding boxes for remaining images in the group.

3.2.2 Evaluation Metrics

Following the literature [78][24], we use Jaccard Similarity (Jacc.) and Accuracy (Acc.) for segmentation evaluation. Jaccard Similarity is defined as the intersection divided by union of ground-truth and the segmentation result. Accuracy is defined as the percentage of correctly labeled pixels. Similarly, CorLoc score has been used for evaluation of localization which is defined as percentage of images that satisfy the condition: $\frac{area(B_{gt} \cap B_{co})}{area(B_{gt} \cup B_{co})} > 0.5$, where B_{gt} and B_{co} are ground-truth and computed bounding boxes, respectively.

3.2.3 Segmentation Evaluation

In Tables 3.2-3.4, we compare our results of both original and efficient methods with state-of-the-art co-segmentation methods on different datasets. It can be seen that our methods obtain competitive performance compared to the existing methods. Note that our method is much faster than state-of-the-art [24]. Specifically, running on the same PC with Intel Core i5-3470@3.20 GHz CPU and 32 GB RAM, [24] (using their own source codes in Matlab) takes 29.2 hours to complete the entire segmentation process on MSRC dataset. However, our method (also in Matlab codes) takes only 4.9 hours. Also, we show some examples in Figure 3.9 where our method performs better than other methods including [24]. Another thing to note here is that performance difference is quite narrow between our original and efficient methods, which suggests that our assumption of warping process being precise is a viable

Table 3.2: Comparison on Coseg-Rep dataset using overall values of Jaccard Similarity (Jacc.) and Accuracy (Acc.)

	Jacc.	Acc.
Co-segmentation&Co-sketch [20]	0.67	90.2
Ours (original)	0.73	91.9
Ours (efficient)	0.72	91.3
Ours (tuned/group)	**0.76**	**92.8**

Table 3.3: Comparison on Internet image dataset using overall values of Jaccard Similarity (Jacc.) and Accuracy (Acc.)

	Car		Horse		Airplane	
	Jacc.	Acc.	Jacc.	Acc.	Jacc.	Acc.
[39] (reported in [78])	0.37	58.7	0.30	63.8	0.15	49.2
[40] (reported in [78])	0.35	59.2	0.29	64.2	0.12	47.5
[78]	0.63	83.4	0.54	83.7	0.56	86.1
Ours (original)	0.71	87.0	0.57	84.7	0.55	85.7
Ours (efficient)	0.71	86.4	0.56	84.2	0.54	85.2
Ours (tuned/group)	**0.73**	**88.4**	**0.61**	**88.1**	**0.59**	**88.4**

Table 3.4: Comparison on MSRC and iCoseg datasets using overall values of Jaccard Similarity (Jacc.) and Accuracy (Acc.)

	MSRC		iCoseg	
	Jacc.	Acc.	Jacc.	Acc.
Discriminative [39]	0.45	70.8	0.39	61.0
Multi-Class [40]	0.51	73.6	0.43	70.2
Object Discovery [78]	0.68	87.7	0.69	89.8
Composition [24]	0.73	89.2	**0.73**	**92.8**
Ours (original)	0.72	88.9	0.67	89.3
Ours (efficient)	0.71	88.1	0.66	88.9
Ours (tuned/group)	**0.74**	**89.7**	0.72	91.8

Table 3.5: Comparison on large scale dataset ImageNet using overall values of Jaccard Similarity (Jacc.) and Accuracy (Acc.)

Methods	Jacc.	Acc.
[44]	-	77.3
[28]	0.57	84.3
Ours(efficient)	0.56	84.1
Ours(tuned/group)	**0.59**	**86.4**

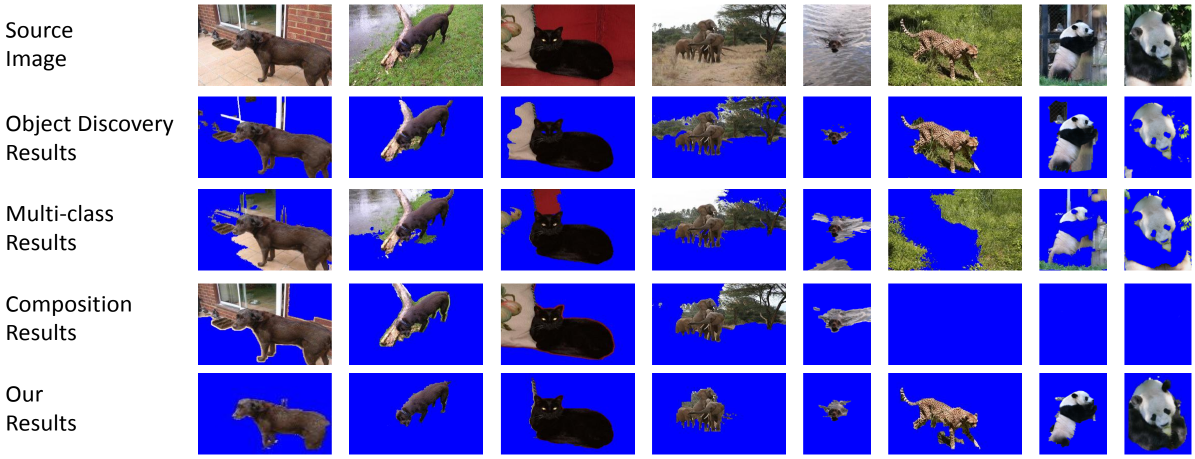


Figure 3.9: Sample segmentation results where our method performs better than other methods of co-segmentation

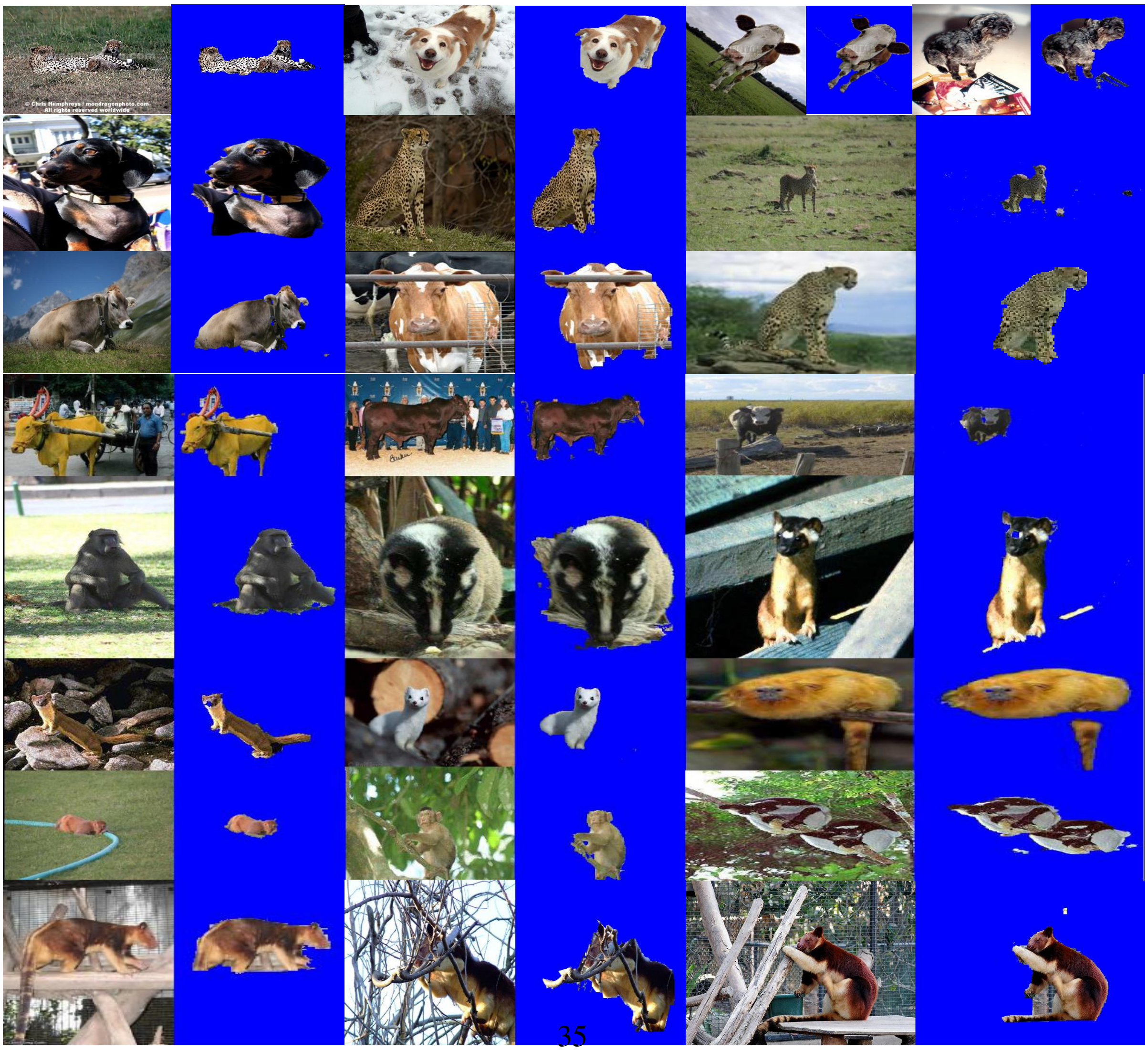


Figure 3.10: Sample segmentation results from ImageNet dataset

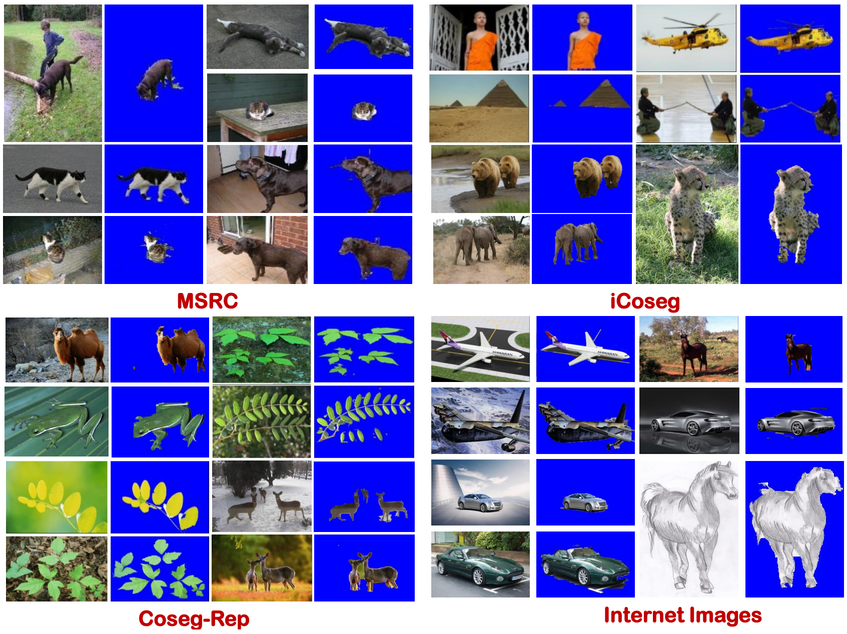


Figure 3.11: Sample segmentation results from MSRC, iCoseg, Coseg-Rep, and Internet Images datasets.

assumption. Given this, we compare segmentation results of our efficient method with existing state-of-the-art results on ImageNet (large-scale dataset) in Table 3.5. We achieve comparable performance here as well. Note that results reported in other methods are obtained either by tuning parameters or by undertaking some parameter learning. For the fair comparison, we also tune our parameter τ per group and show the resultant performance. See Figure 3.10-3.11 for sample segmentation results that we obtain on different datasets. It can be seen in these figures that proposed method is able to accurately segment both simple and complex images because we are able to effectively guide the co-saliency estimation using our saliency quality measurement.

3.2.4 Localization evaluation

In this section, we discuss how we evaluate the proposed method for localization application on both unsupervised and supervised setups.

Unsupervised Setup: In the unsupervised setup, we compare our results with existing methods in Table 3.6 on both ImageNet and Internet images datasets. We achieve 21.8% and 10.3% improvements over [90] on ImageNet and Internet images datasets, respectively. Since

Table 3.6: CorLoc comparison on ImageNet and Internet images datasets in unsupervised setup

	ImageNet	Internet
[78](U)	-	75.2
[90](U)	53.2	76.6
[17]	-	84.2
Ours (efficient)(U)	**64.6**	**84.5** (tuned/group)

Table 3.7: CorLoc comparison on ImageNet dataset in supervised setup

	CorLoc
[27](S)	58.5
[95](S)	66.5
[95]*(S)	68.3
Ours (efficient)(S)	**71.1**
Ours (efficient)(U)	68.7

other methods tune their parameters on Internet images dataset, we also tune our threshold parameter (τ) per group and report the results for this particular dataset. [17] also evaluates on Internet images dataset, and proposed method could marginally outperform [17] as well. However, the vital difference between [17] and proposed method is in terms of speed. Our method is much simpler and faster, and therefore it has large scale application as demonstrated on the ImageNet.

Supervised Setup: In the supervised setup, the problem that we try to address here is similar to the “Self" case in [95], where only images within the same class are used as source sets. In Table 3.7, we compare our results on the target sets with two previous attempts in [27] and [95] to populate ImageNet with bounding boxes in a supervised manner. We achieve 21.4% and 6.9% improvement over [27] and [95], respectively. [95] also reports results using state-of-the-art features and object proposals, which we denote as [95]*. We achieve 4.1% improvement over state-of-the-art [95]* as well. Considering that proposed method does not essentially need bounding boxes, unlike [95], we report our unsupervised results (Proposed Method(U)) as well, where we do not use any ground-truth bounding boxes of even images belonging to source sets. Interestingly, we still obtain comparable results to [95]*(S).

Figure 3.12 shows sample localization results obtained on ImageNet dataset. In addition, we show our results (red) along with the ground-truth (green) for visual comparison in Fig-

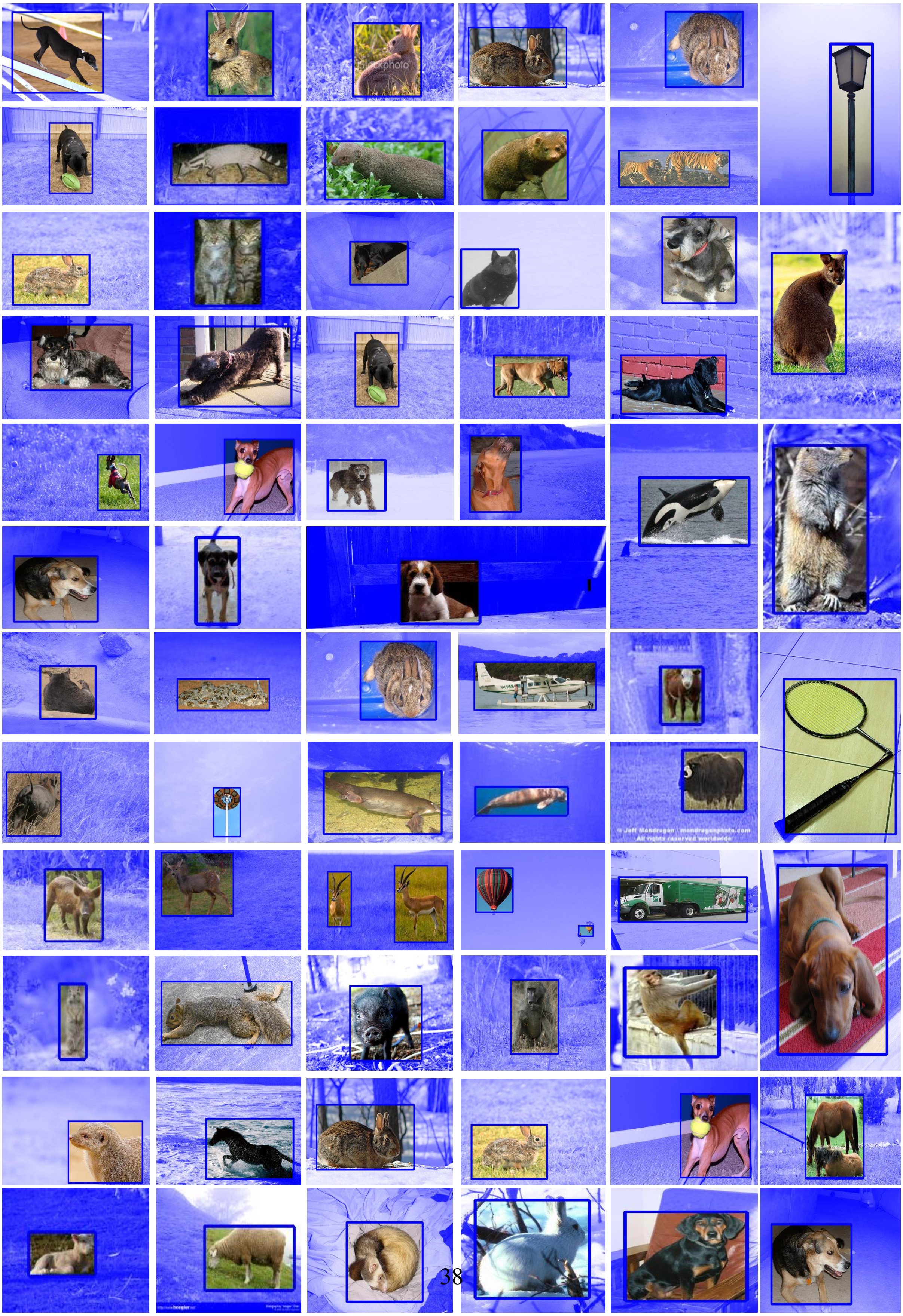


Figure 3.12: Sample localization results from ImageNet dataset

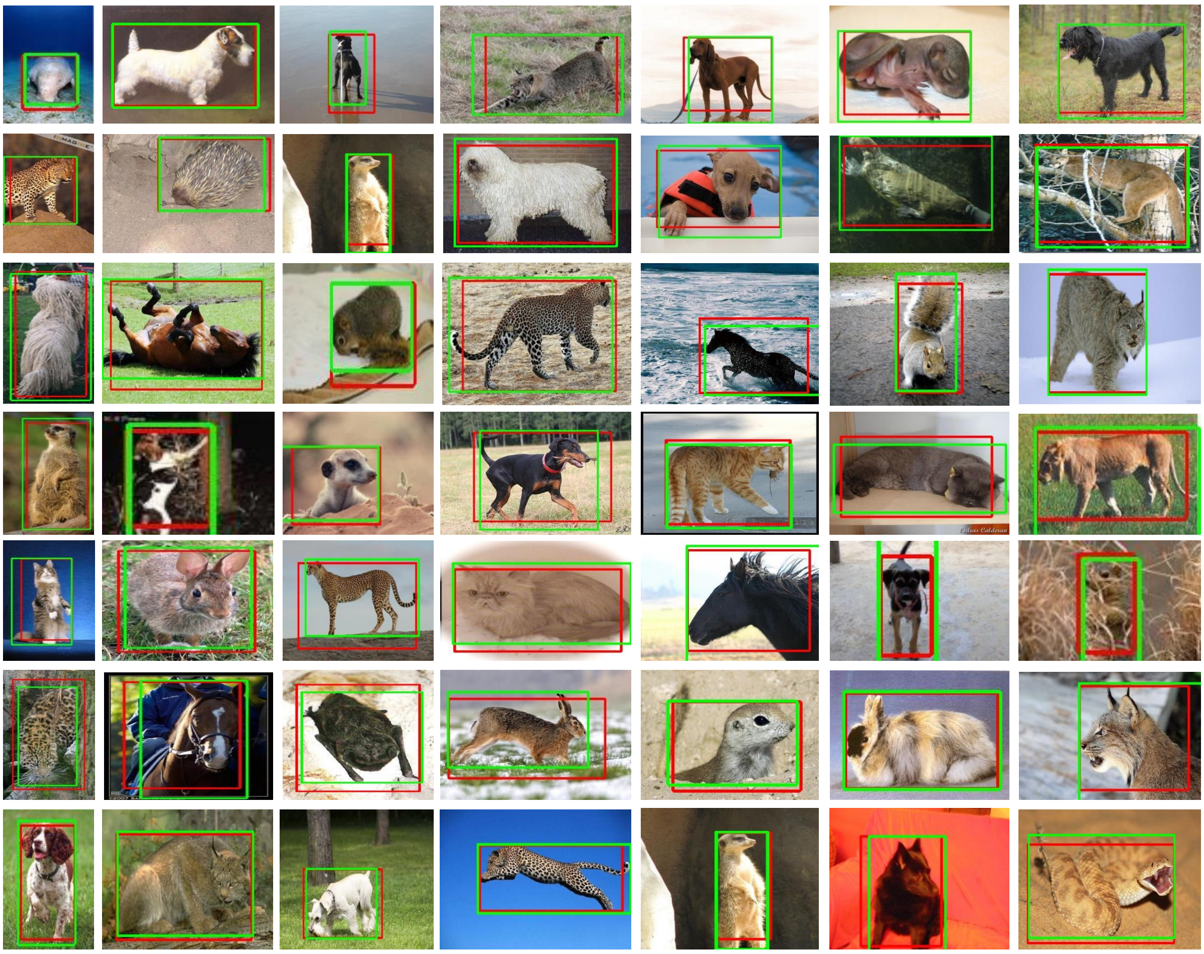


Figure 3.13: Sample visual comparison between groundtruth (green) and our results (red)

Table 3.8: Average δ and total time taken on various datasets for original and efficient interaction strategy (for one iteration).

	δ	$\delta_{g.t.}$	Time (mins.) (Original)	Time (mins.) (Efficient)
MSRC	0.0026	0.0023	85	18
iCoseg	0.0017	0.0014	116	28
Coseg-Rep	0.0029	0.0029	186	39
Weizmann Horses	0.0039	0.0027	80	17
Internet Images	0.0028	0.0027	4306	901

ure 3.13. Thanks to our quality-guided approach to the joint processing, proposed method is able to accurately provide bounding boxes for both simple and complex images here also.

3.3 Discussion

ϵ-setting and δ-effectiveness: In a group Z_n, usage of color feature depends upon $\delta(Z_n) < \epsilon$ criterion. In order to set the ϵ value properly, we show average δ values obtained for 5 datasets in Table 3.8, and only iCoseg dataset out of them requires color feature. As expected, a notable difference can be observed between iCoseg dataset and rest of the datasets in terms of their δ values. Noting this, we comfortably set $\epsilon = 0.0020$. We also show $\delta_{g.t}$ values where we make use of groundtruth maps in the place of saliency maps. A high correlation of 0.796 between δ and $\delta_{g.t}$ suggests that our δ measurement is a good indicator.

Efficiency Comparison: It can also be observed in Table 3.8 how the efficient strategy greatly reduces the time taken for interaction to $20\% - 25\%$ compared to the original strategy. Therefore, proposed modifications have certainly made large-scale application feasible while keeping the performance somewhat competitive (as we see in Tables 4.4-4.6).

Limitations: Proposed method fails when our assumption (common object or its parts are salient in general, if not in every image) fails. Therefore, it pretty much depends on the association of the image for our method to succeed. For example, only the beak portion of goose gets segmented or localized in Figure 3.14(i), because other body parts are salient neither in the considered image nor in the association. The second limitation is caused by poor warping process, i.e. when it struggles to align objects of very different sizes (a case that can easily

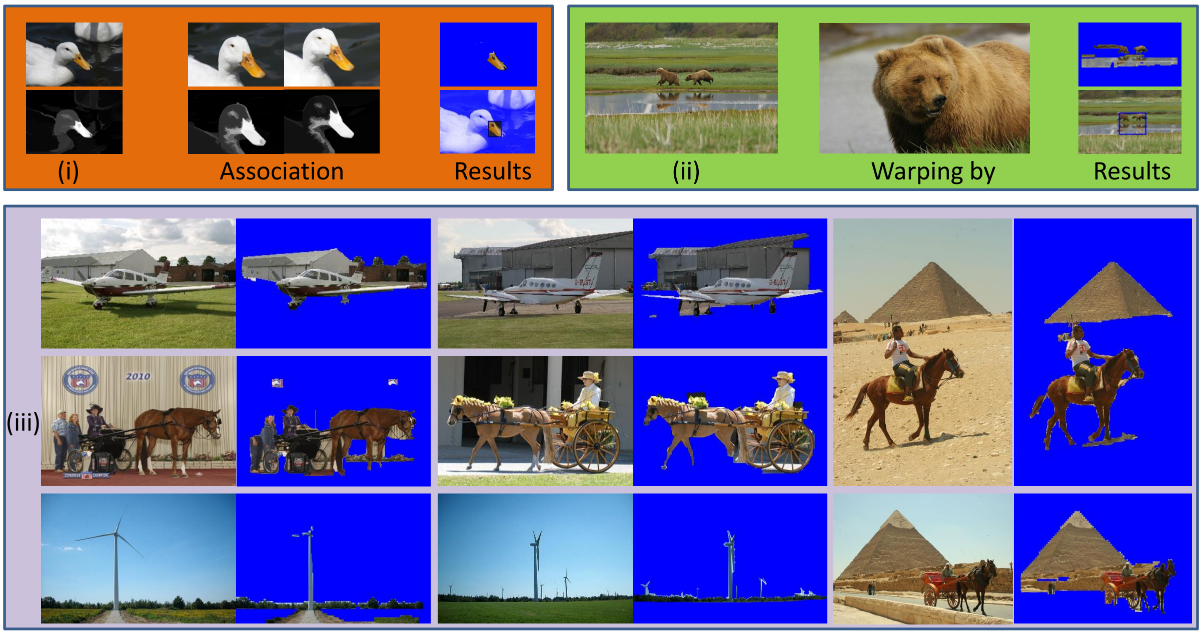


Figure 3.14: Our method fails in three scenarios: (i) Wrong association, (ii) Difficult warping, and (iii) Multiple common objects.

arise due the poor choice of clustering parameter during group formation, especially if hardly 10 images are there). For example, high size variation in Figure 3.14(ii) produces poor results. The third limitation is that our method may end up segmenting multiple object classes in some images, while groundtruth masks may consist of only one object class. This can happen due to two reasons, one is that all images in a particular group (cluster) contain multiple object classes, and another one is that saliency quality couldn't be improved by fusion against the (already) high quality original saliency map. In such cases, our result may not match well with groundtruth masks. Note how in Figure 3.14(iii) our method captures multiple object classes. On the contrary, groundtruth masks capture only one object class: plane, horse, windmill, or pyramid in their corresponding images.

