Thèse

THESE INSA Rennes
sous le sceau de l'Université européenne de Bretagne
pour obtenir le titre de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Traitement du signal et de l'image

présentée par
Wenbin ZOU

ECOLE DOCTORALE : *MATISSE*
LABORATOIRE : *IETR CNRS UMR 6164*

Semantic-oriented Object Segmentation

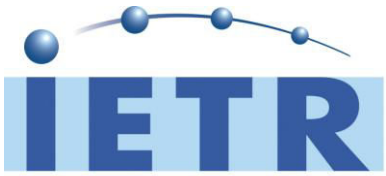
Frédéric JURIE
Professeur des Universités, Université de Caen / Rapporteur
Joseph RONSIN
Professeur des Universités, l'INSA de Rennes / Directeur de thèse (invité)

Thèse soutenue le 13.03.2014
devant le jury composé de :

Lotfi SENHADJI
Professeur des Universités, Université de Rennes1 / Président
Philippe CARRE
Professeur des Universités, Université de Poitiers / Rapporteur
Zhi LIU
Professeur des Universités, Shanghai University / Examinateur
Kidiyo KPALMA
Maître de conférence (HDR), l'INSA de Rennes / Co-encadrant

Semantic-oriented Object Segmentation

Wenbin ZOU



To My Parents

To Rong

Thank Dr. Kidiyo KPALMA!

Thank Prof. Joseph RONSIN!

Thank China Scholarship Council!

Thank my mother, Quanying ZOU!

Thank my father, Rencai ZOU!

Thank my wife, Rong YOU!

Thank all my friends!

Contents

1 Introduction

1.1 Background

Image segmentation, one of the fundamental problems in computer vision and image processing, is the process of grouping pixels of image into multiple sets, such that pixels within the same set share certain visual characteristics. A wide range of practical applications, such as semantic web, intelligent video coding, mobile robots, medical imagery and military surveillance, benefit from image segmentation.

In semantic web system, one of the biggest barriers is to associate visual content with a semantic label which describes a category of objects. According to the comprehensive study of InfoTrends [1], in the U.S. alone, 11 billions digital images were shared on social networking sites (e.g., Facebook, Twitter and Flickr) in 2010, and this amount will double in 2015. It is laborious and impracticable to manually annotate such huge volume of images. Image segmentation enables automatic object categorization.

In video coding system, globally lower bite-rate coding can be achieved by adaptively allocating more bits to highlight the desired objects like faces in the scenario of video conference, and less bits to background which is considered less important compared to target objects. In addition, image segmentation also helps to find the best matching reference frames/blocks for current coding frame/block and to improve the encoding efficiency.

Another application can be found in robotic system. A mobile robot is typically equipped with a camera to perceive environments where it evolves. Floor

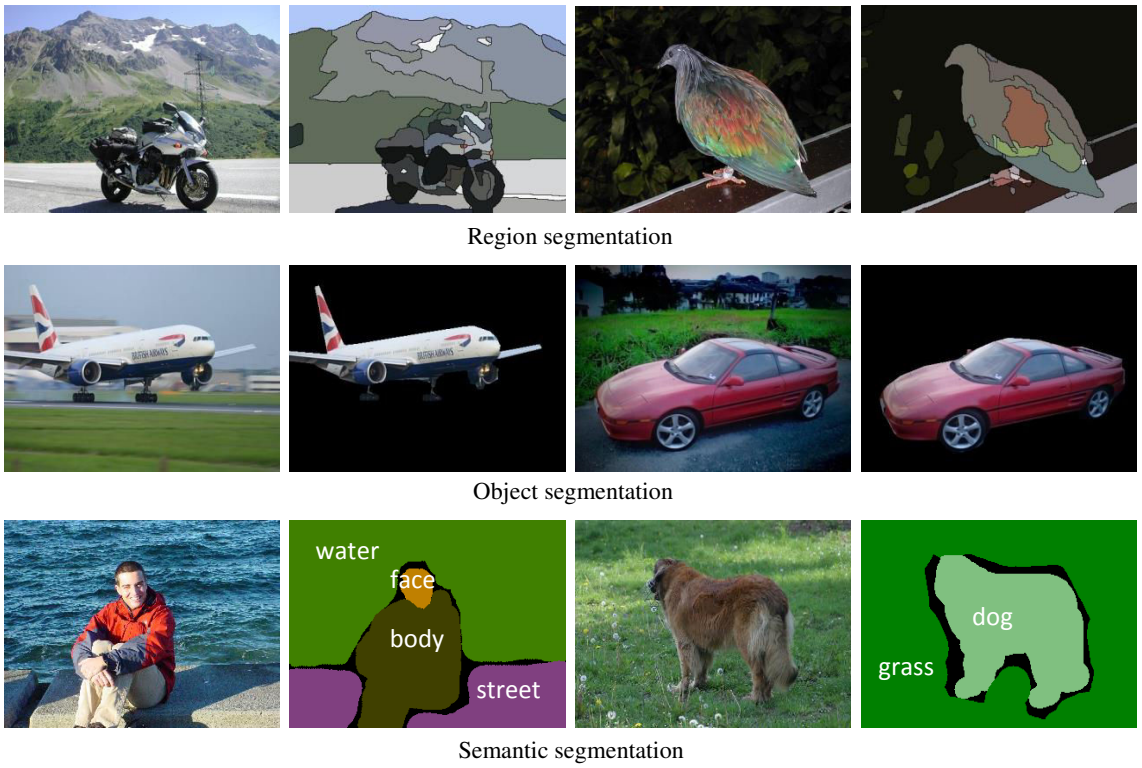


Figure 1.1: Examples of different segmentation categories. Top: region segmentation fuses pixels into homogeneous regions. Middle: object segmentation extract foreground objects. Bottom: semantic segmentation assigns a meaningful label to pixels of image.

segmentation is essential for robot navigation. To manipulate specific objects, like medical instruments in the operating room, the robot needs to know exactly which pixels belong to the object. In addition, the precisely segmented objects masks are also useful for identifying objects.

According to the goals of segmentation, existing approaches can be broadly classified into three categories: *region* segmentation, *object* segmentation and *semantic* segmentation. As shown in Figure 1.1, region segmentation partitions an image into a set of homogeneous regions; object segmentation, also termed as *figure-ground* segmentation, aims at separating objects from background; and semantic segmentation intends to assign a meaningful label, which describes a category of objects, to each pixel in a image. Region segmentation has been extensively studied for several decades and a number of approaches have been proposed, e.g. watersheds [2], active contours [3], mean-shift [4] and graph-based

segmentation [5, 6]. Object segmentation and semantic segmentation are more challenging than the region segmentation and have not been fully investigated. This thesis mainly focuses on object segmentation and semantic image segmentation.

1.2 Overview of the thesis

Depending on whether or not training on the manually labeled images or human intervention is required, object segmentation methods also can be broadly classified as *unsupervised* or *supervised* segmentation.

In practice, it is still not very feasible, in a fully unsupervised manner, to segment any objects in any images since what is defined as object depends on specific context and applications. Therefore, we only focus on salient object segmentation in an unsupervised manner, i.e., segmenting relatively outstanding objects from background by modeling the low-level data of image itself without using any top-down cues. Moreover, we also address a more difficult case with the objective to extract all foreground objects in an image by leveraging a set of manually segmented exemplar images. As the objects to be segmented maybe never appear in the exemplar images, this approach can be considered as a weakly supervised segmentation approach. Both of the aforementioned approaches produce a binary segmentation mask, where one label indicates objects and the other label represents background.

In addition, we also address the problem of assigning a meaningful label (like cat, dog, car or road) to each pixel in the image, which is so-called semantic segmentation. In this connection we propose a feature representation method to bridge the gap between local features and semantics. Semantic segmentation requires a set of semantically pixel-wise labeled images for semantic prediction, and each pixel in a test image only can be assigned to one of the pre-defined categories. Such an approach is categorized to the supervised segmentation.

In the following subsections, we overview the content of this thesis and its main contributions for the purpose of leading readers to understand it. Two proposed

approaches to object segmentation will be briefly introduced in Section 1.2.1 and Section 1.2.2, respectively. Then the proposed semantic segmentation approach is summarized in Section 1.2.3. Finally the key aspects in our conclusion are presented in Section 1.2.4.

1.2.1 Chapter 2: Saliency-based object segmentation

Chapter 2 presents a novel saliency detection model which aims at identifying relatively outstanding regions in an image, and a unified framework for jointly addressing unsupervised salient object segmentation and saliency boosting.

For the saliency detection, we propose a segmentation driven low-rank matrix recovery (SLR) model, in which a prior matrix, derived from region segmentation, is proposed to highlight potential salient objects and suppress typical background regions, and is efectively exploited to modulate low-rank matrix recovery model. In such a model, the feature matrix input image is decomposed into a low-rank matrix which naturally represents background regions, and a sparse matrix which potentially captures salient objects. The output of the saliency detection model is a saliency map, in which each pixel is labeled by a real value within the range of [0, 1] to indicate its saliency probability.

For object segmentation, we derive object location information from the detected saliency map and seamlessly integrate it into our object segmentation model which is formulated within Markov random field (MRF) framework. Moreover, observing the fact that saliency detection model might generate an imperfect saliency map where background regions may be highlighted and object regions may be suppressed as well, we propose a saliency boosting model which aims to improve the quality of saliency map by effectively exploiting the segmentation result. The boosted saliency map is then used as a new constraint for object segmentation. Therefore, iteratively optimizing the segmentation model and the saliency boosting model leads to the optimal saliency map and final segmentation.

Extensive evaluations on MSRA-B dataset and PASCAL-1500 dataset, demonstrate that the proposed approach outperforms the state-of-the-art techniques for both saliency detection and salient object segmentation. Figure 1.2

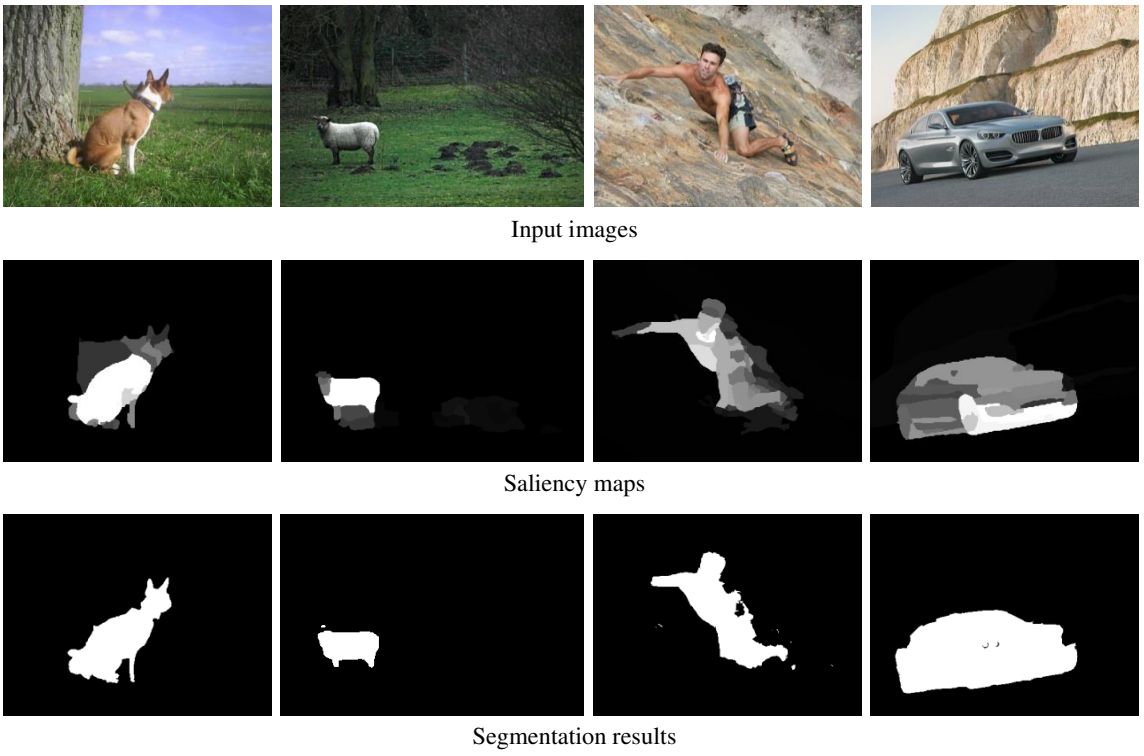


Figure 1.2: Some examples of saliency maps and segmentation results generated by the proposed saliency detection model and segmentation approach. Top: input images. Middle: saliency maps. Bottom: segmentation results.

shows some saliency maps and the corresponding segmentation results generated by the proposed approach.

The main contributions of this chapter are:

- a saliency detection model based on low-rank matrix decomposition, which is shown to outperform the state-of-the-art saliency detection models in both objectively and subjectively evaluations.
- a unified framework for joint saliency-based object segmentation and saliency boosting, which iteratively and mutually addresses one of the two tasks and leads to optimal object segmentation and saliency detection.
- a collected publicly available dataset containing 1500 images with ground truths for the performance evaluation of saliency detection and salient object segmentation.

1.2.2 Chapter 3: Exemplar-based object segmentation

Chapter 3 presents an exemplar-based object segmentation approach. The underlying idea of this approach is to transfer segmentation masks of similar exemplar images into the input image.

So the first and critical problem is how to find the most matching exemplar images for segmentation transfer. We propose a novel high-level image representation method named as *object-oriented descriptor* (OOD) which is able to highlight local objects and to represent global geometric layout of the image. By using this descriptor, a set of exemplar images globally and locally (glocally) similar to the query image is retrieved. Then, the exemplar images are segmented into regions, and a discriminative classifier of support vector machine (SVM) is learned on-the-fly from these exemplar regions and is subsequently used to predict foreground probability for each region in the query image. Therefore, we can obtain a probabilistic SVM map, from the SVM classifier, which carries important semantic information. Such an SVM map is exploited to define a robust segmentation model based on Markov random field (MRF) framework, which associates each pixel with a random variable indicating "object" or "background", and the final segmentation is achieved by finding the maximum a posteriori (MAP) configuration in the MRF.

The proposed approach has been extensively evaluated on three challenging datasets including Pascal VOC 2010, VOC 2011 segmentation dataset and iCoseg dataset. Experiments demonstrate that: the proposed approach outperforms the state-of-the-art object segmentation methods, and has the potential to segment large-scale images containing unknown objects, which never appeared in the exemplar images. Figure 1.3 shows some segmentation results generated by the proposed approach.

The main contributions of this chapter are:

– A novel high-level image descriptor which enable finding most matching exemplar images for segmentation transfer.
– A generic object segmentation framework combining online prediction and

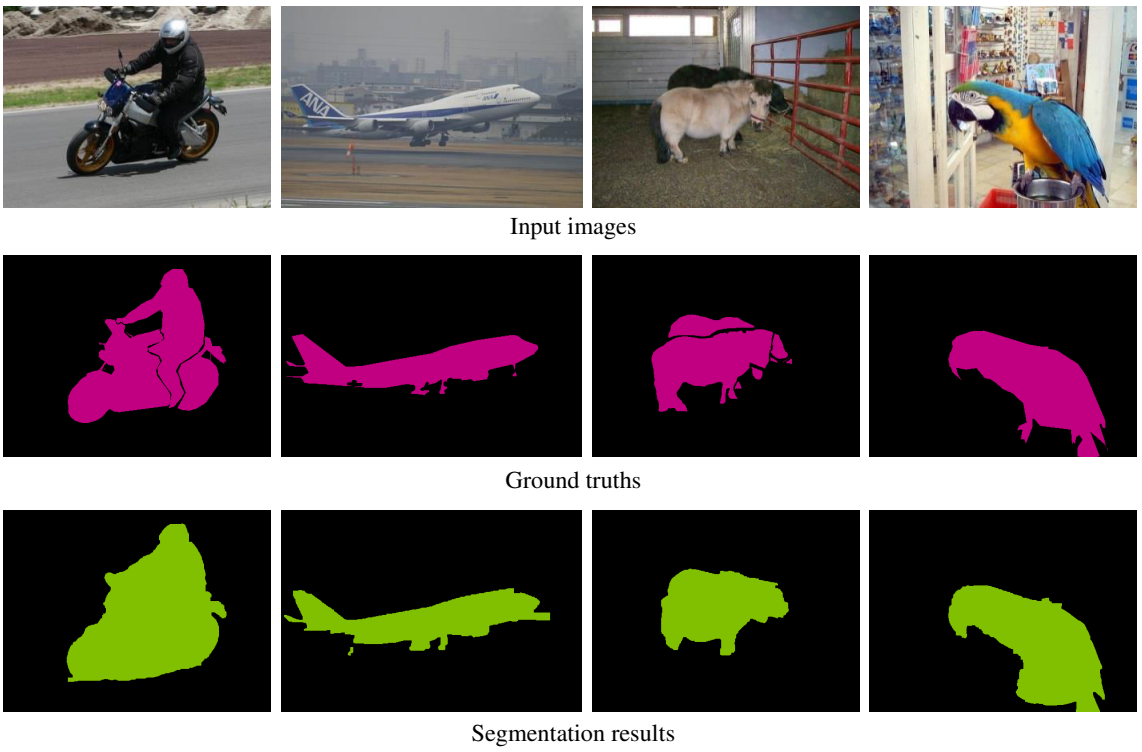


Figure 1.3: Some example segmentation results produced by the proposed approach. Top: input images. Middle: manually segmented ground truths. Bottom: our object segmentation results.

MRF energy optimization, which is shown to be superior to existing exemplar-based object segmentation approaches.

– Potential application to extracting objects in large-scale internet images by leveraging a set of available exemplars.

1.2.3 Chapter 4: Semantic image segmentation

Chapter 4 presents a unified framework for semantic image segmentation which aims to assign a semantic label to each pixel in an image.

For the purpose of capturing objects as completely as possible using unsupervised region segmentation, we generate a region bank for the input image and for training images, respectively. The region bank is a collection of regions generated from multiple scales of hierarchical segmentation. The region hierarchy provides a natural constraint for feature extraction. For robust and compact region representation, we propose a *sparse coding* method which represents each local feature within a region

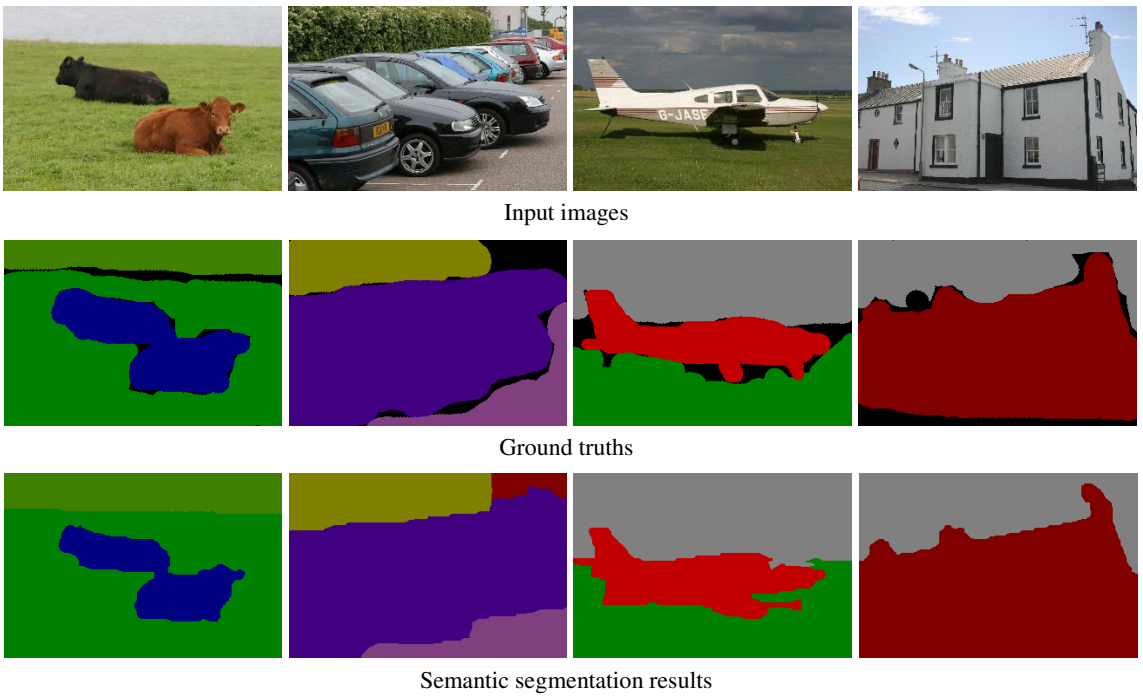


Figure 1.4: Some example semantic segmentation results produced by the proposed approach. Top: input images. Middle: manually annotated ground truths where each object is labeled by a unique color and black indicates void area for accuracy computing. Bottom: our semantic segmentation results.

by several basic vectors in a learned visual dictionary, and each region is described by a sparse histogram. With the sparse-based region description, we apply SVM classifiers for semantic inference and fuse all regions in the region bank to generate a semantic labeling map.

The proposed approach is evaluated on the standard Microsoft Research Cambridge (MSRC 21) dataset. Experiments demonstrate that the proposed approach obtains competitive performance compared to recent studies reported in the literature. Figure 1.4 shows some examples of semantic segmentation results generated by the proposed approach.

The main contributions of this chapter are:

– A simple yet effective framework for semantic image segmentation, which obtains competitive results on a standard evaluation dataset.
– A sparse-based image representation method which compacts local feature descriptors into a single histogram and is shown more robust compared to the

traditional bag of visual-words method.