3.4 Summary

In this chapter, we have proposed a novel quality-guided fusion-based co-saliency estimation method, where saliency maps of different images are simply fused using dense correspondence technique. More importantly, this joint processing is guided by our proposed saliency quality measurement system, which helps us decide whether to choose the original or fused saliency map as the final one. Idea is to choose the saliency map with well-separated foreground and background, as well as a concentrated foreground. In this way, we attempt to address the individual versus joint processing issue. Our evaluation of final saliency maps w.r.t. segmentation and localization applications on several benchmark datasets including the large-scale dataset, Imagenet, show that proposed framework is able to achieve very competitive results.

Chapter 4

Image Co-segmentation *via* Saliency Co-fusion

Image co-segmentation refers to the task of extracting common objects from a set of images, which is very useful for many vision and multimedia applications such as object-based image retrieval, image classification, and object recognition. It can be considered as one type of weakly supervised segmentation methods, which makes use of the weak prior that there exist common objects across different images in the set. This is quite different from single image segmentation. The existing single image object-level segmentation methods can only exploit either the prior from human supervision, which requires human interactions such as GrabCut, or the prior from single image-based visual saliency, which might fail at complex images with cluttered background or non-salient foreground. In contrast, image co-segmentation goes beyond single image segmentation in the sense that it can exploit not only the intra-image priors, but also the inter-image priors. Furthermore, it also brings in the new challenges of how to find the right inter-image priors and how to make use of them.

The concept of co-segmentation was first introduced in [77]. Later, many co-segmentation algorithms have been proposed in the literature, ranging from early image pair co-segmentation [30][70], multiple image co-segmentation [39][43][48][65] to the recent multiple objects co-segmentation [69][56][59][42], noisy image set co-segmentation [78] and large-scale co-segmentation [28][34].

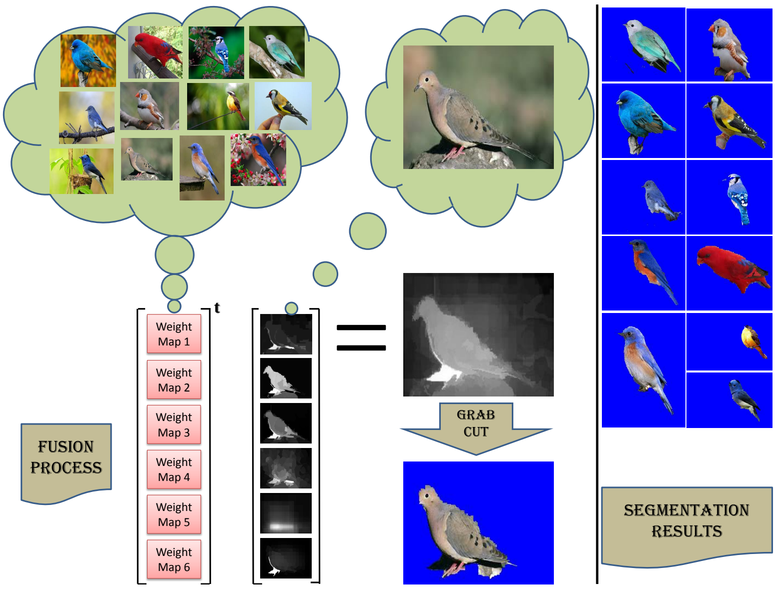


Figure 4.1: Fusion of multiple saliency maps of an image generated by different saliency extraction methods to enhance the common foreground object while suppressing background saliency. The fusion process is essentially weighted summation of different saliency maps at superpixel level.

Despite the great progress made by the existing co-segmentation algorithms, they still have some major limitations. First, most of the state-of-the-art co-segmentation algorithms require fine-tuning of quite a few parameters and the co-labelling of multiple images simultaneously, which are very complex and time-consuming, especially for large diverse datasets. Second, as seen in the existing works [78][97], co-segmenting images might not perform better than single image segmentation for some datasets. This might be due to the additional energy term commonly used to enforce inter-image consistency, which often results into unsmooth segmentations in individual images.

In this chapter, we focus on binary image co-segmentation, i.e. extracting a common foreground from a given image set. Instead of following the conventional way of co-labelling multiple images, we aim to exploit inter-image information through co-saliency, and then perform single-image segmentation on each individual image. Thus, no additional energy terms get added while performing the segmentation. Differently from the co-saliency idea presented in the previous chapter, in order to make the system robust and avoid heavy dependence on one single saliency extraction method for generating co-saliency, we here propose to apply

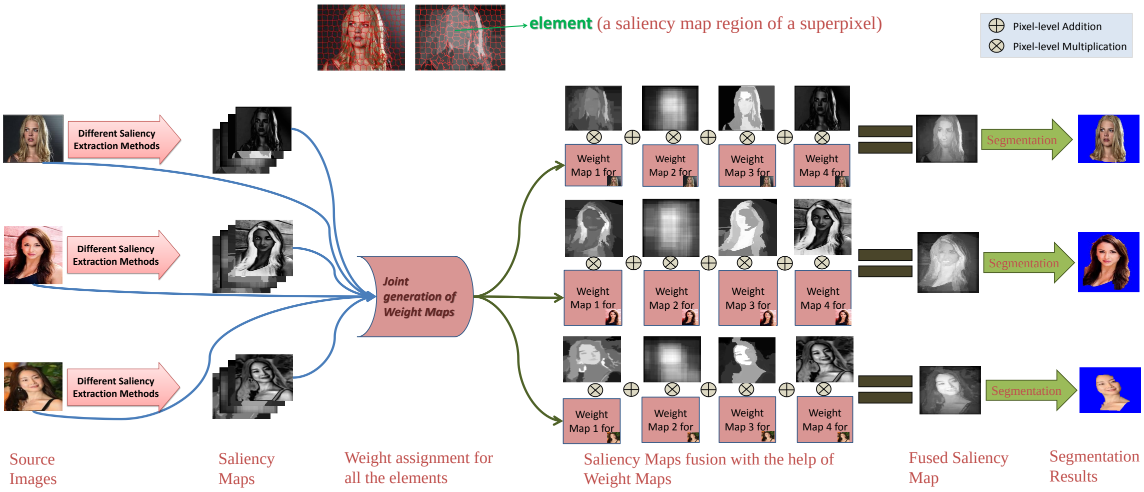


Figure 4.2: Flowchart of the proposed saliency co-fusion based image co-segmentation where multiple images are used to generate weight maps for fusing different saliency maps of images to extract common foreground. Element, the basic processing unit of our process, is defined as a saliency map region of a superpixel.

multiple saliency extraction methods on each image. Eventually, an enhanced saliency map is generated for each image by fusing its various saliency maps via weighted summation at superpixel level, where the weights are optimized by exploiting inter-image information, as shown in Figure 4.1. We call the proposed method saliency co-fusion, whose objectives include: (1) boosting the saliency of common foreground regions; and (2) suppressing the saliency of background regions.

Figure 4.2 illustrates the process flow of the proposed saliency co-fusion based image co-segmentation. The key component lies in the developed saliency co-fusion process, which is performed at the superpixel level. Particularly, we define each saliency map region (produced by one saliency detection method) of one superpixel as an element (see Figure 4.2 top), and give a weight for each element. We formulate the weight selection as an energy minimization problem, where we incorporate saliency recommendations from similar elements, foreground/background priors through similar element voting, and neighbor smoothness constraints. Finally, the fused saliency for a superpixel is just a weighted summation of the corresponding elements. Experimental results show that our saliency co-fusion based co-segmentation achieves competitive performance even without fine-tuning the parameters, i.e., at default setting, compared with the state-of-the-art co-segmentation algorithms. In addition, our co-fused saliency maps are much cleaner compared to the co-saliency maps generated in previous chapter. Here, saliency details do not get lost through regularization, which was required in the previous chapter.

4.1 Proposed Method

In this section, we first formulate our saliency co-fusion problem. Then we give a detailed description of individual terms as well as implementation details.

4.1.1 Problem Formulation

Considering a set of N images $\mathcal{I} = \{I^1, I^2, ..., I^N\}$, denote $\mathcal{S}^n = \{S_1^n, S_2^n, ..., S_M^n\}$ the set of M saliency maps (normalized to range 0-1) for image I^n obtained using M different existing saliency extraction methods. Also, denote $\mathcal{P}^n = \{P_1^n, P_2^n, ..., P_{|\mathcal{P}^n|}^n\}$ the set of superpixels in image I^n obtained using [1]. Defining a saliency map region of superpixel as an element e, which is the basic processing unit in our method, we have total $N_e = \sum_{n=1}^{N} M|\mathcal{P}^n|$ elements. Let $z(n,k,m)$ denote the associated weight for element $e(n,k,m)$ that belongs to image n, superpixel k, and saliency map m. The weight maps depicted in Figure 4.1 and Figure 4.2 are basically constructed using these associated weights.

We stack all the weights into a vector $\mathbf{z} = [z_1, z_2, \ldots, z_{N_e}]^t$ for simplicity and use u or v as the element indices for referencing purposes. We mix the usage of the element vector index with its corresponding matrix index (n,k,m) since one can be converted to the other easily.

Our goal is to find the optimal weight for each of the elements in order to jointly fuse various saliency maps of similar images at superpixel level such that common foreground saliency gets boosted up and background saliency is suppressed in final fused saliency maps. In particular, we treat saliency co-fusion as a weight selection problem. On one hand, we want to give higher weights to elements with higher confidence. On the other hand, we want to have certain consistency in the weight selection among neighboring elements. Considering the constraint that the resultant fused saliency map values should occur in the range [0,1], we formulate our task as a quadratic programming problem:

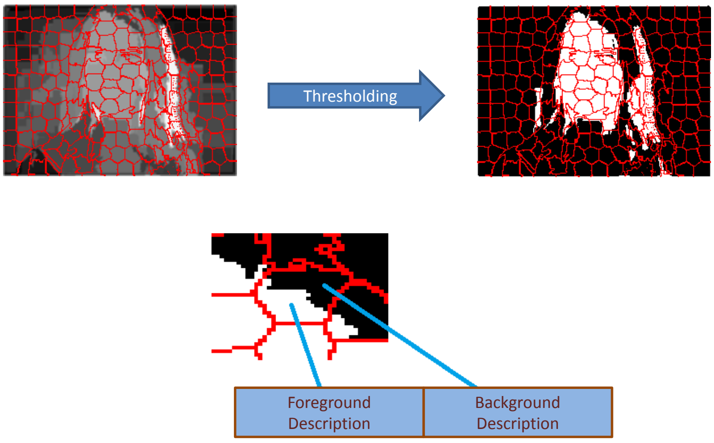


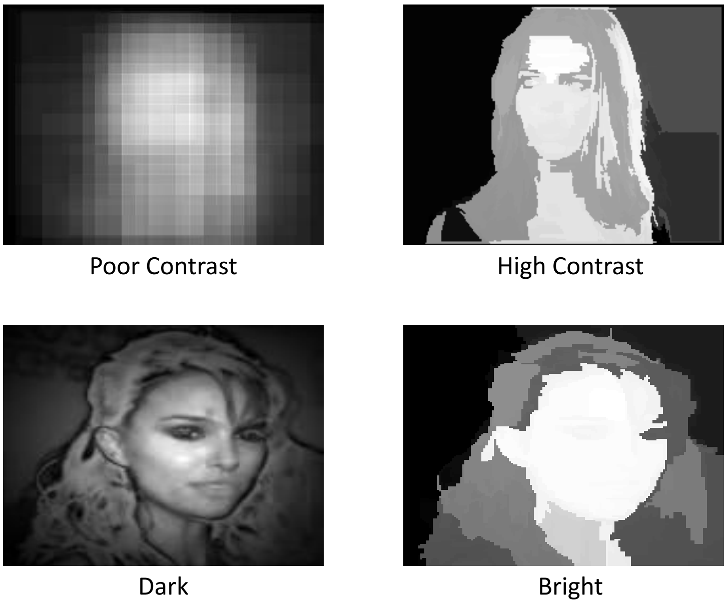
Figure 4.3: Feature Description: Each element is divided into foreground and background regions after global thresholding and two sets of features are extracted from these two regions separately, one for foreground and the other for background.

$$\begin{aligned} \min_{\mathbf{z}} \quad & Y^t\mathbf{z} + \lambda \mathbf{z}^t G\mathbf{z} \\ \text{s.t.} \quad & 0 \le z_u \le 1, \ \forall u \in [1, N_e], \\ & \sum_{m=1}^{M} z(n,k,m) = 1, \ \forall I^n \in \mathcal{I}, P_k^n \in \mathcal{P}^n \end{aligned} \tag{4.1}$$

where there are two terms traded off by a balancing parameter λ. The first term ($Y^t\mathbf{z}$) is a prior term to enforce global commonness and co-saliency, where the prior term coefficient vector $Y \in \mathbb{R}^{N_e \times 1}$. The second term ($\mathbf{z}^t G\mathbf{z}$) is a pairwise smoothness term to encourage neighborhood elements to take similar weights, where the smoothness term coefficient matrix $G \in \mathbb{R}^{N_e \times N_e}$. The constraints in Eq. (4.1) are there to ensure that individual weights range between 0 and 1, and the summation of all the weights for one superpixel is equal to one. Once $\mathbf{z}$ is determined by minimizing Eq. (4.1), the fused saliency map F^n for a pixel $p \in P_k^n$ can be simply computed as

$$F^n(p) = \sum_{m=1}^{M} z(n,k,m) \times S_m^n(p), \tag{4.2}$$

where B_m^n is the m-th saliency map for an image I^n.



Poor Contrast

High Contrast

Dark

Bright

Figure 4.4: There is no uniformity among saliency maps obtained by different methods. Some saliency maps are of high contrast, while some are of poor contrast. Some are bright, while some are dark.

4.1.2 Feature Description and Similarity

Unlike other methods [78][25][36] where features for matching are extracted from images independent of saliency maps, we develop a saliency map based feature descriptor because our processing units are elements (defined as a saliency map region of a superpixel), instead of pixels or superpixels. We consider the fact that there is no uniformity among saliency maps obtained by different methods. For instance, some saliency maps are of high contrast, while others are of poor contrast. Some are bright, while others are dark. It can be noticed in Fig. 4.4. This can cause serious problems in the process if saliency values are directly taken as features. We tackle it by distinguishing potential foreground pixels from potential background pixels in an element using the classical Otsu's method as shown in Figure 4.3. For each group (both the potential foreground group and the potential background group in the element), we construct a feature descriptor which consists of the average dense SIFT descriptor, and also the average color values in RGB, HSV, and Lab spaces. However, for each element, we have two feature descriptors with each having dimensions $d = 128 + 3 + 3 + 3 = 137$. We concatenate them as the feature descriptor for one element. In this way, different elements of the same superpixel obtain different feature descriptors, depending upon the foreground/background distributions in each element.

$X_f, X_b \in \mathbb{R}^{N_e \times d}$ and $X \in \mathbb{R}^{N_e \times 2d}$ denote the data matrices that stack the foreground descriptors, the background descriptors and the foreground background concatenated descriptors of all the elements as its rows, respectively. We construct similarity matrices κ_f, κ_b and κ, all of $N_e \times N_e$ dimensions that record the potential foreground similarity, the potential background similarity, and the total similarity respectively between all the element pairs:

$$\kappa_f(u,v) = exp\Big(-\gamma \sum_{q=1}^{d} \frac{\big(X_f(u,q) - X_f(v,q)\big)^2}{X_f(u,q) + X_f(v,q)} \Big) \tag{4.3}$$

$$\kappa_b(u,v) = exp\Big(-\gamma \sum_{q=1}^{d} \frac{\big(X_b(u,q) - X_b(v,q)\big)^2}{X_b(u,q) + X_b(v,q)} \Big) \tag{4.4}$$

$$\kappa(u,v) = exp\Big(-\gamma \sum_{q=1}^{2d} \frac{\big(X(u,q) - X(v,q)\big)^2}{X(u,q) + X(v,q)} \Big) \tag{4.5}$$

where γ is a parameter set to $\frac{1}{300}$.

Note that the potential foreground similarity is set to zero if all the pixels in the element belong to the background group and vice versa. If elements u and v belong to the same image, $\kappa_f(u,v)$, $\kappa_b(u,v)$, and $\kappa(u,v)$ are all set to zero since we aim at exploiting similar elements from other images.

Based on the total similarity matrix S, similar elements for each element are identified if the corresponding similarity values are large than a similarity threshold θ (θ is set to 0.75). For one element, its similar elements provide recommendations *via* different cues, based on which we then derive the appropriate weight for the considered element so as to encourage or discourage its role in the final fused saliency map. Details are elaborated below.

4.1.3 Prior Term

We define our prior term coefficient vector Y in Eq. (4.1) as

$$Y = Y_s + Y_f + Y_c \tag{4.6}$$

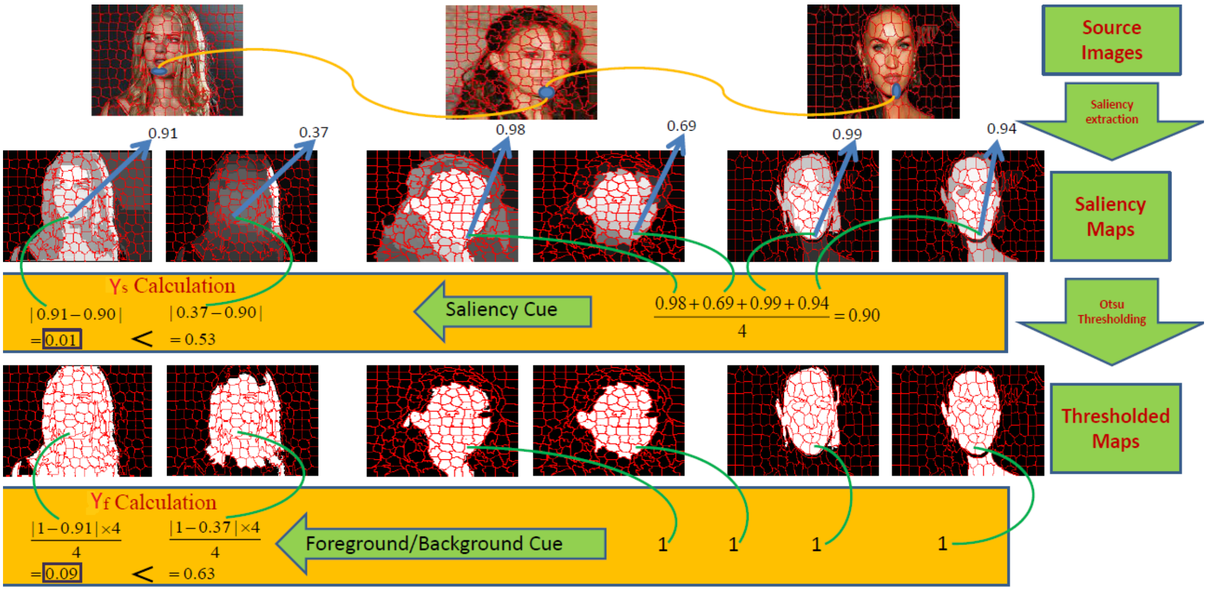


Figure 4.5: Illustration of how similar elements from other images help in determining better elements *via* Y_s (saliency cue) and Y_f (foreground/background cue) calculations in the first image. NOTE: The numerical value that an element is pointing to is the average saliency value of the element. Y_s signifies how close the saliency value of an element is to the recommended saliency value by its similar elements, whereas Y_f signifies the average punishment of an element for deviating from the foreground/background recommendations from each of its similar elements. The lesser the Y_s and Y_f are for an element, the higher weight the element will get. For example, the element covering the chin area in the first saliency map is considered as a better one than that in the second saliency map because of having lower values for both of the cues.

which includes three cues: saliency cue (Y_s) from similar elements, foreground/background prior cue (Y_f) from similar elements and centerness cue (Y_c) based on the spatial location of the element.

Saliency Cue: Following the idea of co-saliency or common saliency, we compare the average saliency of similar elements with the average saliency value of the considered element to decide whether the element should be emphasized or not (give high weight or not). Let $T = [T_1, T_2, ..., T_{N_e}]^t$ denote the vector where each entry is the average saliency value of an element. On the other hand for an element u, we compute the average saliency recommended by its similar elements as

$$E_u = \frac{\sum_{v=1}^{N_e} T_v \delta\big(\kappa(u,v) > \theta\big)}{\sum_{v=1}^{N_e} \delta\big(\kappa(u,v) > \theta\big)} \tag{4.7}$$

where $\delta\big(\cdot\big)$ is the indication function, equal to one if the condition $\big(\cdot\big)$ is true (otherwise 0), which is used to determine whether element v is a similar one or not. Let $E = [E_1, E_2, ..., E_{N_e}]^t$ be the vector comprising of the recommended average saliencies of elements. We then simply define the saliency cue as

$$Y_s = |E - T|. \tag{4.8}$$

Essentially, Eq. (4.8) suggests that if $T(u)$ is very different from $E(u)$, then the corresponding weight z_u is encouraged to be small by Eq. (4.1), which means such elements are not so important in defining final co-fused saliency value, hence the punishment. Figure 4.5 illustrates how similar elements from other images help to determine better elements.

Foreground/Background Cue: Another cue similar elements can provide is to recommend the given element to be foreground or background. For an element u and one of its similar elements v, if their foreground feature descriptors are more similar than the background descriptors, v recommends foreground with a saliency punishment of $(1 - T(u))$ to u; otherwise, it recommends background with a punishment of $(T(u) - 0)$, i.e.

$$\begin{aligned} R_u(v) &= 1 - T(u), \text{ if } \kappa_f(u,v) > \kappa_b(u,v) \\ R_u(v) &= T(u) - 0, \text{ if } \kappa_f(u,v) < \kappa_b(u,v) \end{aligned} \tag{4.9}$$

where $R_u(v)$ denotes the saliency punishment recommended by v to u. Considering all the similar elements, we define foreground/background cue Y_f for an element u as

$$Y_f(u) = \frac{\sum_{v=1}^{N_e} \delta\big(\kappa(u,v) > \theta\big) R_u(v)}{\sum_{v=1}^{N_e} \delta\big(kappa(u,v) > \theta\big)} \tag{4.10}$$

where $\delta\big(\cdot\big)$ is the indication function, equal to one if the condition $\big(\cdot\big)$ is true (otherwise 0), so as to include only similar elements. Figure 4.5 also illustrates how similar elements from other images provide the foreground/background cue.

Centerness Cue: In addition to the above mentioned saliency and foreground/background cues, we also take advantage of the general observation that objects are often located at the center, and such central bias is quite prevalent in several benchmark datasets as pointed out in [52]. Therefore as an extra measure, saliency maps that emphasize center regions are encouraged to be given higher weights at central regions. To account for central bias, a spatial weight mask for each image is created using normalized Gaussian function which is centered at the image center. Specifically, for a pixel p in I^n (of size $width_n \times height_n$) with coordinates (x, y) and with its origin at the image center, the central weight mask is defined as

$$\eta^n(p) = exp\Big(-\frac{x^2}{0.2 \times width_n^2} - \frac{y^2}{0.2 \times height_n^2}\Big). \tag{4.11}$$

For an element u or $e(n, k, m)$, its central bias is calculated by averaging the spatial weights of all its pixels, i.e.

$$\omega_u = \frac{\sum_{p \in P_k^n} \eta^n(p)}{\sum_{p \in P_k^n} 1}. \tag{4.12}$$

Let $\omega = [\omega_1, \omega_2, ..., \omega_{N_e}]^t$ denote the vector consisting of the central bias weights of all the elements. Thus, we now define the centerness cue Y_c for an element u as

$$Y_c(u) = \omega(u) \times |\omega(u) - T(u)|, \tag{4.13}$$

which essentially measures how the saliency of an element deviates from its central bias weight. Central bias weight is also multiplied so that influence of this deviation in minimizing Eq. (4.1) depends upon the spatial location of the element.

Note that our centerness cue is different from other methods like [25], which deliberately emphasize the center regardless of whether an object is present or not in the center. On the contrary, our method emphasizes the center only if a salient object is present in the center. Our centerness cue provides additional support when the saliency and foreground/background cues fail to recommend something substantial because of lack of support from other images due to too much intra-class variation or pose differences. In such case, if there is a salient object at the center, it will be supported by the centerness cue.