1.2.4 Chapter 5: Conclusion

Chapter 5 concludes this thesis by summarizing the proposed approaches for both object segmentation and semantic image segmentation. Some reflections for further improvement based on our approaches will be presented in this chapter as well.

2 Saliency-based object segmentation

2.1 Introduction

Saliency detection and object segmentation are two of the fundamental problems of computer vision. The problem of detecting visual saliency consists of identifying what mostly captures the human perceptual attention in an image. This is a core problem with a variety of applications such as image quality evaluation, image summarization and picture collage. The object segmentation aims to segment out foreground objects from background. This often serves as a key preprocessing step for many applications, e.g., object recognition, object tracking, content-based image retrieval, etc.

While the saliency detection and object segmentation are seemingly independent; in fact, an inextricable connection exists between them. Since objects are generally salient relative to their surrounding background regions in terms of visual properties, solving any one of them leads to addressing the other one either explicitly or implicitly. Indeed, many object segmentation approaches are built on the result of saliency detection model which is the so-called saliency map. A typical solution is to derive location information of objects from the saliency map for the segmentation. For instance, we can at least appropriately localize objects by thresholding the saliency map.

On the contrary, object segmentation may be helpful to identify saliency in the image as well. As a matter of fact, accurate object segmentation is the ultimate goal of object-level saliency detection. If object segmentation model is sufficiently robust,

saliency can be boosted by highlighting regions of the segmented objects. However, to the best of our knowledge, none of previous work exploited object segmentation cues for saliency detection, perhaps it is difficult to judge whether an unsupervised segmentation is sufficiently good or not.

In this chapter, we investigate to jointly address the saliency detection and object segmentation together by exploiting beneficial cues from each of them. To achieve this goal, we propose a system consisting of two key components and also corresponding to our two main contributions. The first one is a saliency detection model, called segmentation driven low-rank matrix recovery (SLR) model and originally appeared as [7]. This model proposed a bottom-up segmentation prior, which highlights potential objects in the image, to guide the feature matrix decomposition from which salient regions can be discovered. The second one is a unified scheme which jointly addresses saliency boosting and object segmentation. This scheme works iteratively and mutually to improve the quality of saliency map and to segment out objects from background.

This chapter is organized as follows: in Section 2.2, we briefly survey the relevant literature in saliency detection and saliency-based object segmentation. Then we describe, in detail, a saliency detection model which generates initial saliency map, and a unified framework for jointly addressing saliency boosting and object segmentation in Section 2.3 and Section 2.4, respectively. Experimental evaluation and discussion are presented in Section 2.5. Finally, we conclude this chapter in Section 2.6.

2.2 Related work

This section briefly introduces the related work on saliency detection and saliency-based object segmentation.

2.2.1 Saliency detection

Existing saliency models can be broadly classified into two categories: biological models and computational ones.

The biological model is pioneered by Koch and Ullman [8] who derived visual saliency from a set of topographical maps of elementary features like orientation of edges, color and luminance. The biological model is usually implemented using the center-surround scheme with different formulations on a set of features [9–11]. As the objective of biological models is to find some points that mostly catch human attention, the resulting saliency maps are typically sparse and blurry, and limit their applications mainly for prediction of eye fixations.

Instead, the computational models, inspired by the biological models, aim at discovering objects standing out from surrounding regions. A number of computational models measure the saliency based on global, local and regional contrasts with different forms [12–16]. A variety of theories and methods, including information theory [17, 18], graph theory [19, 20], machine learning [21, 22], statistical model [23–25], Bayesian model [26], frequency domain analysis [27–29], have been exploited to build saliency models. Recently, some saliency models such as [13, 16, 23, 30, 31] benefit from measuring the saliency on the basis of region segmentation to effectively incorporate global information at region level, in order to improve the saliency detection performance. Besides, some recent saliency models also exploit some form of priors such as background prior [32], generic objectness [33] and object-level shape prior [15]. A recent benchmark on saliency detection performance of different saliency models can be found in [34]. Although these saliency models may work well for images with consistent scenes like the images in MSRA-1000 dataset [12], they still lack robustness to detect objects in complex images with cluttered background and/or low contrast between objects and background.

Recently, a new trend is to formulate the problem of saliency detection with low-rank matrix recovery (LRMR) model, in which an image is decomposed into a low-rank matrix which corresponds to the background, and a sparse one which links to salient objects. In [35], sparse coding is used as an intermediate representation of image features and then fits to LRMR model to recover salient objects. As pointed out in [36], the sparse coding can not guarantee that, in the entire image representation, the sparse codes of salient objects are sparse and those

of the background are of low-rank. Therefore, [36] proposed to modulate the image features with the learned transform matrix and high-level priors to meet the low-rank and sparse properties. This sounds reasonable and remarkable experiment results have been demonstrated. Unfortunately, the supervised training is required and the learned transform matrix is somewhat biased toward the training dataset, therefore it suffers from the limited adaptability.

Based on the aforementioned issues, we present an unsupervised LRMR model for saliency detection. The key idea is to derive a bottom-up prior to constrain image features so that they can fit well to the LRMR model. For this purpose, we propose a novel generic prior named as *segmentation prior* which is created from a bottom-up segmentation. The segmentation prior softly separates objects from background so that the objects are highlighted and the background is highly redundant in the feature domain.

2.2.2 Object segmentation

A number of methods derived useful information from saliency map for unsupervised object segmentation. In [37], salient regions are selected in a saliency map by a trained support vector machine using image segment features, and then the selected regions are merged to form the object segmentation result. In [38], segmentation seeds are derived from the saliency map and standard Markov random field framework is applied to object segmentation by integrating image features in terms of color, luminance and edge orientation. In [12], salient objects are extracted by using a thresholding method, in which those segments with average saliency values greater than the twice the mean of saliency values in the entire saliency map are composed of objects, and the other segments are considered as background.

Many methods are built under the framework of graph cuts [39], in which a graph is associated with each pixel and is constructed based on a data term, which measures the likelihood of a pixel to be labeled as object, and a smoothness term, which ensures overall label smoothing. The binary segmentation is achieved by finding the minimum cut between object and background nodes in the graph. In [40], the data

term is defined as the summation of saliency and color similarity, and conditional random field learning is employed to train the balance weight between the data term and smoothness term. In [23], segmentation process consists of two phases, in which saliency map is exploited to generate an initial segmentation, and then iterative graph cuts with adaptive seed adjustment and parameter refinement leads to the final segmentation. In [15], graph cuts is also performed iteratively using a histogram-based data term and a shape constraint smoothness term. In [13], an initial segmentation is obtained by thresholding the saliency map at a fixed value which gives 95% recall rate in the total dataset, and final segmentation result is obtained by using GrabCut [41], which iteratively predicts foreground/background appearance with Gaussian Mixture Models and segments image with the graph cuts.

Although a number of approaches exploited saliency map for object segmentation in different forms, a key fact seems to be ignored that the saliency map itself may contain noises and might lead to error propagation in the process of object segmentation, and there is no feedback from the segmentation result to gauge the saliency map. In contrast to the previous unidirectional saliency-based object segmentation methods, in which only the detected saliency map is utilized to guide object segmentation, we also aim to boost the quality of saliency map by leveraging the object segmentation results. Our hypothesis is that saliency optimization and object segmentation can be mutually reinforced, as objects can be localized more accurately with both of them. Therefore, our framework interactively performs object segmentation and saliency boosting. mutually optimizing the defined objective functions of saliency boosting model and object segmentation model leads to the optimal saliency map and the final object segmentation result.

2.3 Saliency detection model

In this section, we present a saliency detection model called as segmentation driven low-rank matrix recovery (SLR) model which is used to generate a saliency map from input image.

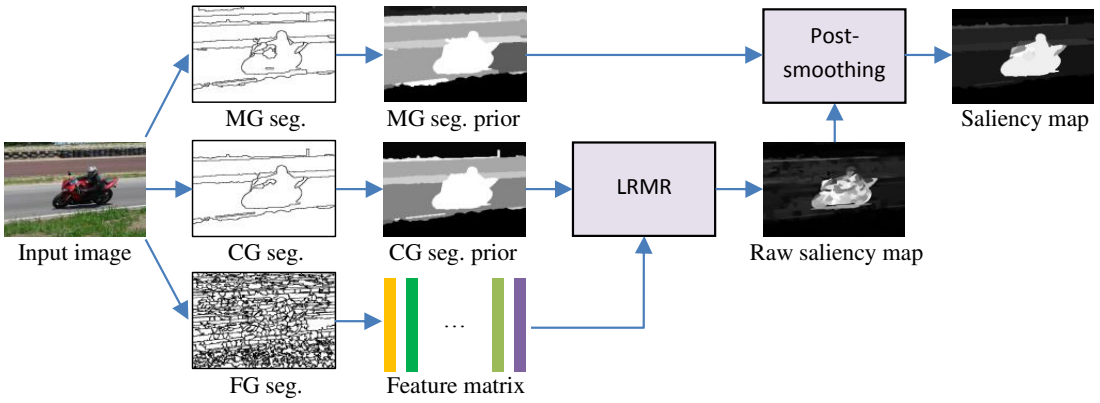


Figure 2.1: Framework of the proposed saliency model. Input image is firstly segmented into three levels. Feature descriptors are accumulated within superpixels of fine-grained (FG) segmentation. Segmentation priors are derived from the medium-grained (MG) segmentation and coarse-grained (CG) segmentation, respectively. The final saliency map is obtained by smoothing the raw saliency map generated by LRMR model with the MG segmentation prior.

Figure 2.1 presents the framework of the proposed segmentation driven low-rank matrix recovery model. An input image is firstly segmented into three-level segmentations: *fine-grained* (FG), *medium-grained* (MG) and *coarse-grained* (CG). The FG segmentation significantly over-segments the image into a number of superpixels (to avoid confusion, the segments of FG segmentation are called "superpixels rather than "regions"used in MG and CG segmentations). The MG segmentation also over-segments the image but generates regions as few as possible. The CG segmentation aims at maximally separating objects from the background, thus the image may be over-segmented or under-segmented. Based on these three-level segmentations, image features are extracted from the superpixels, and segmentation priors are derived from the MG and CG segmentations. Then, the low-rank matrix recovery (LRMR) model is applied to generate the raw saliency map. Finally, the raw saliency map is smoothed by using the MG segmentation prior to generate an optimal saliency map.

2.3.1 Low-rank matrix recovery model

Given an input image $\mathbf{I}$, let $\mathbf{P} = \{p_1, p_2, \cdots, p_N\}$ be a set of N superpixels created by FG segmentation, and $\mathbf{a}_n \in \mathbb{R}^{d\times 1}$ be the feature vector of the superpixel p_n, where d is the dimension of feature descriptor. The image $\mathbf{I}$ is represented by a feature matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N] \in \mathbb{R}^{d\times N}$.

In real images, the background pixels generally show similar appearance, and have strong correlation in the feature space. This suggests that the feature matrix $\mathbf{A}$ might have low-rank property, and it can be decomposed into two parts, a low-rank matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N] \in \mathbb{R}^{d\times N}$, and a sparse one $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_N] \in \mathbb{R}^{d\times N}$

$$\mathbf{A} = \mathbf{U} + \mathbf{E} \tag{2.1}$$

Applying this model to saliency detection, the background is naturally represented by the low-rank matrix $\mathbf{U}$, and the objects might be captured by the sparse matrix $\mathbf{E}$.

To recover the matrices $\mathbf{U}$ and $\mathbf{E}$, the problem can be formulated with the Lagrangian representation

$$\begin{aligned} \min \quad & rank(\mathbf{U}) + \lambda\|\mathbf{E}\|_0 \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{U} + \mathbf{E} \end{aligned} \tag{2.2}$$

where λ is a coefficient to balance $\mathbf{U}$ and $\mathbf{E}$, and $\|\cdot\|_0$ indicates l_0-norm. Unfortunately, this is a NP-hard problem as the matrix rank and l_0-norm are not convex. Recent theoretic analysis in [42] shows that, under rather weak assumptions, the low-rank matrix $\mathbf{U}$ and the sparse matrix $\mathbf{E}$ can be exactly recovered by

$$\begin{aligned} \min \quad & \|\mathbf{U}\|_* + \lambda\|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{U} + \mathbf{E} \end{aligned} \tag{2.3}$$

where $\|\cdot\|_*$ is the nuclear norm of matrix $\mathbf{U}$ (the sum of singular values of $\mathbf{U}$), and $\|\cdot\|_1$ indicates l_1-norm. The regularization of l_1-norm ensures to produce a

sparse matrix $\mathbf{E}$. The optimal matrices $\mathbf{U}$ and $\mathbf{E}$ can be obtained by alternatively minimizing (2.3) over one while keeping the other one fixed.

With the optimal sparse matrix $\mathbf{E}$, the saliency value of superpixel p_n is given by the l_1 energy of the vector $\mathbf{e}_n$

$$s_n = \sum_{i=1}^{d} |\mathbf{e}_n(i)| \tag{2.4}$$

The saliency value s_n represents the probability of superpixel p_n to belong to an object, i.e., a larger value indicates a higher probability, and vice versa. The saliency map of image $\mathbf{I}$ is then generated by assigning the saliency value of each superpixel to all pixels in the superpixel.

2.3.2 LRMR with segmentation prior

Directly fitting the LRMR model to the problem of saliency detection is under the assumption that the background is homogeneous and has a high contrast with objects. In the reality, however, many backgrounds are cluttered and objects may be similar to part of the background regions. This results in false positive detection results. To improve the robustness of saliency detection, a feasible method is to adopt high-level priors to modulate input features [36, 43], so that the feature matrix has a lower rank. The underlying idea of the modulation is to give small weights to feature vectors of those superpixels which are more likely to be background and large weights to those corresponding to objects.

Many priors have been proposed for saliency detection, such as center prior, color prior and learnt transform prior [36]. The main drawback of these priors is the lack of adaptability, since they are either obtained by empirical statistics or trained from the annotated images. For example, center prior assumes that objects always appear in the center of image, and color prior believes that the objects are in warm colors. Obviously, these assumptions are not always valid in practice. In addition, the learnt transform prior tends to fail when the test image has a high difference with the training images.

Here we introduce a bottom-up segmentation driven prior, named as

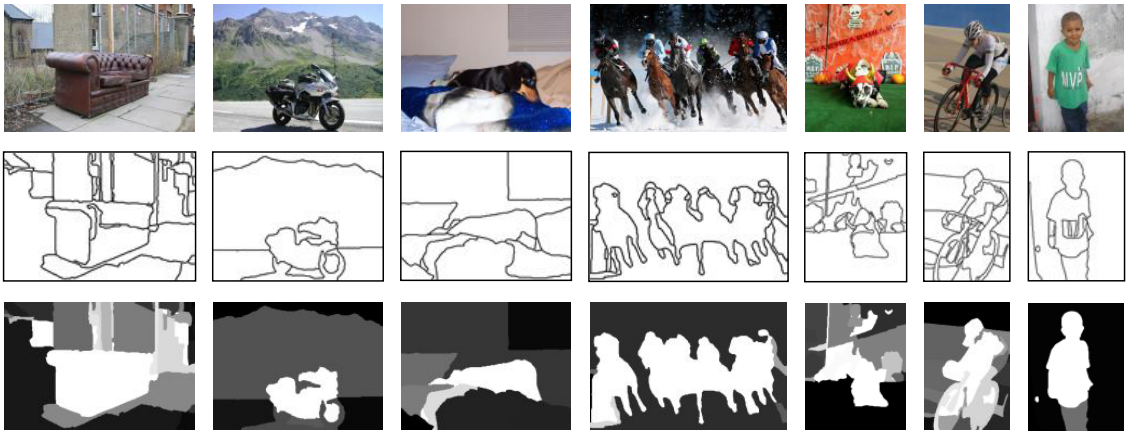


Figure 2.2: Examples of segmentation prior. First row: input images; second row: bottom-up segmentation results; last row: segmentation prior where white indicates a higher weight and black represent a lower weight.

segmentation prior. Firstly, let us take a look at the images and their CG segmentations in Figure 2.2. Salient objects locate at diversity of positions: center, bottom, left, right and corner. Both background and objects are typically segmented into several regions, thus, the bottom-up segmentation can not be expected to totally separate objects from background. However, the segmented regions of background have very high probability of connecting with the border of the image, while very few object regions link to it. Even if an object is truncated on the border, like the bike and the child of the two right-most images, border regions of object are small compared to the whole object in the image. In contrast, the border regions of the background are usually large, as background appears more uniform, like sky, road, tree, wall, etc. This observation implies that objects can be roughly separated from the background by the bottom-up segmentation. Therefore, we propose the segmentation prior according to the connectivity between each region and image border. Let r_m be a segmented region of image $\mathbf{I}$, the segmentation prior of region r_m is defined as

$$h_m = \exp(-\frac{|r_m \cap C|}{\sigma\psi_m}) \tag{2.5}$$

where $|\cdot|$ denotes the length of intersection, C is the border of image $\mathbf{I}$, ψ_m is the

outer perimeter of region r_m, and σ is a balance parameter which is set to 0.3 in our experiments. Clearly, if a region touches the image border, its prior value is in the range of $(0, 1)$, otherwise it is equal to 1. In other words, the segmentation prior gives a small weight to the region touching the image border. By using (2.5), segmentation priors of all regions can be computed, and form the prior of the input image.

In Figure 2.2, one might observe that, on one hand, there are still some regions of the background without connection with the image border, on the other hand, some regions of objects are inevitably merged with the background. Indeed, such a strategy can not perfectly separate the objects from the background. However, the segmentation prior derived from CG segmentation can serve as a guidance cue for LRMR model to address the task of saliency detection.

Suppose an input image $\mathbf{I}$ is segmented into N superpixels by FG segmentation, and represented by a feature matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N]$. Let $\mathbf{H}_c = [h_1^c, h_2^c, \cdots, h_N^c]$ denote a set of CG segmentation prior values of the superpixels. In order to recover salient objects well with the LRMR model, the feature matrix $\mathbf{A}$ is firstly modulated by the CG segmentation prior $\mathbf{H}_c$

$$\mathbf{B} = [h_1^c \mathbf{a}_1, h_2^c \mathbf{a}_2, \cdots, h_N^c \mathbf{a}_N]. \tag{2.6}$$

Then, the modulated feature matrix $\mathbf{B}$ is used as the input of the standard LRMR model

$$\begin{aligned} \min \quad & \|\mathbf{U}\|_* + \lambda \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{B} = \mathbf{U} + \mathbf{E} \end{aligned} \tag{2.7}$$

As the segmentation prior assigns small weights to most of background feature vectors in $\mathbf{B}$, the l_1 energies of the corresponding vectors in the recovered matrix $\mathbf{E}$ are inclined to be small. Therefore, objects are highlighted more effectively in the matrix $\mathbf{E}$.

2.3.3 Post-smoothing

Raw saliency map generated by the LRMR model might still contain some noises: some large saliency values in background area and/or small values in objects. There are mainly two reasons for this phenomenon: on one hand, some superpixels of background might be strongly similar to those of objects in the feature domain; on the other hand, the LRMR model decomposes the feature matrix without considering spatial constraint. To remove the noises, the raw saliency map is smoothed at two scales: FG and MG levels.

Let $\mathbf{S} = \{s_1, s_2, \cdots, s_N\}$ denote the saliency values of all superpixels $\mathbf{P} = \{p_1, p_2, \cdots, p_N\}$ in the image. At the FG level, the saliency of each superpixel p_n is impacted by its adjacent superpixels

$$s'_n = s_n + \alpha \sum_{j \in \mathcal{N}} s_j \cdot \exp(-\|\mathbf{a}_n - \mathbf{a}_j\|_2^2) \tag{2.8}$$

where $\mathcal{N}$ is a set of adjacent neighbours of superpixel p_n, $\|\cdot\|_2$ denotes l_2-norm. The weight α is used to balance the impact of neighbours on the current superpixel, and is set to 0.5 in our experiments. Obviously, neighbours with appearance more similar to the current superpixel are considered to give more contribution to compute the saliency, and vice versa. Therefore, the FG level smoothing ensures the saliency of each superpixel is coherent with its neighbours showing a high similarity on features.

The FG level smoothing might be still far from labelling saliency at object level. We also perform a MG level processing on the FG smoothed saliency map $\mathbf{S}'$. To do this, segmentation prior of the MG segmentation is also computed.

Let $\mathbf{S}' = \{s'_1, s'_2, \cdots, s'_K\}$,$\mathbf{R} = \{r_1, r_2, \cdots, r_L\}$ and $\mathbf{H}_m = \{h_1^m, h_2^m, \cdots, h_K^m\}$ denote the FG smoothed saliency map, MG segmentation and MG segmentation prior of the input image, respectively, where K is the number of pixels, L is the number of regions. The saliency value of region r_l is computed by

$$\overline{s}_l = \frac{1}{T_l} \sum_{k \in r_l} s''_k \tag{2.9}$$

where T_l is the number of pixels in the region r_l, s''_k is the weighted saliency value

of pixel k

$$s''_k = h_k^m s'_k. \tag{2.10}$$

Therefore, the final saliency map of image is obtained by distributing saliency values of all regions into corresponding pixels. Notice that, the parameters of MG segmentation are set to ensure an image is over-segmented with as few regions as possible. Thus, this process generates more smooth saliency map than assigning saliency values based on superpixels; in addition, object contours are also preserved well.

2.4 Joint object segmentation and saliency boosting

In this section, we describe a unified scheme which jointly segments foreground objects from background and optimizes the saliency map obtained in the previous section.

Given an input image $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}, \mathbf{x}_n \in \mathbb{R}^3$ and its saliency map $\mathbf{S} = \{s_1, s_2, \cdots, s_N\}, s_n \in \mathbb{R}^1$, where N is the number of pixels in the image, our goal is, (i) to generate a new saliency map $\mathbf{S}^* = \{s_1^*, s_2^*, \cdots, s_N^*\}, s_n^* \in \mathbb{R}^1$ in which objects are more highlighted and irrelevant background regions are more suppressed, (ii) and to find a label array $\mathbf{L} = \{l_1, l_2, \cdots, l_N\}, l_n \in \{0, 1\}$, which represents a segmentation of the input image X such that

$$l_n = \begin{cases} 1 & \text{if pixel } n \text{ belongs to the foreground} \\ 0 & \text{otherwise} \end{cases}$$

As illustrated in Figure 2.3, we address both object segmentation and saliency optimization jointly as follows:

1. Propose a candidate solution for image saliency map $\mathbf{S}$.
2. Using the saliency map $\mathbf{S}$, segment out objects from background by using the segmentation model described in Section 2.4.1.

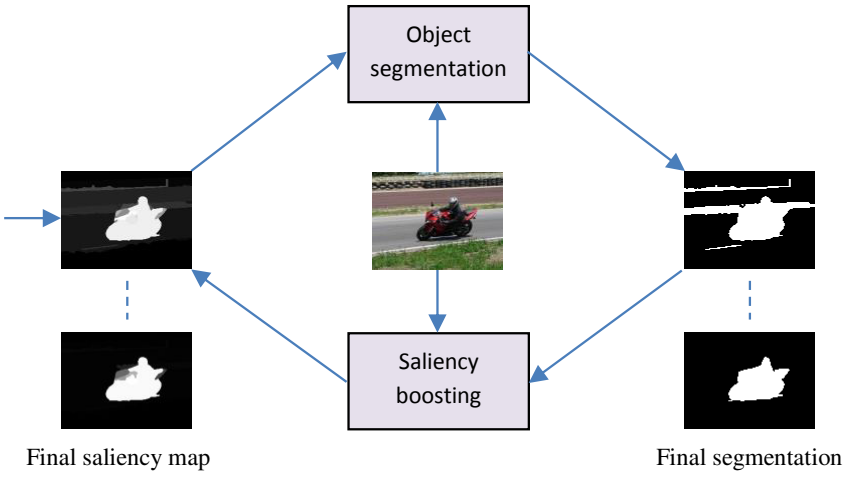


Figure 2.3: Unified framework of joint object segmentation and saliency boosting.

3. Based on the segmentation result, optimize the saliency map **S** by using the saliency boosting model presented in Section 2.4.2.

4. Repeat the processing from step 2 until convergence or at the maximal iterations.

In the following three subsections, we firstly detail the object segmentation model and the saliency optimization model respectively, and then summarize the procedure of the iterative and interactive optimization scheme.

2.4.1 Object segmentation model

The widely acknowledged standard object segmentation methods, e.g. [39, 41, 44], are based on Markov random field (MRF) framework, which associates each pixel with a random variable representing the segmentation label of a pixel. The optimal segmentation is achieved by minimizing a binary pairwise energy function defined over the random variables, which can be efficiently solved via graph cuts [39]. We also employ the MRF framework for object segmentation. In contrast to previous methods, we mainly focus on how to obtain a robust segmentation model by the optimum use of saliency information. In the rest of this subsection, we briefly introduce the basic knowledge of MRF for object segmentation, and then we show how to leverage useful saliency information to derive a robust segmentation model.

MRF for object segmentation

An input image is a discrete array $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}, \mathbf{x}_n \in \mathbb{R}^3$ of color pixels. The MRF framework, with a neighborhood system $\mathcal{N}$, defines a random variable l_n over each pixel n and each random variable l_n represents a segmentation label which is either 1 or 0. The neighborhood system $\mathcal{N}$ of MRF consists of a set of $\mathcal{N}_n, \forall n \in \mathcal{P}$, where $\mathcal{N}_n$ denotes a set of all 4-connectivity or 8-connectivity neighbors of pixel n and $\mathcal{P}$ denotes the set of all pixels in the image. An energy function E is defined with respect to the set of segmentation labels (random variables) $\mathbf{L} = \{l_1, l_2, \cdots, l_N\}$, so that its minimum should lead to the optimal segmentation by finding the maximum a posteriori (MAP) configuration in an MRF. The standard form of MRF model for object segmentation is defined as:

$$E(\boldsymbol{L}) = \sum_{n \in \mathcal{P}} \Lambda_n(l_n) + \sum_{\{n,j\} \in \mathcal{N}} \Theta_{n,j}(l_{n,}, l_j) \tag{2.11}$$

where Λ_n is the data term which measures the consistency between a segmentation label l_n and a pixel n by evaluating the extracted data (like color feature) from the image, $\Theta_{n,j}$ is the smoothness term which ensures the overall segmentation smoothing by penalizing neighboring pixels assigned with different labels.

Object segmentation with saliency information

Though the MRF model is shown to be successful to address the problem of object segmentation, its two components, i.e. data term and smoothness term, should be defined appropriately. Our data term and smoothness term are described as follows.

Data term The data term Λ_n is a function measuring the negative log of likelihood degree of labeling pixel n as foreground or background. Typically it is computed from the appearance model of foreground/background. We propose a new data term which consists of a location model Φ and an appearance model Ω

$$\Lambda_n(l_n) = -\log \Phi_n(l_n) - \log \Omega(\mathbf{x}_n | l_n). \tag{2.12}$$

The location model Φ estimates the probability of pixel n to be labeled as foreground/background according to its location in the image. As objects appear more salient relative to surrounding background, we derive the location model from saliency map. Accordingly, foreground location model is defined as

$$\Phi_n(l_n = 1) = \max\left\{s_n, l_n^{t-1}\right\} \tag{2.13}$$

where, s_n is the saliency value of pixel n of the saliency map $\mathbf{S}$, and l_n^{t-1} denotes the segmentation label of pixel n produced by the previous segmentation in the iterative optimization, which is set to 0 in the initialization (see Section 2.4.3). Similarly, the background location model is defined as

$$\Phi_n(l_n = 0) = 1 - \Phi_n(l_n = 1). \tag{2.14}$$

The appearance model Ω computes the probability of pixel n to be labeled as foreground/background based on color distributions in the image. We adopt two Gaussian Mixture Models (GMMs) to formulate the foreground/background appearance. The GMM is a parametric probability density function represented as a weighted sum of Gaussian densities

$$\Omega(\mathbf{x}_n|\vartheta) = \sum_{i=1}^{Q} w_i g(\mathbf{x}_n|\mu_i, \Sigma_i) \tag{2.15}$$

where $\mathbf{x}_n \in \mathbb{R}^3$ is a color pixel value vector, Q is the number of Gaussian components, $\vartheta = \{w_i, \mu_i, \Sigma_i\}, i = (1, \cdots, Q)$ is a set of GMM parameters, $g(\mathbf{x}_n|\mu_i, \Sigma_i)$ is a Gaussian probability density function

$$g(\mathbf{x}_n|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^3|\Sigma_i|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \mu_i)'\Sigma_i^{-1}(\mathbf{x}_n - \mu_i)\right\} \tag{2.16}$$

here $\mu_i \in \mathbb{R}^3$ is the mean vector of data vectors in the same Gaussian component, $\Sigma_i \in \mathbb{R}^{3\times3}$ is the covariance matrix. w_i is the weight of Gaussian component, such that the sum of all components weights is unity. The GMM parameters for foreground/background apparence modeling are learned from the initially or

prevously separated foreground/background pixels.

Smoothness term While foreground objects might be partly similar to background regions, the data term along is not sufficient to separate objects from background. The smoothness term promotes the pixels to be labeled smoothly, and thus coupling with the data term enables to group pixels into real objects. Following previous works [39, 41] the smoothness term is defined as

$$\Theta_{n,j}(l_n, l_j) = \begin{cases} 0 & \text{if } l_n = l_j \\ \Psi(n,j) & \text{otherwise} \end{cases} \tag{2.17}$$

where $\Psi(n,j)$ is a function defined based on color contrast

$$\Psi(n,j) = \frac{\varphi}{\text{dis}(n,j)} \exp\Big\{ -\beta \|\mathbf{x}_n - \mathbf{x}_j\|^2 \Big\} \tag{2.18}$$

here $\text{dis}(\cdot)$ indicates spatial Euclidean distance between neighboring pixels, the constant parameter φ is moderately set to 50, β is color contrast adaptive parameter defined as

$$\beta = \frac{1}{2 \cdot \text{mean}\Big\{ \|\mathbf{x}_n - \mathbf{x}_j\|^2 \cdot \text{dis}(n,j) \Big\}}. \tag{2.19}$$

2.4.2 Saliency boosting model

In this subsection, we present how to improve the quality of saliency map by taking object segmentation result into consideration. We assume that objects are at least partly extracted by the segmentation model described in the previous subsection. Pixels spatially near to the labeled foreground regions, and pixels similar to the labeled foreground regions in appearance should be assigned with a higher saliency value, and vice versa. Based on this assumption, the saliency boosting model is defined as

$$\mathbf{S}^* = \mathbf{S} \odot (\mathbf{M} + \mathbf{C}) \tag{2.20}$$

where, $\odot$ indicates element-wise multiplication, $\mathbf{M}$ is the spatial prior matrix and $\mathbf{C}$ is the appearance prior matrix.

Recall that the saliency map $\mathbf{S}$, generated by the segmentation driven low-rank matrix recovery (SLR) model described in Section 2.3, is finally computed based on medium-grained (MG) segmentation. We also compute the spatial prior and appearance prior for each region of MG segmentation, and assemble the spatial/appearance priors of all regions to form spatial/appearance prior of the entire image.

Let $\mathbf{R} = \{r_1, r_2, \cdots, r_K\}$ denote the MG segmentation of image $\mathbf{X}$, and $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_P\}$ denote a set of the separated foreground objects and $\mathcal{B}$ denote the background in the segmentation result $\mathbf{L}$ generated by using the method described in Section 2.4.1, where K is the number of MG regions, and P is the number of the segmented objects. We want to compute a set of spatial priors $\mathbf{M} = \{m_1, m_2, \cdots, m_K\}$ and a set of appearance priors $\mathbf{A} = \{a_1, a_2, \cdots, a_K\}$.

Spatial prior

The spatial prior of region r_k is defined as

$$m_k = \frac{1}{P} \sum_{p}^{P} \exp\Big\{ - \alpha \cdot \rho \cdot \eta \cdot \mathcal{D}(r_k, \mathcal{O}_p) \Big\} \tag{2.21}$$

where

α is a constant balance parameter, and set to 10 in our experiments,

$\rho = \frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}}(s_n)$ is the average of saliency values of background regions, where $|\cdot|$ indicates the number of elements,

$\eta = \frac{|\mathcal{B}| + \sum_{p=1}^{P} |\mathcal{O}_p|}{\sum_{p=1}^{P} |\mathcal{O}_p|}$ is the ratio of the image size to total area of all foreground objects,

$\mathcal{D}(\cdot)$ is a spatial distance function.

The spatial distance function $\mathcal{D}(r_k, \mathcal{O}_p)$ computes the normalized Euclidean squared distances between boundary pixels in region r_k and the centroid of object $\mathcal{O}_p$

$$\mathcal{D}(r_k, \mathcal{O}_p) = \frac{d'_{kp}}{\max\{d'_{1p}, d'_{2p}, \cdots, d'_{Kp}\}} \tag{2.22}$$

where

$$d'_{kp} = \frac{1}{|\mathcal{E}_k|} \sum_{j \in \mathcal{E}_k} \|\mathbf{z}_j - \mathbf{c}_p\|_2^2 \tag{2.23}$$

where $\mathcal{E}_k$ is a set of boundary pixels in region r_k, $\mathbf{z}_j$ is the coordinate of boundary pixel j , and $\mathbf{c}_p$ is the centroid of object $\mathcal{O}_p$.

From Eq. (2.21), we can observe that the spatial prior is adaptive to the quality of saliency map and object size. On one hand, the quality of saliency map is measured by ρ. The smaller ρ means higher quality of saliency map as most background pixels are assigned with small saliency values, which leads to larger value of the spatial prior. On the other hand, object size information is represented by η. A large η means small objects in the image, and spatial distance function $\mathcal{D}(\cdot)$ is multiplied by a large weight. Thus, spatial prior is more sensitive to the distance between region to object centroid.

Appearance prior

The appearance prior computes the similarity between regions and the segmented objects. For appearance representation, we use the CIE L*a*b* and hue color histograms, where channels L*, a*, b* and hue are quantized into 8, 16, 16 and 4 bins, respectively. Thus each region/object is represented by a $(8 \times 16 \times 16 \times 4)$-dimensional histogram which is normalized to unity. Let $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_P\}$ represent the color histograms of the segmented objects $\{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_P\}$, and $\mathbf{h}_k^t$ denote the color histogram of region r_k, the non-normalized appearance prior of r_k is defined as

$$a'_k = \sum_{p=1}^{P} |\mathcal{O}_p| \cdot \mathcal{K}(\mathbf{h}_k^t, \mathbf{h}_p) \tag{2.24}$$

where $\mathcal{K}(\cdot)$ is a similarity kernel function. In our experiment, the intersection kernel is adopted, thus

$$\mathcal{K}(\mathbf{h}_k^t, \mathbf{h}_p) = \sum_{i=1}^{T} \min \left\{ \mathbf{h}_k^t(i), \mathbf{h}_p(i) \right\} \tag{2.25}$$

where T is the dimensionality of the histogram. By using Eq. (2.24), the non-normalized appearance priors of all regions are computed. Then the final appearance

prior of region r_k is computed by a normalization function, i.e.,

$$a_k = \frac{a'_k}{\max\{a'_1, a'_2, \cdots, a'_K\}}. \tag{2.26}$$

Notice that, in Eq. (2.24), the similarity kernel $\mathcal{K}(\cdot)$ is weighted by object size $|\mathcal{O}_p|$. This implies that larger objects give more contribution to the appearance prior when there are multiple objects. In addition, this also significantly decreases the impact of segmentation noises, in which very few pixels form a region and are labeled as foreground. Therefore, taking the object size into consideration improves the robustness of appearance prior.

2.4.3 Iterative and joint optimization

The proposed scheme for joint object segmentation and saliency optimization works in an iterative manner, and is summarized in Algorithm 1. To launch the overall process of joint object segmentation and saliency optimization, the initial saliency map $\mathbf{S}$ is thresholded to obtain the initial label map $\mathbf{L}$, which can roughly separate foreground pixels from background. The threshold is set to a value that ensures those pixels occupying 75% saliency of the whole image to be labeled as foreground. During the iterative optimization process, the saliency map $\mathbf{S}$ and the label map $\mathbf{L}$ are mutually refined with the update of GMM parameters, data term for graph cuts, spatial prior and appearance prior. If the label map $\mathbf{L}$ does not change any more, the iterative optimization reaches the convergence and output the optimal saliency map and final segmentation result. For reducing time consumption, we set the maximal iterations to 4.

The advantages of the proposed scheme are twofold. On one hand, the iterative MRF energy minimization allows to make use of previous segmentation results to learn the refined parameters for the next round of segmentation. On the other hand, interaction between object segmentation and saliency optimization enables to derive more reliable object cues, and promote both to achieve the optimality.

Algorithm 1 Joint object segmentation and saliency boosting

– **Input**: image $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}, \mathbf{x}_n \in \mathbb{R}^3$ and its initial saliency map $\mathbf{S} = \{s_1, \cdots, s_N\}, s_n \in \mathbb{R}^1$.
– **Output**: a labeled map $\mathbf{L} = \{l_1, \cdots, l_N\}, l_n \in \mathbb{R}^1$, and an optimized saliency map $\mathbf{S}^*$.

Initialization

1. Initialize $\mathbf{L}$ by thresholding $\mathbf{S}$.
2. Compute the pairwise smoothness term Θ with (2.18).

Iterative Optimization

1. Compute the location probability Φ based on $\mathbf{S}$.
2. Learn a set of GMM parameters ϑ based on $\mathbf{L}$.
3. Compute the data term Λ with (2.12).
4. Segment image $\mathbf{X}$ by minimizing (2.11) and update $\mathbf{L}$.
5. If the sum of object centroid shifts is more than ξ pixels (typically, $\xi = 5$)
   (a) By using $\mathbf{L}$, compute spatial prior and appearance prior with (2.21) and (2.24), respectively.
   (b) Compute a new saliency map $\mathbf{S}^*$ with (2.20) and assign it to $\mathbf{S}$.
6. Stop the iteration if the convergence is reached or the number of iteration is greater than the predefined threshold, which is set to 4 in our experiments.

2.5 Experimental evaluation

The proposed joint saliency detection and object segmentation approach is evaluated on two datasets including the popular MSRA-B [21] and the newly introduced but more challenging PASCAL-1500 [7].

MSRA-B dataset [1] includes 5000 images, most of which contain a single salient object typically appearing in the center of the image. The original MSRA-B dataset annotates salient objects with bounding boxes and suffers from the limited accuracy in the performance evaluation of saliency detection and object segmentation. Thus, we use the pixel-wise segmented ground truths [2] provided by [22] for an accurate evaluation.

1. http://research.microsoft.com/en-us/um/people/jiansun
2. http://www.jianghz.com/projects/saliency_drfi/index.html

While MSRA-B may have the limited variations of salient objects, we also validate the performance of saliency detection and object segmentation on a more challenging PASCAL-1500 dataset [3]. This dataset contains 1500 real-world images from PASCAL VOC 2012 segmentation dataset [45], in which each image is accurately annotated at pixel-level for performance evaluation. In PASCAL-1500 dataset, many images show highly cluttered background, and about 40% of the images contain multiple objects (on average 3 objects) which appear at a variety of locations and scales.

In the rest of subsections, we first give the relevant implementation details of the proposed approach, and then discuss the results of saliency detection and object segmentation, respectively.

2.5.1 Implementations

For image description in saliency detection, we use three visual features including color, responses of steerable pyramids filters [46] and responses of Gabor filters [47].

Color: R, G, B color values, saturation and hue components are computed for each pixel, thus this forms a 5-dimensional color feature vector. To make it more discriminative, we also perform a mean normalization, i.e., each color feature vector is subtracted by the mean of all color feature vectors in the image.

Steerable pyramids filters: the input color image is first transferred to gray-scale image and decomposed into 3 pyramid scales, and then derivative operations are applied at 4 orientations to each pyramid scale. This results in a 12-dimensional feature vector.

Gabor filters: Gabor filters are performed on the gray-scale image with 3 wavelet scales and 12 filter orientations, which yields a 36-dimensional feature vector. The wavelength of smallest scale filter is 6, and the scaling factor between successive filters is 2.

These three visual features are accumulated (by average pooling) within the superpixel and stacked together to form a (5 + 12 + 36)-dimensional feature vector to represent the superpixel.

3. http://wzou.perso.insa-rennes.fr

For region generation, hierarchical segmentation of gPb [48] (globalized probability of boundary) is employed to generate medium-grained (MG) and coarse-grained (CG) segmentations. The output of gPb is a real-valued ultrametric contour map (UCM). The MG and CG segmentations are generated by thresholding the UCM, which is normalized from 0 to 1, at 0.125 and 0.25 respectively. As gPb generally preserves global contours of objects, and it fits well to MG and CG segmentations. However, it can not apply to the fine-grained (FG) segmentation very well, as it tends to group uniform areas into a large region. This makes feature descriptors extracted from superpixels of background to be insufficiently redundant, and thus they lack the low-rank property which is essential for low-rank matrix recovery model. Therefore, the segment-size controllable Mean-shift [4] is used to obtain FG segmentation, where the minimum segment area is set to 200 pixels.

The low-rank matrix recovery model is solved by the augmented Lagrange multiplier method proposed in [49], and the balance parameter λ of the model is set to 0.05. The MRF energy minimization for object segmentation is solved via standard graph cuts [39].

2.5.2 Performance evaluation of saliency detection

In this subsection, we first present evaluation metrics for saliency detection, and then we analyze different components of the segmentation driven low-rank matrix recovery (SLR) model, described in Section 2.3, and show how the SLR-based saliency boosting (SB) model, described in Section 2.4.2, helps to achieve higher-quality saliency map. After that, the proposed SLR and SB models are compared with the state-of-the-art saliency detection models.

Evaluation Metrics

To objectively evaluate the performance of saliency detection, we adopt the widely used metrics: receiver operator characteristic (ROC) curve to measure the similarity between the saliency map and the ground-truth, and the area under the

curve (AUC) for quantitative comparison between different models. To obtain the ROC curve, saliency maps are normalized from 0 to 255 and thresholded using integer values within [0, 255]. Then for each thresholding, the average true positive rate and the average false positive rate over all test images are computed. Finally, the ROC curve is generated by plotting the true positive rate values on the y-axis against false positive rate values on the x-axis.

Performance analysis

We analyze, in both objectively and subjectively, the contributions of different components in SLR model and validate the further improvement achieved by the saliency boosting (SB) model.