4.1.4 Smoothness Term

Since in our prior term we have made discrete conditions using θ to select similar elements, there is a certain possibility of inconsistencies in weight distribution. A smoothness term is necessary to curb inconsistencies in weight distribution among neighbor elements. Here we define neighbor elements as those which are similar in not only the feature space but also the saliency space. If a pair of elements have very similar saliency and are very close in the feature space as well, they should be encouraged to have similar weights. Thus, the smoothness term $\mathbf{z}^t G\mathbf{z}$ is introduced to ensure that these neighbor elements in both feature space and saliency space take similar weights. However, we use the conventional normalized Laplacian matrix for defining smoothness term coefficient G in Eq. (4.1), similar to [90], i.e.

$$G = A - \pi^{-\frac{1}{2}} V \pi^{\frac{1}{2}} \tag{4.14}$$

where A is the identity matrix, V is neighborhood matrix, and π is the diagonal matrix composed of row sums of matrix V. In addition, different from the similarity matrix S defined in Eq. (4.5), V takes into account similarity in both feature space and saliency space, i.e.

$$V(u,v) = exp\Big(-\gamma \frac{\sum_{q=1}^{2d} \frac{\left(X(u,q)-X(v,q)\right)^2}{X(u,q)+X(v,q)}}{2d} - |T(u) - T(v)|\Big) \tag{4.15}$$

where γ is a normalization parameter set to $\frac{1}{300}$, which is the same as that in Eqs. (4.3), (4.4) and (4.5).

4.1.5 Implementation Details

For optimization, since G is positive semi-definite and the constraints are linear, the objective function defined in (4.1) is essentially a quadratic programming problem, which is solved by the interior-point convex algorithm [5][93] provided in Matlab.

Once the fused saliency map is available, different single-image segmentation algorithms can be applied for segmentation. In this research, we adopt two segmentation methods as two variations. One is the classical Otsu's method, which is an optimal threshold based method. The other one is GrabCut algorithm [76] with some modification. Specifically, by noticing the final fused saliency map containing certain boundary information, following [28], we modify the GrabCut energy equation and add another localization potential to ensure that segmentation is guided not only by color, but also by the location prescribed by the object prior contained in the fused saliency map. The foreground (FG) and the background (BG) seed locations are determined by

$$p \in \begin{cases} FG, & \text{if } F^n(p) > \tau \\ BG, & \text{if } F^n(p) < \upsilon^n \end{cases} \tag{4.16}$$

where υ^n is a global threshold value automatically determined by the classical OtsuâĂŹs method and τ is a parameter (by default $\tau = 0.75$). It should be noted that other single-image segmentation methods such as [15] can also be used for the final segmentation.

4.2 Experimental Results

We conducted extensive experiments on five existing benchmark co-segmentation datasets (MSRC [88], iCoseg [7], Coseg-Rep [20], Internet image dataset [78], and FlickrMFC dataset [42]). As mentioned in the introduction, the existing methods often require fine tuning of quite a few parameters. In order to demonstrate the effectiveness of our method, we make two types of settings in our experiments: (1) default parameter settings for all the categories in the datasets and (2) tuning parameter τ over categories for a fair comparison with other methods. Same as the ones defined in previous chapter for the segmentation evaluation, we again adopt two evaluation metrics: (i) Jaccard Similarity (Jacc.) [31] and (ii) Accuracy (Acc.). Specifically, if A_p^f, A_p^b, A_g^f and A_g^b are denoted as proposed foreground pixels set, proposed background pixels set, groundtruth foreground pixels set and groundtruth background pixels set, respectively, the Jaccard Similarity is computed as $\frac{|A_p^f \cap A_g^f|}{|A_p^f \cup A_g^f|}$, and Accuracy is computed as $\frac{|A_p^f \cap A_g^f| + |A_p^b \cap A_g^b|}{|A_g^f \cup A_g^b|} \times 100$. We use eight saliency extraction methods [102][103][51][87][61][3][16][38] to generate various saliency maps as the input to our method. In the following subsections, we first briefly introduce the datasets used, followed by individual experiments, discussions and comparisons.

4.2.1 Datasets

MSRC, iCoseg, Coseg-Rep and Internet images datasets are the same ones discussed in the previous chapter. One thing to note here is that Coseg-Rep dataset contains a special category named “Repetitive" that has several instances of the same type of object within one image (e.g., an image containing multiple horses), and we evaluate how well the proposed method

Table 4.1: Evaluation on MSRC dataset using Jaccard-Similarity metric where individual saliency maps and fused saliency maps are segmented using Otsu's method

class	[102]	[103]	[51]	[87]	[61]	[3]	[16]	[38]	AVG	MAX	Ours
Car	0.541	0.466	0.381	0.469	0.510	0.598	0.507	0.629	0.660	0.666	**0.696**
Sheep	0.745	0.699	0.736	0.612	0.776	0.615	0.697	0.744	0.793	0779	**0.810**
Cow	0.670	0.670	0.673	0.603	0.734	0.658	0.653	0.736	0.742	0.729	**0.794**
Flower	0.705	0.679	0.556	0.627	0.694	0.625	0.641	0.688	0.721	0.726	**0.768**
Cat	0.439	0.526	0.560	0.597	0.573	0.565	0.539	0.609	0.651	0.624	**0.714**
Sign	0.746	0.619	0.552	0.567	0.646	0.669	0.570	0.796	0.775	0.743	**0.812**
Tree	0.636	0.655	0.471	0.400	0.632	0.601	0.561	0.606	0.681	0.669	**0.738**
House	0.586	0.613	0.486	0.389	0.528	0.630	0.450	0.640	0.670	0.669	**0.712**
Dog	0.527	0.559	0.503	0.520	0.469	0.452	0.503	0.627	0.628	0.582	**0.643**
Bird	0.529	0.590	0.573	0.535	0.590	0.459	0.583	0.611	0.644	0.589	**0.662**
Bike	0.377	0.416	0.297	0.3827	0.436	0.453	0.463	0.420	0.488	0.473	**0.548**
Chair	0.546	0.588	0.566	0.474	0.595	0.530	0.496	0.563	**0.638**	0.588	**0.638**
Face	0.515	0.411	0.395	**0.582**	0.463	0.446	0.367	0.548	0.565	0.567	0.571
Plane	0.420	0.437	0.399	0.297	0.505	0.505	0.475	**0.542**	0.535	0.469	0.518
Avg	0.570	0.566	0.510	0.504	0.582	0.558	0.536	0.626	0.656	0.634	**0.688**

can handle such a repetitive scenario. MSRC, Coseg-Rep and Internet images datasets exhibit intra-class variation. As a result, we do not use the color feature here for matching the elements as it will be unreliable. Flickr MFC dataset (which we didn't discuss earlier) contains multiple common objects that might not appear in every image. It has 14 categories and 263 images in total. For this one and iCoseg dataset, since the same objects appear frequently across the images, we include the color features in our method.

Note that we first perform k-means clustering using GIST descriptor [16] and the proposed saliency co-fusion is then applied to each cluster independently. This is to reduce the intra-class variation. Otherwise, a wide diversity might cause unnecessary difficulties in the co-fusion process. Empirically, we set the target cluster size to be 10, i.e. on average each cluster contains 10 images.

4.2.2 Performance Improvement by Co-fusion

The key point of our proposed saliency co-fusion process is to generate a fused saliency map that can better highlight the common object while suppressing the background saliency. To

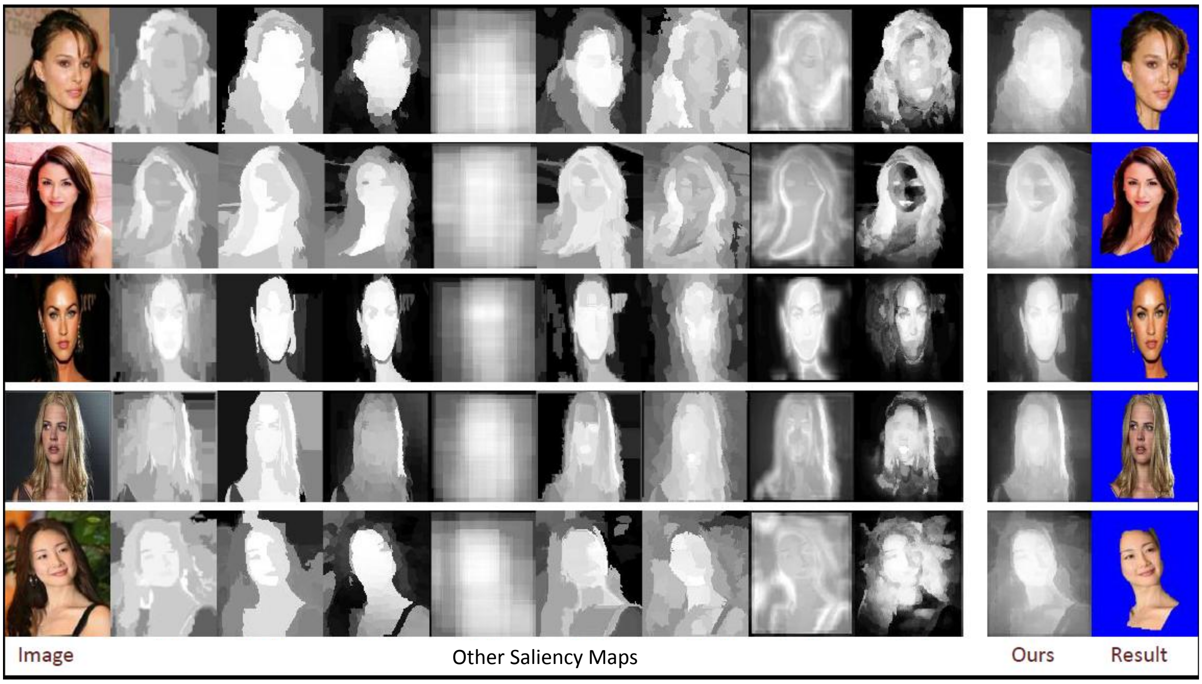


Figure 4.6: Examples to illustrate the advantages of the fused saliency maps over the input saliency maps.

compare the quality of the fused saliency map with other saliency maps, we apply the simple segmentation approach, Otsu's method, on individual saliency maps of images in MSRC dataset, and report segmentation results in Table 4.1. It can be seen that our method achieves about 10% gain over that of the best saliency extraction method [38]. Table 4.1 also shows the results of simple averaging or taking the maximum of those individual saliency maps at the pixel level also outperform the best single saliency map, clearly suggesting the advantage of using multiple saliency maps. Our method outperforms the simple average function and the max function by about 6% and 8%, respectively. Note that the Avg Jaccard Similarity value of 0.688 on MSRC dataset by using simple Otsu's method on the fused saliency map (without any parameter tuning) is even better than the result of 0.68 (see Table 4.6) obtained by [78] which used complex co-labelling, parameter tuning, and Grabcut.

Figure 4.6 shows the visual comparison of individual saliency maps used and our fused saliency map. It can be seen that pixels pertaining to the woman (the common object) obtain boosted saliency values, while the background regions get suppressed saliency values in the final fused saliency maps which lead to clean segmentation results. Figure 4.7 provides more

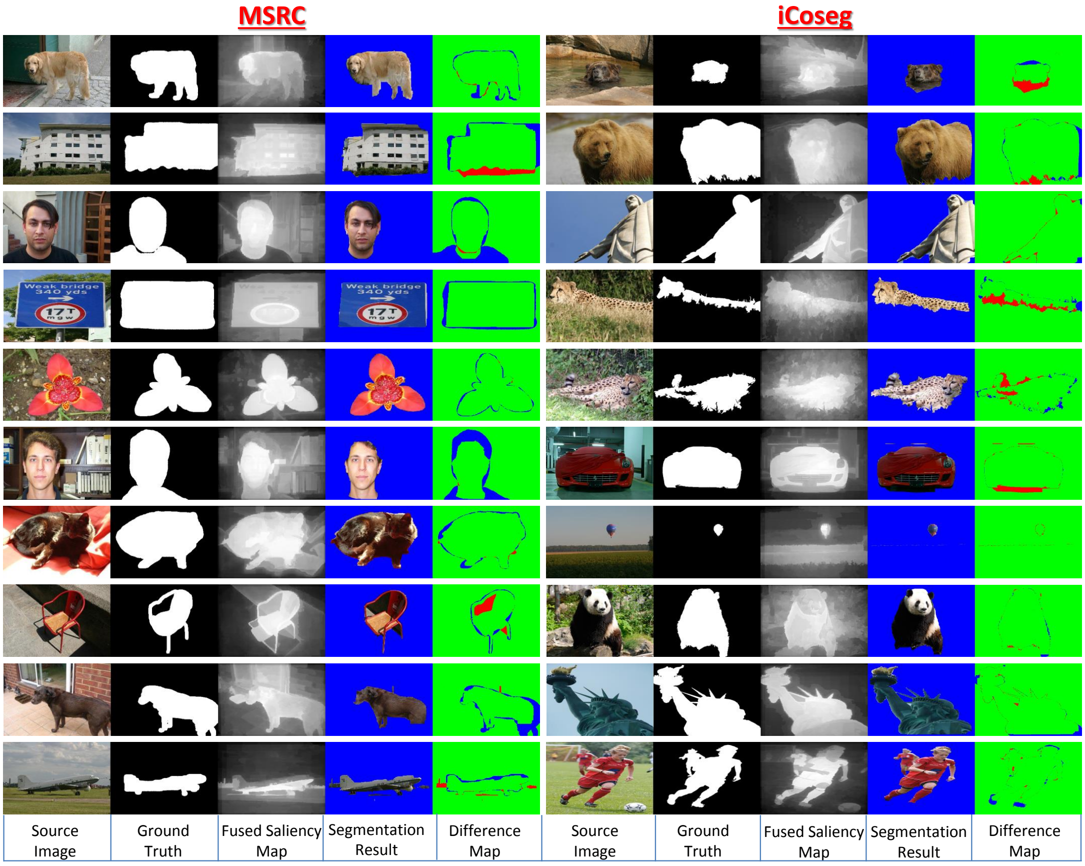


Figure 4.7: Sample examples of ground-truth images, fused saliency maps, our segmentation results and the difference maps (our results minus the corresponding ground-truth images) on MSRC and iCoseg datasets. Note that for the difference maps, green, red and blue correspond to 0, 1, -1, respectively.

Table 4.2: Overall Jaccard-Similarity (Jacc.) and Accuracy (Acc.) results on different datasets using our methods that respectively incorporate Otsu's method and GrabCut method with the default setting [$\tau = 0.75$] for segmentation.

	Otsu's method		GrabCut	
	Jacc.	Acc.	Jacc.	Acc.
MSRC	0.69	86.7	0.70	87.9
iCoseg	0.65	87.0	0.70	89.7
Coseg-Rep	0.71	89.5	0.76	92.7
Car	0.70	85.3	0.69	86.0
Horse	0.49	78.5	0.55	83.9
Airplane	0.52	82.6	0.56	86.8
FlickrMFC	0.60	83.5	0.67	87.0

Table 4.3: Performance results by varying region-size parameter of SLIC [1] on MSRC dataset

	20	40	60	80	100
Jacc.	0.6878	0.6877	0.6870	0.6869	0.6865
Acc.	86.31	86.30	86.27	86.26	86.25

examples of fused saliency maps and the corresponding segmentation results on MSRC and iCoseg datasets. Furthermore, it also shows the difference maps against the ground-truths.

4.2.3 Discussion on the Parameters

In Table 4.2, we report our results obtained by fixing the parameter τ in Eq. (4.16) to 0.75 on all the datasets with GrabCut segmentation, and also the results obtained using simple Otsu's method. Due to the fact that categories of Internet images dataset are quite large, their results on each category are separately shown. We can see that even the simple Otsu's method is able to produce decent results with our fused saliency maps. This can be attributed to the high-quality saliency maps produced by our saliency co-fusion approach. By using GrabCut for segmentation, the performance of our method can be further improved. For parameter λ in Eq. (4.1), we empirically set it to 9. Also, we empirically set parameter γ in Eqs. (4.3), (4.4), (4.5), and (4.15) to 1/300, and parameter θ in Eqs. (4.7) and (4.10) to 0.75. Parameter υ^n in Eq. (4.16) is automatically computed using Ostu's method.

In order to examine the sensitivity of our method on different superpixel extraction methods and different parameter settings, we further conducted experiments using irregular superpixels generated by [94]. The results on MSRC dataset show that use of the superpixels of [94] with the global thresholding achieves the average Jaccard similarity of 0.6876. However, this is almost same as the result of 0.6875 obtained by using SLIC [1]. We also vary the region-size parameter of SLIC [1]. By varying the region-size parameter of SLIC [1] from 20 to 100, the results can be seen in Table 4.3. It can be seen that the performance decreases only slightly with the increase of the region size. Therefore, these experiments indicate that the proposed method is robust to different super-pixel methods/settings.

Table 4.4: Comparison on Coseg-Rep dataset using overall values of Jaccard-Similarity (Jacc.) and Accuracy (Acc.)

	Jacc.	Acc.
Cosegmentaton&Cosketch [20]	0.67	90.2
Geometric Mean Saliency [36]	0.73	92.2
Ours (tuned)	**0.77**	**93.4**

4.2.4 Experiments for Comparison

For different datasets, we compare our method with the methods that report the state-of-the-art performance on the datasets. We denote "Ours (default)" as our method with the setting $\tau = 0.75$ using GrabCut while denoting "Ours (tuned)" as the one where we tune parameter τ with a step size of 0.03 from 0.60 to 0.99 over each category and report the best results, which is similar to other methods. Our method outperforms the state-of-the-art methods on two of the single object co-segmentation datasets (Coseg-Rep and Internet Images) as shown in Tables 4.4 and 4.5. Also, some sample visual results of our method on Coseg-Rep dataset and Internet images dataset are shown in Figure 4.8 and Figure 4.9, respectively.

Note that for the Internet image dataset, since each of its categories consists of large number of images, we tune parameter τ per cluster. It can be seen from Tables 4.4 and 4.5 that, in terms of Jaccard Similarity metric, our method achieves about 5% on Coseg-Rep dataset, 13%, 11%, and 9% improvements on Car, Horse and Airplane categories of Internet image dataset, respectively, when compared with the best results reported in [36] (our initial work from Chapter 3) and [78] for CosegRep and Internet Images datasets, respectively. Table 4.6 compares the results of our method with those of state-of-the-art methods on MSRC and iCoseg datasets. It can be seen that our results are competitive to the best one by [24], while our method is much faster than [24]. Specifically, running on the same PC with Intel Core i5-3470@3.20 GHz CPU and 32 GB RAM, [24] (using their own source codes in Matlab) takes 29.2 hours to complete the entire segmentation process on MSRC dataset. However, our method (also in Matlab codes) takes only 8.5 hours. These durations include the time taken for pre-processing steps as well like generating proposals in [24] and generating saliency maps in our method.

It is interesting to see that our method can also well handle Flickr MFC dataset that contains multiple common objects across the images and the repetitive category of Coseg-Rep dataset

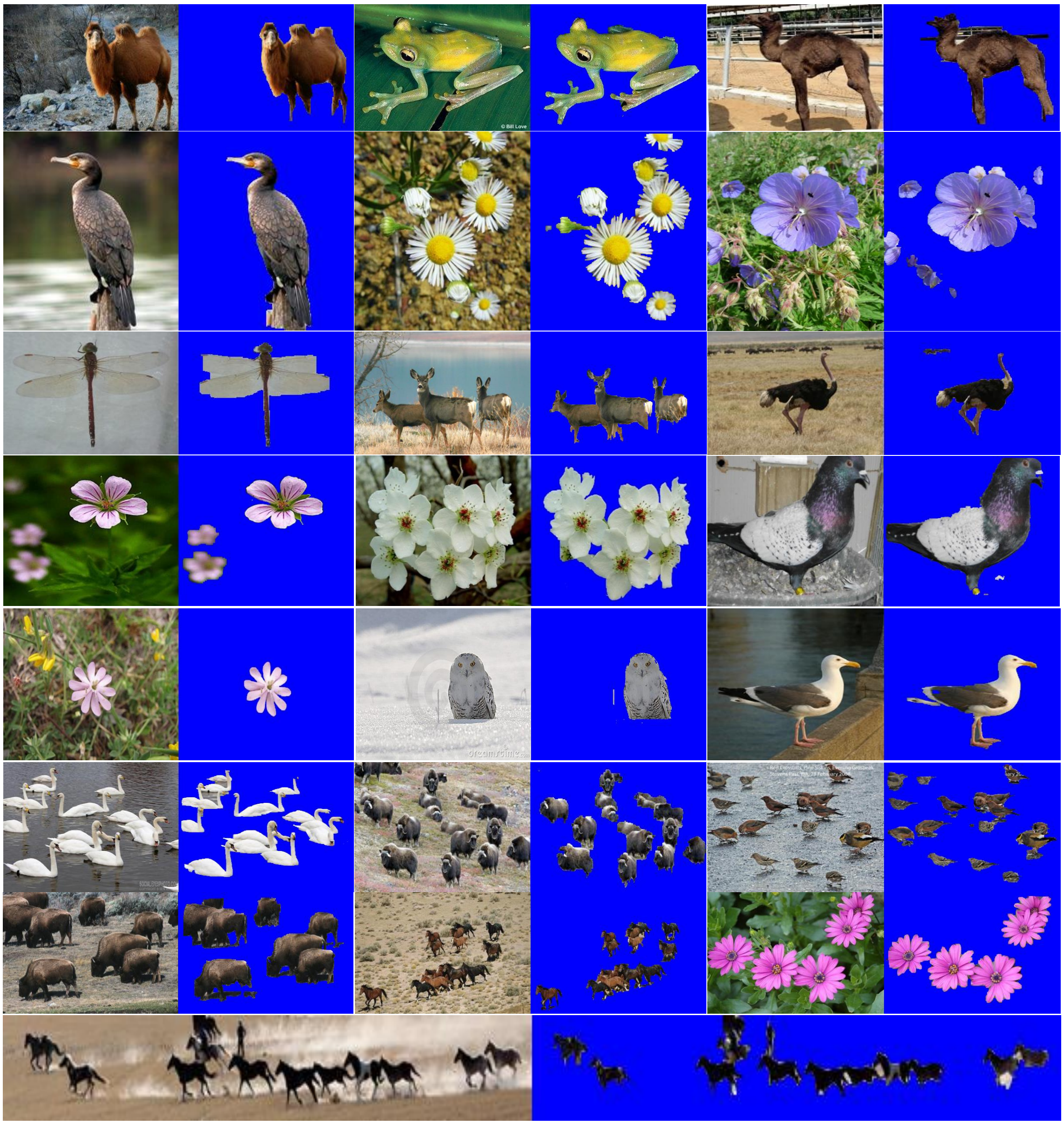


Figure 4.8: Sample segmentation results on Coseg-Rep dataset

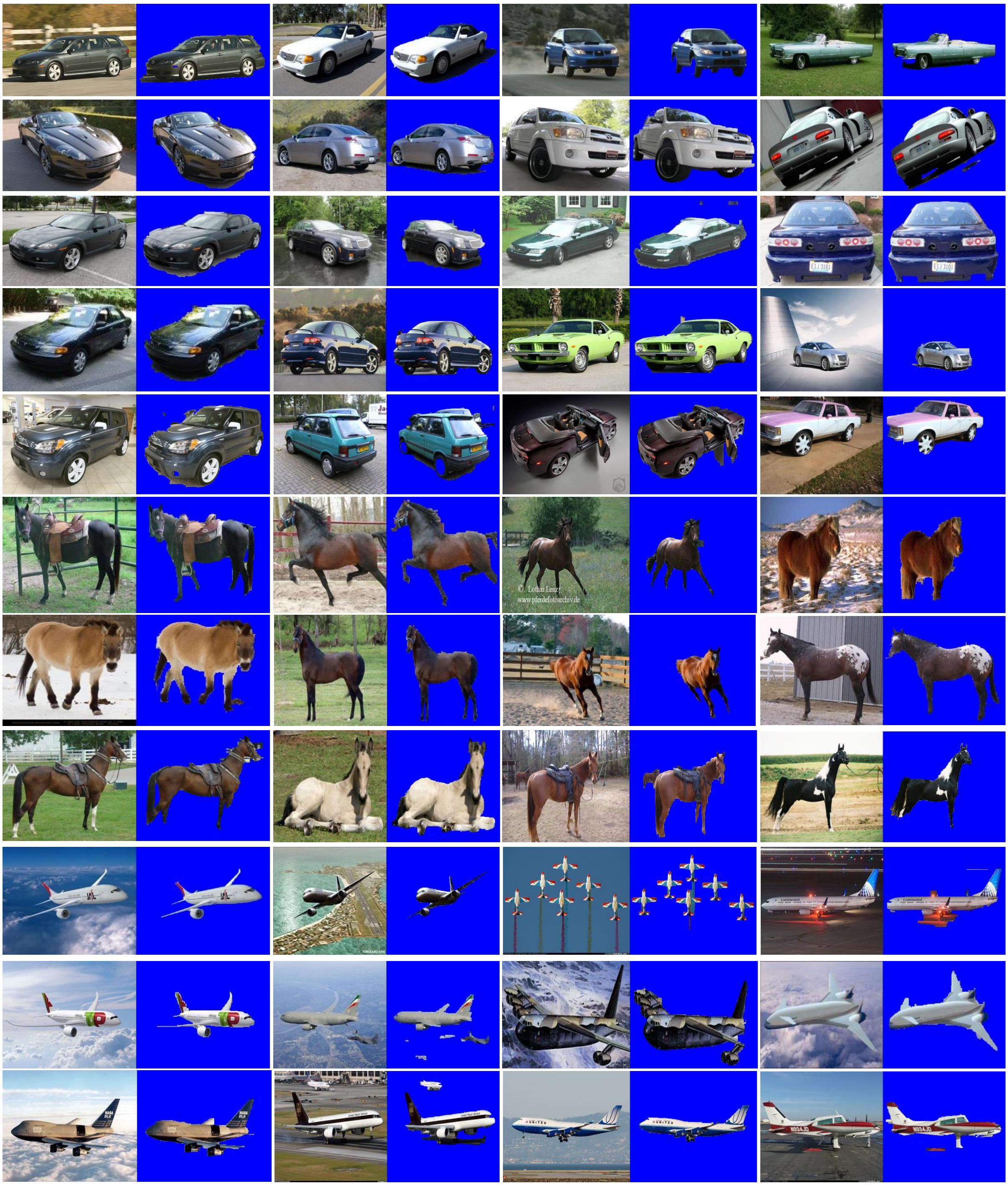


Figure 4.9: Sample segmentation results on Internet image dataset containing three categories:(i) Car, (ii) Horses and (iii) Airplane

Table 4.5: Comparison with state-of-the-art methods on Internet image dataset using overall values of Jaccard-Similarity (Jacc.) and Accuracy (Acc.)

	Car		Horse		Airplane	
	Jacc.	Acc.	Jacc.	Acc.	Jacc.	Acc.
[39] (reported in [78])	0.37	58.7	0.30	63.8	0.15	49.2
[40] (reported in [78])	0.35	59.2	0.29	64.2	0.12	47.5
[78]	0.63	83.4	0.54	83.7	0.56	86.1
Ours (default)	0.69	86.0	0.55	83.9	0.56	86.8
Ours (tuned)	**0.71**	**88.0**	**0.60**	**88.3**	**0.61**	**90.5**

Table 4.6: Comparison with state-of-the-art methods on MSRC and iCoseg datasets using overall values of Jaccard-Similarity (Jacc.) and Accuracy (Acc.)

	MSRC		iCoseg	
	Jacc.	Acc.	Jacc.	Acc.
Discriminative [39]	0.45	70.8	0.39	61.0
Multi-Class [40]	0.51	73.6	0.43	70.2
Object Discovery [78]	0.68	87.7	0.69	89.8
Geometric Mean Saliency [36]	0.70	88.4	0.72	91.6
Composition [24]	**0.73**	**89.2**	**0.73**	**92.8**
Ours (tuned)	0.71	88.7	0.72	91.9

Table 4.7: Comparison on FlickrMFC dataset using overall Jaccard Similarity (Jacc.) value. (U) means unsupervised method and (S) means Supervised method

Methods	Jacc.
Multiple Foreground Cosegementation (U) [42]	0.322
Multiple Foreground Cosegementation (S) [42]	0.482
Discriminative Clustering (U) [40]	0.414
Directed Graph Clustering (U) [69]	0.547
Graph Transduction (S) [59]	0.626
w/o NON RIGID Mapping (U) [56]	0.589
with NON RIGID Mapping (S) [56]	0.647
Ours (U) (default)	0.667
Ours (U) (tuned)	**0.684**

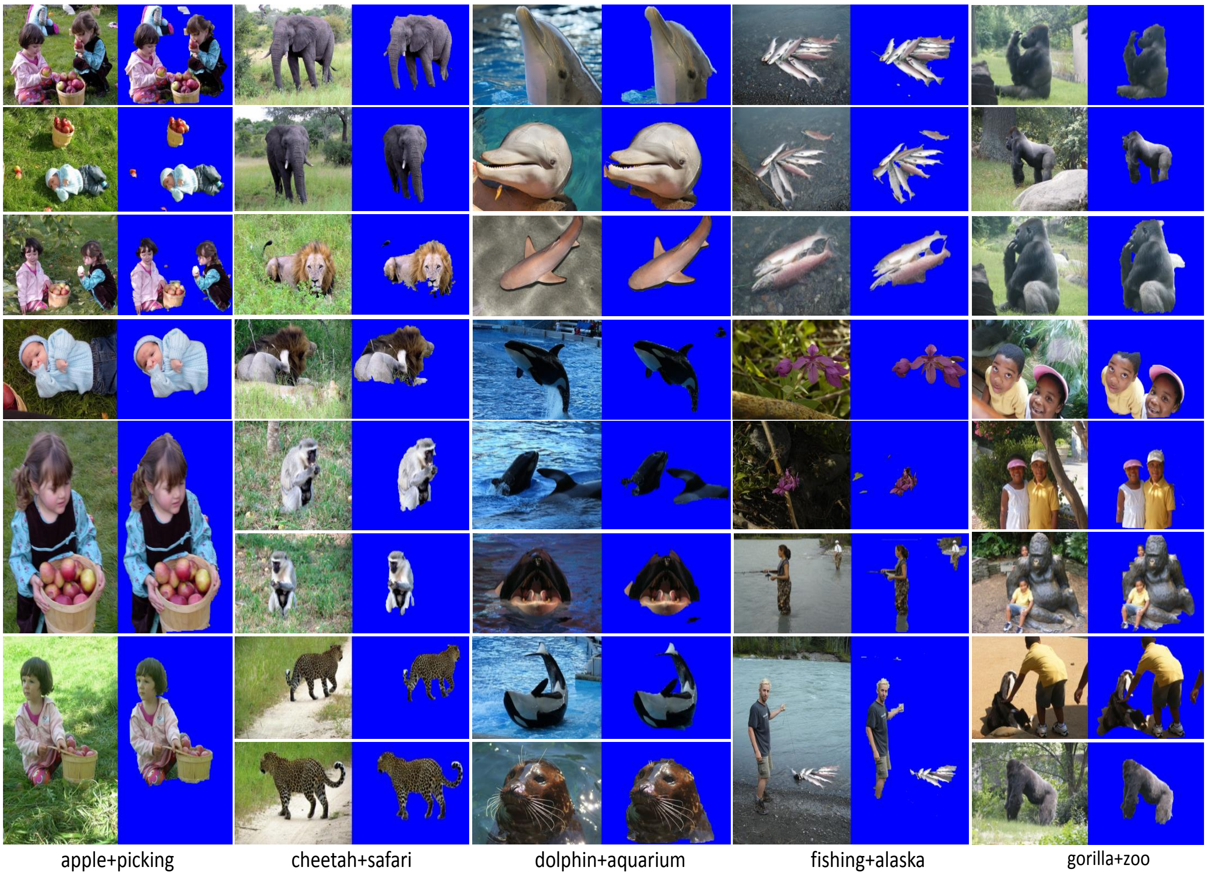


Figure 4.10: Sample segmentation results on FlickrMFC dataset.

Table 4.8: Class-wise Jaccard Similarity performance on MSRC dataset

	car	sheep	cow	flower	cat	sign	tree	house	dog	bird	bike	chair	face	plane
[78]	0.667	0.789	0.794	0.714	0.662	0.823	0.699	0.727	0.675	0.673	0.541	0.622	0.583	0.567
[36]	0.704	0.799	0.801	0.723	**0.760**	0.839	**0.772**	0.764	0.683	0.628	0.462	0.650	0.604	0.543
[24]	0.710	**0.850**	**0.880**	**0.790**	0.700	**0.850**	0.760	**0.840**	0.690	**0.680**	**0.580**	**0.730**	**0.630**	**0.580**
ours	**0.713**	0.811	0.812	0.770	0.734	0.831	0.769	0.752	**0.699**	0.665	0.544	0.671	0.608	0.552

Table 4.9: Class-wise Jaccard Similarity performance on Coseg-Rep dataset

	repet	blue	camel	cormo	cranes	deer	desert	dragon	egret	fire	flea	forget
	-itive	-flagris		-rant	-bill		-rose	-fly		-pink	-bane	-menot
[20]	0.754	0.890	0.641	0.493	0.842	0.450	0.880	0.380	0.463	**0.902**	**0.888**	**0.867**
[36]	0.747	0.823	0.688	0.592	0.854	0.634	0.826	**0.550**	0.499	0.781	0.829	0.842
ours	**0.776**	**0.903**	**0.702**	**0.613**	**0.863**	**0.636**	**0.841**	0.542	**0.601**	0.884	0.851	0.849
	frog	geran	ostrich	pear	piegon	seagull	seastar	silen	snow	white	wild	
		-ium		blossom				-clorata	-owl	campion	beast	
[20]	0.484	0.897	0.605	0.777	0.427	0.464	0.631	**0.835**	0.355	0.739	0.839	
[36]	0.714	0.852	0.668	0.775	0.624	0.681	0.762	0.766	0.736	0.794	0.776	
ours	**0.741**	**0.912**	**0.747**	**0.791**	**0.675**	**0.719**	**0.821**	0.828	**0.748**	**0.901**	**0.877**	

Table 4.10: Class-wise Jaccard Similarity performance on iCoseg dataset

	base	bear2	brown	cheetah	Christ	elephant	ferrari	goose	gymna	gymna	gymna	helico
	ball		bear						-stic1	-stic2	-stic3	-pter
[78]	0.657	0.653	0.736	0.697	0.770	0.688	**0.724**	0.742	0.948	0.839	0.896	0.803
[36]	**0.756**	0.701	0.662	0.754	0.795	0.735	0.703	0.773	0.910	**0.897**	**0.911**	0.766
[24]	0.610	**0.720**	**0.920**	0.670	**0.870**	0.670	0.680	**0.870**	0.970	0.820	0.900	**0.820**
ours	0.703	0.675	0.725	**0.780**	0.757	**0.799**	0.708	0503	**0.976**	0.831	0.892	0.803
	liver	monk	panda	panda	pyramid	skate	skate	skate	statue	stone	taj	track&
	pool		1	2			2	3		-henge	mahal	field
[78]	**0.541**	0.681	0.759	0.625	0.611	0.735	**0.910**	0.449	0.799	0.595	0.460	0.519
[36]	0.512	0.688	**0.806**	**0.718**	**0.686**	0.737	0.866	0.297	0.813	0.714	0.587	0.632
[24]	0.470	**0.800**	0.700	0.550	0.580	**0.910**	0.690	0.160	0.770	**0.910**	**0.840**	**0.660**
ours	0.470	0.683	0.722	0.614	0.595	0.769	0.900	**0.491**	**0.863**	0.781	0.516	0.595
	hotba	kendo	kendo2	wind	women	women						
	-lloon			mill	soccer	soccer2						
[78]	0.657	0.778	0.826	0.492	0.661	0.530						
[36]	0.763	0.862	0.893	0.316	0.657	**0.538**						
[24]	**0.880**	0.890	**0.960**	**0.570**	0.660	0.460						
ours	0.802	**0.896**	0.921	0.531	**0.699**	0.526						

Table 4.11: Class-wise Jaccard Similarity performance on FlickrMFC dataset

	apple picking	baseball kids	butterfly blossom	cheetah safari	cow pasture	dog park	dolphin aquarium	fishing alaska	gorilla zoo	liberty statue	parrot zoo
[59]	0.540	0.640	0.620	**0.850**	0.580	0.550	0.580	0.320	0.570	**0.900**	0.450
[56]	0.661	0.655	0.641	0.683	0.586	0.570	0.618	0.449	0.609	0.563	0.590
ours	**0.720**	**0.783**	**0.729**	0.800	**0.694**	**0.700**	**0.717**	**0.663**	**0.631**	0.614	**0.640**
	stone henge	swan zoo	thinker robin								
[59]	**0.960**	0.360	**0.840**								
[56]	0.476	0.504	0.642								
ours	0.594	**0.604**	0.682								

that contains repeated instances of objects, as shown in Tables 4.7 and 4.9, respectively. Our method with tuning per category outperformed the best one [56] (with supervised information) by 6% in terms of Jaccard similarity metric despite being an unsupervised method. In fact, our method's default setting itself outperforms the state-of-the-art method on Flickr MFC dataset. It should be noted that the comparison here is in terms of foreground/background segregation, and not multi-label segmentation. Figure 4.10 shows some sample segmentation results in such multiple-foreground scenario. It can be seen that although different multiple objects are present in one category of the dataset, our method successfully extracts the foreground. As far as the repetitive scenario is concerned, our method obtained a Jaccard Similarity value of 0.776 in comparison to 0.754 obtained by [20] on the repetitive category of the Coseg-Rep dataset (see bottom three rows of Figure 4.8 for such sample visual results).

Tables 4.8-4.11 list out the detailed Jaccard similarity results of our method as well as the state-of-the-art methods on individual categories of the four datasets. It was seen earlier that our method performs worse than [24] on MSRC and iCoseg datasets as far as the overall average performance is concerned. The main reason could be that our method relies on saliency co-fusion. If the common object cannot be identified as salient by any of the saliency extraction methods, our method would not be able to segment it out. Interestingly, these tables reveal that despite such slightly inferior overall performance, our method outperforms [24] in 4 out of 14 and 15 out of 30 categories in MSRC and iCoseg datasets, respectively. Figure 4.11 gives some visual examples of those categories, where our results look better than those of [24].

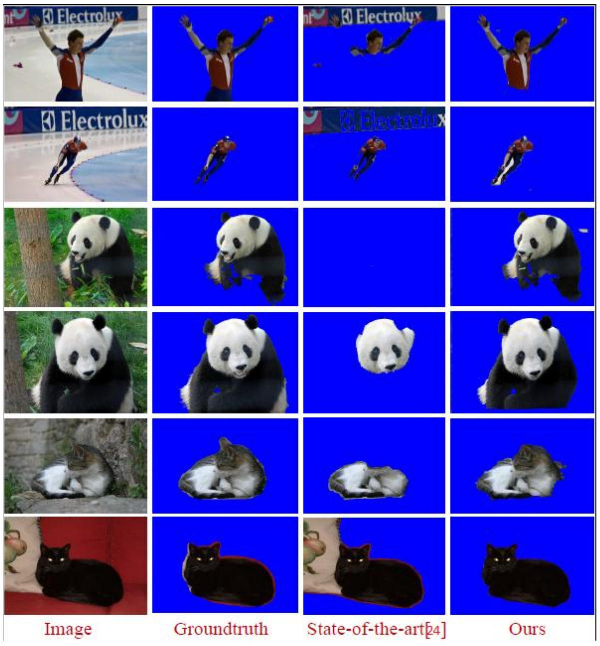


Figure 4.11: Sample segmentation results where our method outperforms the state-of-the-art[24].

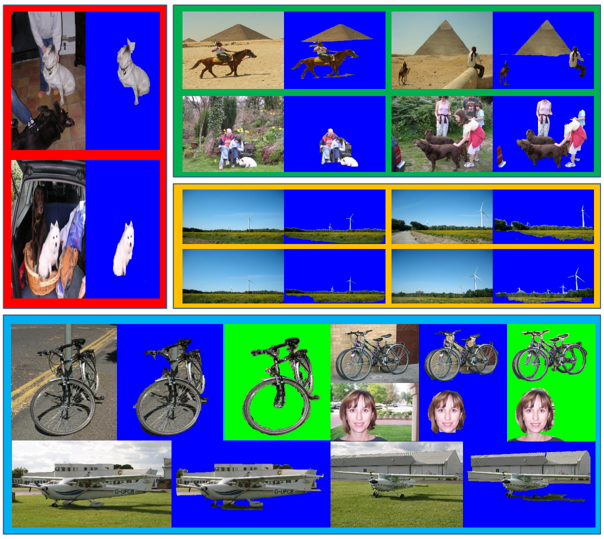


Figure 4.12: Failure cases: Our method fails (i) (red-box) when a common object (black dog) is not salient in any of the saliency maps; (ii) (green-box) when multiple foregrounds are present and the goal is to extract a particular foreground and (iii) (yellow-box) when very similar images are grouped for the co-segmentation process. (iv) (blue-box): Examples to show the limitations of our method in some specific categories in MSRC where our methods tend to segment convex shapes instead of thin rods in the bike class, miss segmenting the unsalient shoulder in the face class, and include the airport in the airplane class. NOTE: Segmentations with blue background are our results and those with green background are the ground-truth results.

4.2.5 Limitations and Discussions

Although our method performs well on the benchmark datasets in general, there are some failure cases: (i) As shown in the red-box of Figure 4.12, our method only segments out one dog and misses the other. This is because one of the dogs is extremely salient in all the saliency maps, while the other dog is not very salient in any of the saliency maps. (ii) Another case is when there are multiple common salient objects in the images, while the goal of benchmark dataset is to segment out only one common object. For such case, our method will segment out all the salient common objects as shown in the green-box of Figure 4.12. (iii) Similar to almost all the co-segmentation methods, our method requires sufficient background variations across the images in one cluster. If very similar images are being included in one cluster, our method will fail to distinguish background from the foreground, as illustrated in the yellow-box of Figure 4.12.

The blue box in Figure 4.12 gives some class-specific examples where our method does not perform well. For example, (a) Bicycles in the bike category need segmentation of thin rods and tyres whereas our method segments such bicycles into convex shapes such as triangles and disks due to using GrabCut; (b) Our method misses segmenting out shoulders in most of the images in the face category, because shoulders are not so salient; and (c) Many images in the plane category also include airports along with the planes, thus making it difficult to segment out the planes clearly.

4.3 Summary

In this chapter, we have proposed a novel saliency co-fusion approach for the purpose of image co-segmentation which uses the association of similar images to fuse multiple saliency maps of the same image in order to boost up common foreground saliency and suppress background saliency. Experimental results on five benchmark datasets show that our method while co-fusing eight different saliency maps, achieves very competitive performance, compared to the state-of-the-art methods of image co-segmentation.

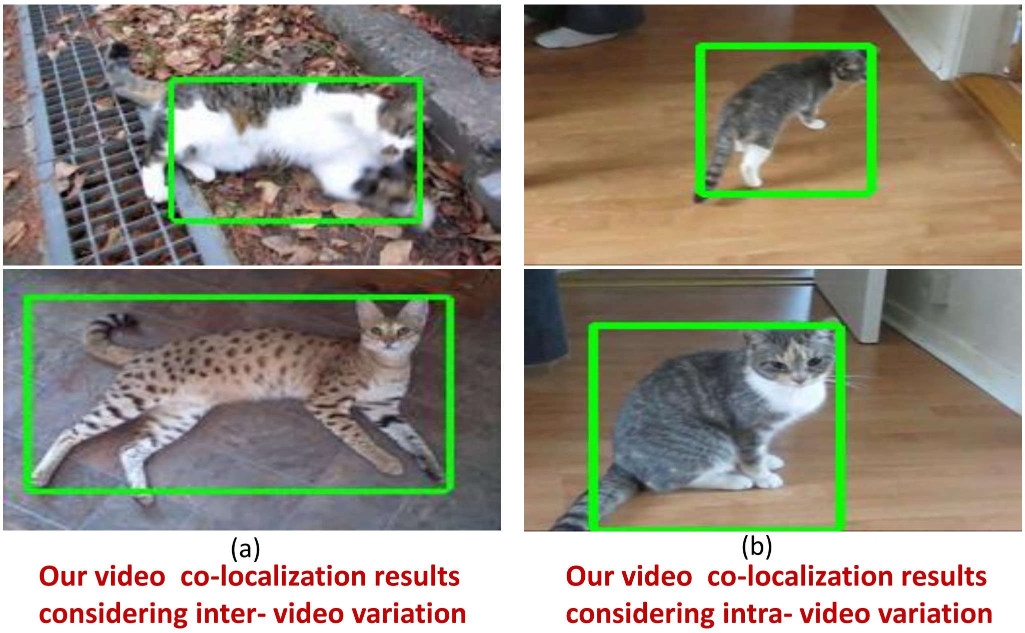
Chapter 5

CATS: Co-saliency Activated Tracklet Selection for Video Co-localization

Localizing the common object in a video is an important task in computer vision since it facilitates many other vision tasks such as object recognition and action recognition. Recent research interests have been shifted from single-video object localization to video co-localization [41, 45], which aims at jointly localizing common objects across videos by exploiting shared attributes among videos as weak supervision.

Video co-localization is a challenging problem due to the following reasons. First, for a large diverse video dataset, it is non-trivial to discover the related videos that contain semantically similar objects. Second, even for videos from the same semantic class, their common objects may exhibit large inter-video variations (see Figure 5.1(a)). Third, even within one video, objects could also have large variations due to viewpoint/pose changes (see Figure 5.1(b)).

A few video co-localization works [41, 75] have been proposed in literature. In particular, [41] proposed to co-select bounding box proposals, and [75] proposed to co-select tubes across the videos. Both methods try to localize common objects in multiple videos simultaneously. Surprisingly, such joint processing methods did not outperform the individual video processing based framework [74]. One reason could be the inability of both methods to handle large variations of objects across the videos in the same class. Such an observation that co-processing might not be better than individual processing has also been reported in some relevant studies [41, 78, 97]. This motivates us to propose a framework to divide video co-location into two


(a)
Our video co-localization results considering inter- video variation

(b)
Our video co-localization results considering intra- video variation

Figure 5.1: Variations of cats (a) across the videos as well as (b) within the video make the co-localization problem very challenging.

steps: exploiting inter-video relationship to find the common object prior and then locating the common object separately in each individual video, in other words, we propose a guided single video-based framework. Similar to this idea, recently [45] developed a two-step framework for video co-localization, where they iteratively discover common objects across neighboring videos and then incorporate the prior into individual video localization. However, [45] relies on bounding box proposals independently extracted at every sampled frame, which itself could be quite noisy.

Instead of relying on large number of bounding box proposals, in this chapter we propose to leverage *co-saliency activated tracklets* for video co-localization. In particular, we first explore inter-video commonness, intra-video commonness, and motion saliency to generate the co-saliency maps and then fuse them to extract object prior masks for uniformly sampled key frames. We then make use the object prior to select only a small set of proposals at each key frame and use them to activate the tracklets to be generated across subsequent frames. Finally, we separately generate the best tube for each video by selecting optimal tracklets based on confidence and consistency between adjacent tracklets using dynamic programming.

Experimental results on the benchmark YouTube Object dataset show that our proposed method outperforms state-of-the-art methods.

We would like to point out that our work is also motivated by benefits of co-saliency and tracklets. Co-saliency research [107][98][57] has recently demonstrated significant contribution in object discovery problems. On the other hand, tracklets developed through trackers [58][62][4][63][64][6] are quite spatio-temporally consistent and reliable already for short video intervals. In addition, tracklet processing is much more efficient than bounding box based processing [41] [45].

The main contributions of this chapter are twofold: 1) exploring inter-video, intra-video and motion information for tracklet activations; 2) leveraging tracklets for video co-localization.

5.1 Proposed Method

Our framework consists of three major steps: co-saliency based object prior generation, tracklet activation and generation, and tube generation, as shown in Figure 5.2. First of all, each video is uniformly cut into short-interval video trunks and in each video trunk, we generate some tracklets, each of which is a sequence of bounding boxes across consecutive frames, hoping to locate the common object with high recall. Since each tracklet needs an initial bounding box at its starting frame (we call such starting frame an activator), the first step of our framework is to generate a co-saliency map for each activator so as to provide some object prior information. The second step is to make use of the object prior mask to generate good initial bounding boxes and the corresponding tracklets, from which we generate a set of tracklets between every two adjacent activators. Finally, the third step of our framework is to select one tracklet per set to form a tube which localizes the object. We name our framework *co-saliency activated tracklet selection* (CATS).

5.1.1 Co-saliency Based Object Prior Generation

To generate good object prior, our basic idea is to combine the following three type of co-saliency. 1) Inter-video co-saliency: since one video of a common object often contains similar background, it is needed to introduce other videos of similar objects that are likely to have

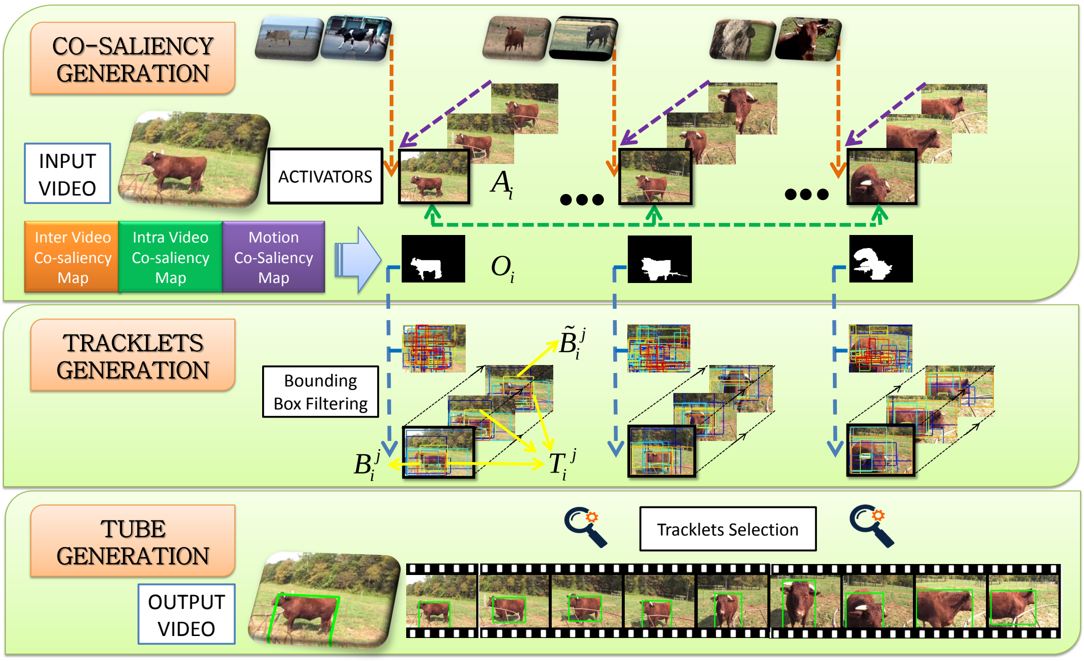


Figure 5.2: Overview of the proposed *co-saliency activated tracklet selection* (CATS) for video co-localization, which consists of three main components: co-saliency generation, tracklet generation and tube generation. NOTE: 3 different co-saliency processes are represented in 3 different colors: (1) inter-video (orange), (2) intra-video (green), and (3) motion (violet). Bounding boxes of same color across a video trunk denote a tracklet.

different backgrounds. Thus, we exploit the activators from different videos of similar objects to obtain inter video co-saliency. 2) Intra video co-saliency: Sometimes the activators from the same video could also contain diverse backgrounds, from which we could highlight intra-video co-saliency. 3) Total motion saliency: Since motion clues are always critical for video analysis, we want to use motion to identify co-saliency among consecutive frames. Once the three co-saliency maps are obtained, we fuse them by averaging followed by segmentation to obtain a co-saliency based object mask for each activator for the subsequent tracklet generation.