In Figure 2.4, the dashed ROC curves for saliency maps show the saliency performance of SLR model using different components, and the solid ROC curve shows the saliency performance of SB model. As demonstrated in Figure 2.4, the dashed ROC curves are gradually elevated while integrating more components to the SLR model, thus higher AUC scores are obtained. If raw feature extracted from image is directly used as the input of low-rank matrix recovery (LRMR) model, the saliency detection performance is low. With the CG segmentation prior, the AUC scores increase significantly: from 83.9% to 93.3% on MSRA-B, and from 74.0% to 88.0% on PASCAL-1500. By further integrating the post-smoothing component (i.e., full SLR model), the AUC score on MSRA-B increases with 1.4%, while on PASCAL-1500 it increases with 3.1%, compared to when using CG segmentation prior. From Figure 2.4, we can also observe that solid ROC curve of SB model is higher than the ROC curve of full SLR model, and AUC scores on MSRA-B dataset and PASCAL-1500 dataset increase to 95.2% and 91.9%, respectively. This demonstrates that the SB model improves the performance of saliency detection from SLR model.

Figure 2.5 shows some examples of saliency maps generated by the SLR model using different configurations and by the SB model. Some observations can be derived from these examples. First, the saliency maps in the 2nd column of Figure 2.5, generated by using raw feature as the input of LRMR model, are

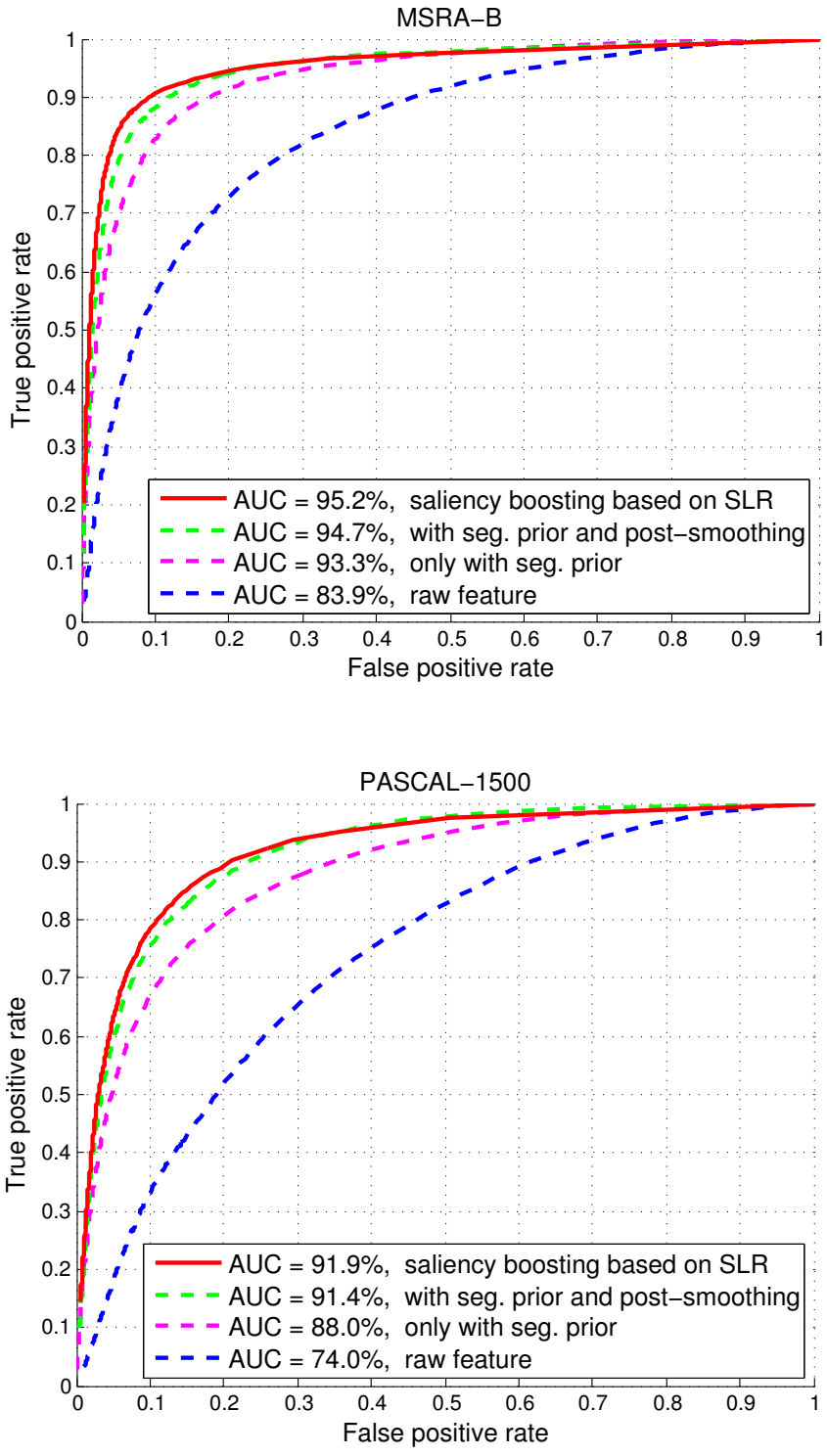


Figure 2.4: ROC curves and AUC scores OF the proposed model with different configurations on MSRA-B (top) and PASCAL-1500 (bottom) datasets. the dashed curves show the performance of SLR model using different components. The solid curve shows the performance of SB model.

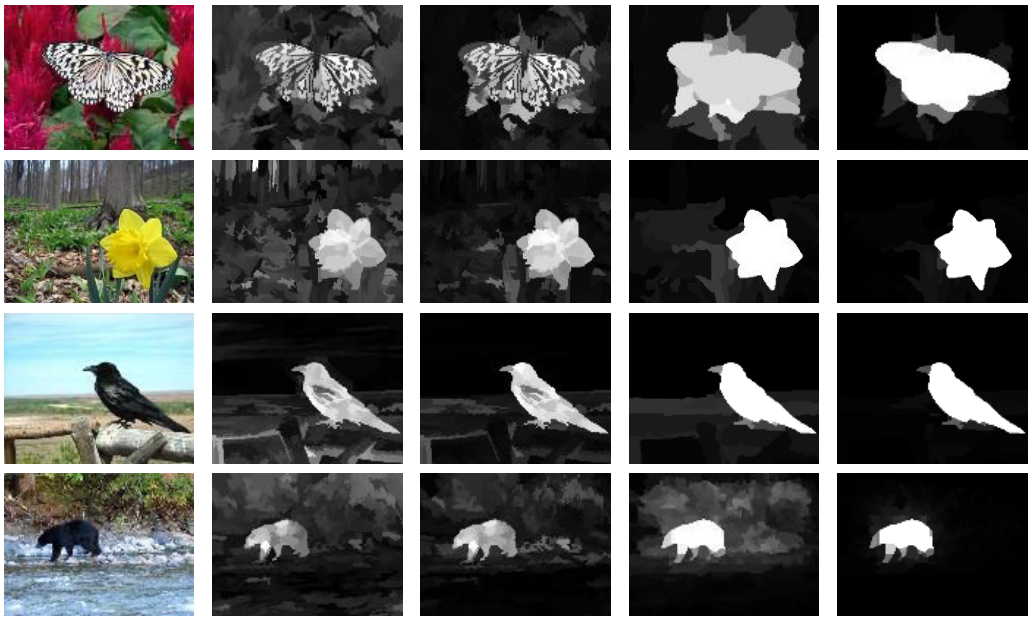


Figure 2.5: Some examples of saliency maps generated by the proposed saliency model with different configurations. (a) input image; (b) results of LRMR model using raw feature only; (c) results of LRMR model with segmentation prior; (d)results of full SLR model: LRMR model with segmentation prior and post-smoothing component; (e) results of SLR-based saliency boosting (SB) model.

typically very sparse, thus it is far from to highlight the total salient objects. Second, the LRMR model with segmentation prior suppresses some background regions, especially those regions connecting with image border (see the 3rd column). Third, adding post-smoothing component to the LRMR model with segmentation prior ensures the overall saliency map smoothing, and the salient objects are more completely highlighted (see the 4th column). Last but not the least, the saliency boosting model is able to effectively suppress difficult background regions, and ensures to discovery salient objects from complex scenes (see the last column).

Comparison with the state-of-the-art saliency models

For performance comparison, we first consider the top five saliency models ranked in the benchmarking report [34], i.e.,

- fusing generic objectness (GO) model [33] which integrates object detection measure of Objectness [50] in into a graphical model for saliency evaluation,
- context-aware (CA) model [14] which combines local contrast, global uniqueness of color feature and some visual organization rules for saliency measuring.
- context and shape prior based (CBS) model [15] which generates object-level saliency maps by modeling regional context and object shape prior from multi-scale segmentations,
- region contrast based model (RC) [13] which computes saliency from global contrast and spatial weighted regional contrast,
- kernel density estimation based (KDE) model [23] which associates color saliency and spatial saliency with a set of KDE models constructed from over-segmented regions.

Moreover, we also compare our SLR and SB models with three recent saliency models, which are not evaluated in the benchmarking report [34], including

- Bayesian saliency (BS) model [26] which evaluates saliency from convex hull analysis on interest points and Laplacian subspace clustering on superpixels,
- hierarchical saliency (HS) model [31] which exploits hierarchical inference to fuse three saliency maps computed from multiple scales of region segmentation,
- training based low-rank matrix recovery (TLR) model [36] which integrates transformation prior learnt from MSRA-B dataset, semantic prior (face detection), color prior and center prior to LRMR model for saliency detection.

Therefore, there are eight reference saliency models in total. The most relative to our saliency models is TLR model which required supervised learning. In contrast, our models only use a single bottom-up segmentation prior, without using any supervisory information.

The proposed SLR and SB models are compared with the eight state-of-the-art models on MSRA-B dataset and PASCAL-1500 dataset in Figure 2.6 and Figure 2.7, respectively. The ROC curves and AUC socres of these baselines are computed using authors' publicly available codes or their results. Clearly, the proposed SLR model

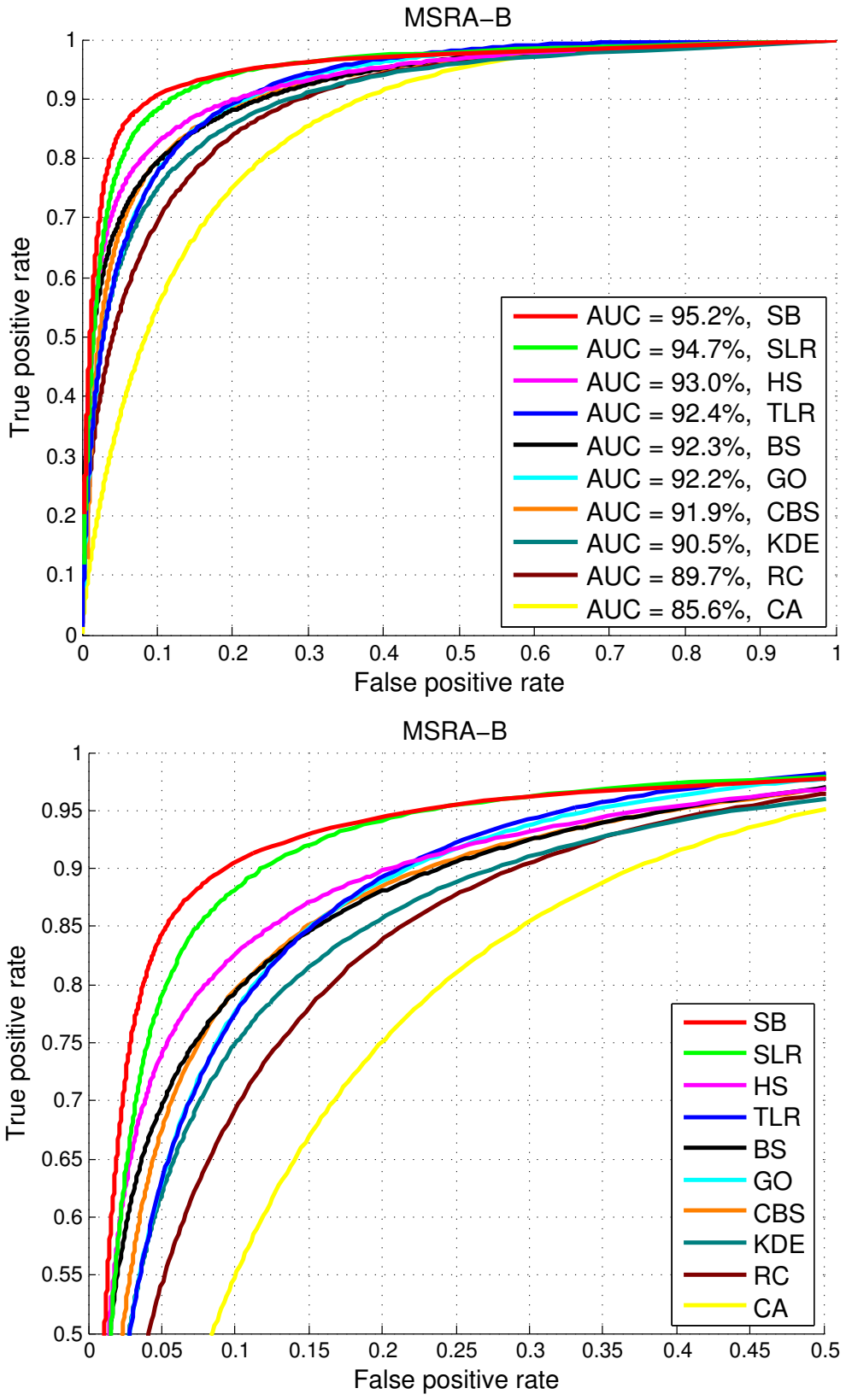


Figure 2.6: ROC curves and AUC scores of different models on MSRA-B dataset. Top: complete ROC curves; Bottom: the zoomed top left corner of ROC curves. The models are ranked based on AUC scores in the legend.

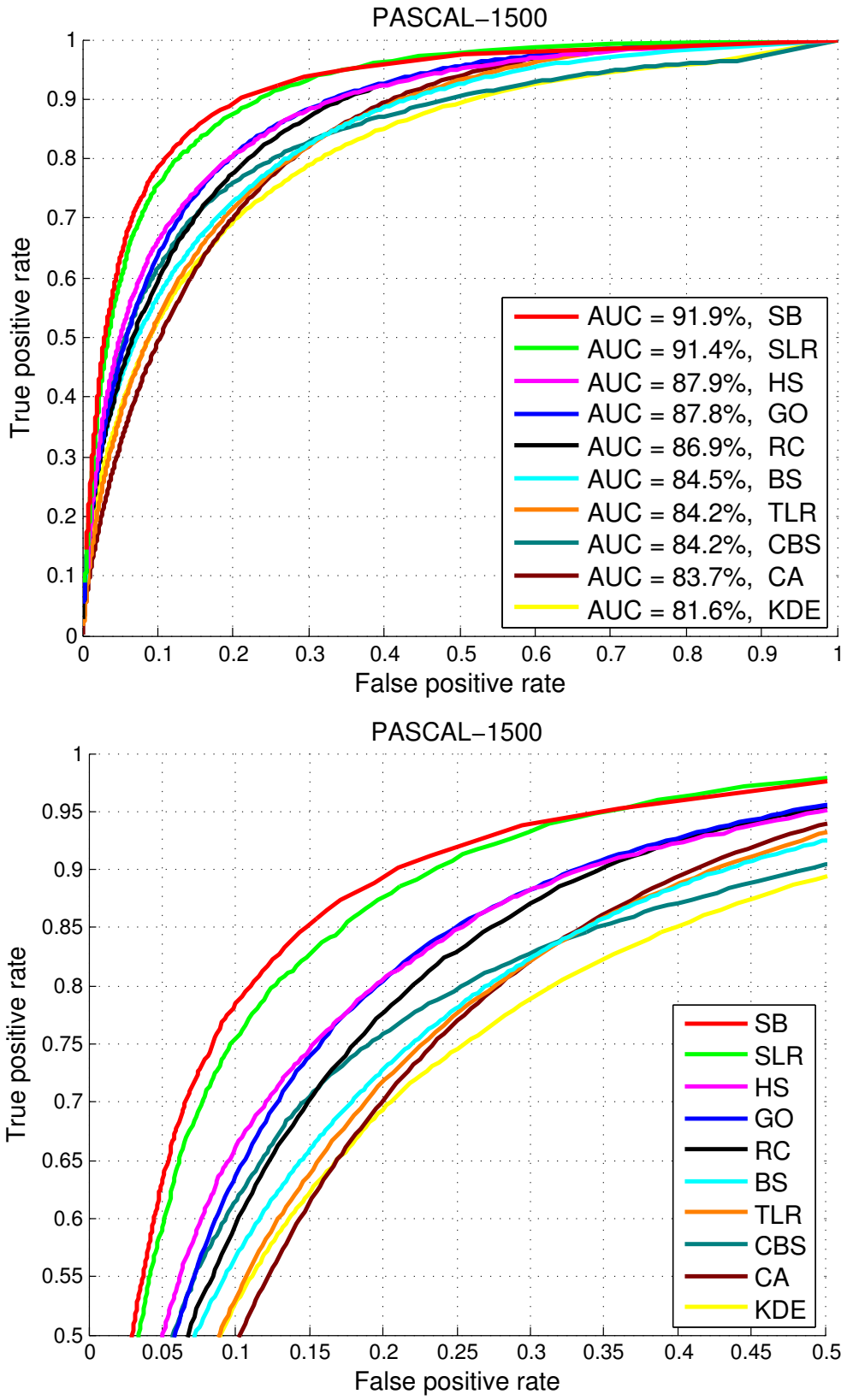


Figure 2.7: ROC curves and AUC scores of different models on PASCAL-1500 dataset. Top: complete ROC curves; Bottom: the zoomed top left corner of ROC curves. The models are ranked based on AUC scores in the legend.

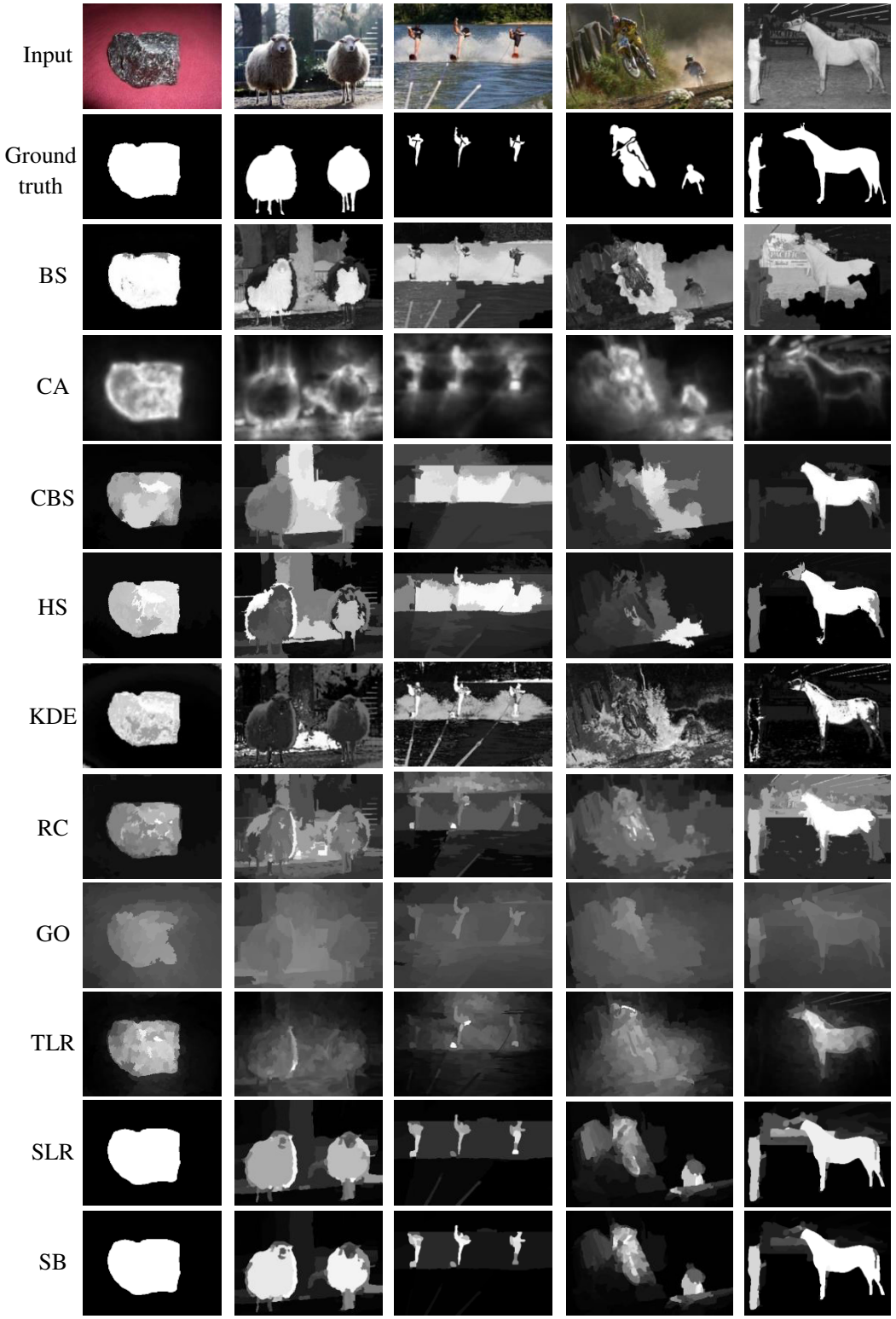


Figure 2.8: Examples of saliency maps generated using the eight state-of-the-art models and the proposed SLR and SB models (in the last two rows).

already outperforms all reference models with substantial margin. It achieves 94.7% and 91.4% AUC scores on MSRA-B and PASCAL-1500 datasets, and obtains 1.7% and 3.5% improvements respectively, compared to HS model which gets the best performance in the reference models. Moreover, our SB model improves the SLR model further, and reaches 95.2% and 91.9% AUC scores on MSRA-B and PASCAL-1500 datasets, respectively.