Inter video co-saliency: Let $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ be a set of n activators (uniformly sampled) in a video $\mathcal{V}$ such that $\mathcal{A} \subseteq \mathcal{V}$, where $\mathcal{V}$ is the set of all the frames in the video. Let $\mathbb{V}$ be the set of similar videos (containing a similar semantic object) such that $\mathcal{V} \in \mathbb{V}$. For each activator, say A_i, we search for its matched activators from other videos in $\mathbb{V}$ to create a externally matched activators set $\mathcal{N}_i^{ext} = \{\mathbb{A} | \zeta(A_i, \mathbb{A}) < \epsilon, \mathbb{A} \in \mathbb{V} \backslash \mathcal{V}\}$, where $\mathbb{A}$ denotes externally matched activator and ζ denotes distance function. Particularly, we extract the GIST descriptor [72] from each activator weighted by its initial saliency map [38]. The distance $\zeta(A_i, \mathbb{A})$ between a pair of activators is measured as the l_2 distance between their weighted GIST features. Such distance computation is essentially to find the activators that contain similar saliency regions. For an activator A_i, once its externally matched activators set $\mathcal{N}_i^{ext}$ is obtained, we compute the inter-video co-saliency M_i^{ext} as

$$M_i^{ext} = \frac{\mathcal{S}(A_i) + \sum\limits^{\mathbb{A}\in\mathcal{N}_i^{ext}} \mathcal{W}_{\mathbb{A}}^{A_i}\big(\mathcal{S}(\mathbb{A})\big)}{|\mathcal{N}_i^{ext}| + 1} \tag{5.1}$$

where $\mathcal{S}(\cdot)$ denotes the initial saliency map filter, $\mathcal{W}_{\mathbb{A}}^{A_i}\big(\ \cdot\ \big)$ denotes warping function from $\mathbb{A}$ to A_i, and $|.|$ denotes cardinality. We use the masked dense SIFT correspondence (SIFT flow) [55, 78] to find pixel correspondences for the warping. Eq. (5.1) essentially computes the joint saliency of the matched object points in different activators by such average of own saliency and warped saliency maps. If $\mathcal{W}_{\mathbb{A}}^{A_i}(\mathbb{A}(p)) = A_i(p + \xi(p))$,

$$\begin{aligned}\varphi(\xi; \mathcal{S}(\mathbb{A}), \mathcal{S}(A_i)) = \sum_{p\in domain(A_i)} & \mathcal{S}(\mathbb{A}(p))(\mathcal{S}(A_i(p + \xi(p)))) \\ ||\Omega_{\mathbb{A}}(p) - \Omega_{A_i}(p + \xi(p))||_1\} & + (1 - \mathcal{S}(A_i(p + \xi(p)))L_0 \\ + \sum_{q\in neighbor(p)} & \alpha||\xi(p) - \xi(q)||_2)\end{aligned} \tag{5.2}$$

where warping has been weighted by the available saliency maps to match salient pixels well. While Ω denotes SIFT feature vector, L_0 is just a large number.

Intra video co-saliency: We obtain intra-video co-saliency in a similar way as that for inter-video co-saliency. Particularly, we first group the activators in one video into different clusters using k-means based on weighted GIST descriptor as discussed before. Then, for an activator, other activators in its cluster are considered as its matches. Therefore, internally matched activators set $\mathcal{N}_i^{int} = \{A_j | A_j \in Z_k \backslash A_i, A_i \in Z_k\}$ is basically all other activators in cluster Z_k to which A_i belongs after the clustering. The intra-video co-saliency M_i^{int} for activator A_i is also computed as the average of its own saliency and the warped saliency maps

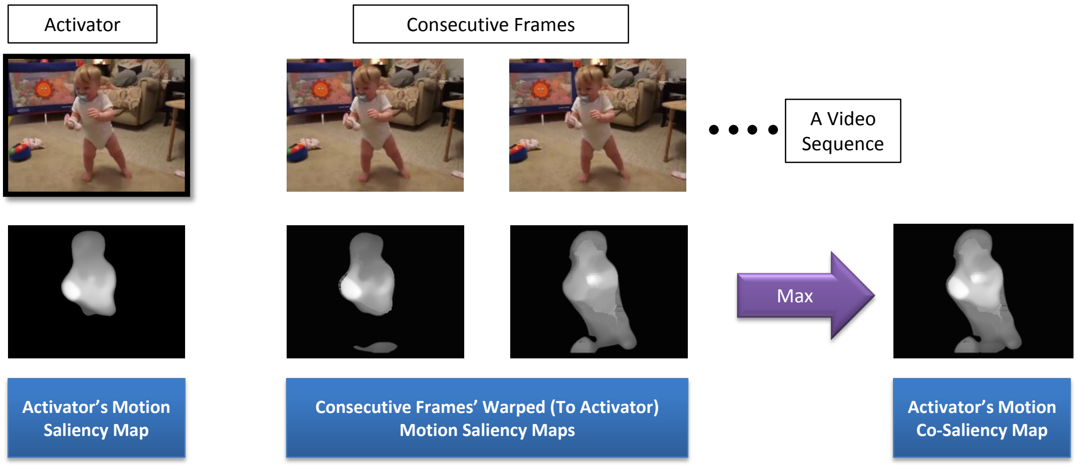

Activator
Consecutive Frames
A Video Sequence
Max
Activator's Motion Saliency Map
Consecutive Frames' Warped (To Activator) Motion Saliency Maps
Activator's Motion Co-Saliency Map

Figure 5.3: Motion co-saliency: Considering non-rigid object motion, max pooling motion saliency of different parts at different frames help develop a proper object prior.

of its matches, i.e.

$$M_i^{int} = \frac{\mathcal{S}(A_i) + \sum^{A_j \in \mathcal{N}_i^{int}} \mathcal{W}_{A_j}^{A_i}(\mathcal{S}(A_j))}{|\mathcal{N}_i^{int}| + 1} \tag{5.3}$$

where definitions of $\mathcal{S}$ and $\mathcal{W}$ remain same as defined previously. Here, for applying SIFT flow to find pixel correspondences for warping, we use not only SIFT feature but also color features (RGB, HSV, and Lab) since the common object in one video is likely to be of similar color.

Motion Co-saliency: For an activator, many subsequent frames are generally similar to it, typically with some variations due to object movements. We adopt the $\omega - flow$ method in [33] to extract the motion saliency map for each frame in a video trunk. Considering that for deformable objects, parts of the object could move while other parts might remain still (see Figure 5.3 for example), we propose to use max pooling to collect motion saliency from an activator and its consecutive frames after warping, which we call *motion co-saliency* M_i^{mot}, defined as

$$M_i^{mot} = \max\left(\mathcal{M}(A_i), \max_{I_j \in \mathcal{N}_i^{mot}} \left(\mathcal{W}_{I_j}^{A_i}(\mathcal{M}(I_j))\right)\right) \tag{5.4}$$

for activator A_i, where $\mathcal{M}$ denotes the motion saliency filter, set $\mathcal{N}_i^{mot} = \{I_j | I_j \in \mathcal{V}[A_i, A_{i+1}]\}$ denotes consecutive frames of activator A_i, i.e. between A_i and A_{i+1}, and $\max(\cdot)$ denotes pixel-level maximum function. Figure 5.4 shows why max pooling is preferred over average

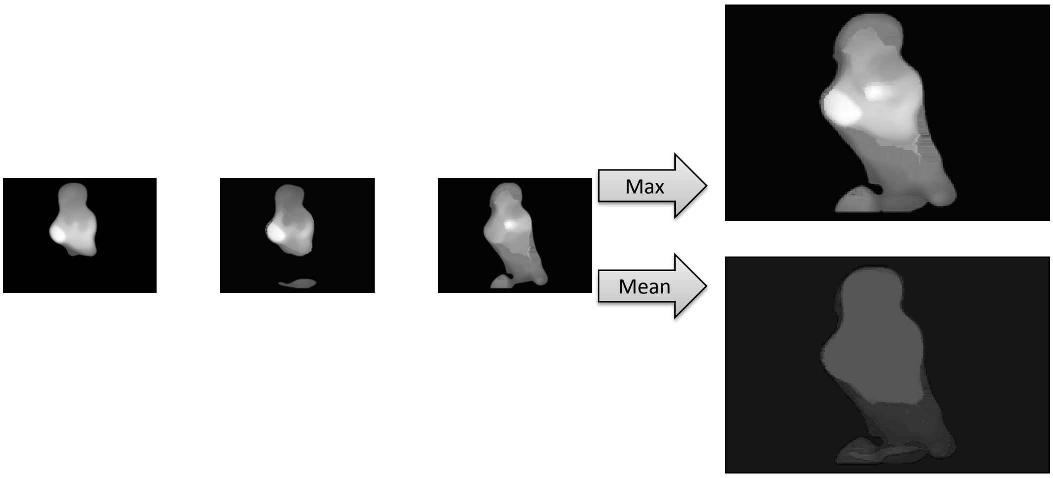


Figure 5.4: Max pooling is preferred over average pooling. The important motion clues of objects (such as legs) can be missed by average pooling because that clue may not be present in every subsequent frame.

pooling. The important motion clues of objects (such as legs) can be missed by average pooling because that clue may not be present in every subsequent frame.

Generating object prior: We simply fuse the three co-saliency maps, namely inter video co-saliency map (M_i^{ext}), intra video co-saliency map (M_i^{int}) and motion co-saliency map (M_i^{mot}), through averaging so that possible saliency defects which may exist in the individual maps can get subdued in the fused one. Once the final fused co-saliency map is available (see Figure 5.5 for examples), we apply the GrabCut [76] to obtain a binary segmentation mask, denoted as object prior O_i, for activator A_i.

5.1.2 Tracklet Activation and Generation

Bounding box filtering: We need an initial bounding box at the activator to activate a tracklet which then ends at next activator. Following state-of-the-art methods [45, 74], we also use bottom-up object proposal techniques, particularly [2], to generate initial bounding boxes. However, to ensure a high object detection rate, the existing general object proposal technique typically requires to generate at least hundreds of proposals, which makes the subsequent tracklet generation and tube generation infeasible. Thus, we propose to make use of our generated co-saliency based object prior to greatly trim down a large number of object proposals.

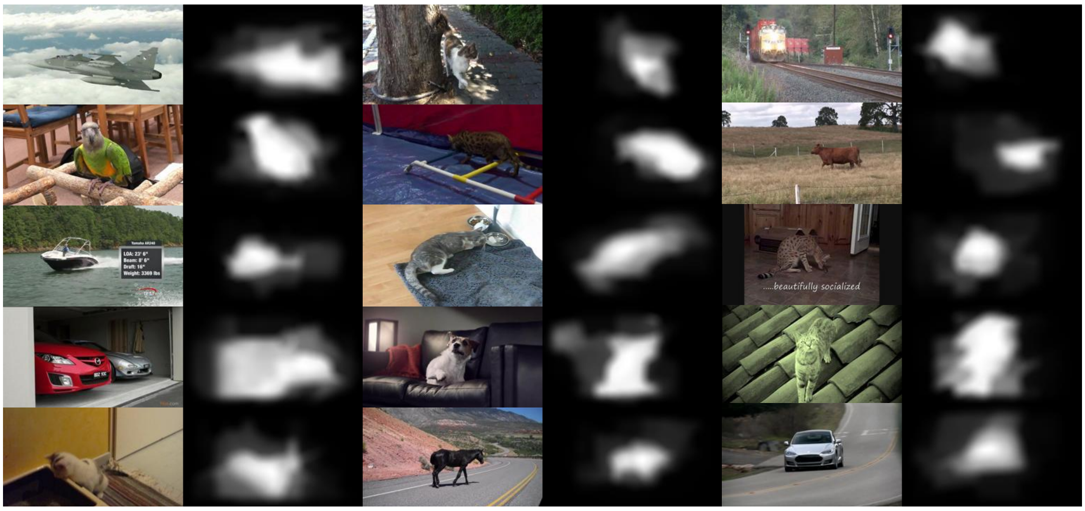


Figure 5.5: Final fused co-saliency maps for some activator samples in YouTube-Object dataset

Particularly, we rank each object proposal by its objectness score [2] and its overlap with the tight bounding box of the co-saliency based object prior. Let B_i^o denote the tight bounding box of the largest component in the object prior O_i and B_i^j be an object proposal in activator A_i. We calculate an object confidence score Ω for proposal B_i^j as

$$\Omega(B_i^j) = \Omega_o(B_i^j) + J(B_i^j, B_i^o) \tag{5.5}$$

where $\Omega_o(B_i^j)$ is the objectness score (between 0 and 1) directly obtained from [2] and $J(\cdot)$ is Jaccard similarity function (also called IoU, intersection over union). We then select the top-m proposals with highest confidence scores.

Tracklet confidence scores: Once m candidate bounding box proposals are selected at the activator, tracklets are obtained using the existing tracker [6] starting from these proposals at the activator and ending at the next activator, which we call *co-saliency activated tracklets*. Let T_i^j denote a tracklet activated at A_i by B_i^j and ending at A_{i+1} with bounding box $\tilde{B}_i^j$. To facilitate the subsequent tube generation via tracklet selections, for tracklet Δ_i^j, we define two confidence scores based on its IoU values with the object prior bounding boxes at A_i and A_{i+1}, respectively:

$$\Omega_f(\Delta_i^j) = J\big(B_i^j, B_i^o\big), \tag{5.6}$$

$$\Omega_l(\Delta_i^j) = J\big(\tilde{B}_i^j, B_{i+1}^o\big) \tag{5.7}$$

where Ω_f and Ω_l are defined as first and the last confidence scores of a tracklet based on our object priors at its two ends, respectively. Since we don't have objectness score (Ω_o) for the last bounding box produced by tracking, we omit the use of objectness score here altogether, even for first bounding box, although available.

5.1.3 Tube Generation

Given the n sets of tracklets from n activators in a video, we need to select one tracklet from each set to create a spatio-temporal consistent tube which localizes the common object with high confidence. Let $\cdot = \{\Delta_1, \Delta_2, \ldots, T_n\}$ be a possible tube. Our goal is to find the best tube for every video that minimizes the following criterion, i.e.

$$\min \sum_{i=1}^{n-1} - \log\Big(\Omega_l(\Delta_i)\Omega_f(\Delta_{i+1})\Big) - \lambda \log\Big(J\big(\tilde{B}_i, B_{i+1}\big)\Big) \tag{5.8}$$

where tracklet Δ_i starts with B_i and ends with $\tilde{B}_i$, and λ is a trade-off parameter. At any activator (A_{i+1}), both the selected adjacent tracklets (Δ_i, Δ_{i+1}) should have high confidence scores. Therefore, the first term in Eq. (5.8) is to measure how confidently a pair of adjacent tracklets Δ_i and Δ_{i+1} contain the object w.r.t. the object prior B_{i+1}^o. The selected adjacent tracklets (Δ_i, Δ_{i+1}) should also overlap well with each other to form a consistent tube. Therefore, the second term in Eq. (5.8) is to measure the smoothness between the adjacent tracklets via their IoU value. While one term signifies the reliance on co-saliency, another term signifies the reliance on temporal consistency between activated tracklets, to perform what we call as *co-saliency activated tracklet selection* (CATS), resulting in video co-localization. This problem of Eq. (5.8) can be well solved using dynamic programming.

5.2 Experimental Results

We evaluate our method on the benchmark YouTube Object Dataset using the evaluation metric of CorLoc (same as in Chapter 3), which is defined as the percentage of frames that satisfies the IoU condition: $\frac{area(B_{gt} \cap B_{co})}{area(B_{gt} \cup B_{co})} > 0.5$, where B_{gt} and B_{co} are ground-truth and computed bounding boxes, respectively. YouTube Object Dataset consists of videos downloaded from YouTube and is divided into 10 object classes. Each object class consists of several video shots of the objects belonging to the class. We treat each shot as a video sequence and group all the shots in one class as a weakly supervised scenario for video co-localization.

5.2.1 Implementation Details

Activators are chosen at the interval of 50 frames. While calculating inter video co-saliency, we wanted to ensure that at least 10 best matched activators should be available, therefore we used K-NN instead of ϵ-NN algorithm. For intra-video co-saliency, we set the number of clusters as $\{n/10\}$ where n is the total number of activators in a video and $\{\cdot\}$ denotes the rounding function. We use [38] to generate saliency maps for individual activators. We choose $m = 10$ at bounding-box filtering step, and sample every 5^{th} frame between activators to generate total motion saliency map for preceding activator to avoid repetitiveness. The parameter λ introduced in Eq. (5.8), i.e. weight for temporal consistency, is set to 2, same as [45]. For the off shelf techniques we adopt including tracklets [6], motion saliency [33] and GrabCut [76], we use their default settings.

5.2.2 Co-localization Performance

Results under weakly supervised scenarios: Table. 5.1 shows the CorLoc performance on YouTube Object Dataset under weakly supervised scenarios using our full-fledged CATS method (*ext*+*int*+*mot*), where *ext*, *int* and *mot* refer to using inter-video co-saliency, intra-video co-saliency and motion co-saliency respectively for obtaining the final co-saliency map. We compare with state-of-the-art methods on video co-localization. It can be seen that we almost double the average performance of the frameworks [75] and [41] that simultaneously locate

Table 5.1: CorLoc results of video co-localization on YouTube Object Dataset under weakly supervised scenarios.

	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	avg
[75]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
[41]	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
[45]	56.5	**66.4**	58.0	**76.8**	39.9	**69.3**	50.4	**56.3**	**53.0**	31.0	55.7
ext	62.4	43.3	63.8	50.9	51.9	63.8	61.7	43.4	30.0	45.7	51.7
int+mot	64.7	48.1	60.9	54.5	51.2	64.0	58.9	42.5	27.0	**46.6**	51.8
ext+int+mot	**65.7**	59.6	**66.7**	72.3	**55.6**	64.6	**66.0**	50.4	39.0	42.2	**58.2**

the common object in multiple videos. This suggests that single video localization with an incorporated object priors from other videos is better than directly performing co-localization on multiple videos, since inter-video variations could be huge. Moreover, thanks to our proposed co-saliency generation and the adoption of consistent tracklets, we achieve 4.5% improvement for the average performance over the state-of-the-art [45], which uses bounding box proposals at every frame and optimizes over them to obtain consistency. Compared to bounding box proposals, using tracklets significantly reduces the computational complexity as the number of nodes to deal with are drastically reduced. For example, for a video of 1000 frames, [45] would need to deal with 100×50 nodes (according to their settings of 100 selected proposals per key frame and 1 sampled key frame per 20 frames), whereas we only need to deal with only 10×20 nodes (default 10 proposals/activator and 1 sampled activator per 50 frames). Considering that more noisy nodes are eliminated, it results in more reliable results. In addition, [45] is an iterative approach and needs 5 iterations to achieve as good as 55.7% score beginning with nearly 38% score at the first iteration, whereas our method achieves 58.2% score in just one shot.

In addition to our full-fledged method, in Table. 5.1 we also show the results of the variants (*ext*, *int+mot*) that use different combinations of the co-saliency maps. The results of *ext* show how much we can explore other videos to help the localization in the considered video. The results of *int+mot* show how much we can benefit from the single video itself. We can see that the combination of all the three co-saliency maps achieves the best performance.

In Figure 5.6, we show the localization results (red) on some of the frames in the dataset along with their ground truths (green). It can be seen that our proposed method is able to effectively localize the dominant objects with various poses and shapes across the videos. In

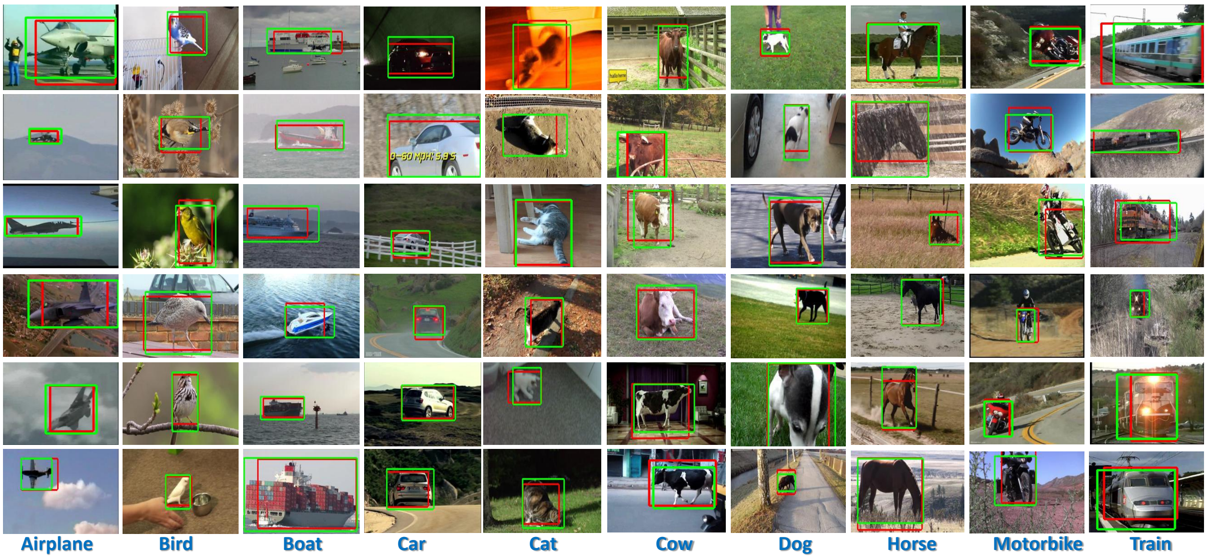


Figure 5.6: Sample localization results (red) along with groundtruths (green) on YouTube Objects dataset.

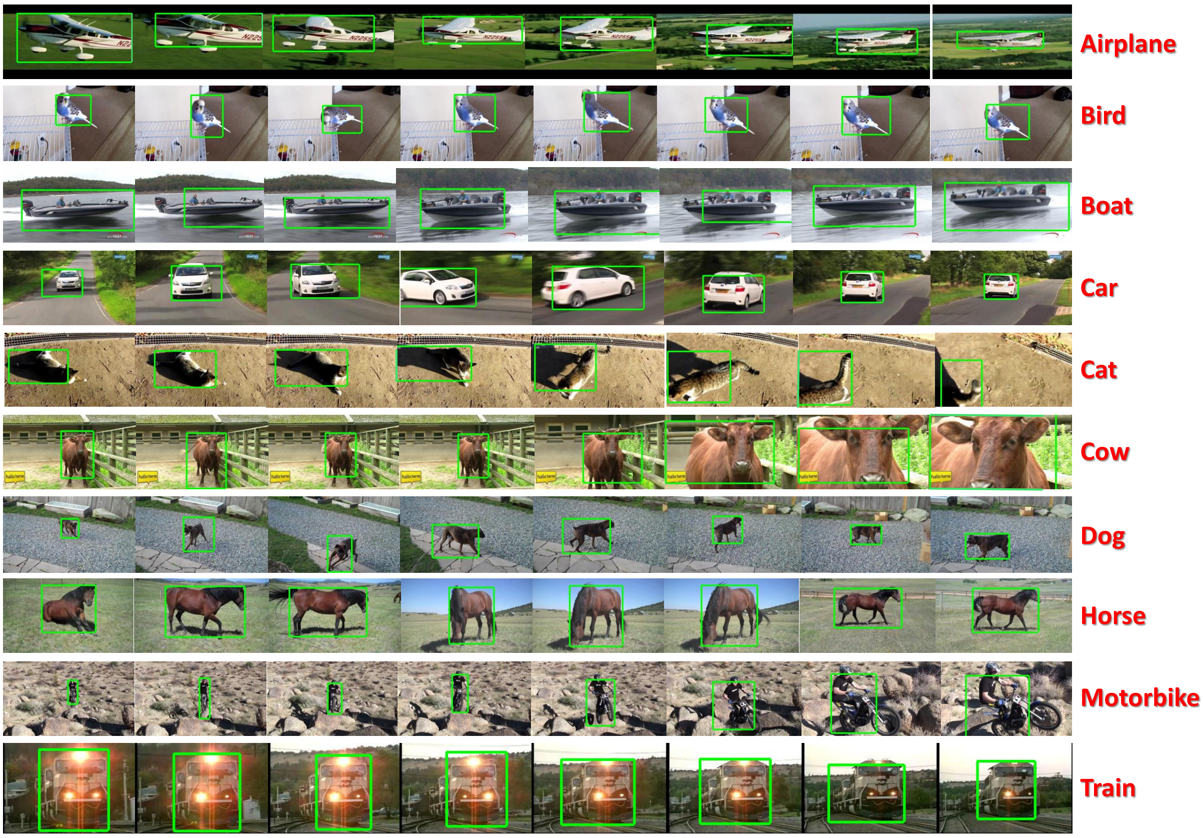


Figure 5.7: Our video co-localization results on YouTube Object Dataset. It can be seen that our method can handle variations in size (for airplane, cow and motorbike), position (for dog), pose (for car, cat and horse), and mobility (negligible motion for bird).