Figure 2.8 shows some examples of saliency maps generated using the eight state-of-the-art models and the proposed SLR and SB models. At least three observations can be derived from these examples. To begin with, most models obtain pretty good results when the input image is with high contrast between object and background appearing near uniformly, like the image in the first column. Moreover, the state-of-the-art models are weak to detect multiple salient objects in the image, such as the 2nd-4th columns. However, the proposed SLR and SB models show their potential to identify all of them. Last but not the least, the proposed models are able to detect objects in the cluttered scenes, while the reference models suffer from limited robustness. For example, in the 2nd-4th columns, the sheep, the persons and the horse are within cluttered backgrounds. The reference models either fail or only partly highlight these objects, but our models show the abilities to discover them.

2.5.3 Performance evaluation of object segmentation

In this subsection, we first present the evaluation metrics for object segmentation, then, we compare our approach with the state-of-the-art methods.

Evaluation metrics

To objectively evaluate the performance of object segmentation, we adopt the widely used measures of average precision, recall and F-score for the entire dataset. The average precision (AvP) and average recall (AvR) are computed as

$$\text{AvP} = \frac{1}{T}\sum_{t=1}^{T}\frac{S_t \cap G_t}{S_t} \tag{2.27}$$

$$\text{AvR} = \frac{1}{T}\sum_{t=1}^{T}\frac{S_t \cap G_t}{G_t} \tag{2.28}$$

where, T is the number of images in the dataset, S_t is the segmented salient objects of image t and G_t is the ground-truth of image t. The average F-score (AvF) is defined as

$$\text{AvF} = \frac{1}{T}\sum_{t=1}^{T}\frac{(1+\beta)P_t R_t}{\beta P_t + R_t} \tag{2.29}$$

where P_t and R_t are precision and recall of image t respectively, the coefficient β balances the importance between precision and recall. As in previous works [13, 51], β is set to 0.3 in our experiments.

Comparison with the sate-of-the-art segmentation approaches

We compare the proposed segmentation method with three state-of-the-art approaches for salient object segmentation, i.e., (i) KDEseg [23] in which saliency detection model is based on kernel density estimation and object segmentation is achieved using two phases graph cuts with adaptive seed adjustment; (ii) CBSseg [15] in which saliency detection is based on regional context as well as object shape prior, and histogram-based iterative graph cuts is employed for object segmentation; and (iii) RCseg [13] in which saliency is computed from bothe global and local region contrast and standard GrabCut [41] is used for object segmentation. The results of these methods for comparison are provided by authors or produced by their publicly available implementations with the best parameters.

Figure 2.9 compares the AvP, AvR and AvF of different saliency-based object segmentation methods on MSRA-B and PASCAL-1500 datasets. First of all, let us take a look at the common ground in all approaches. The AvP in these approaches is typically higher than AvR, perhaps because precision is more important in many applications, such as attention detection. Then we see the performance difference in these methods. From Figure 2.9, we can observe that the proposed method consistently outperforms other methods in both of the evaluation datasets. Compared to the best performance in the reference methods, our approach obtains

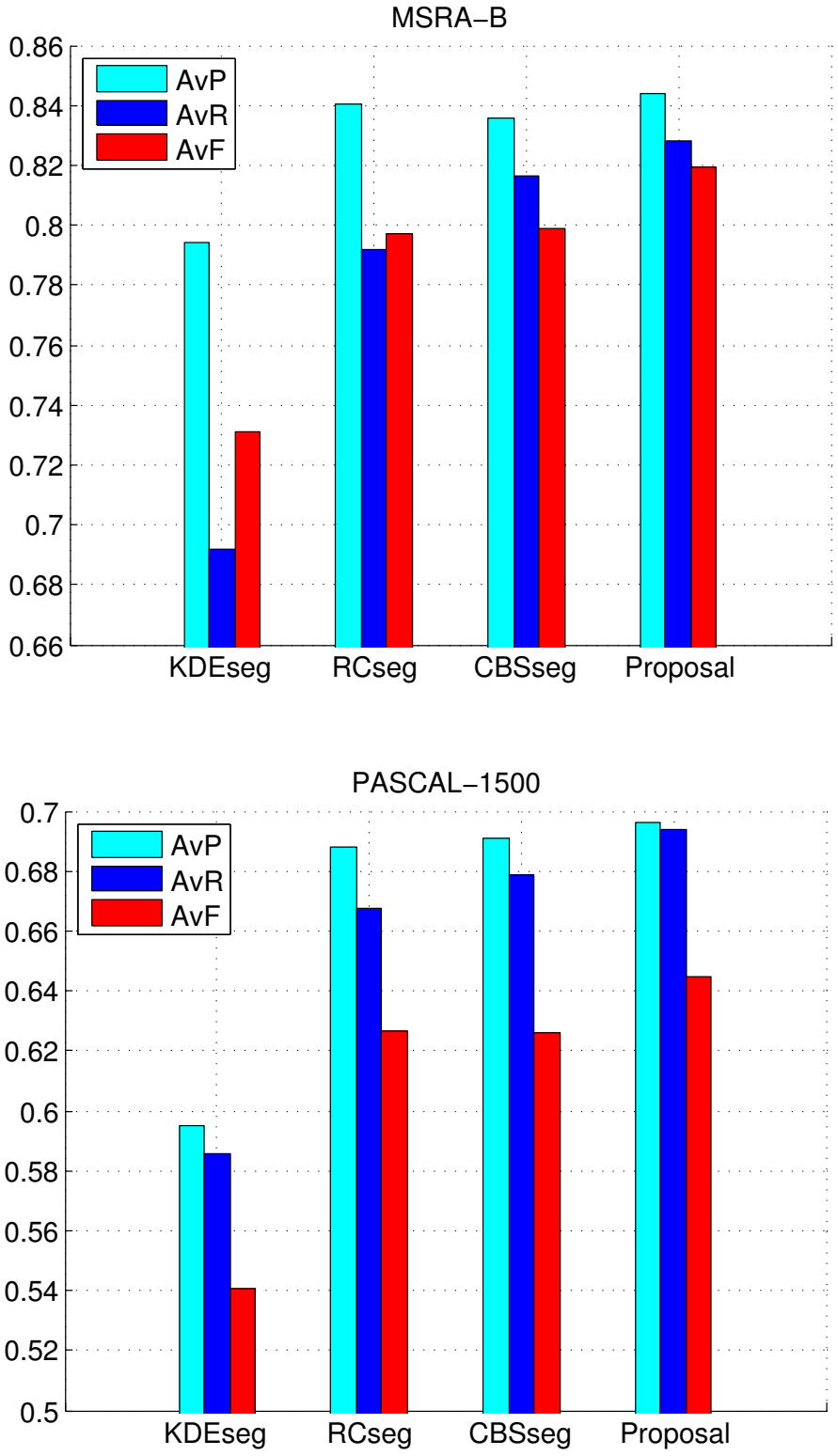


Figure 2.9: Average precision, recall and F-score for different saliency-based segmentation methods on MSRA-B (top) and PASCAL-1500 (bottom) datasets.

Figure 2.10: Examples of segmentation results generated using three state-of-the-art methods and the proposed approach.

2.06% and 1.86% improvement of AvF on MSRA-B and PASCAL-1500 datasets, respectively.

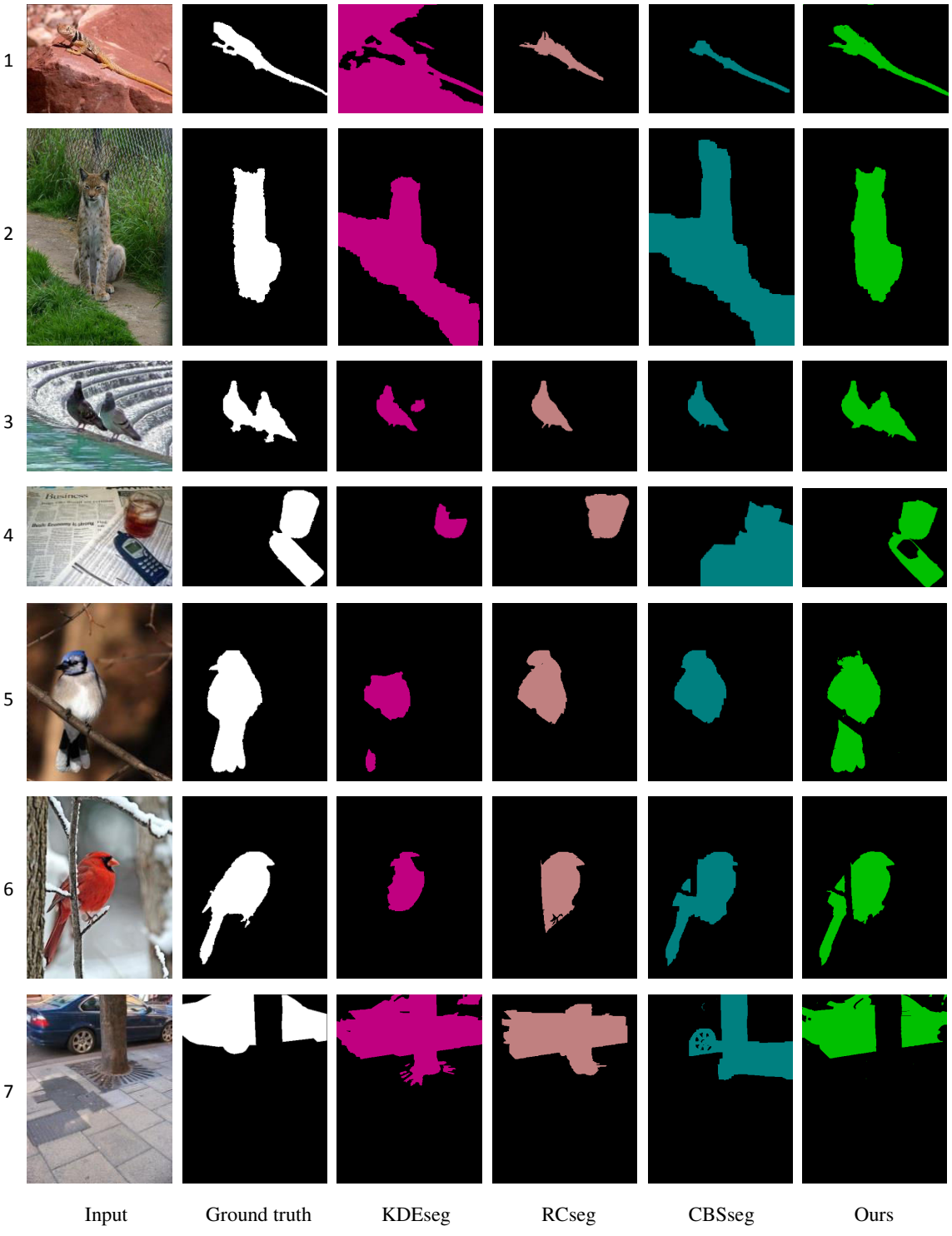


Figure 2.11: Examples of segmentation results generated using three state-of-the-art methods and the proposed approach.

Figures 2.10 and 2.11 show some examples of segmentation results generated using the three state-of-the-art methods and the proposed approach. We make some observations from these examples. First, for images containing a single salient object showing obvious contrast with relatively simple background, like the 1st-3rd rows in Figure 2.10, all approaches successfully segmented the objects with well preserving their contours. Second, for images with low contrast between salient objects and background regions, such as the 4th-7th rows in Figure 2.10 and 1st-2nd rows in Figure 2.11, the three reference methods either merge objects into background or totally/partly fail to extract the objects. In contrast, our approach separates the objects from most irrelevant background regions. Third, for images containing multiple salient objects, e.g., the 3rd 4th rows in Figure 2.11, the three reference methods tend to segment only a single object which is considered as the most salient in their saliency models. However, our approach can extract all salient objects in the image. Finally, for images with object occlusion, e.g. the last three rows Figure 2.11, the three reference methods only segment part of object regions or merge background regions into objects, while the proposed method shows its ability to extract well the occluded objects. Therefore, the proposed approach is much more robust and outperforms the state-of-the-art methods.

2.6 Conclusion

In this chapter, we have detailed a novel approach for jointly addressing the problem of saliency detection and object segmentation.

As the first contribution, a segmentation driven low-rank matrix recovery (SLR) model is proposed to detect salient object in an image. The key idea of this model is to decompose an image feature matrix into a low-rank matrix and a sparse one, where the decomposed low-rank matrix naturally corresponds to the background, and the sparse one captures salient objects. In order to improve the robustness of low-rank matrix recovery model for saliency detection, a bottom-up prior called segmentation prior, which is defined base on region's connectivity with image border, is proposed as an important constraint cue for the matrix

decomposition and is shown to significantly improve the saliency detection performance. In addition, a simple yet effective post-smoothing method is presented to ensure the overall saliency smoothing and to generate visually higher-quality saliency map. Moreover, a challenging dataset named as PASCAL-1500, which contains 1500 images with pixel-wise ground truth, is introduced to evaluate the performance of saliency detection.

Second, a unified scheme is proposed to jointly segment foreground objects from background and to boost saliency map generated by SLR model. On one hand, the segmentation model is based on the standard Markov random field (MRF) framework which consists of a data term and a smoothness term. We have proposed a robust data term via the optimum use of saliency information. On the other hand, the saliency boosting (SB) model improves the quality of saliency map by effectively leveraging object location and appearance information from the segmentation result. Mutually performing object segmentation and saliency optimization promotes to obtain a better segmentation result and a higher-quality saliency map.

To validate the performance of saliency detection and object segmentation, extensive evaluation has been carried out on two datasets, including MSRA-B containing 5000 images and the newly introduced PASCAL-1500 (6500 images in total for the two datasets). Experiments demonstrate that: i) SLR model already outperforms the state-of-the-art saliency models, ii) SB model further improves the saliency detection performance, iii) the proposed segmentation approach is superior to the state-of-the-art object segmentation methods as well.

3 Exemplar-based object segmentation

3.1 Introduction

In the previous chapter we have discussed the salient object segmentation. However, objects are not always salient in real images and saliency-based segmentation approaches may suffer limited robustness to segment un-salient objects. In this chapter, we concentrate on the objective of extracting all foreground objects from the background, which is usually called as figure-ground segmentation. The figure-ground segmentation is essential for many applications, e.g., image editing [41], object recognition [52], image retrieval [53], target tracking [54, 55], adaptive compression [56], etc.

According to the number of object classes within an image, existing figure-ground segmentation algorithms can be broadly classified into two categories: class-specific and class-independent. The class-specific segmentation [57–62] requires the input images to contain only a single class of objects. One of the main solutions for class-specific segmentation uses the learned top-down priors of specific category (e.g., shape templates and object parts configuration) to guide bottom-up segmentation. Even though impressive results are demonstrated, the class-specific segmentation lacks adaptivity which limits the range of its applications. Recently, growing attention has been paid to class-independent segmentation due to the rising demands of applications like large-scale object annotation [63]. The class-independent segmentation is a generic approach which aims at segmenting out any class of objects from background. This

is a more difficult case: challenges mainly come from intra and inter variations of objects, object occlusion and truncation.

There are mainly two strategies to address the class-independent segmentation. The first one is based on multiple segmentations or hierarchical segmentation [64–66]. Typically, a large set of regions is generated by varying segmentation parameters, and then offline learned ranking model is used to select regions likely to cover objects. This strategy exclusively depends on the bottom-up segmentation which usually lacks robustness, and thus it is more applicable to consistent scenes. The second strategy is exemplar-based segmentation transfer, such as the recently proposed approaches [67, 68]. In [67], exemplars are picked from the annotated training images which are geometrically similar to the query image. Then the object locations are predicted, by merging segmentation masks of exemplars, to serve as seeds for graph cuts [39] to create spatially consistent segmentation. Unfortunately, retrieving exemplars by scene layout matching suffers from the limited robustness of global image descriptor (e.g., GIST [69]) to handle geometric deformations [70]. More importantly, their location model is sensitive to object variations, e.g., rotation, scale and position. Instead, in [68], exemplars are gathered from windows predicted by object detection, and then segmentation masks of exemplar windows are transferred into the query image. The window-based segmentation transfer is instinctively determined by the performance of generic object detection algorithm. However, it is not easy to obtain a reliable class-independent detection model, since the state-of-the-art object detection approaches still have difficulties to handle cluttered background.

Based on the aforementioned issues, this chapter proposes a novel exemplar-based segmentation approach named as *Online glocal transfer*. As illustrated in Figure 3.1, it aims to retrieve better exemplars for segmentation transfer: the exemplars not only have global similar scene structure but also contain local objects with appearance similar to those in the query image. Our hypothesis is that *glocally* (globally and locally) similar images generally have similar segmentations. The contributions of this chapter are two-fold.

– Motivated by our ultimate application of object segmentation, a new high-level

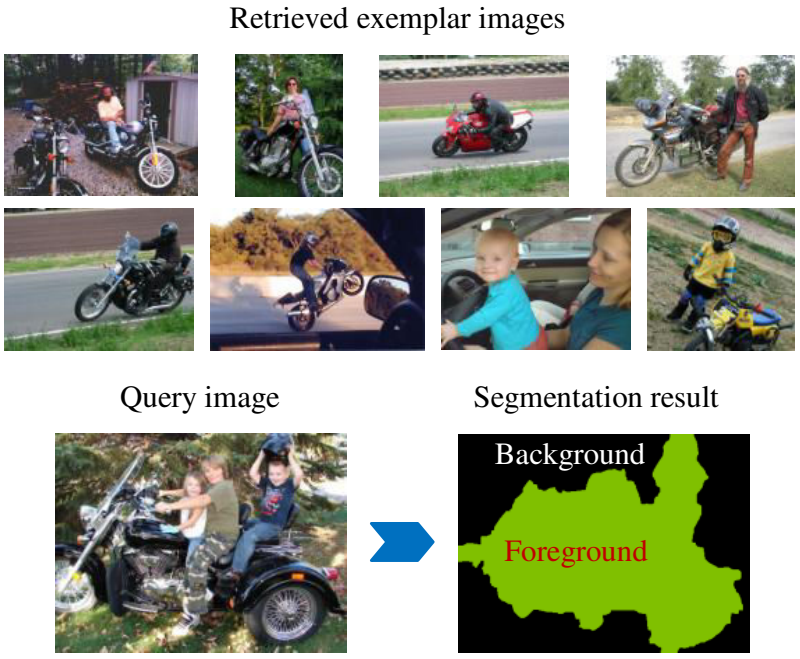


Figure 3.1: Given a query image (bottom left), we first retrieve glocally similar exemplars in the annotated dataset (top) and then transfer their segmentation masks to the query image (bottom right).

image representation method, object-oriented descriptor (OOD), is proposed to implicitly represent geometric information and to highlight objects in the image. Therefore OOD enables to effectively find glocally similar images.

– A novel scheme is proposed to obtain the optimal segmentation that harmoniously combines online prediction and Markov random field (MRF) energy optimization. A discriminative classifier is learned on-the-fly from the retrieved k nearest neighbors. The classifier initially predicts foreground probability of the query image which serves as high-level prior for further pixel-wise segmentation. While the online learning has been shown successful in exemplar-based image classification [71], to the best of our knowledge, it has not been applied to the figure-ground segmentation yet.

The proposed approach has been extensively evaluated on three challenging datasets including Pascal VOC 2010, VOC 2011 [72], and iCoseg [73]. Experiments demonstrate that the proposed approach outperforms the state-of-the-art methods and has the potential to segment large-scale images containing unknown objects,

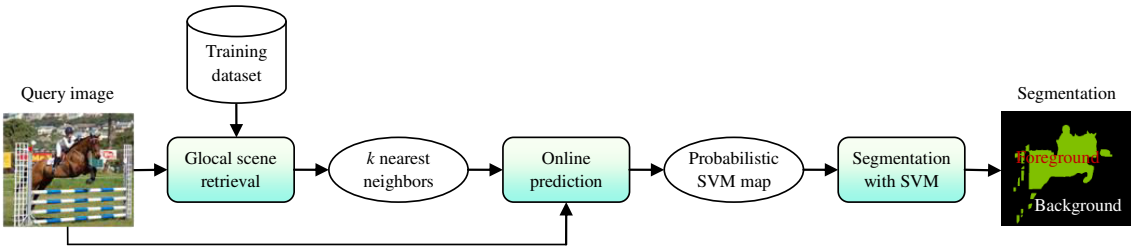


Figure 3.2: Generic framework of online glocal transfer which consists of three core algorithmic modules: glocal scene retrieval, online prediction and segmentation with SVM prior.

which never appear in the exemplar images.

3.2 Overview

The objective of this chapter is to automatically segment out all objects from background. The underlying idea is to transfer foreground/background labels of glocally similar exemplars to a query image. Figure 3.2 shows the framework of the proposed approach. There are three core algorithmic modules:

1. *Glocal scene retrieval*, by using the proposed object-oriented descriptor (OOD), finds a set of glocally nearest neightbors for the query image. In such neighbors, both the appearance of objects and the scene layouts are similar to those in the query image.