Table 5.2: CorLoc results on YouTube Object Dataset in an unsupervised scenario where we do not use class labels.

	aeroplane	bird	boat	car	cow	cat	dog	horse	motorbike	train	avg
[10]	53.9	19.6	38.2	37.8	32.2	21.8	27.0	34.7	45.4	37.5	34.8
[74]	65.4	**67.3**	38.9	65.2	46.3	40.2	**65.3**	48.4	39.0	25.0	50.1
[45]	55.2	58.7	53.6	**72.3**	33.1	58.3	52.5	**50.8**	**45.0**	19.8	49.9
ext+int+mot	**66.7**	48.1	**62.3**	51.8	**49.6**	**60.6**	58.9	41.9	28.0	**47.4**	**51.5**

Figure 5.7, we demonstrate our localization results on different videos. It can be seen that our method is able to effectively handle various pose variations in the videos of the car, cat and horse, the size variation in cow and motorbike, and the location variation in dog video. At the same time, our method is also able to handle objects that do not move much such as in the video of bird. These results clearly demonstrate the robustness of our method in different scenarios.

Results under unsupervised scenario: Table. 5.2 presents the CorLoc results obtained when we do not make use of any weak supervision provided by class labels. We consider entire YouTube Object Dataset as a whole and apply the proposed method on it. We basically rely upon kNN method to find good matching activators from other videos. We compare with other methods which reported such unsupervised results as well as other single video localization methods. It can be seen that our full-fledged method also achieves the best performance in such unsupervised scenario.

5.2.3 Evaluation on Bounding Box Filtering

In this subsection, we evaluate the effectiveness of the proposed bounding box filtering. We generate 300 bounding box proposals using [2] and select Top-k proposals based on either the objectness scores [2] or our confidence scores defined in Eq. (5.5). The recall rates are shown in Table 5.3. It can be seen that by incorporating the co-saliency based object prior for bounding box selection, our method greatly improves the recall rate. Even with only one proposal generated by our method, it has 45.5% probability to be overlapped with the ground truth bounding box with IoU large than 0.5, almost double of that in [2].

Table 5.3: The recall performance on YouTube Object Dataset using either the existing objectness scores $\Omega_o(B_i^j)$ [2] or the proposed object confidence scores $\Omega(B_i^j)$ in (5.5) for bounding box selection.

	Top-1	Top-3	Top-5	Top-10	Top-20
$\Omega_o(B_i^j)$ [2]	22.8	50.8	64.4	77.9	86.1
$\Omega(B_i^j)$ (5.5)	**45.5**	**65.8**	**74.0**	**80.9**	**87.1**

Table 5.4: Comparison with the objectness baseline with different m values.

	$m = 1$	$m = 3$	$m = 5$	$m = 10$	$m = 20$
Baseline	23.0	35.6	40.9	42.4	41.1
Proposed Method	**45.5**	**55.3**	**57.7**	**58.2**	**55.2**

5.2.4 Evaluation on Co-saliency Prior and Tracklet Selection

In order to show the improvement in performance by using the developed co-saliency prior, we compare our method with an objectness based baseline, which is essentially our method but with top objectness-ranked bounding boxes using [2] instead of using our co-saliency prior. Table 5.4 shows the CorLoc results under different $m \in \{1, 3, 5, 10, 20\}$ in Table 5.4. It can be seen that our method achieves better overall CorLoc scores than the baseline for all m, which suggests that our co-saliency prior plays the key role here. When $m = 1$, the result signifies benefit of co-saliency alone, which can be compared with the result obtained by Hough match alone in [45] (referring to the foreground saliency based on appearance only, i.e. F(A) at 1^{st} iteration. Kindly refer to [45] for more details). Ours is 45.5 compared to their 32. It can also be observed that as m increases, i.e. considering multiple candidate tracklets, the performance increases. This indicates that the co-saliency alone ($m = 1$ case) is not sufficient. Only when we combine the co-saliency prior with the tracklet generation and selection, we achieve the best performance. In the tracklet selction, we have the tradeoff parameter λ balancing the confidence and smoothness terms. In Figure 5.8, we show that when λ is set in range 1 to 6, performance varies between 55 and 58, which is somewhat stable. After $\lambda = 6$, performance drops because smoothness overweighs the confidence.

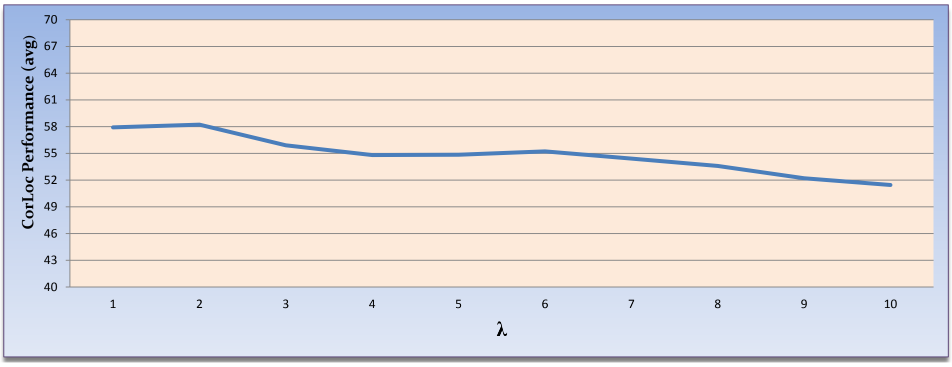


Figure 5.8: Performance variation as λ in the Eq. (5.8) varies.

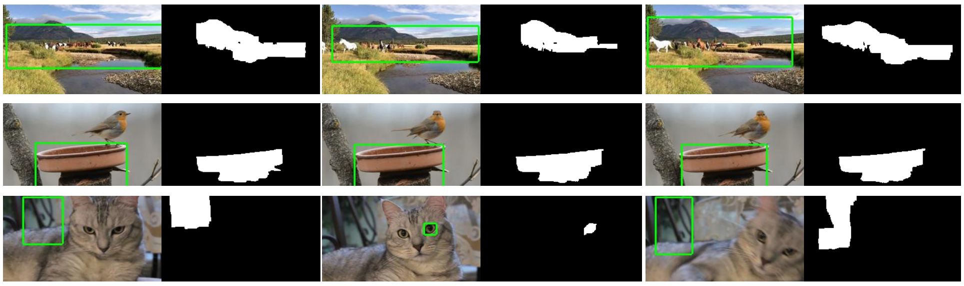


Figure 5.9: Failure examples of videos where most of the activators failed to obtain good co-saliency based object prior (O_i) resulting in poor highest scored bounding box proposals

5.2.5 Limitations and Discussions

Although we consider objectness measure alongside with our object mask for selection of bounding boxes, incase co-saliency map based object prior is not good. In addition, we rely on the consistency of adjacent tracklets to negate the effect of few bad object priors. But it is quite possible that most of the activators fail in obtaining good co-saliency based object prior in a particular video. In such cases, proposed method is quite likely to fail. In Figure 5.9, we show such failure examples of videos where most of the activators failed to obtain good co-saliency maps resulting in poor highest scored bounding box proposals.

Also, there are a few reasons for the relatively low performance of our method at some categories, as can be observed in Table 5.1. First, our method heavily relies on the co-saliency object prior. For some categories such as horse or motorbike, human beings often appear on horse or motorbike on several videos, which also get highlighted in our co-saliency maps and included in our results, while they are excluded in the groundtruths of the two categories. Second, our parameters are all set globally instead of calibrated for individual categories. Thus, it is likely that for some other parameter setting, we might achieve better results. For example, in the case of bird category, if we select 8 bounding boxes instead of the default 10, we can improve the CorLoc result from 59.6% to 62.5%.

Execution Time: Our algorithm takes nearly 16 hours for co-localizing the entire YouTube Object dataset on PC with Intel Core i5-3470 (3.20 GHz, 4 cores) CPU. Whereas [45] takes 60 hours (from [45]) on PC with Xeon CPU (2.6 GHz, 12 cores). Therefore, our method is relatively faster.

5.3 Summary

In this chapter, we have proposed a new video co-localization method named *co-saliency activated tracklet selection* (CATS) where we activate several tracklets with the help of co-saliency maps at regular intervals. We then select optimal tracklets from these sets for forming a tube to localize the common object. In contrast to previous methods, we proposed a guided single video-based framework which is non-iterative and computationally efficient. In the proposed approach, co-saliency plays the key role in guiding the activation and selection of our processing units called *co-saliency activated tracklets*, different from bounding box proposals or tube proposals used previously for the video co-localization problem. We obtain state-of-the-art localization results on YouTube Objects dataset in both weakly supervised and unsupervised scenarios through the proposed approach.

Chapter 6

Object Co-skeletonization with Co-segmentation

Our main objective in this chapter is to exploit joint processing to extract objects' skeletons in images of the same category. We call it *object co-skeletonization*. By objects, we mean something which interests the image viewer more compared to the stuff like sky, roads, mountains, sea, etc, in its presence. Automatic skeletonization of such objects has many applications such as image search, image synthesis, generating training data for object detectors, etc. Existing methods either need pre-segmentation [18, 84] of the object in the image or groundtruth skeletons for the training images to learn [83, 91] to perform skeletonization on test images. In this chapter, we attempt weak supervision to approach the problem, *i.e.*, co-skeletonization. However, it is difficult to solve this problem as a standalone task, because it requires object's shape information as well. Even the recent deep learning based method [86] requires not only the skeleton location information but also the skeleton scale information as groundtruths to account for shape information. The skeleton scale is basically the distance between a skeleton point and the nearest boundary point of the object. In our joint processing context, we leverage the existing idea of object co-segmentation to help it out so that co-skeletonization can be performed more effectively. In fact, it turns out that co-skeletonization can also help co-segmentation in return by providing good scribbles. In this way both co-skeletonization and co-segmentation benefit each other synergistically. We couple these two tasks to achieve what we call "*Object Co-skeletonization with Co-segmentation*" as shown in Figure 6.1.

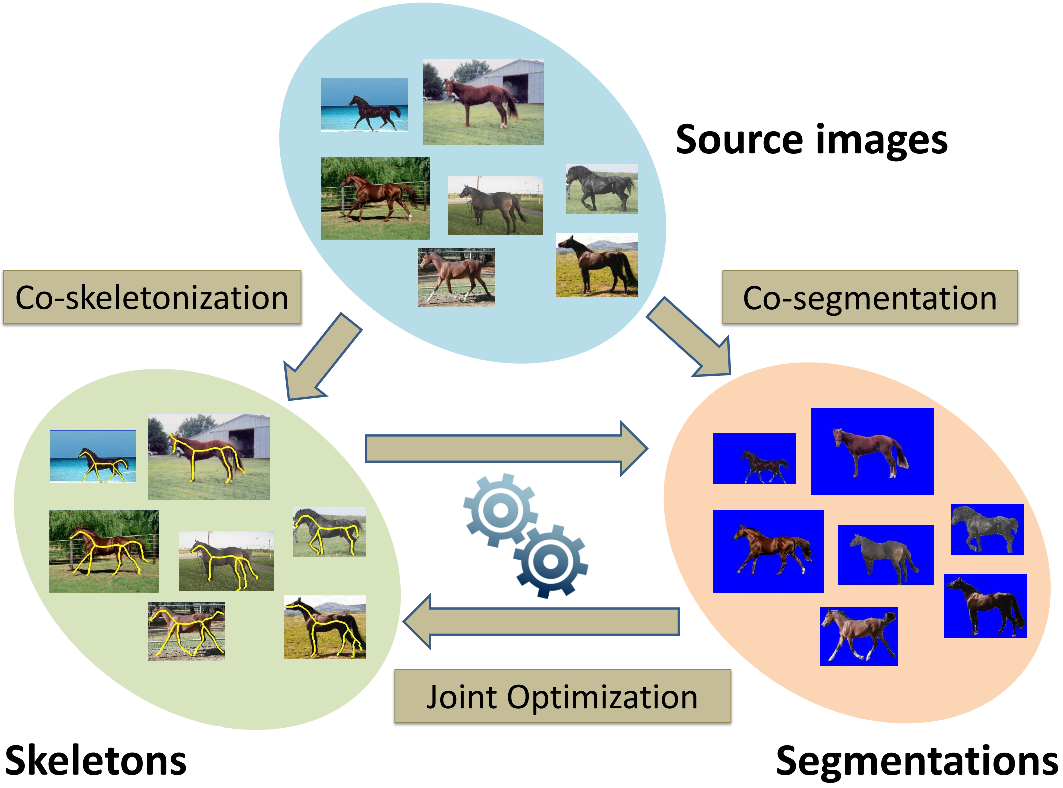


Figure 6.1: Object co-skeletonization with co-segmentation. Skeletons are in yellow.

There are several challenges involved in performing co-skeletonization and such a coupling. First, existing skeletonization algorithms [18, 80, 82, 84] can yield a good skeleton if a good and smooth shape is provided, but they are quite sensitive to the given shape, as shown for the image in Figure 6.2(a) which has unsmooth segmentation. The skeleton produced by [84] in Figure 6.2(a) has too many unnecessary branches, while more desirable skeleton to represent the cheetah would be our skeleton in Figure 6.2(c). Thus, the quality of the provided shape becomes crucial , considering that co-segmentation may not provide good and smooth shapes due to its complicated way of co-labeling many images. Second, joint processing of skeletons across multiple images is quite tricky. Because most of the skeleton points generally lie on homogeneous regions as shown in Figure 6.2(d) and (e), they are not so easy to detect and describe for the purpose of matching. Third, how to couple the two tasks so that they can synergistically assist each other is another challenge.

Our key observation is that we can exploit the inherent interdependencies of two tasks

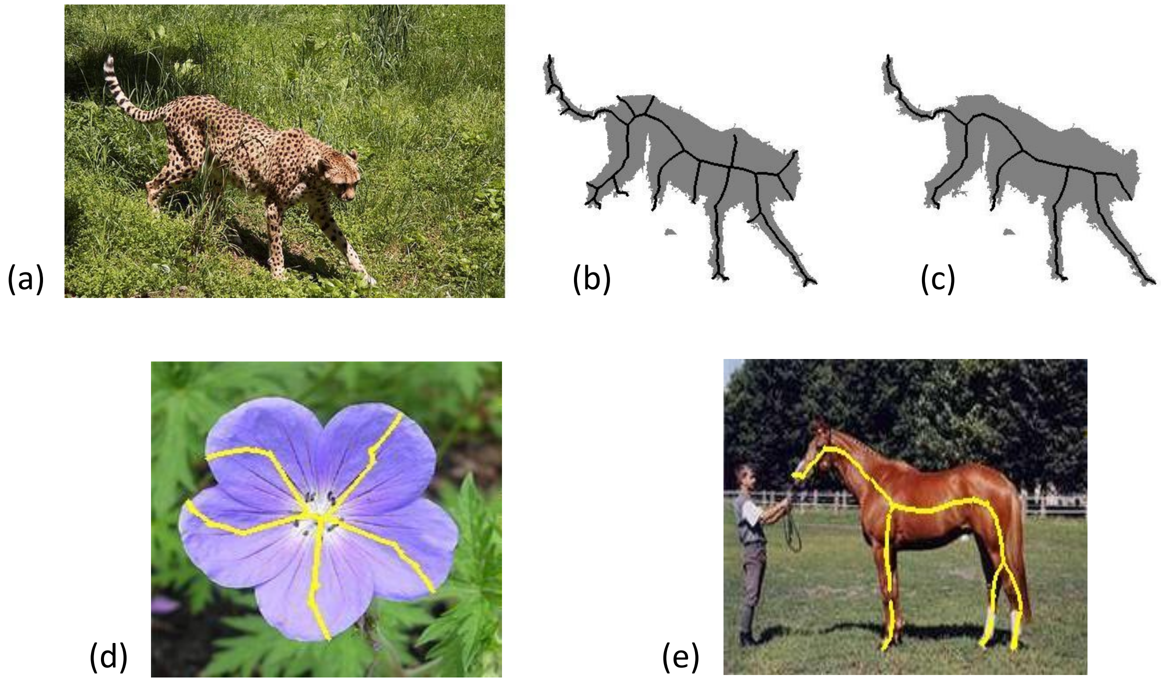


Figure 6.2: Example challenges of co-skeletonization. The quality of segmentation affects the quality of skeletonization. (b) The result of [84] for (a). (c) Our result. Skeletons lie on homogeneous regions, such as in (d) and (e), thus not easy to detect and describe.

to achieve better results jointly, as shown in Figure 6.3. Since most of the skeleton pixels still remain on the horse in bad co-segmentation at the beginning, they gradually improve the segmentation by providing good seeds for segmentation in the subsequent iterations of joint processing, and in turn co-skeletonization is benefited and becomes better as the co-segmentation improves.

Therefore, we propose a joint framework for co-skeletonization and co-segmentation where we try to address the challenges. First of all, we not only rely on the shape but also the jointly processed prior to perform skeletonization. We also build upon a skeleton pruning process [84] to better handle the unsmooth shapes. Second, structure preserving quality of dense correspondence is exploited for overcoming the skeleton matching challenge. We follow the same strategy to generate a co-segment prior, and use an interactive segmentation framework [76] to perform co-segmentation. The scribbles are generated with the help of the skeleton to make co-segmentation well informed of the current skeleton location. And in turn the resultant segmen-

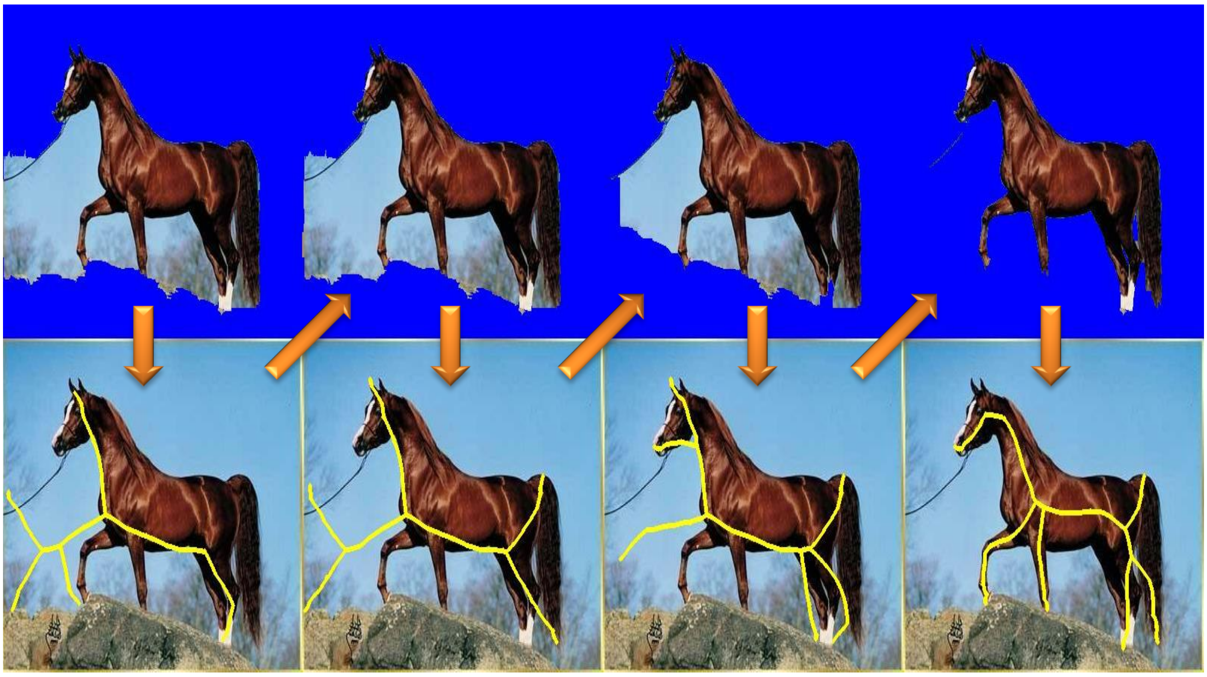


Figure 6.3: Inherent interdependencies of co-skeletonization and co-segmentation can be exploited to achieve better results through a coupled iterative optimization process.

tation provides shape information to the skeleton pruning process. It is an iterative approach and our framework is initialized with the help of visual saliency. It can also be initialized with groundtruths for training images, and this allows us to make fair comparison with fully supervised learning based methods [46, 47, 89, 91, 99, 108]

To the best of our knowledge, there is only one dataset where co-skeletonization could be performed in a weakly supervised manner, i.e. WH-SYMMAX dataset [83], and it only contains horse images. To extensively evaluate co-skeletonization, we construct a new benchmark dataset called CO-SKEL dataset, which consists of images ranging from animals, birds, flowers to humans classified into total 26 categories. Experiments show that our approach achieves state-of-the-art co-skeletonization performance in weakly supervised setting.

6.1 Proposed Method

In this section, we discuss our joint framework of co-skeletonization and co-segmentation in detail.

6.1.1 Overview of Our Approach

Given a set of m similar images belonging to the same category, denoted by $\mathcal{I} = \{I_1, I_2, \cdots, I_m\}$, we aim to provide two output sets: $\mathcal{K} = \{K_1, K_2, \cdots, K_m\}$ and $\mathcal{O} = \{O_1, O_2, \cdots, O_m\}$, comprising of skeleton masks and segmentation masks, respectively, where $K_i(p), O_i(p) \in \{0, 1\}$ indicating whether a pixel p is a skeleton pixel ($K_i(p) = 1$) and whether it is a foreground pixel ($O_i(p) = 1$).

Our overall objective function for an image I_i is defined as

$$\begin{aligned}&\min_{K_i, O_i} \lambda\Theta_{pr}(K_i, O_i|\mathcal{N}_i) + \Theta_{in}(K_i, O_i|I_i) + \Theta_{sm}(K_i, O_i|I_i)\\&s.t.\ K_i \subseteq \mathbf{ma}(O_i)\end{aligned} \tag{6.1}$$

where the first term Θ_{pr} accounts for the priors from the set of neighbor images denoted as $\mathcal{N}_i$, the second term Θ_{in} is to enforce the interdependence between the skeleton K_i and the shape / segmentation O_i in image I_i, the third term Θ_{sm} is the smoothness term to enforces the smoothness among neighboring pixels, and λ is a parameter to control the influence of the inter-image prior term. The constraint in (6.1) means the skeleton must be a subset of medial axis ($\mathbf{ma}$) [18] of the shape.

We resort to the typical alternative optimization strategy (such as [76]) to solve (6.1), i.e. dividing (6.1) into two sub-problems and solve them iteratively. In particular, one sub-problem is: given the shape O_i, we solve co-skeletonization by

$$\begin{aligned}&\min_{K_i}\ \lambda\Theta^k_{pr}(K_i|\mathcal{N}_i) + \Theta^k_{in}(K_i|O_i, I_i) + \Theta^k_{sm}(K_i|I_i)\\&s.t.\ K_i \subseteq \mathbf{ma}(O_i)\end{aligned} \tag{6.2}$$

The other sub-problem is: given the skeleton K_i, we solve co-segmentation by

$$\min_{O_i} \lambda\Theta^o_{pr}(O_i|\mathcal{N}_i) + \Theta^o_{in}(O_i|K_i, I_i) + \Theta^o_{sm}(O_i|I_i). \tag{6.3}$$

If we treat both the inter-image prior term Θ^k_{pr} and the shape prior term Θ^k_{in} as a combined prior, Eq. (6.2) turns out to be a skeleton pruning problem and can be solved using the approach similar to [84], where branches in the skeleton are iteratively removed as long as it reduces the energy. Similarly, if we combine both the inter-image prior Θ^o_{pr} and the skeleton prior Θ^o_{in} as the data term, Eq. (6.3) become a standard MRF-based segmentation formulation, which can be solved using GrabCut [76]. Thus, compared with the existing works, the key differences of our formulation lie in the designed inter-image prior terms as well as the interdependence terms, which link the co-skeletonization and co-segmentation together.

Iteratively solving (6.2) and (6.3) requires a good initialization. We propose to initialize $\mathcal{O}$ by Otsu thresholded saliency maps and $\mathcal{K}$ by the medial axis mask [18]. Algorithm 1 summarizes our approach, where $(\Theta_{pr} + \Theta_{in} + \Theta_{sm})^{(t)}$ denotes the objective function value of Eq. (6.1) at the t^{th} iteration and $\Theta_{pr} = \Theta^k_{pr} + \Theta^o_{pr}$, $\Theta_{in} = \Theta^k_{in} + \Theta^o_{in}$, $\Theta_{sm} = \Theta^k_{sm} + \Theta^o_{sm}$.

```
**Algorithm 1** Our approach for solving Eq. (6.1)
**Data:** An image set $\mathcal{I}$ containg images of the same category
**Result:** Sets $\mathcal{O}$ and $\mathcal{K}$ containing segmentations and skeletons of images in $\mathcal{I}$
**Initialization:** $\forall I_i \in \mathcal{I}$, $O_i^{(0)}$ = Otsu thresholded saliency map and $K_i^{(0)} = \mathbf{ma}(O_i^{(0)})$;
**Process:** $\forall I_i \in \mathcal{I}$,
**do**
    1) Obtain $O_i^{(t+1)}$ by solving Eq. (6.3) using [76] with $\mathcal{O}^{(t)}$ and $K_i^{(t)}$.
    2) Obtain $K_i^{(t+1)}$ by solving Eq. (6.2) using [84] with $\mathcal{K}^{(t)}$ and $O_i^{(t+1)}$, $s.t.\ K_i^{(t+1)} \in \mathbf{ma}(O_i^{(t+1)})$.
**while** $(\lambda\Theta_{pr} + \Theta_{in} + \Theta_{sm})^{(t+1)} \leq (\lambda\Theta_{pr} + \Theta_{in} + \Theta_{sm})^{(t)}$;
$\mathcal{O} \leftarrow \mathcal{O}^{(t)}$ and $\mathcal{K} \leftarrow \mathcal{K}^{(t)}$
```

6.1.2 Object Co-skeletonization

As shown in Algorithm 1, the step of object co-skeletonization is to obtain $K^{(t+1)}$ by minimizing (6.2), given the shape $O^{(t+1)}$ and the previous skeleton set $\mathcal{K}^t$. Considering the constraint of $K_i^{(t+1)} \in \mathbf{ma}(O_i^{(t+1)})$, we only need to search skeleton pixels from the medial axis pixels. We build up our solution based on [84], but with our carefully designed individual terms for (6.2) as explained below.