2. *Online prediction* predicts foreground probability for the query image. The retrieved k nearest neightbors as well as the query image are over-segmented into regions. A discriminative classifier of support vector machine (SVM), which is learned on-the-fly from the regions of the retrieved exemplars, predicts initial foreground probability for each region of the query image.

3. *Segmentation with SVM prior* produces the optimal segmentation by combining the probabilistic SVM map, created by the online prediction, and the Markov random field (MRF) energy optimization.

The proposed framework is generic, since any algorithm that fits the above modules can be plugged into the framework. For instance, the typical PHOG [74] (pyramid of

histograms of oriented gradients) can be applied for scene retrieval. The performance of using PHOG has been evaluated in experiments (see Section 3.7.1) and is shown to outperform the previous approaches as well. Moreover, we use the SVM for online prediction, and random forest may be an alternative choice for this.

The remainder of this chapter is organized as follows: image features used in the proposed approach are firstly introduced in Section 3.3. After that, the three key algorithmic modules are detailed in Section 3.4, Section 3.5 and Section 3.6, respectively. Then experimental evaluations are presented in Section 3.7. Finally, the chapter is concluded in Section 3.8.

3.3 Image features

In this section, low-level and middle-level features used in our approach are briefly introduced.

3.3.1 Low-level features

The methods of low-level image representation have been significantly advanced in the past years. Nevertheless, it is fare to say that none is perfect for all types of images. To propose a generic solution, we make use of the following three descriptors:

- Color GIST descriptor [69]. The GIST is computed from Gabor filters responses on a 4×4 grid over the entire image. They are extracted at 3 scales, with 8, 8 and 4 orientation bins respectively from each of the CIE L*, a* and b* channels. Thus, the GIST descriptor is a 960-dimensional vector $(3 \times (8 + 8 + 4) \times (4 \times 4))$.
- Scale-invariant feature transform (SIFT) [75]. Histograms of gradients are computed, with 8 orientation bins, on a 4×4 grid over a patch. This results in a 128-dimensional vector $(8 \times 4 \times 4)$. The SIFT descriptors are extracted densely with a step size of 2 pixels.
- Self-similarity feature (SSIM) [76]. The SSIM descriptor computes correlation values between a 5×5 patch and a larger surrounding one which is 20×20 in our experiments. They are firstly transformed into the log-polar space, then

quantized into 32 bins (8 orientations with 4 radial intervals). Hence an SSIM descriptor is a 32-dimensional vector. The SSIM descriptors are also extracted densely with a step size of 2 pixels.

In this chapter, the three descriptors are used for image representation in the module of glocal scene retrieval. Moreover, SIFT and SSIM are also used for region representation in the module of online prediction.

3.3.2 Middle-level representation

The SIFT and SSIM descriptors are further represented by the standard bag-of-visual-words (BOV) [77]. K-means is used to create visual dictionaries for SIFT and SSIM with the sizes of 2000 and 800, respectively. To capture global geometric layout or object configuration, spatial pyramid is also adopted to accumulate visual words, where 3 levels are applied to image description and 2 levels to region representation. We have empirically observed that increasing the pyramid levels does not improve the performance in our application.

3.4 Glocal scene retrieval

In this section, we describe how to retrieve a set of glocally nearest neighbors for a query image. First, a novel high-level image descriptor, named as *object-oriented descriptor* (OOD), is presented, and then the retrieval method is introduced by using this new descriptor.

3.4.1 Object-oriented descriptor

The middle-level image representation based on BOV might be not sufficient to capture semantic meaning; researchers, therefore, propose to transform the middle-level representation into high-level descriptor by leveraging machine learning techniques. A recent method is *attribute descriptor* [78, 79] that describes an object by its parts (e.g., mouth), shape (e.g., cylindrical) and materials (e.g., furry). The attribute descriptor of an image is a set of responses of discriminative

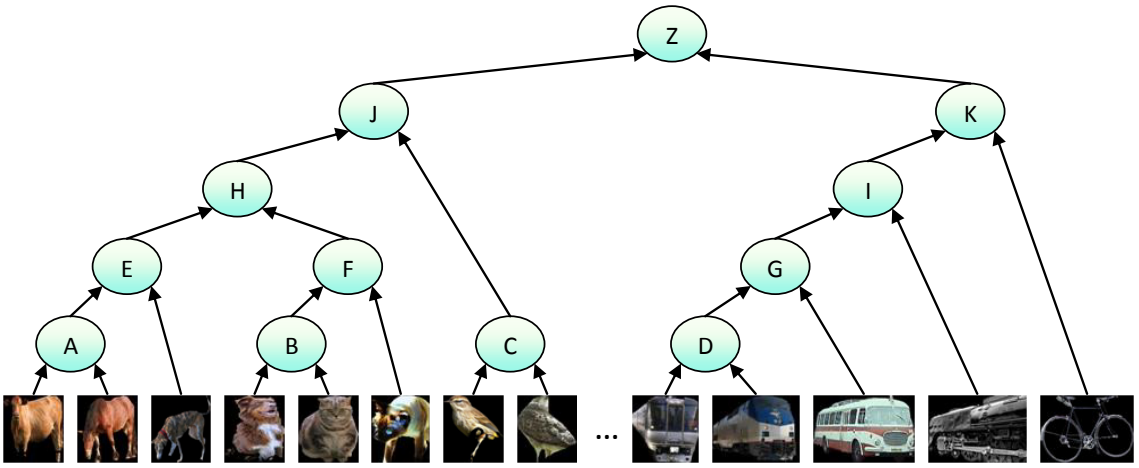


Figure 3.3: Creating pseudo-categories by hierarchical clustering. Objects are clustered by measuring appearance similarity regardless of real category.

classifiers which are learned from the training images with attributes richly annotated. However, neither the attribute nor category information is available in the task of class-independent segmentation. Thus, we propose a generic high-level representation method.

Firstly, a *pseudo-category* is created to gather objects sharing similar appearance together. Objects within manually segmented training images are extracted, and each of them is represented by a BOV vector. The BOV vectors of all objects are collected together and classified into N subsets. Clearly, objects within the same subset share similar appearance. However, it does not mean that they belong to the same real category, since objects of intra-category may show high variation (e.g., chair), and objects from different categories may be similar in appearance (e.g., horse and cow). So we call this subset *pseudo-category*. To classify objects, as shown in Figure 3.3, we make use of agglomerative hierarchical clustering [80], in which each object forms a cluster and pairs of clusters are grouped together to form a new one moving up through the hierarchy. In order to decide which clusters should be merged, one has to define a distance function for measuring similarity between clusters. We employ χ^2 distance defined as

$$\chi^2(\mathbf{f}_i, \mathbf{f}_j) = \sum_{d=1}^{D} \frac{(\mathbf{f}_i(d) - \mathbf{f}_j(d))^2}{\mathbf{f}_i(d) + \mathbf{f}_j(d)} \tag{3.1}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are BOV vectors of a pair of objects, D is the dimension of BOV vector. The main reason accounting for choosing the hierarchical clustering rather than typical K-means is the fact that the K-means does not support χ^2 distance metric, which is powerful for clustering histograms of BOV vectors.

Secondly, N SVM classifiers of pseudo-categories are learned to compute the similarities of an image to each pseudo-category. The SVM classifiers are built by setting images containing objects of specific pseudo-category as positive examples, and the others as negative ones. Note that, since an image may contain objects from different pseudo-categories, one image may belong to positive example for several classifiers. With the learned SVM classifiers, an input image is represented by a score vector $\mathbf{v}_i$ consisting of N SVM classification scores which are typically within the range of $[-3, 3]$. These classification scores naturally represent probabilities of an image belonging to each of N pseudo-categories, i.e., the larger score indicates the higher probability, and vice versa.

Finally, the high-level image descriptor is created by normalizing the SVM score vector. We normalize each score vector $\mathbf{v}_i$ by exploiting the distribution of the score vectors extracted from all training images. Let $\mathbf{V}_t = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_P\}$ denote a set of score vectors of P training images. The normalization is performed as follows

$$\mathbf{h}_i = \frac{\mathbf{v}_i^{'}}{\|\mathbf{v}_i^{'}\|_2} \tag{3.2}$$

where $\|\cdot\|_2$ indicates l_2-norm, $\mathbf{v}_i^{'}$ is the vector difference between $\mathbf{v}_i$ and the mean of all score vectors computed from the training images

$$\mathbf{v}_i^{'} = \mathbf{v}_i - \frac{1}{P}\sum_{p=1}^{P}\mathbf{v}_p \tag{3.3}$$

Here, the normalized score vector $\mathbf{h}_i$ is termed as object-oriented descriptor (OOD). The number of SVM classifiers N is determined by the appearance distributions of objects in the training images. If the objects show high variations in the appearance space, N should be set to a relative larger value. In our experiments, N is moderately set to 40.

Some properties can be observed from OOD:

– As separating hyperplane of SVM is typically very sparse, SVM classifiers simultaneously perform feature selection and classification.

– The feature selection along with spatial pyramid local descriptor aggregation enables OOD to capture global geometric layout and also to highlight local objects in an image.

– With the learned SVM classifiers, it is simple to compute OOD from BOV descriptor, since only a multiplication and a normalization are needed to be performed.

3.4.2 Glocal nearest neighbor retrieval via OOD

With the OOD representation, the key problem of retrieving a set of glocally similar exemplars is to define a distance function for similarity measure, which is still an active research area. We have evaluated the OOD with l_2 distance and l_1 distance, and have experimentally observed that, l_2 distance gives more relevant exemplars when query image is simple and contains only a single object; however, l_1 distance obtains better exemplars when the query image is complex and consists of multiple objects from different categories. As our objective is to find exemplars with objects similar to the query image possibly containing multiple objects, we chose l_1 distance for retrieval.

3.5 Online prediction

The objective of this module is to initially predict foreground probability for a query image. As similar images generally share similar segmentation, we make use of the retrieved k nearest neighbors as reference samples to predict the foreground probability. This module first segments the query image and its k nearest neighbors into regions, and then a region-based figure-ground classifier is trained to predict foreground probability for each region of the query image.

For region generation, we make use of the contour-based hierarchical segmentation algorithm gPb [48] (globalized probability of boundary), which

provides the output as a probability-of-boundary map, so-called ultrametric contour map (UCM). Like other generic bottom-up segmentation methods, gPb is far from perfectly separating objects from background, mainly due to the lack of top-down knowledge about specific objects and image context. To generate regions, we threshold the UCM, normalized from 0 to 1, at 0.125 to ensure that an image is over-segmented.

The figure-ground classifier is learned on-the-fly by using a set of regions segmented from k nearest neighbors. To learn the classifier, we follow our previous work [66] and employ support vector machine with multiple kernel learning (SVM-MKL) proposed in [81]. Positive examples for training are the exemplar regions that mainly belong to objects, and negative examples are the rest of exemplar regions corresponding to background. Let $\mathbf{f}_Q = \{\mathbf{f}_q^1, \mathbf{f}_q^2, \cdots, \mathbf{f}_q^U\}$ denote a set of BOV vectors of a test region and $\mathbf{f}_T = \{\mathbf{f}_t^1, \mathbf{f}_t^2, \cdots, \mathbf{f}_t^U\}$ denote a collection of BOV vectors of a training region, where U is the number of appearance descriptors. The classification function of an SVM in kernel formulation is expressed as

$$C(\mathbf{f}_Q) = \sum_{n=1}^{N} y_n a_n K(\mathbf{f}_Q, \mathbf{f}_T^n) + b \tag{3.4}$$

where $y_n \in \{+1, -1\}$ indicates foreground/background label of the training region, N is the number of training regions, and $K(\cdot, \cdot)$ is the positive definite kernel, calculated as a linear combination of feature kernels

$$K(\mathbf{f}_Q, \mathbf{f}_T^n) = \sum_{u=1}^{U} w_u \Psi(\mathbf{f}_q^u, \mathbf{f}_t^u) \tag{3.5}$$

The kernels $\Psi(\cdot, \cdot)$ are generally chosen by experiments. Typical histogram kernels are from three types: linear, quasi-linear(e.g., intersection and χ^2) and non-linear (e.g., Radial basis function).

The SVM-MKL learns a set of coefficients $\mathbf{a} = \{a_1, a_2, \cdots, a_N\}$, a threshold b and a set of non-negative feature weights $\mathbf{w} = \{w_1, w_2, \cdots, w_U\}$. The learned coefficient vector $\mathbf{a}$, usually termed as separating hyperplane, is typically sparse which suggests that only a representative subset of training features is used for classification. The weight vector $\mathbf{w}$ emphasizes more discriminative features and depresses those of less

discriminative features. For instance, SIFT is generally much more discriminative than SSIM, so it is usually assigned with a larger weight.

With the learned parameters, each region of the query image can obtain an SVM classification score from the classification function (3.4), which is typically within the range of $[-3, 3]$. Such an SVM classification score naturally links to the probability of a region belonging to foreground. For post-processing, the SVM classification scores of all regions are converted to probabilistic values by fitting a sigmoid function to them [82]. Thus an SVM map of input image is generated by assigning the probabilistic values of regions to their corresponding pixels.

A naive approach to segment an image is to threshold its SVM map. Unfortunately, the SVM classification scores are not always reliable and may lead to noisy segmentation. The reasons are mainly two fold. On one hand, the bottom-up gPb segmentation only partitions an image into homogeneous regions and is far from separating objects from background. Thus some features extracted from the regions are not sufficiently distinctive for the SVM classification. On the other hand, the region-based prediction handles each region separately and might cause unsmoothing labeling. Some segmentation results produced by using SVM only can be found in the fifth column of Figure 5.

3.6 Segmentation with SVM prior

To make our approach more robust and obtain coherent segmentation, we leverage SVM scores as a prior for energy optimization of Markov random field (MRF), which not only considers how likely a pixel belongs to an object but also the labels of its neighboring pixels. In this way, the SVM scores provide soft constraint and supplementary information for the MRF optimization through the following segmentation model.

3.6.1 Segmentation model

For segmentation, we use the standard MRF model [39], which defines a Markov random field on pixels of image with a neighborhood system. In such a model, each

pixel is associated with a random variable, which corresponds to its segmentation label. The optimal segmentation is achieved by finding the maximum a posteriori (MAP) configuration in an MRF and is addressed by minimizing the energy function of a pairwise MRF

$$E(\mathbf{L}) = \sum_{n \in \mathcal{P}} \Lambda_n(l_n) + \sum_{\{n,j\} \in \mathcal{N}} \Theta_{n,j}(l_n, l_j) \tag{3.6}$$

where $\mathcal{P}$ denotes the set of all image pixels, $\mathcal{N}$ corresponds the neighborhood system defined on the pixels which is chosen to be 4 or 8 connecting neighborhood, $\mathbf{L} = \{l_1, l_2, \cdots, l_N\}$ is an array of labels (random variables) at pixels, n is the single index of image, $l_n = \{0, 1\}$ with 0 indicating background and 1 indicating foreground objects, Λ_n is the data term and $\Theta_{n,j}$ is the smoothness term.

3.6.2 Data term

The data term measures consistency between the pixel and its label, and is generally defined as the negative log of the likelihood of a foreground/background label being assigned to a pixel, i.e.,

$$\Lambda_n(l_n) = -\log(\Omega(\mathbf{x}_n|l_n)) \tag{3.7}$$

where $\mathbf{x}_n \in \mathbb{R}^3$ is the color feature vector, Ω is an appearance model predicting the foreground or background probability by modeling color distributions in the image. However the color feature is not very discriminative and may lead to inaccurate segmentation. To overcome this problem, we propose a novel data term which incorporates an SVM prior and an appearance model

$$\Lambda_n(l_n) = -\log(\Phi(l_n) \cdot \Omega(\mathbf{x}_n|l_n)) \tag{3.8}$$

where the SVM prior $\Phi(l_n)$ is computed from the figure-ground SVM classification scores. Given the probabilistic SVM map $\mathbf{S} = \{s_1, s_2, \cdots, s_N\}, s_n \in \mathbb{R}^1$, of input image, which is normalized to $[0, 1]$, the SVM prior of pixel n for foreground model

is defined as

$$\Phi(l_n = 1) = s_n. \tag{3.9}$$

Similarly, the SVM prior of pixel n for background model is defined as

$$\Phi(l_n = 0) = 1 - s_n. \tag{3.10}$$

Note that, as described in Section V, the SVM figure-ground classifier is online learned from a set of the most similar images, the SVM prior $\Phi(l_n)$ naturally links each pixel to the foreground/background of its nearest neighbors. Therefore, the proposed data term carries both intra and inter pixel attributes. This suggests that Eq. (3.8) promotes pixels more similar to foreground objects in the exemplar images to be labeled as foreground, and encourages other pixels more similar to the background in those images to be labeled as background.

The appearance model is defined by two Gaussian mixture models (GMMs), where one is for foreground modeling and the other one for background modeling. The GMM is a parametric probability density function represented as a weighted sum of Gaussian densities

$$\Omega(\mathbf{x}_n|\vartheta) = \sum_{i=1}^{Q} w_i g(\mathbf{x}_n|\mu_i, \Sigma_i) \tag{3.11}$$

where Q is the number of Gaussian components (typically $Q = 5$), w_i is the mixture component weight with the constraint that the sum of all component weights equals 1, $g(\mathbf{x}_n|\mu_i, \Sigma_i)$ is a Gaussian probability density function

$$\begin{aligned} & g(\mathbf{x}_n|\mu_i, \Sigma_i) \\ & = \frac{1}{\sqrt{(2\pi)^3|\Sigma_i|}} \exp\{-\frac{1}{2}(\mathbf{x}_n - \mu_i)'\Sigma_i^{-1}(\mathbf{x}_n - \mu_i)\} \end{aligned} \tag{3.12}$$

where $\mu_i \in \mathbb{R}^3$ is the mean vector of data vectors in the same Gaussian component, and $\Sigma_i \in \mathbb{R}^{3\times3}$ is the covariance matrix.

Algorithm 2 Segmentation with SVM prior

Input: test image $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}, \mathbf{x}_n \in \mathbb{R}^3$ and its SVM map $\mathbf{S} = \{s_1, s_2, \cdots, s_N\}, s_n \in \mathbb{R}^1$, where N is the number of pixels.
Output: labeled image $\mathbf{L} = \{l_1, \cdots, l_N\}, l_n \in \{0, 1\}$.

Initialization

- Compute the smoothness term Θ with (3.13).
- Compute an SVM prior Φ with (3.9) and (3.10).
- Initialize $\mathbf{L}$ by thresholding $\mathbf{S}$ with (3.15).

Iterative Optimization

- Learn a set of GMM parameters ϑ based on $\mathbf{L}$.
- Compute appearance model with (3.11).
- Compute data term Λ with (3.8).
- Segment image $\mathbf{X}$ by minimizing (3.6) and update $\mathbf{L}$.
- Stop the iteration if the convergence is reached or the number of iterations is greater than a predefined threshold.

3.6.3 Smoothness term

The smoothness term is defined within the neighborhood system which consists of all pairs of adjacent pixels. Its goal is to ensure the overall label smoothing by penalizing neighboring pixels assigned with different labels. Like in [39, 41], the smoothness term is defined based on the spatial distance and color contrast between neighboring pixels

$$\Theta_{n,j}(l_n, , l_j) = \frac{\varphi}{\mathrm{dis}(n, j)}[l_n \neq l_j] \exp\{-\beta \|\mathbf{x}_n - \mathbf{x}_j\|_2^2\} \tag{3.13}$$

where $\mathrm{dis}(\cdot)$ is the spatial Euclidean distance of neighboring pixels, $\|\cdot\|_2$ indicates l_2-norm. The balance parameter φ is set to 50 which has been proved to be suitable for most real images [83]. The constant β is a contrast-oriented weight. When β is 0, all neighboring pixels are smoothed with fixed degree determined by φ. To make the smoothness adaptive to global contrast of neighboring pixels, β is chosen to be

$$\beta = \frac{1}{2 \cdot \mathrm{mean}((\|\mathbf{x}_n - \mathbf{x}_j\|_2^2) \cdot \mathrm{dis}(n, j))}. \tag{3.14}$$

3.6.4 Overall segmentation algorithm

The overall algorithm of object segmentation is summarized in Algorithm 2. The segmentation procedure consists of two key steps including initialization and iterative optimization.

In the initialization step, we pre-compute the smoothness term by considering both color contrast and spatial distance between neighboring pixels, an SVM prior by using the SVM map $\mathbf{S}$, and initially separate foreground pixels from background to launch the subsequent iterative optimization. To obtain an initial segmentation, we propose a new method by thresholding the SVM map $\mathbf{S}$. The threshold is self-adaptively computed by

$$\eta = \min(\tau \cdot \text{mean}(\mathbf{S}), \varpi \cdot \max(\mathbf{S}))) \tag{3.15}$$

where τ and ϖ are predefined parameters, and are set to 0.8 and 0.6 respectively in our experiments.

The initial segmentation result generated by thresholding the SVM map may be too coarse, so in the second step we refine the segmentation result based on an iterative optimization scheme. First, GMM parameters for foreground/background modeling are estimated via expectation-maximization (EM) [84] using the separated foreground/background pixels. Then an appearance model is computed based on the learned GMM parameters and is coupled with the SVM prior to compute a data term. With the well-prepared data term and smoothness term, a new segmentation is estimated by minimizing the energy function (3.6) via efficient graph cuts [39]. The obtained segmentation result is employed to learn a set of more precise GMM parameters, and then a new round of optimization is launched accordingly. Although the above iterative optimization model is guaranteed to converge at least to a local minimum energy [41], we still limit the maximum number of the iterations to 10 for saving the computation cost.