Prior Term (Θ^k_{pr}): In the object co-skeletonization, a good skeleton pixel will be the one which is repetitive across images. To account for this repetitiveness, we need to find corresponding skeleton pixels in other images. However, skeleton pixels usually lie on homogeneous regions (see Figure 6.2(d)&(e)) and are thus difficult to match. Thus, instead of trying to match sparse skeleton pixels, we make use of dense correspondences using SIFT Flow [55], which preserve the skeleton and segmentation structures well, as shown in Figure 6.4.

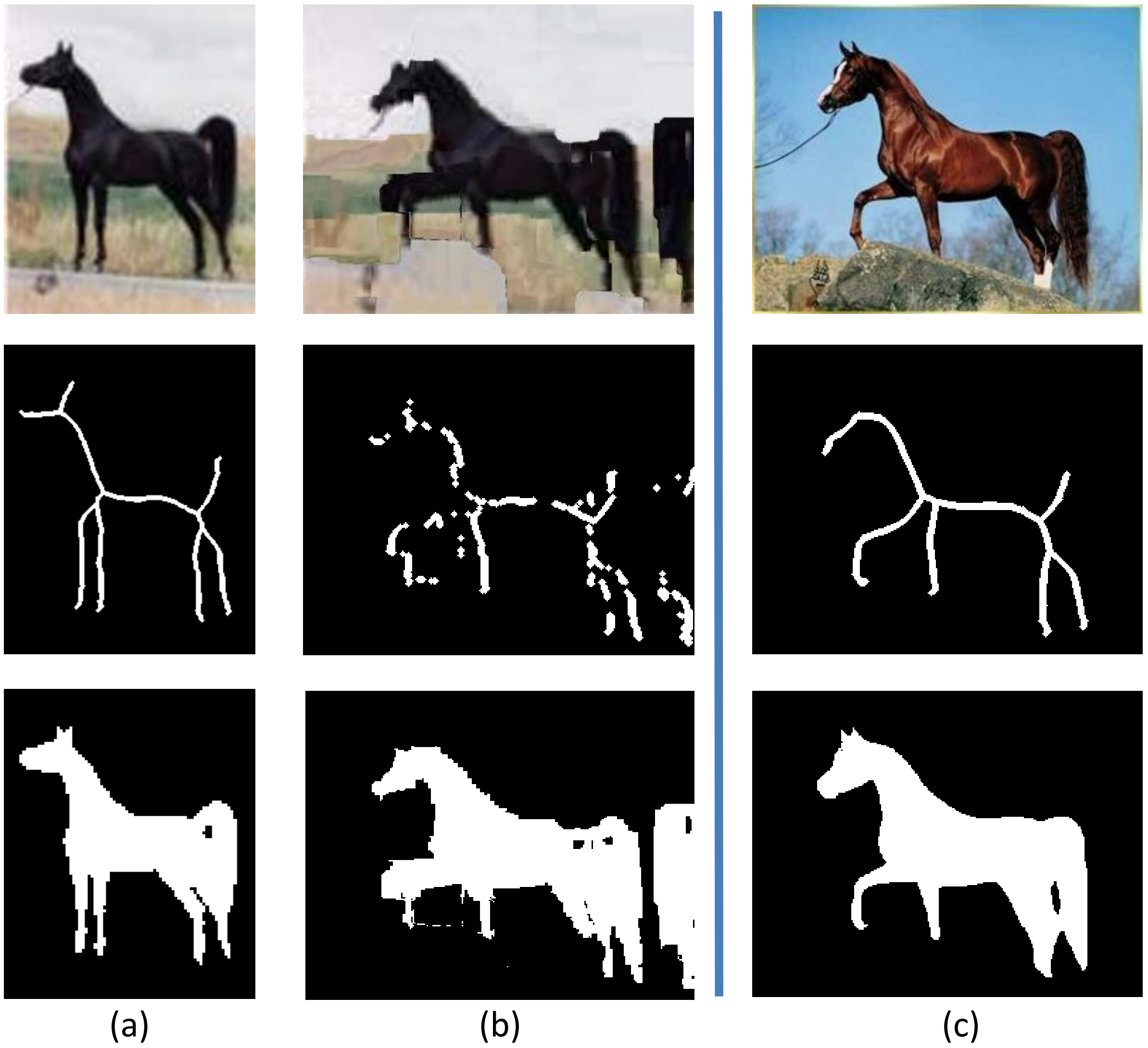


Figure 6.4: Dense Correspondences preserve the skeleton and segmentation structures roughly. Here (a) is warped to (b) to be used as prior for (c).

Once dense correspondence is established, we utilize the warped skeleton pixels from neighboring images to develop the prior term. Particularly, we align all the neighboring images' t^{th} iteration's skeleton maps to the concerned image I_i, and generate a co-skeleton prior at $(t+1)^{th}$ iteration as

$$\widetilde{K}_i^{(t+1)} = \frac{K_i^{(t)} + \sum_{I_j \in \mathcal{N}_i} \mathbf{W}_j^i(K_j^{(t)})}{|\mathcal{N}_i| + 1} \tag{6.4}$$

where we align other skeleton maps using warping function $\mathbf{W}_j^i$ [55] and then average them with I_i's own skeleton map. Note that the neighborhood $\mathcal{N}_i$ is developed simply based the GIST distance [72]. For simplicity, we drop the superscriptions such as $(t+1)$ in all the following derivations.

Considering that the corresponding skeleton pixels from other images may not exactly align with the skeleton pixels of the considered image, we define our prior term as

$$\Theta_{pr}^k(K_i|\mathcal{N}_i) = \sum_{p \in \mathbf{ma}(O_i)} -K_i(p) \log\Big(1 + \sum_{q \in \mathbb{N}(p)} \widetilde{K}_i(q)\Big). \tag{6.5}$$

Eq. (6.5) essentially measures the consistency between the image I_i's own skeleton mask and the recommended skeleton mask from its neighbor images. Note that we accumulate the co-skeleton prior scores in a certain neighborhood $\mathbb{N}(p)$ for each pixel p to account for the rough skeleton alignment across the images.

Interdependence Term (Θ_{in}^k): Our interdependence term serves the traditional data term in skeleton pruning, enforcing that skeleton should provide good reconstruction of the given shape, which medial axis already does. However, a medial axis often contains spurious branches, while the noisy shapes obtained from imperfect co-segmentation only make this worse. To avoid spurious branches, we prefer simplified skeleton, whose reconstructed shape is expected to be smooth while still preserving the main structure of the given shape (see Figure 6.5 for example). On the other hand, we do not want over-simplified skeleton, whose reconstructed shape is likely to miss some important parts (see the 4th column of Figure 6.5).

Therefore, we expect the reconstructed shape from skeleton to match the given shape, but not necessary to be exactly the same as the given shape. In this spirit, we define our interdependence term Θ_{in}^k as

$$\Theta_{in}^k(K_i|O_i) = -\alpha \log \frac{|\mathbf{R}(K_i, O_i) \cap O_i|}{|\mathbf{R}(K_i, O_i) \cup O_i|} \tag{6.6}$$

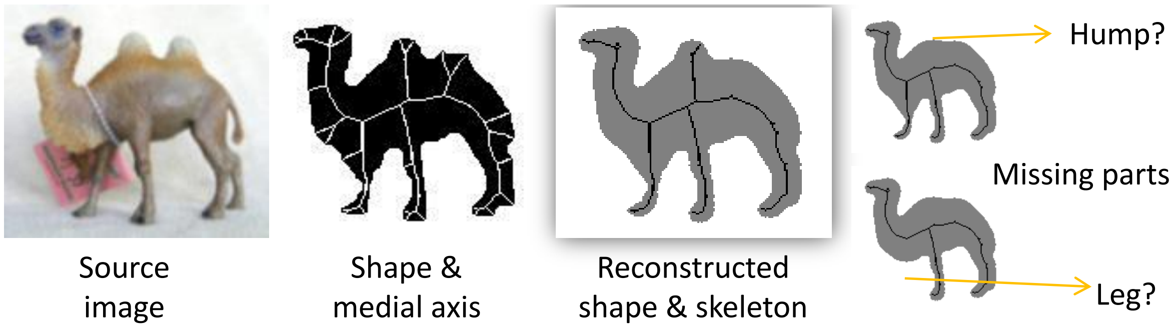


Figure 6.5: Shape reconstruction from skeleton. Compared to the reconstructed shape from medial axis (2nd column), the reconstructed shape (3rd column) from our simplified skeleton is simpler and smoother while still preserving the main structure. Nevertheless, we do not want over-simplified skeleton, which will result in missing important parts in the corresponding shape reconstruction (4th column).

where we use IoU to measure the closeness between the reconstructed shape $\mathbf{R}(K_i, O_i)$ and the given shape O_i, and α is the normalization factor as defined in [84]. The reconstructed shape $\mathbf{R}(K_i, O_i)$ is basically the union of maximal disks at skeleton pixels [84], i.e.

$$\mathbf{R}(K_i, O_i) = \bigcup_{p \in \mathbf{ma}(O_i)} d(p, O_i) \tag{6.7}$$

where $d(p, O_i)$ denotes the maximal disk at skeleton pixel p for the given O_i, and the maximal disk is the disk that exactly fits within O_i with skeleton pixel p as the center.

Smoothness Term (Θ^k_{sm}): To ensure a smoother and simpler skeleton, we aim for skeleton whose: (i) branches are less in number and (ii) branches are long. Our criteria discourage skeletons with spurious branches while at the same time encouraging skeletons with structure-defining branches. This is different from the criteria in [84] which only aims for less number of skeleton pixels. Specifically, we define the smoothness term Θ^k_{sm} as

$$\Theta^k_{sm}(K_i) = |\mathbf{b}(K_i)| \times \sum_{u=1}^{|\mathbf{b}(K_i)|} \frac{1}{length\Big(b_u(K_i)\Big)} \tag{6.8}$$

where $\mathbf{b}(K_i) = \{b_1(K_i), \cdots, b_{|\mathbf{b}(K_i)|}(K_i)\}$ denotes the set of branches of the skeleton K_i. In this way, we punish skeletons with either large number of branches or short-length branches.

6.1.3 Object Co-segmentation

The object co-segmentation problem here is: given the skeleton K_i, find the optimal O_i that minimizes the objective function defined in (6.3). The individual terms in (6.3) are defined in the following.

Prior Term (Θ^o_{pr}): We generate an inter-image co-segment prior, similar to that for co-skeleton. In particular, we align segmentation masks of neighboring images and fuse them with that of the concerned image, i.e.

$$\widetilde{O}_i = \frac{O_i + \sum\limits_{I_j \in \mathcal{N}_i} \mathbf{W}^i_j(O_j)}{|\mathcal{N}_i| + 1} \tag{6.9}$$

where $\mathbf{W}^i_j$ is the same warping function from image j to image i. Then, we use $\widetilde{O}_i$ as a guidance, and define our inter-image prior term as

$$\begin{aligned}\Theta_{pr}(O_i|\mathcal{N}_i) = \sum_{p \in D_i} -\Bigg(& O_i(p) \log\Big(\frac{1}{|\mathbb{N}(p)|} \sum_{q \in \mathbb{N}(p)|} \widetilde{O}_i(q)\Big) \\ & + \Big(1 - O_i(p)\Big) \log\Big(1 - \frac{1}{|\mathbb{N}(p)|} \sum_{q \in \mathbb{N}(p)|} \widetilde{O}_i(q)\Big)\Bigg)\end{aligned} \tag{6.10}$$

which encourages the shape to be consistent with the guidance $\widetilde{O}_i$ from neighboring images. Here again we account for pixel correspondence errors by neighborhood $\mathbb{N}(p)$ averaging.

Interdependence Term (Θ^o_{in}): For co-segmentation process to benefit from co-skeletonization, our basic idea is to build up foreground and background appearance models based on given skeleton K_i. Particularly, we use GMM for appearance models. The foreground GMM model is learned using K_i (i.e. treating skeleton pixels as foreground seeds), whereas the background GMM is learned using the background part of the reconstructed shape $R(K_i)$ of the skeleton. In this manner entire appearance model is developed entirely using the skeleton. Note that at the beginning it is not robust to build up the GMM appearance models in this manner since the initial skeleton extracted based on saliency is not reliable at all. Thus, for initialization, we develop the foreground and background appearance models based on the inter-image priors $\widetilde{K}_i$ and $\widetilde{O}_i$, respectively.

Denoting $\theta(K_i, I_i)$ as the developed appearance models, we define the interdependence term Θ^o_{in} as

$$\Theta^o_{in}(O_i|K_i, I_i) = \sum_{p \in D_i} -\log\left(P\Big(O_i(p) \,|\, \theta(K_i, I_i), I_i(p)\Big)\right) \tag{6.11}$$

where potential $P\Big(O_i(p) \,|\, \theta(K_i, I_i), I_i(p)\Big)$ denotes how likely a pixel of color $I(p)$ will take the label $O_i(p)$ given the appearance model $\theta(K_i, I_i)$. Θ^o_{in} is similar to the data term in the interactive segmentation method [76].

Smoothness Term (Θ^o_{sm}): For ensuring smooth foreground and background segments, we simply adopt the smoothness term of GrabCut [76], i.e.

$$\Theta^o_{sm} = \gamma \sum_{(p,q) \in E_i} [O_i(p) \neq O_i(q)] \exp(\Theta||I_i(p) - I_i(q)||^2) \tag{6.12}$$

where E_i denotes the set of neighboring pixel pairs in the image I_i, and γ and Θ are segmentation smoothness related parameters as discussed in [76].

6.1.4 Implementation Details

We use saliency extraction method [16] for initialization of our framework in all our experiments. We use the same default setting as that in [76] for the segmentation parameters γ and Θ in (6.12) throughout our experiments. For the parameters of SIFT flow [55], we follow the setting in [78] in order to handle the possible matching of different semantic objects. The parameter λ in both (6.2) and (6.3), which controls the influence of joint processing, is set to 0.1.

6.2 Experimental Results

6.2.1 Datasets and Evaluation Metrics

Datasets: There is only one publicly available dataset, i.e. WH-SYMMAX dataset [83], on which weakly supervised co-skeletonization can be performed, but it contains only horse category of images. In order to evaluate co-skeletonization task extensively, we develop a new

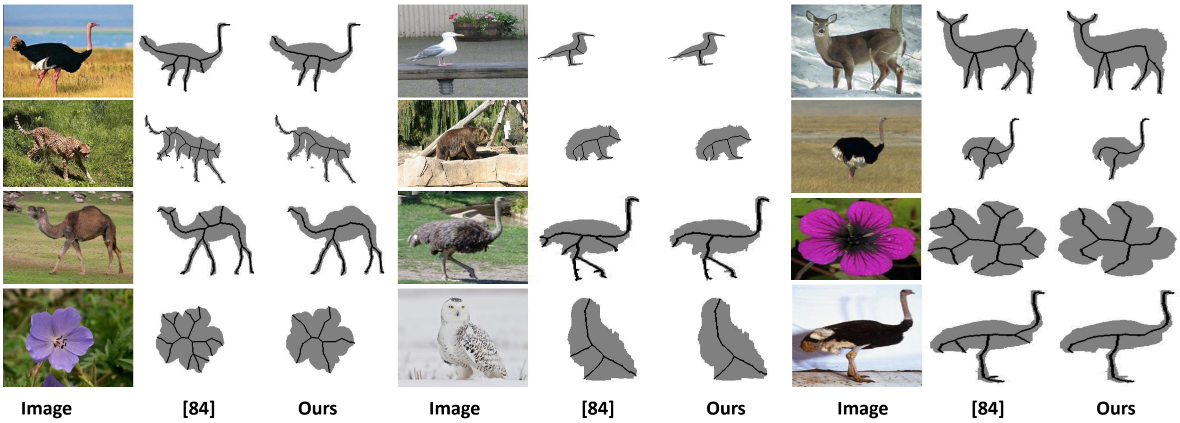


Figure 6.6: Given the shape, we improve skeletonization method [84] using our improved terms in their objective function. It can be seen that our skeletons are much smoother and better in representing the shape. We use these improved results as groundtruths in our CO-SKEL dataset.

benchmark dataset called CO-SKEL dataset. It consists of 26 categories with total 353 images of animals, birds, flowers and humans. These images are collected from MSRC dataset, CosegRep, Weizmann Horses and iCoseg datasets along with their groundtruth segmentation masks. Then, we apply [84] (with our improved terms) on these groundtruth masks, in the same manner as the WH-SYMMAX dataset has been generated from Weizmann Horses dataset [9]. Figure 6.6 shows some example images, and their skeletons using [84] and our improvement of [84][1]. It can be seen that our skeletons are much smoother and better in representing the shape.

Since our method searches k-nearest neighbors first and then performs joint processing, our method can work in an unsupervised way as well as long as there are sufficient number of images of same category objects or visually similar objects. Thus, our method can also be applied to datasets like SK506 dataset [86], which consists of many uncategorized images.

Metrics: For evaluation of skeletonization and segmentation, we calculate F-measure (including precision and recall) and Jaccard Similarity, respectively. Considering it is very difficult to get a resultant skeleton mask exactly aligned with the groundtruth, if a resultant skeleton pixel is nearby a groundtruth skeleton pixel, it should be considered as a hit. Therefore, we consider a resultant skeleton as correct if it is at a distance of $d \in \{0, 1, 2, 3\}$ from a groundtruth

[1] We will make our dataset with groundtruths and code publicly available.

Method	F^0	F^1	F^2	F^3	J
Ours$^{(0)}$	0.095	0.229	0.282	0.319	0.412
Ours (w/o Θ_{in})	0.168	0.337	0.391	0.434	0.649
Ours	**0.189**	**0.405**	**0.464**	**0.506**	**0.721**

Table 6.1: Comparisons of the co-skeletonization and co-segmentation results of our method and its two baselines on WH-SYMMAX dataset. Ours$^{(0)}$: our initialization baseline using Otsu thresholded saliency maps [16] for segmentation and [84] for skeleton. Ours (w/o Θ_{in}): our method without the interdependence terms, i.e. running co-segmentation followed by skeletonization.

	F^0	F^1	F^2	F^3	J
Ours$^{(0)}$	0.129	0.306	0.371	0.416	0.600
Ours (w/o Θ_{in})	0.236	0.426	0.484	0.522	0.725
Ours	**0.237**	**0.435**	**0.495**	**0.535**	**0.741**

Table 6.2: Comparisons of the co-skeletonization and co-segmentation results of our method and its two baselines on our CO-SKEL dataset.

skeleton pixel, for which we denote F^d as the corresponding F-measure. Jaccard Similarity (denoted as J) is basically the IoU of groundtruth and our segmentation result.

6.2.2 Co-skeletonization Results

We report our overall co-skeletonization and co-segmentation results on WH-SYMMMAX and our CO-SKEL datasets in Table 6.1 and 6.2, respectively. Note that since we do not perform any kind of training, we combine both training and test images of WH-SYMMMAX dataset, and then obtain the results. It can be seen that our method greatly improves over our initialization baseline. To demonstrate the importance of considering the interdependence between co-segmentation and co-skeletonization, we also compare with another baseline, Ours (w/o Θ_{in}), where we remove the interdependence, i.e. running co-segmentation first and then doing skeletonization from the resultant foreground segments.

It can be seen that our method outperforms this baseline on both of the datasets. Marginal improvement on CO-SKEL dataset may be due to already good initialization. Specifically, it can be seen that J for initialization is already 0.600 in CO-SKEL dataset compared to 0.412 in WH-SYMMAX dataset, suggesting that there is less room for improvement.

	m	F^0	F^1	F^2	F^3	J
bear	4	0.075	0.1714	0.213	0.246	0.846
iris	10	0.363	0.600	0.658	0.698	0.837
camel	10	0.224	0.353	0.395	0.432	0.674
cat	8	0.118	0.360	0.469	0.523	0.733
cheetah	10	0.078	0.221	0.287	0.335	0.735
cormorant	8	0.351	0.545	0.606	0.642	0.768
cow	28	0.142	0.437	0.580	0.669	0.789
cranesbill	7	0.315	0.619	0.670	0.696	0.935
deer	6	0.214	0.366	0.407	0.449	0.644
desertrose	15	0.360	0.662	0.721	0.759	0.934
dog	11	0.122	0.356	0.457	0.522	0.746
egret	14	0.470	0.642	0.669	0.693	0.760
firepink	6	0.416	0.685	0.756	0.805	0.918
frog	7	0.163	0.358	0.418	0.471	0.734
geranium	17	0.299	0.633	0.716	0.764	0.940
horse	31	0.217	0.435	0.490	0.529	0.726
man	20	0.144	0.246	0.274	0.295	0.385
ostrich	11	0.298	0.530	0.592	0.634	0.752
panda	15	0.037	0.102	0.140	0.174	0.696
pigeon	16	0.181	0.326	0.361	0.382	0.590
seagull	13	0.257	0.461	0.520	0.562	0.662
seastar	9	0.440	0.649	0.681	0.702	0.750
sheep	10	0.078	0.249	0.342	0.401	0.769
snowowl	10	0.089	0.222	0.268	0.306	0.543
statue	29	0.306	0.506	0.542	0.564	0.681
woman	23	0.305	0.463	0.503	0.533	0.674
variance		0.015	0.028	0.029	0.030	0.016

Table 6.3: Categorywise number of images and our Co-skeletonization results on the Co-skel dataset

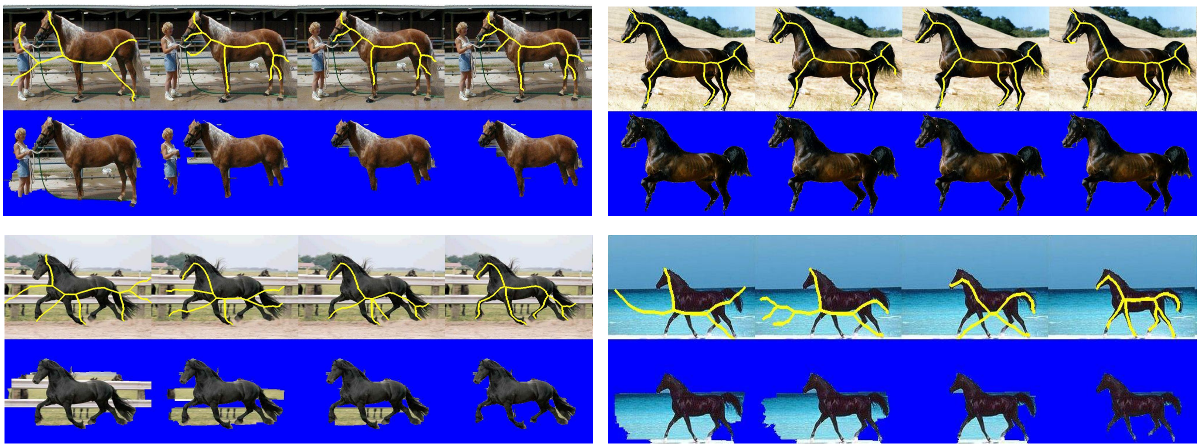


Figure 6.7: Some examples of steadily improving skeletonization and segmentation after each iteration. Top-right example shows that our model continues to reproduce similar results once the optimal shape and skeleton are obtained.

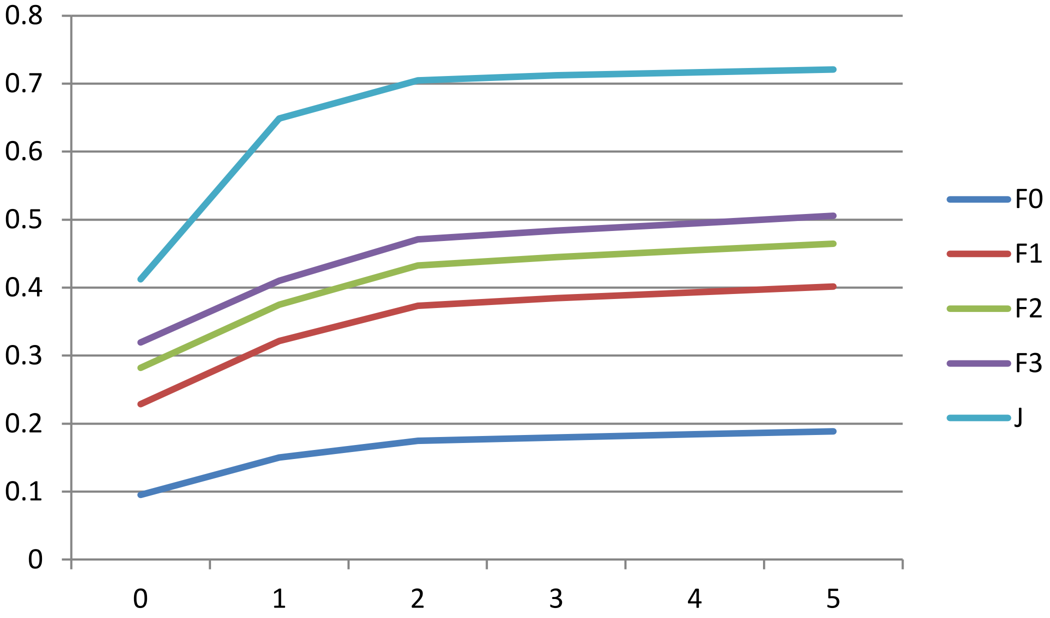


Figure 6.8: Performance v/s Iteration plot. It can be seen that the performance improves swiftly at first and then becomes steady.

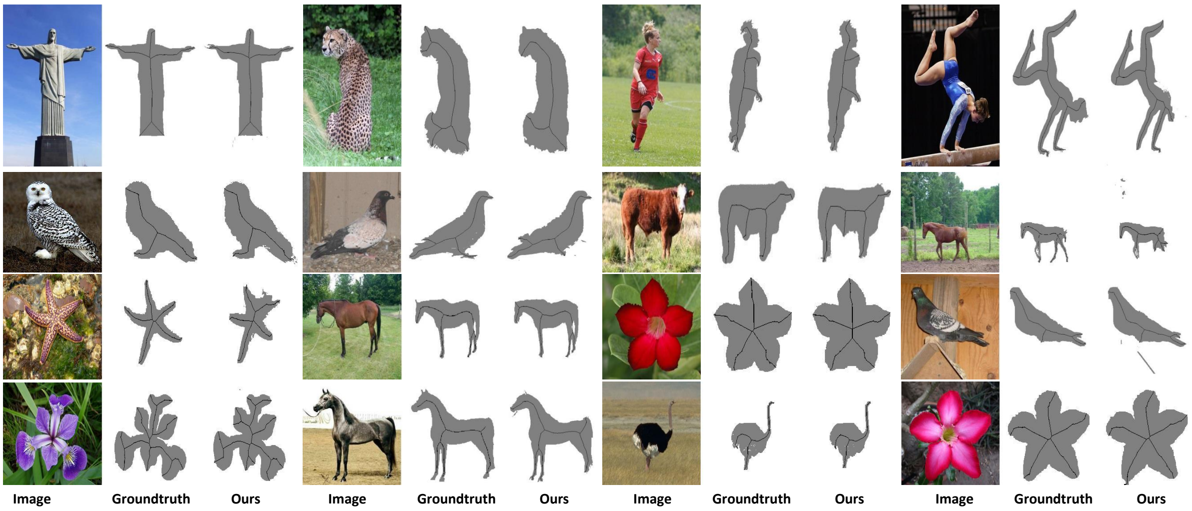


Figure 6.9: Sample co-skeletonization results along with the our final shape masks. It can be seen that both are quite close to the groundtruths.

We also evaluate how our method performs at different iterations in Figure 6.8 on WH-SYMMAX dataset. It can be seen that our method first improves the performance swiftly and then it becomes somewhat steady. This suggests that 2-3 iterations are good enough for our method. Please refer to Figure 6.7 for examples where the results improve steadily with each iteration. Figure 6.9 shows some sample results of our method along with groundtruths from WH-SYMMMAX and CO-SKEL datasets.

We also show our results on individual categories and the variance in performance across the categories of our CO-SKEL dataset in the Table. 6.3. Low variance for both F^d and J metrics suggests that our method is quite reliable.

6.2.3 Supervised Co-skeletonization Results

In order to fairly compare with existing supervised skeletonization methods, we follow the original process but with a change in the initialization. We replace the saliency initialization with ground truth initialization for training images. This will help develop better joint processing priors for remaining images which are the test images. We do the comparisons on test images of WH-SYMMAX and SK506 datsets in Table. 6.4. Note that to make the distinction between our supervised method (groundtruth initialization) and our weakly supervised method

Methods	WH-SYMMAX	SK506
[47]	0.174	0.218
[46]	0.223	0.252
[99]	0.334	0.226
[89]	0.103	-
[91]	0.365	0.392
[108]	0.402	-
Ours$^{(0)}$	0.322	0.261
Ours	0.530	0.483
Ours (S)	**0.594**	**0.523**

Table 6.4: Comparisons of the results of F^d of our methods with supervised methods. Ours$^{(0)}$: our initialization baseline. Ours (S): our method with groundtruth initialization on training images. Note that here $d = 0.0075 \times \sqrt{width^2 + height^2}$ following [86].

(with saliency initialization), we denote the results of our supervised approach as “Ours(S)". It can be seen that not only our supervised method comfortably outperforms all the traditional supervised methods, but also our weakly supervised (unsupervised for SK506) approach is able to do so. Note that performance values reported here are directly taken from [86]. We would like to point out that the recently developed deep learning based supervised method [86] reports much better performance. We did not compare with it since our method essentially is a weakly supervised approach.

6.2.4 Limitations

Our method has some limitations. First, for initialization, our method requires common object parts to be salient in general across the neighboring images if not in all. Therefore, it depends on the quality of the neighboring images. The second limitation lies in the difficulty in the warping process. For example, when the neighboring images contain objects at different sizes or at different viewpoints, the warping processing will have difficulty to align the images. Such situation will not be crucial when there is a large number of images to select. Another problem is that smoothing the skeleton may cause missing out some important short branches.

6.3 Summary

The major contributions of this chapter lie in the newly defined co-skeletonization problem and the proposed joint co-skeletonization and co-segmentation framework, which nicely exploits inherent interdependencies between the two so as to assist each other synerergistically. Extensive experiments demonstrate that proposed method achieves very competitive results on a few benchmark datasets.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have proposed several co-saliency based visual object co-segmentation and co-localization methods. In the process, we try to address several crucial challenges of joint processing such as complexity, parameter tuning, joint v/s single processing issue, etc.

Firstly, we propose to develop co-saliency maps by simple fusion of warped saliency maps while keeping an eye on the quality of saliency maps involved trying to resolve the joint v/s single processing issue. For measuring the quality, foreground-background separability and foreground concentratedness have been used. Also, it has been shown how we can segment and localize the objects using these co-saliency maps to perform co-segmentation and co-localization, respectively. The proposed method is simple and could be easily extended for large scale application. Moreover, it can be utilized in both supervised and unsupervised scenarios. Extensive experiments conducted on different co-segmentation and co-localization datasets, including ImageNet, demonstrated promising results.

Secondly, we propose another kind of saliency fusion technique for co-saliency generation called saliency co-fusion (to perform image co-segmentation eventually), where we fuse multiple saliency maps of the same image (to benefit from various saliency fronts instead of relying on just one) while exploiting its association. Unlike others algorithms, our processing units here are elements, which are basically the super-pixel projection on saliency maps. We develop optimization problem of assigning optimal weights to these elements such that good elements are encouraged and bad elements are suppressed during the fusion process based on similar elements' recommendations and elements' positions. The resultant co-saliency maps were much cleaner and supported co-segmentation of even multiple and repetitive objects. Experiments on five benchmark datasets with eight saliency extraction methods show that our saliency co-fusion based approach achieves competitive performance even without any parameter fine-tuning when compared with the state-of-the-art methods.

Thirdly, we extend our saliency fusion idea to perform video co-localization. Noting that developing co-saliency for each frame could prove to be a costly affair, and tracklets are somewhat reliable for short intervals, we activate sets of tracklets using co-saliency maps generated at regular intervals. For generating the final co-saliency maps, we first generate three types of co-saliency maps: inter-video co-saliency, intra-video co-saliency and motion co-saliency maps, and then fuse them by simple averaging. These tracklets are activated from only those bounding box proposals which overlap well with the co-saliency priors and have good objectness scores. After that, an optimized tube is generated where one tracklet from each set are selected such that their confidence scores are high and they are spatially consistent at the joints. The proposed method is not only faster than state-of-the-art video co-localization method but it also produces better results.

Fourth, we propose a new joint processing problem called co-skeletonization, where we generate joint skeleton priors by fusing the skeleton prior maps of different images, and we also integrate it with co-segmentation task so that the two tasks can help each other. The skeleton can provide good scribbles for segmentation, and skeletonization, in turn, needs good segmentation. We exploit this interdependence and develop a coupled framework for these two tasks. We employ an alternative optimization strategy to solve the optimization problem. Since it is a new problem, we also construct a benchmark dataset for co-skeletonization task, named CO-SKEL dataset. Extensive experiments demonstrate that proposed method achieves very competitive results, even against the completely supervised skeletonization methods.

Our methods are primarily based on saliency fusion idea and highly scalable as shown by experiment on 1 million images of ImageNet, thanks to our modification for making the model efficient. We report the time taken for our saliency fusion idea using both the original fusion and the efficient fusion in Table 7.1. Empirically, it can be seen that the time taken is almost linear to the number of images, and efficient method is at least 4-5 times faster. However,

Table 7.1: Time-taken by our saliency fusion approach is almost linear to the number of images

Dataset	Number of images	Time Taken (mins)
Weizmann Horses	328	117
Coseg-Rep	572	216
Internet Images	2470	806

Table 7.2: Co-segmentation performance comparison among our methods on Weizmann Horses dataset

Methods	Saliency Fusion	Saliency Co-fusion	Object Co-skeletonization
IoU	0.684	0.733	0.721

it should be noted that our method can be easily parallelized after the neighborhood retrieval step, which is performed using either kNN or clustering. This can reduce the shown time-taken drastically. As far as memory constraints are concerned, saliency co-fusion will need more memory space compared to the others for storing the saliency information from multiple sources. Comparatively, memory requirement order will be following:

Saliency Fusion<Object Co-skeletonization<Video Co-localization<Saliency Co-fusion.

It is good to compare how our methods perform w.r.t to each other. Therefore, we evaluate saliency fusion, saliency co-fusion and object co-skeletonization methods on Weizmann Horses dataset for co-segmentation task. Table 7.2 shows IoU results of these methods. It can be seen that saliency co-fusion method outperforms saliency fusion and object co-skeletonization method. This is because the method benefits from multiple fronts of saliency whereas others benefit from initialization of one kind of saliency map only. However, it is significant that integrating co-skeletonization with co-segmentation has benefited the co-segmentation in improving the performance from 0.684 to 0.721. This suggests that saliency co-fusion is our best method but at the cost of memory as discussed earlier. And if memory is a constraint, object co-skeletonization is the better option.

7.2 Future Work

This thesis has primarily focused on fusing the saliency priors while respecting correspondences and the generality. There are still some untouched joint processing problems in the line of ideas developed here, and then there are some new directions this work can take.

The untouched problems are as follows. First, in videos, we have explored co-localization problem only. Thanks to tracklets (or series of bounding boxes), we didn't require to develop fused priors for every frame for this problem. But how to accomplish this in the case of co-segmentation or co-skeletonization, may be through propagation as done for the large scale application, or can we do it in a better way? Second, joint processing in the case of multiple objects is a challenging problem. Although, saliency co-fusion tries to tackle this but it tries to give same foreground labels to all the objects, instead of giving different labels to different objects or instances. Exploiting image level labels of the objects in images may help to solve this. Third, the depth feature in RGBD images and videos gives good information about the objects. There are already some works on RGBD co-segmentation such as [26]. It would be interesting to see how depth cue can be exploited in our fusion-based approaches of joint processing. Fourth, joint processing can help only up to certain extent, so can we automatically detect which are the images that need human interaction and which can be taken care off by computers itself (either through individual processing or joint processing). This is something in lines with the idea proposed in [29].

Many times image level labels contain not only the information about the semantic class of the objects in the image but also its attributes, for e.g. “black dog", “metallic chair" etc. While existing joint processing algorithms mainly exploit semantic commonness across the images, the extra information about the object in the form of attributes hasn't been exploited. Using these attributes can give us clue about what features to use for matching in a given set of images. [67] does recognize the necessity of using different features for different cases but uses weak prior of saliency to facilitate this.

In the past few years, deep learning based methods such as fully convolution networks (FCN) have achieved state-of-the-art performance on semantic image segmentation. Recently, [53] proposed a joint framework to combine interactive segmentation with FCN based semantic segmentation [81] so as to help each other. We can try to reduce the human interaction required using the joint processing techniques.

Semantic part segmentation[92] using joint processing is another interesting research direction that can be pursued. It may require some human interaction in few examples to indicate different parts that are desired to be labeled, and some new features will be needed as color and texture can hardly separate different parts of the body. Another idea could be using skeletonization to take help in separating the parts.

Despite making our fusion idea efficient, it's still not real-time. In order to overcome this, group saliency maps as proposed in our saliency fusion idea can be stored in the memory. These maps are quite good and can be just warped and fused to any new relevant and neighboring image in a weighted fashion; the time taken is as good as just one SIFT flow. However, all this is at the cost of memory. So, efficient use of the memory is challenge here.

Bibliography

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 34(11):2274–2282, 2012.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 34(11):2189–2202, Nov 2012.

[3] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 34(11):2189–2202, 2012.

[4] N. Alt, S. Hinterstoisser, and N. Navab. Rapid selection of reliable templates for visual tracking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1355–1362. IEEE, 2010.

[5] Anna Altman and Jacek Gondzio. Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. *Optimization Methods and Software*, 11(1-4):275–302, 1999.

[6] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1830–1837. IEEE, 2012.

[7] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2010.

[8] Dhruv Batra, Devi Parikh, Adarsh Kowdle, Tsuhan Chen, and Jiebo Luo. Seed image selection in interactive cosegmentation. In *International Conference on Image Processing (ICIP)*, pages 2393–2396. IEEE, 2009.

[9] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *European Conference on Computer Vision (ECCV)*, pages 109–122. Springer Berlin Heidelberg, 2002.

[10] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision (ECCV)*, pages 282–295. Springer, 2010.

[11] Joao Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3241–3248. IEEE, 2010.

[12] Yuning Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *Internation Conference on Computer Vision (ICCV)*, pages 2579–2586. IEEE, 2011.

[13] Kai-Yueh Chang, Tyng-Luh Liu, and Shang-Hong Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2136. IEEE, 2011.

[14] Hwann-Tzong Chen. Preattentive co-saliency detection. In *International Conference on Image Processing (ICIP)*, pages 1117–1120. IEEE, 2010.

[15] Ming-Ming Cheng, Victor Adrian Prisacariu, Shuai Zheng, Philip HS Torr, and Carsten Rother. Densecut: Densely connected crfs for realtime grabcut. In *Computer Graphics Forum*, volume 34, pages 193–201. Wiley Online Library, 2015.

[16] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 409–416. IEEE, 2011.

[17] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1210. IEEE, 2015.

[18] Wai-Pak Choi, Kin-Man Lam, and Wan-Chi Siu. Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition*, 36(3):721 – 729, 2003.

[19] Maxwell D Collins, Jia Xu, Leo Grady, and Vikas Singh. Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1656–1663. IEEE, 2012.

[20] Jifeng Dai, Ying Nian Wu, Jie Zhou, and Song-Chun Zhu. Cosegmentation and cosketch by unsupervised learning. In *International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[21] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[23] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] Alon Faktor and Michal Irani. Co-segmentation by composition. In *International Conference on Computer Vision (ICCV)*, pages 1297–1304. IEEE, 2013.

[25] Huazhu Fu, Xiaochun Cao, and Zhuowen Tu. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing (T-IP)*, 22(10):3766–3778, 2013.

[26] Huazhu Fu, Dong Xu, Stephen Lin, and Jiang Liu. Object-based rgbd image co-segmentation with mutex constraint. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[27] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3209. IEEE, 2012.

[28] Matthieu Guillaumin, Daniel KÃijttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision (IJCV)*, 110(3):328–348, 2014.

[29] Danna Gurari, Suyog Jain, Margrit Betke, and Kristen Grauman. Pull the plug? predicting if computers or humans should segment images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[30] Dorit S Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *International Conference on Computer Vision (ICCV)*, pages 269–276. IEEE, 2009.

[31] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 9:37–50, 1912.

[32] David E Jacobs, Dan B Goldman, and Eli Shechtman. Cosaliency: Where people look when comparing images. In *ACM Symposium on User Interface Software and Technology*, pages 219–228. ACM, 2010.

[33] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2555–2562. IEEE, 2013.

[34] K. R. Jerripothula, J. Cai, and J. Yuan. Group saliency propagation for large scale and quick image co-segmentation. In *International Conference on Image Processing (ICIP)*, pages 4639–4643. IEEE, 2015.

[35] K. R. Jerripothula, J. Cai, and J. Yuan. Image co-segmentation via saliency co-fusion. *IEEE Transactions on Multimedia*, 18(9):1896–1909, Sept 2016.

[36] Koteswar Rao Jerripothula, Jianfei Cai, Fanman Meng, and Junsong Yuan. Automatic image co-segmentation using geometric mean saliency. In *International Conference on Image Processing (ICIP)*, pages 3282–3286. IEEE, 2014.

[37] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Cats: Co-saliency activated tracklet selection for video co-localization. In *European Conference on Computer vision (ECCV)*, pages 187–202. Springer, 2016.

[38] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2083–2090. IEEE, 2013.

[39] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1943–1950. IEEE, 2010.

[40] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 542–549. IEEE, 2012.

[41] Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer vision (ECCV)*, pages 253–268. Springer, 2014.

[42] Gunhee Kim and Eric P Xing. On multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 837–844. IEEE, 2012.

[43] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *International Conference on Computer Vision (ICCV)*, pages 169–176. IEEE, 2011.

[44] Daniel Kuettel, Matthieu Guillaumin, and Vittorio Ferrari. Segmentation propagation in imagenet. In *European Conference on Computer Vision (ECCV)*, pages 459–473. Springer, 2012.

[45] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3173–3181, Dec 2015.

[46] T. S. H. Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *2013 IEEE International Conference on Computer Vision*, pages 1753–1760, Dec 2013.

[47] A. Levinshtein, S. Dickinson, and C. Sminchisescu. Multiscale symmetric part detection and grouping. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2162–2169, Sept 2009.

[48] H. Li, F. Meng, and K. N. Ngan. Co-salient object detection from multiple images. *IEEE Transactions on Multimedia (T-MM)*, 15(8):1896–1909, Dec 2013.

[49] H. Li and K. N. Ngan. A co-saliency model of image pairs. *IEEE Transactions on Image Processing*, 20(12):3365–3375, Dec 2011.

[50] Hongliang Li and King Ngi Ngan. A co-saliency model of image pairs. *IEEE Transactions on Image Processing (T-IP)*, 20(12):3365–3375, 2011.

[51] Xiaohui Li, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via dense and sparse reconstruction. In *International Conference on Computer Vision (ICCV)*, pages 2976–2983. IEEE, 2013.

[52] Yin Li, Xiaodi Hou, Christian Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 280–287. IEEE, 2014.

[53] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[54] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.

[55] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 33(5):978–994, 2011.

[56] Zhao Liu, Jianke Zhu, Jiajun Bu, and Chun Chen. Object cosegmentation by nonrigid mapping. *Neurocomputing*, 135:107–116, 2014.

[57] Zhi Liu, Wenbin Zou, Lina Li, Liquan Shen, and O. Le Meur. Co-saliency detection based on hierarchical segmentation. *IEEE Signal Processing Letters*, 21(1):88–92, Jan 2014.

[58] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[59] Tianyang Ma and Longin Jan Latecki. Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1955–1962. IEEE, 2013.

[60] Long Mai and Feng Liu. Comparing salient object detection results without ground truth. In *European Conference on Computer vision (ECCV)*, pages 76–91. Springer International Publishing, 2014.

[61] Ran Margolin, Ayellet Tal, and Lihi Zelnik-Manor. What makes a patch distinct? In *Computer Vision and Pattern Recognition (CVPR)*, pages 1139–1146. IEEE, 2013.

[62] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 26(6):810–815, 2004.

[63] Xue Mei and Haibin Ling. Robust visual tracking using l1 minimization. In *International Conference on Computer Vision (ICCV)*, pages 1436–1443. IEEE, 2009.

[64] Xue Mei, Haibin Ling, Yi Wu, E.P. Blasch, and Li Bai. Efficient minimum error bounded particle resampling l1 tracker with occlusion detection. *IEEE Transactions on Image Processing (T-IP)*, 22(7):2661–2675, July 2013.

[65] F. Meng, H. Li, G. Liu, and K. N. Ngan. Object co-segmentation based on shortest path algorithm and saliency model. *IEEE Transactions on Multimedia (T-MM)*, 14(5):1429–1441, Oct 2012.

[66] Fanman Meng, Jianfei Cai, and Hongliang Li. On multiple image group cosegmentation. In *Asian Conference on Computer Vision (ACCV)*, pages 258–272. Springer, 2014.

[67] Fanman Meng and Hongliang Li. Complexity awareness based feature adaptive cosegmentation. In *Internation Conference on Image Processing (ICIP)*, pages 4059–4063. IEEE, 2013.

[68] Fanman Meng, Hongliang Li, Guanghui Liu, and King Ngi Ngan. Object cosegmentation based on shortest path algorithm and saliency model. *IEEE Transactions on Multimedia (T-MM)*, 14(5):1429–1441, 2012.

[69] Fanman Meng, Bing Luo, and Chao Huang. Object co-segmentation based on directed graph clustering. In *Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2013.

[70] Lopamudra Mukherjee, Vikas Singh, and Chuck R Dyer. Half-integrality based algorithms for cosegmentation of images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2028–2035. IEEE, 2009.

[71] Lopamudra Mukherjee, Vikas Singh, and Jiming Peng. Scale invariant cosegmentation for image groups. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1881–1888. IEEE, 2011.

[72] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision(IJCV)*, 42(3):145–175, 2001.

[73] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[74] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *International Conference on Computer Vision (ICCV)*, pages 1777–1784. IEEE, 2013.

[75] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3282–3289. IEEE, 2012.

[76] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[77] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Computer Vision and Pattern Recognition(CVPR)*, pages 993–1000. IEEE, 2006.

[78] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1939–1946. IEEE, 2013.

[79] Jose C Rubio, Joan Serrat, Antonio López, and Nikos Paragios. Unsupervised cosegmentation through region matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 749–756. IEEE, 2012.

[80] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3 – 12, 2016. Special Issue on Skeletonization and its Application.

[81] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016.

[82] Wei Shen, Xiang Bai, Rong Hu, Hongyuan Wang, and Longin Jan Latecki. Skeleton growing and pruning with bending potential ratio. *Pattern Recognition*, 44(2):196 – 209, 2011.

[83] Wei Shen, Xiang Bai, Zihao Hu, and Zhijiang Zhang. Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition*, 52:306 – 316, 2016.

[84] Wei Shen, Xiang Bai, XingWei Yang, and Longin Jan Latecki. Skeleton pruning as trade-off between skeleton simplicity and reconstruction error. *Science China Information Sciences*, 56(4):1–14, 2013.

[85] Wei Shen, Yan Wang, Xiang Bai, Hongyuan Wang, and Longin Jan Latecki. Shape clustering: Common structure discovery. *Pattern Recognition*, 46(2):539 – 550, 2013.

[86] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Zhijiang Zhang, and Xiang Bai. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[87] Xiaohui Shen and Ying Wu. A unified approach to salient object detection via low rank matrix recovery. In *Computer Vision and Pattern Recognition (CVPR)*, pages 853–860. IEEE, 2012.

[88] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision (ECCV)*, pages 1–15. Springer, 2006.

[89] A. Sironi, V. Lepetit, and P. Fua. Multiscale centerline detection by learning a scale-space distance transform. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2697–2704, June 2014.

[90] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1464–1471. IEEE, 2014.

[91] Stavros Tsogkas and Iasonas Kokkinos. Learning-based symmetry detection in natural images. In *European Conference on Computer Vision (ECCV)*, pages 41–54. Springer Berlin Heidelberg, 2012.

[92] Stavros Tsogkas, Iasonas Kokkinos, George Papandreou, and Andrea Vedaldi. Semantic part segmentation with deep learning. *CoRR*, abs/1505.02438, 2015.

[93] Robert J Vanderbei and Tamra J Carpenter. Symmetric indefinite systems for interior point methods. *Mathematical Programming*, 58(1):1–32, 1993.

[94] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European Conference on Computer vision (ECCV)*, pages 705–718. Springer, 2008.

[95] A. Vezhnevets and V. Ferrari. Associative embeddings for large-scale knowledge transfer with self-assessment. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1987–1994. IEEE, 2014.

[96] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *European Conference on Computer Vision (ECCV)*, pages 465–479. Springer, 2010.

[97] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2217–2224. IEEE, 2011.

[98] Le Wang, Gang Hua, Rahul Sukthankar, Jianru Xue, and Nanning Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *European Conference on Computer Vision (ECCV)*, pages 640–655. Springer, 2014.

[99] N. Widynski, A. Moevus, and M. Mignotte. Local symmetry detection in natural images using a particle filtering approach. *IEEE Transactions on Image Processing*, 23(12):5309–5322, Dec 2014.

[100] S. Xie and Z. Tu. Holistically-nested edge detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, Dec 2015.

[101] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems 17*, pages 1537–1544. MIT Press, 2005.

[102] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162. IEEE, 2013.

[103] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3166–3173. IEEE, 2013.

[104] Zeyun Yu and Chandrajit Bajaj. A segmentation-free approach for skeletonization of gray-scale images via anisotropic vector diffusion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–415–I–420 Vol.1, June 2004.

[105] Junsong Yuan, Gangqiang Zhao, Yun Fu, Zhu Li, Aggelos K Katsaggelos, and Ying Wu. Discovering thematic objects in image collections and videos. *IEEE Transactions on Image Processing (T-IP)*, 21(4):2207–2219, 2012.

[106] Dingwen Zhang, Huazhu Fu, Junwei Han, and Feng Wu. A review of co-saliency detection technique: Fundamentals, applications, and challenges. *CoRR*, abs/1604.07090, 2016.

[107] Dingwen Zhang, Junwei Han, Chao Li, and Jingdong Wang. Co-saliency detection via looking deep and wide. In *Computer Vision and Pattern Recognition*, pages 2994–3002. IEEE, 2015.

[108] Q. Zhang and I. Couloigner. Accurate centerline detection and line width estimation of thick lines using the radon transform. *IEEE Transactions on Image Processing*, 16(2):310–316, Feb 2007.