3.7 Experimental evaluations

The proposed approach is evaluated on three datasets including Pascal VOC 2010, Pascal VOC 2011 [72] and iCoseg [73]. In order to objectively evaluate the segmentation performance, we adopt two commonly used objective measures: F-score and average union metric. The F-score is the harmonic mean of precision and recall

$$F\text{-}score = \frac{(1+\alpha)precision \cdot recall}{\alpha \cdot precision + recall} \tag{3.16}$$

where, α is the parameter to balance precision and recall and is set to 1. The precision and recall are computed over the total dataset and are defined as

$$precision = \frac{\sum_{t=1}^{T} P_t \bigcap G_t}{\sum_{t=1}^{T} P_t} \tag{3.17}$$

$$recall = \frac{\sum_{t=1}^{T} P_t \bigcap G_t}{\sum_{t=1}^{T} G_t} \tag{3.18}$$

where T is the number of test images, P_t is the set of predicted foreground pixels in test image t and G_t is the ground-truth of foreground. As in [67, 68], we also compute the average union (AvU) score defined as

$$AvU = \frac{1}{T} \sum_{t=1}^{T} \frac{P_t \bigcap G_t}{P_t \bigcup G_t} \tag{3.19}$$

In the rest of this section, we mainly evaluate in Section 3.7.1 the segmentation performance of the proposed approach on the Pascal VOC 2010, VOC 2011 datasets, and then validate its adaptability on iCoseg dataset in Section 3.7.2.

3.7.1 Pascal VOC experiments

The proposed approach is mainly validated on the Pascal VOC 2010 and VOC 2011 datasets, which are the widely acknowledged difficult datasets for both segmentation and recognition. In this subsection, a brief introduction for the datasets and baselines for performance comparison are given first. Then, we analyse the segmentation performance of the proposed approach with different

configurations. Furthermore, the proposed approach is compared with the state-of-the-art figure-ground segmentation methods both quantitatively and qualitatively. Finally, we discuss some failure cases and analyse the computation cost.

Pascal VOC 2010 and VOC 2011 datasets

The Pascal VOC 2010 and VOC 2011 datasets contain 1928 and 2223 images from 20 object classes, and each image is manually annotated. In each dataset, about one half of images contain multiple objects (on average 3 or 4 objects), and about 30% of images are with occlusion. Both datasets are evenly split into training and validation sets. Note that, we aim at a class-independent segmentation approach, the image category information is not used, therefore, we do not distinguish object classes, but assign all objects as foreground.

As the VOC segmentation datasets are originally designed for the performance evaluation of multi-class object segmentation, each of 20 classes objects is labeled with a unique integer ID within $[1, 20]$ and the background is labeled as 0. Moreover, there are a few other areas labeled as 255, which are from ambiguous objects, significantly truncated objects and boundary pixels separating different objects and background. Here we name them as *difficult areas*. In the previously published works [67, 68], segmentation accuracies are computed by setting these areas as background. However, since some of them actually belong to the foreground; if we ignore them, as recommended by the dataset designers, the segmentation accuracies will be obviously different from setting them as background. For objective evaluation and consistent comparison, the evaluation scores will be computed in two ways: setting the difficult areas as background or ignoring them.

Baselines

We use three state-of-the-art figure-ground segmentation approaches as baselines. The first one is the method proposed in [67], which performs segmentation transfer based on two sets of global similar images. For abbreviation, we call this approach

global transfer. The second approach is *window transfer* proposed in [68], which realizes the segmentation transfer based on windows detected by off-line learned model of Objectness [50]. The third approach is *CPMC* [64], which learns a ranking model to select regions mostly covering objects from multiple segmentations. The former two competitors exactly match with our approach, while CPMC is different from ours as it does not generate a single segmentation mask for all objects in an image. To compare with the CPMC, we report its results of the first ranked segmentation. All results of these three approaches are produced by authors' publicly available codes [1] [2] [3].

Performance analysis

As shown in Figure 3.4, we evaluate the system performance with different configurations by varying the number of nearest neighbors k. The performance of global transfer [67], which is the most relative to our approach, is also presented in Figure 3.4 for comparison.

Validation of OOD: OOD is designed for finding a set of exemplars glocally similar to the query image. Note that, the conventional quantitative retrieval evaluation method is not applicable to OOD for at least two reasons. On one hand, OOD aims at finding object's appearance and object layout similar to that of the query image, rather than to find the same category of the object. On the other hand, as the query image might contain multiple objects from different categories, the retrieved image might exactly match with the object categories in the query image, partially match or totally mismatch. In this case, it is difficult to assess the retrieval accuracy. Therefore, we evaluate OOD by its impact on segmentation performance.

First of all, we compare OOD with PHOG [74] (pyramid of histograms of oriented gradients). For evaluation, we use two sets of nearest neighbors, retrieved by using OOD and PHOG respectively, for segmentation transfer, and then we compute their segmentation results by varying the number of nearest neighbors k.

1. https://sites.google.com/site/amirrosenfeld/
2. http://groups.inf.ed.ac.uk/calvin/software.html
3. http://www2.maths.lth.se/matematiklth/personal/sminchis/code/cpmc/

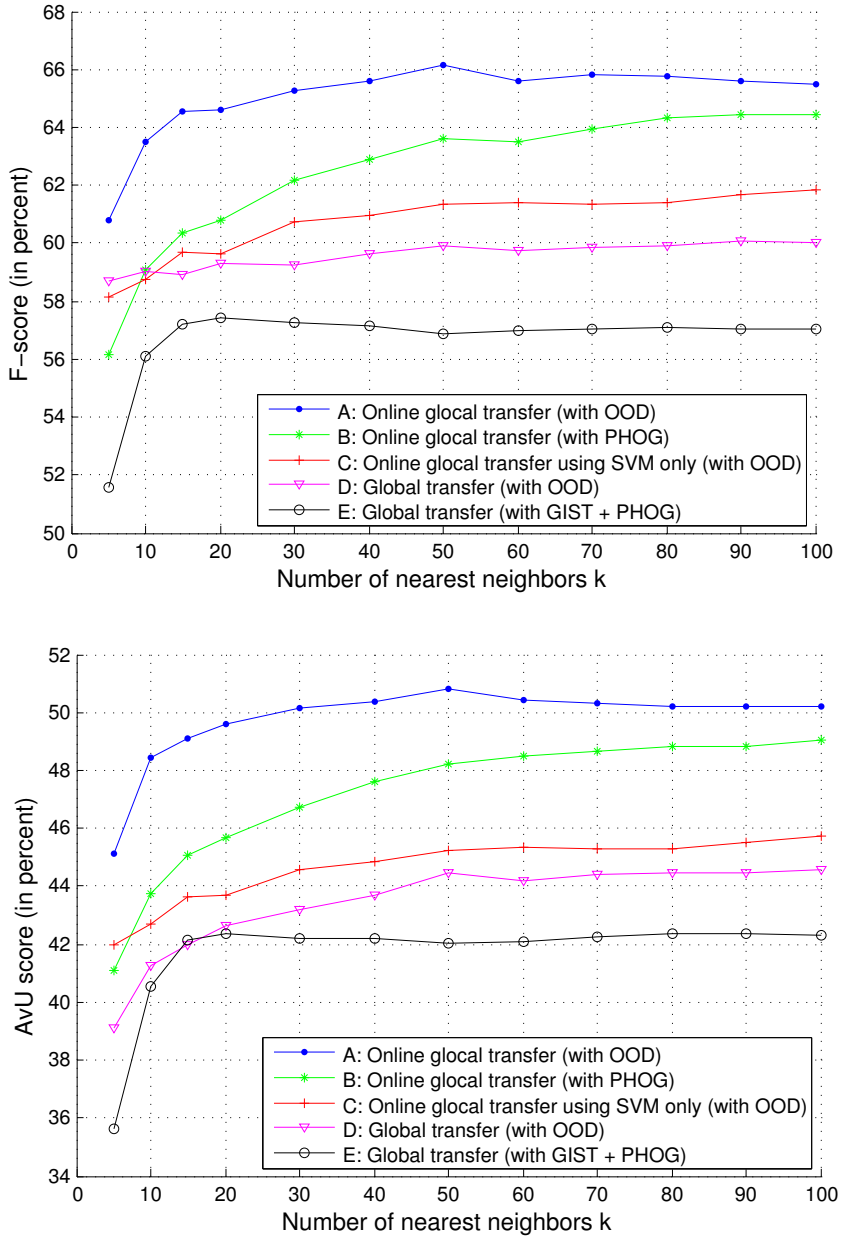


Figure 3.4: F-score (top) and AvU score (down) on Pascal VOC 2011 by varying the number of nearest neighbors k. The curve A shows the performance of our full method of online glocal transfer. The curve B shows the performance of online glocal transfer using PHOG for image retrieval rather than using OOD. The curve C shows the performance of online glocal transfer using only SVM prediction (without MRF optimization). The curve D shows the performance of global transfer [67] with the proposed OOD for image retrieval. The curve E shows the performance of original global transfer [67]. All results are computed by setting difficult areas of ground-truth as background.

As shown in Figure 3.4, the online glocal transfer with OOD (curve A) obtains a higher performance than with PHOG (curve B). The improvement is very obvious when k is small (less than 60). This also implies that using OOD leads to more efficient online learning as less training exemplars are needed.

In addition, OOD is evaluated in global transfer system [67], in which GIST is used to retrieve scenes with similar layout for location modeling, and PHOG is employed to find images with similar content for appearance modeling. To validate the performance of OOD, we replace the neighbors retrieved by OOD with those two sets of neighbors retrieved by GIST and PHOG respectively, and compute the segmentation results. From Figure 3.4, we can observe that, in the stable range of $k \geq 20$, the global transfer with the OOD (curve D) improves performance about 3% on F-score and 2% on AvU score, compared to original global transfer method (curve E).

Validation of the online glocal transfer scheme: to evaluate the novel segmentation scheme, which combins online prediction and MRF-based segmentation model, we compare the online glocal transfer (using OOD or PHOG) with the global transfer using OOD. As shown in Figure 3.4, even though the online glocal transfer with PHOG (curve B) obtains a lower performance than the online glocal transfer with OOD (curve A), it still outperforms the global transfer using OOD (curve D) by a wide margin: about 4% improvement in terms of both F-score and AvU score. This shows that the proposed scheme of online glocal transfer is superior to global transfer [67].

To see how the MRF optimization contributes to the segmentation, results generated by simply thresholding the SVM map are also computed for comparison. As shown in Figure 3.4, with the MRF optimization (curve A), about 4% improvement is obtained in terms of F-score and AvU score in the whole range of k. Figure 3.4 also reveals that our approach without the MRF optimization still outperforms the global transfer with OOD. Thus, the advantage of the online glocal transfer scheme is further demonstrated.

In summary, the compelling performance of the proposed approach stems from the novel image descriptor method OOD and the new scheme combining online

Table 3.1: Segmentation accuracies (in %) on standard validation sets of Pascal VOC 2010 and VOC 2011. (Difficult areas are set to background).

	VOC 2010		VOC 2011	
Approach	F-score	AvU	F-score	AvU
CPMC [64] (first ranked seg.)	44.0	33.2	43.4	32.7
Global transfer [67]	58.3	43.6	57.7	42.4
Window transfer [68]	61.6	47.8	60.7	46.5
SVM only (OOD)	61.1	45.5	61.3	45.3
Online glocal transfer (PHOG)	63.9	49.0	64.3	48.6
Online glocal transfer (OOD)	**66.3**	**51.2**	**66.1**	**50.8**

Table 3.2: Segmentation accuracies (in %) on standard validation sets of Pascal VOC 2010 and VOC 2011. (Difficult areas are ignored).

	VOC 2010		VOC 2011	
Approach	F-score	AvU	F-score	AvU
CPMC [64] (first ranked seg.)	45.9	36.6	45.5	35.1
Global transfer [67]	61.0	47.1	60.5	46.3
Window transfer [68]	63.8	51.3	63.0	50.0
SVM only (OOD)	63.7	49.2	64.1	48.9
Online glocal transfer (PHOG)	66.7	52.7	67.1	52.5
Online glocal transfer (OOD)	**68.5**	**55.0**	**68.7**	**54.7**

prediction and MRF segmentation model. Compared with GIST and PHOG, the OOD retrieves more similar exemplar images, which are crucial for exemplar-based segmentation, and helps to improve segmentation quality. The proposed segmentation scheme computes the data term in Section 3.6.2 considering both intra pixel attributes and similarity to the foreground in the most similar exemplar images, thus the robustness of the segmentation model is enhanced.

Quantitative comparison

In Table 3.1 (difficult areas are set to background) and Table 3.2 (difficult areas are ignored), our segmentation accuracies are compared with the three baselines on the Pascal VOC 2010 and VOC 2011 datasets. Results of these baselines are reported for their best parameters: 80 nearest neighbors for global transfer [67], 100 windows for window transfer [68] and the best segmentation ranked by author's

learned model for CPMC [64]. For our results, the online glocal transfer using PHOG is reported with 80 nearest neighbors; SVM prediction only (without MRF optimization) and our full method of online glocal transfer using OOD are reported with 50 nearest neighbors. As shown in the two tables, all approaches obtain higher accuracies in Table II computed by ignoring the difficult areas. Our approach with only SVM prediction is comparable to the three baselines in terms of F-score. The online glocal transfer with PHOG is shown to outperform these baselines in terms of the two evaluation metrics. Our full method of the online glocal transfer with OOD further improves the performance with more than 2% in average on both F-score and AvU on the two datasets.

Qualitative evaluation

Figure 3.5 shows some segmentation results produced by global transfer [67], window transfer [68] and our approach. We can make some observations from the segmentation results. First of all, combining the MRF optimization with the online SVM prediction achieves a higher segmentation quality than segmenting by directly thresholding the SVM map. As shown in the two rightmost columns, some noise labels are removed while object contours are preserved. Secondly, our approach can recover partially truncated foreground objects. For instance, in the first three rows, the train, the airplane and the motorbike are truncated on the image border, but they are correctly labeled as foreground. Last but not the least, the proposed approach shows its potential to address occlusion and the cluttered scene, which are the most challenging situations in segmentation task. For example, in the fourth, fifth and sixth rows, the car is significantly occluded by the tree trunk, and the cow and the horse are occluded by the barriers; in the last four rows, the car, the sheep, the buses and the boat are in the cluttered environments. Our approach shows the robustness to segment out these objects, while global transfer [67] and window transfer [68] either fail or merge some background regions with objects.

Figure 3.5: Some segmentation results generated by different methods on Pascal VOC 2010 and VOC 2011 segmentation datasets.

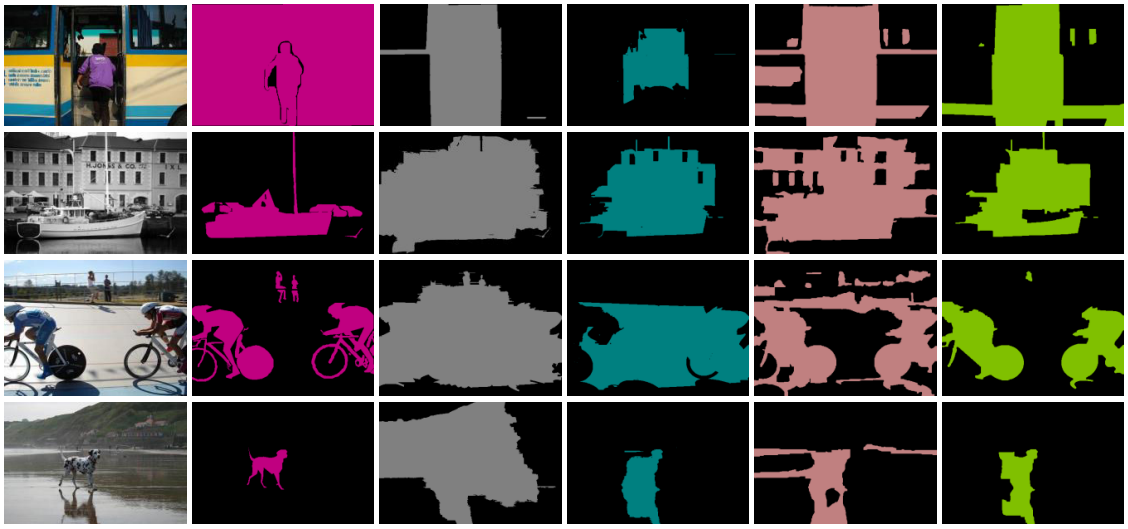


Figure 3.6: Some failure cases.

Failure cases and analysis

Though the proposed approach outperforms the sate-of-the-art figure-ground segmentation methods on both qualitative and quantitative evaluations, our approach can not perform well on some difficult cases shown in Figure 3.6. As shown in the first example, a part of side view of the bus totally covers the whole image. The proposed approach only extracts the most outstanding object (the person) with a part of bus regions. Mainly because the segmentation model generally supposes that in an image both foreground objects and background exist, and less confident regions of the bus are labeled as background. For some objects very similar to background in both color and edge orientation, such as the second example in Figure 3.6, our approach can not completely separate the objects from the background, and it merges a part of background regions with objects. Besides, for images containing multiple objects with significant variations in appearance and scale, the too small and blurry objects are missed in the segmentation, e.g., the bird and cars in the second example, and the two distant people in the third example. In addition, our approach also has difficulty to separate an object from its reflection in the water like the fourth example. However, as shown in Figure 3.6, other approaches also can not perform well on such difficult cases.

Computation cost

To analyze the complexity of the proposed approach, we computed the run-time for each of the three components of our approach. On a laptop with Intel i7 CPU (2.2 GHz) and 8GB RAM, the Matlab implementation takes about 2 seconds for glocal scene retrieval, 108 seconds for online prediction and 4 seconds for segmentation with SVM prior. Obviously, the online prediction occupies the main computation cost, since it includes two time-consuming operations, gPb region generation and SVM prediction. The former takes 97 seconds to segment an image with the configuration that the resizing factor for eigenvector computation is set to 0.5, while the latter takes 11 seconds for region-based foreground prediction. It is clear that the main computation cost of the proposed approach comes from gPb region generation. Fortunately, gPb can be significantly accelerated by using GPU implementation with parallel computing. As reported in [85], the optimized gPb on a NVidia GTX 280 GPU only uses 1.8 seconds to process an image with a resolution of 481×321 (approximate 0.15 Megapixels). The other components also can be accelerated by using a parallel GPU implementation, such as in the glocal scene retrieval, the feature extraction can be done in parallel on the basis of pixel. Therefore, the computation cost of the proposed approach can be substantially reduced with a parallel GPU implementation.

Table 3.3 compares the run-time of different approaches. The global transfer [67] has an obvious advantage in computation cost: it only takes 4 seconds to segment an image, while window transfer [68], our approach and CPMC [64] need 97 seconds, 114 seconds and 230 seconds, respectively.

Table 3.3: Approximate run-time (in second) per image of Matlab implementations.

	CPMC	Global transfer	Window transfer	Ours
Run-time	230	**4**	97	114

3.7.2 Cross-dataset experiments

In the previous experiments, object classes of query image certainly exist in the training set, and exemplar images sharing the same object classes may be retrieved for segmentation transfer. However, obtaining segmentation ground truths is burdensome, and it is impractical in reality to annotate images of all object classes for segmentation transfer. From this observation, the question is: "Is it possible to segment an image with unknown objects by leveraging a set of available exemplar images?". To validate this, we perform segmentation on iCoseg dataset [73] by transferring exemplar segmentations from training set of Pascal VOC 2011. The iCoseg dataset contains 643 images from 38 object classes, most of which never exist in Pascal VOC datasets, such as panda, elephant, tiger, kite, statue, and Stonehenge.

In the rest of this subsection, the baselines including exemplar-based approaches and co-segmentation methods are given first. Then we present and discuss the results generated by exemplar-based approaches. Finally we compare segmentation performance across different methods.

Baselines

We firstly evaluate segmentation performance within exemplar-based figure-ground segmentation approaches, thus global transfer [67] and window transfer [68] are used to compare. To see the performance of different kinds of segmentation approaches, we also compare with two state-of-the-art co-segmentation methods [86, 87], which simultaneously segment several images containing the same object classes.

Results of exemplar-based approaches

Segmentation accuracies obtained by different exemplar-based approaches are shown in Table 3.4. Surprisingly, all approaches provide consistent good results by transferring exemplar segmentations from Pascal VOC 2011 to images in iCoseg dataset. This suggests that the exemplar-based approaches have the potential to

segment large-scale images by using a set of the segmented exemplar images. Among these approaches, the proposed online glocal transfer achieves the best performance. It improves by 5.3% and 3.3% in terms of F-score and AvU score over the second one, window transfer [68].

Some segmentation examples generated by the proposed approach are shown in Figure 3.7. Most objects are extracted well, even though they never appear in exemplar images of Pascal VOC 2011. The reasons are two-fold. On one hand, different object classes might be globally or locally similar in appearance, such as the brown bear may be referred to dog due to their similar color. On the other hand, while it is more difficult to find similar objects across different datasets, retrieving similar background is much more easier, e.g., sky, grass and water may not show significant variations in most images. Both the retrieved similar objects and background scenes are helpful for the segmentation transfer.

Table 3.4: Segmentation accuracies (in percent) on iCoseg dataset.

Approach	F-score	AvU
Global transfer [67]	69.5	56.4
Window transfer [68]	74.8	64.2
Online glocal transfer	**80.0**	**67.5**

Comparison with co-segmentation methods

To validate segmentation performance across different methods, the exemplar-based segmentation approaches are also compared to two state-of-the-art co-segmentation methods [86, 87] on iCoseg dataset. As in [86, 87], we compute AvU score for each object class rather than for the whole dataset, and compare the AvU scores for 10 classes reported in [86, 87]. As shown in Table 3.5, the proposed online glocal transfer achieves the best performance for 7 out of 10 object classes among the five methods. In addition, it also increases the average of the 10 AvU scores to 72.9% and obtains 6.8% improvement compared to the second one [87].

Table 3.5: Average union (AvU) score on iCoseg dataset. The results for [86, 87] are taken from Table 2 in [87].

Object class	*Co-segmentation [86]*	*Co-segmentation [87]*	*Global transfer [67]*	*Window transfer [68]*	*Online glocal transfer*
Baseball player	51.1	62.2	52.7	**69.3**	65.9
Brown bear	40.4	75.6	54.1	73.2	**91.9**
Elephant	43.5	65.5	62.7	63.8	**76.1**
Ferrari	60.5	65.2	51.8	68.1	**73.0**
Football player	38.3	51.1	46.7	46.4	**51.4**
Kite panda	66.2	57.8	81.2	72.3	**93.8**
Monk	71.3	**77.6**	42.7	63.2	67.2
Panda	39.4	55.9	74.2	55.7	**74.9**
Skating	51.1	64.0	58.6	63.5	**78.2**
Stonehenge	64.6	**83.3**	65.1	67.3	56.0
Average	52.6	66.1	59.0	64.3	**72.9**

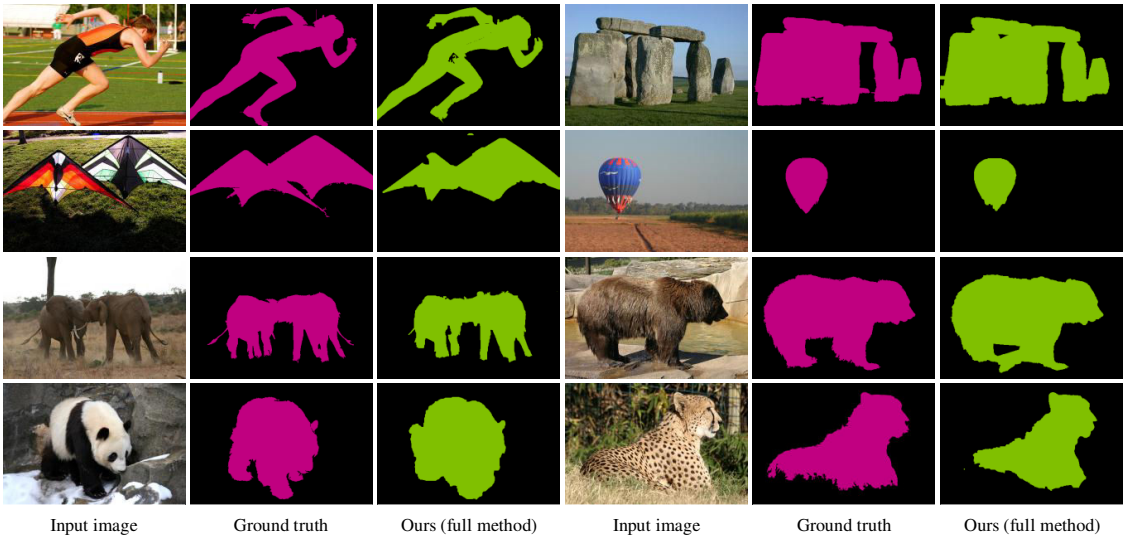


Figure 3.7: Some segmentation results on iCoseg dataset produced by the proposed approach. All results are generated by transferring exemplar segmentations of Pascal VOC 2011 to images of iCoseg dataset.

3.8 Conclusion

We have proposed a novel automatic figure-ground segmentation approach by transferring segmentation masks of glocally similar exemplars into query image. Firstly, object-oriented descriptor (OOD) is proposed as high-level image representation which implicitly encodes geometric information and highlights objects in an image. This descriptor enables to efficiently find better exemplars for segmentation transfer and thus leads to higher segmentation accuracy compared to using the combination of GIST and PHOG descriptors. Secondly, a novel scheme that combines online prediction and energy optimization of Markov random field is proposed to improve the robustness of segmentation model and achieves the optimal segmentation.

Extensive evaluation has been performed on three datasets including Pascal VOC 2010, VOC 2011 segmentation challenges and iCoseg dataset. Experiments demonstrate that: (i) using the scheme of online glocal transfer with typical PHOG for image retrieval can outperform state-of-the-art techniques; (ii) the online glocal transfer with OOD improves the performance further, e.g., compared to the best results of recently proposed window transfer [68], the segmentation

accuracy in terms of F-score criteria increases from 63.0% to 68.7% on Pascal VOC 2011; (iii) the proposed approach has the potential to segment large-scale images containing unknown objects, which never appear in the exemplar images.

4 Semantic image segmentation

4.1 Introduction

In this chapter, we focus on the problem of semantic image segmentation, which aims to assign a semantic label, e.g. “car”and “building”, to each pixel in an image. This has high practical value in many applications, such as image editing, object retrieval, content-based image coding and large-scale internet image management. A number of approaches have been proposed for semantic image segmentation. These methods are either formulated in terms of pixels [88] or regions [44, 89–93]. As a single pixel feature along, e.g. intensity or color, is not sufficiently discriminant for semantic labeling, region-level inferences are generally considered a better choice.

The region-based prediction is typically combined with high-level knowledge to achieve semantic segmentation. In [90], Fisher vector is introduced to describe over-segmented regions and image classification is applied to globally predict object classes in an image. In [91], image tags and scene information are utilized to infer the existence of an object. In [44], bounding boxes, acquired by object detection are used as a prior of the segmentation. A number of researchers also suggested incorporating different cues into a random field (RF) model [92–100]. For example, in [92], probabilistic latent semantic analysis model is integrated to Markov Random Field (MRF) model for the purpose of fusing region-level labels and image-level assumptions. In addition, in [93], image appearances and context information predicted by a set of classifiers are combined within conditional random field model. All these methods suggest that combining different cues might

produce better results.

However, while training prediction models, most of existing region-based approaches for semantic segmentation extract local features directly from objects delineated by ground-truth and or single-level regions generated by over-segmentation; and at the testing step, the features are extracted on single-level regions. As known that low-level segmentation is unstable and cannot precisely separate objects, while local features are only extracted on the single-level regions for recognition, errors from the low-level segmentation might directly migrate to semantic inference.

Based on aforementioned observation, this chapter, which partly appeared as [66, 101], explores to extract local features on multi-level regions for both training and testing steps. The region sets used for training and testing are respectively named as training region bank (TRB) and query region bank (QRB). Our motivation is that by fusing multi-level regions one might have more chance to capture objects or discriminative parts of objects; besides, region hierarchy provides natural spatial constraint for region representation. As the second contribution, we propose sparse coding as the high-level region representation. While it has been shown to lead to high accuracy of image classification [102], the sparse coding has not been applied to semantic image segmentation yet. We demonstrate that, even without using any random field models which are widely used in recent approaches to incorporate multi-cues, our algorithm obtains state-of-the-art results on the standard dataset of semantic segmentation.

This chapter is organized as follows. Section 4.2 overviews the framework of the proposed approach. Then, the three key algorithmic components, including region bank generation, sparse-based region description and semantic prediction, are described in Section 4.3, Section 4.4 and Section A.4.3, respectively. After that, the experimental evaluations are presented in 4.6 and finally, the chapter is concluded in Section A.4.4.

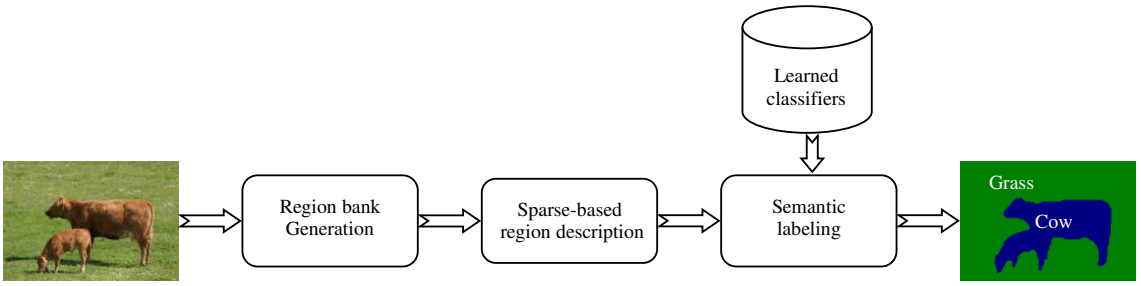


Figure 4.1: Framework of the proposed semantic image segmentation approach.

4.2 Overview

Figure 4.1 illustrates the framework of the proposed approach, which consists of three key algorithmic modules, i.e., region bank generation, sparse-based region description and semantic prediction.

Region bank generation generates a set of multi-level regions from an input image. The motivation of using multi-level regions is based on the observation that the state-of-the-art single-level segmentation algorithm still have difficulty to separate objects from background, however, objects may be captured at certain levels.

Sparse-based region description extracts local invariant features for each region in the region bank, and represents the extracted local features via sparse coding. While many local feature descriptors are available, we emphasize our work on a compact and robust representation of the local feature descriptors using sparse coding, which represents each local feature descriptor with several basis vectors and describes all local feature descriptors in the same region with a *single* histogram.

Semantic labeling assigns each region in the region bank with a predefined semantic label and fuses all labeled regions into a single label map with the same size of original image. We cast the semantic labeling problem as the region classification, which associates a sparse-represented region with a set of classification scores of semantic object categories, and the fusion decision is based on these scores and region size.

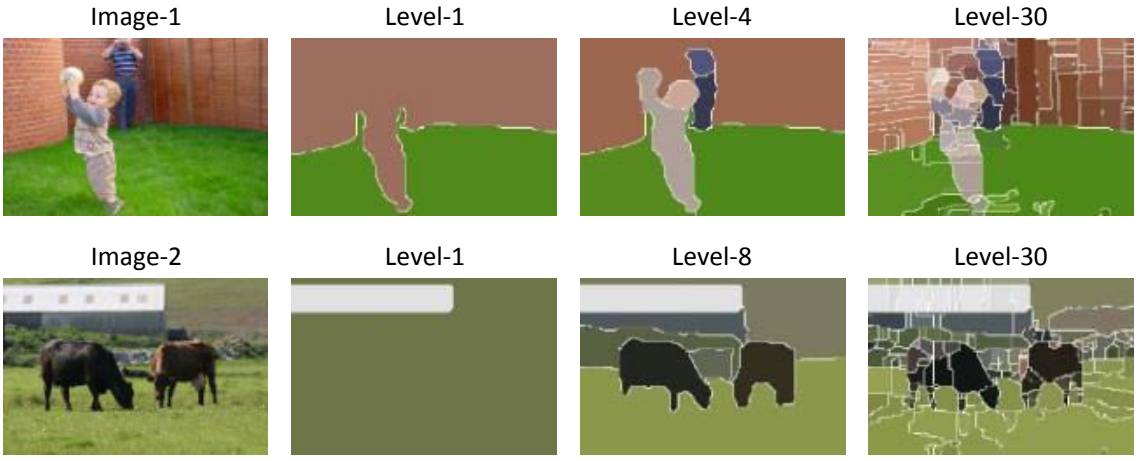


Figure 4.2: Two examples of multi-level segmentations.

4.3 Region bank generation

Region bank is a set of multi-level regions. There are mainly two reasons to use region bank for semantic segmentation. On one hand, single-level segmentation or over-segmentation is unstable and far from precisely separating objects. In most cases, objects are segmented into many regions. On the other hand, hierarchical segmentation might capture objects at some levels, but the optimal segmentation level for objects is unpredictable and may change according to components of images. As shown in Figure 4.2, the best segmentation for Image-1 is at Level-4, where face, bodies, grass and building are near perfectly separated; while for Image-2 the best one is at Level-8, where cows, grass and building are segmented with very few pixel merging. Based on this observation, we leverage the multi-level regions for semantic segmentation.

To generate region banks, we choose contour-based hierarchical segmentation method gPb proposed in [48]. Because it generally preserves object global contour while providing hierarchical regions. The segmentation result of gPb is a valued ultrametric contour map (UCM), where the contour values reflect contrast between neighboring regions. Hierarchical regions are created by thresholding the UCM with a set of thresholds. The key problem of thresholding is how to define the thresholds. Considering the fact that over-segmentation might lead to noisy labeling and under-segmentation might result in two or more objects merging into the same region, the

thresholds should neither be set too small nor too large. In addition, it is inadvisable to fix arbitrarily minimum and maximum thresholds, because the contour values in UCM strongly depend on luminance and contrast of the image. Therefore, we design a self-adapting approach to define the range of thresholding: the minimum and maximum thresholds are computed by multiplying the maximum UCM value of input image by predefined parameters α and β. In our experiments, α and β are set to 0.25 and 0.8 respectively. Contour values in this range are taken as the thresholds to create hierarchical regions. Typically we obtain 5 to 20 thresholds per image. Even such strategy cannot totally avoid the problem mentioned above; we will consider this aspect during the semantic labeling stage.

The region set generated from gPb segmentation for a query image is called as query region bank (QRB); and that generated from gPb segmentation and ground truth segmentation for training images is called as training region bank (TRB).

4.4 Sparse-based region description

After obtained the region banks, we aim to each region in a compact and robust representation. to this purpose, we first extract local features from pixels for each region, and represent the extracted local features by the proposed sparse coding. In the rest of this section, we briefly introduce the local features used in our approach, and describe the sparse coding for region representation.

4.4.1 Local features

In experiments, we use two local features, i.e., Scale-Invariant Feature Transform (SIFT) [75] and self-similarity feature (SSIM) [76].

SIFT descriptors are extracted on a regular grid with a step-size of 6 pixels. And these descriptors are computed for each RGB component. So one SIFT descriptor is represented with a 3×128 dimensional vector. For each grid, the SIFT descriptors are computed respectively at four scales (4, 8, 12, 16 pixel radii).

SSIM descriptors are extracted from a regular grid with step-size of 4 pixels. The SSIM descriptor is generated by computing correlation map of 5×5 pixels patch in

a surrounding 20×20 pixels patch, and then quantizing it into 40 bins (10 angles, 4 radial intervals). Hence one SSIM descriptor is a 40-dimensional vector.

Both SIFT and SSIM features are extracted in a dense approach instead of sparse approach which only computes descriptors on keypoints. This is because keypoint detectors generally have difficulties to detect keypoints in uniform regions, such as sky, calm water and road, and lead to non-assignment on these areas. Therefore, we prefer to compute the local feature descriptors over the entire image and then project them to each region in the image.

4.4.2 Sparse coding

Since a region may contain a great amount of SIFT/SSIM local descriptors, the remainder problem is how to represent these descriptors in a compact manner without loss of representative information. Generally, this is done by using standard bag-of-visual-word (BOV) model, which first learns a visual dictionary and then represents each local feature descriptor with the nearest basic vector in the dictionary in terms of the predefined distance measure. However, the BOV model results in quantization error, since only a single basic vector is used to represent a local feature vector. To address this problem, we introduce sparse coding for region description.

Given a set of local feature vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ in $\mathbb{R}^{M \times N}$, our purpose is to construct a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K]$ in $\mathbb{R}^{M \times K}$, where each column represents a basic vector, and to describe each local feature vector approximately as a weighted linear combination of a few basic vectors

$$\begin{gathered} \mathbf{x}_n \cong \mathbf{D}\mathbf{a}_n \\ \text{such that } \mathbf{a}_n \trianglerighteq 0, \forall n = 1, 2, \cdots, N \end{gathered} \tag{4.1}$$

where $\mathbf{a}_n$ in $\mathbb{R}^{K \times 1}$, is weight vector, in which most entries are zero, $\mathbf{a}_n \trianglerighteq 0$ denotes all elements in $\mathbf{a}_n$ are non-negative. Solving this problem is equivalent to optimizing the cost function

$$\begin{gathered} f(\mathbf{D}, \mathbf{A}) = \min_{\mathbf{D},\mathbf{A}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{D}\mathbf{a}_n\|_2^2 \\ \text{such that } \mathbf{a}_n \trianglerighteq 0, \forall n = 1, 2, \cdots, N \end{gathered} \tag{4.2}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_K]$ in $\mathbb{R}^{K \times N}$, $\| \cdot \|_2$ is the l_2 norm. To do this we apply positive constrained sparse coding [103] to Eq. (4.2)

$$\begin{gathered} \min_{\mathbf{D},\mathbf{A}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{D}\mathbf{a}_n\|_2^2 + \lambda \|\mathbf{a}_n\|_1 \\ \text{such that } \|d_k\|_2 \leq 1, \forall k = 1, \cdots, K, \mathbf{a}_n \trianglerighteq 0, \forall n = 1, 2, \cdots, N \end{gathered} \tag{4.3}$$

where λ is a regularization parameter. l_1 regularization ensures to produce sparse coefficients for a_n [104]. Constraint l_2 norm of vector $\mathbf{d}_k$ less or equal to unity is to prevent $\mathbf{D}$ from taking arbitrarily large values which would due to arbitrarily small values of $\mathbf{A}$. The dictionary $\mathbf{D}$ is obtained by minimizing Eq.(A.25) with respect to $\mathbf{D}$ and $\mathbf{A}$ (i.e. alternatively minimizing over one while keeping the other one fixed). Once dictionary $\mathbf{D}$ is constructed, sparse coefficient vector can be computed by minimizing Eq.(A.25) only with respect to $\mathbf{A}$. Accordingly, each local feature descriptor x_n can be approximated by multiplying the dictionary $\mathbf{D}$ and a sparse coefficient vector a_n. In other words, sparse coding represents one local feature vector with a linear combination of a few basic vectors. We have compared reconstruction performance of sparse coding and BOV methods. The former decreases the Mean Squared Error (MSE) from 6.4 to 2.6 corresponding to 59% reduction in case of reconstructing SIFT feature with a dictionary containing 2000 basic vectors (see Section 4.6).

For compact feature representation, a subset of local feature vectors is randomly chosen to train SIFT and SSIM sparse dictionaries respectively with 2000 and 800 basic vectors (these values are determined experimentally). Then the dictionaries are used to compute sparse vectors of the regions.

4.5 Semantic Labeling

In this chapter, we aim to assign each sparse coded region in the query region bank (QRB) with a semantic label and generate a semantic label map for the query image. To do this, we first associate each region with a similarity score to each of the predefined semantic categories, and then generate the semantic label map by fusing the scored regions.

4.5.1 Region scoring

We now classify sparse coded regions to relevant object classes. Theoretically, any discriminative classifier may be performed for this task. In this study, we prefer Support Vector Machine (SVM) with Multiple Kernel Learning (MKL) [81], as it is easy to train classifiers incorporating several kinds of features even if these features are mapped by different kernels.

For classification, we firstly compute normalized histogram of sparse vectors for each region

$$h_i = \frac{1}{J_i} \sum_{j=1}^{J_i} a_j \tag{4.4}$$

where a_j denotes sparse vectors in each region R_i, J_i denotes the dimensionality of sparse vector.

By using Eq. (A.26), we can compute the histogram of SIFT sparse vectors denoted $\mathbf{h}_i^t$, and that of SSIM sparse vectors denoted as $\mathbf{h}_i^m$. Let $\mathbf{h}_i^c = \{\mathbf{h}_i^t, \mathbf{h}_i^m\}$ define as the combination of feature histograms. So the classification function of an SVM in kernel formulation is expressed as:

$$SVM(h^c) = \sum_{i=1}^{I} y_i a_i K(h^c, h_i^c) + b \tag{4.5}$$

where h_c is feature histogram of a test region; $\{h_i^c \; \forall i = 1, \cdots, I\}$ are feature histograms of I training regions; $y_i \in \{+1, -1\}$ indicates the class label; and K is the positive definite kernel, which is calculated as a linear combination of feature histogram kernels

$$K(h^c, h_i^c) = d_t K(h^t, h_i^t) + d_m K(h^m, h_i^m) \tag{4.6}$$

where d_t and d_m denote non-negative kernels weights. Many kernels can be applied for the histogram-based classification, such as intersection kernel, Chi2 kernel and RBF kernel. In our experiments, Chi2 kernel is used for both the histograms of SIFT and SSIM. MKL learns the kernel weights d_t and d_m and parameters a_i, b for each class. By using Eq. (A.27), a test region can obtain a SVM score, indicating

the likelihood of object class, from each classifier. These scores are then used for labeling regions.

4.5.2 Region labeling

The most direct approach for labeling scored regions of a test image is to assign these regions with the most likely class labels. However it cannot be directly applied to our algorithm, because the hierarchical regions are overlaid or crossed with each other; in addition, as mentioned in Section 4.3, those regions generated by coarse thresholding might merge several objects. Our solution is to combine the effect of SVM scores with that of sizes of regions.

The labeling process mainly consists of three steps. Firstly, the most likely object classes that have the maximum SVM scores are used to pre-label each region. Secondly, these regions are sorted by their increasing SVM scores. Finally, the regions are gradually merged, starting from lower scores, to form a complete labeled image by observing their sizes and SVM scores. Thus when a candidate region R_j, or its part, locates at the same position as labeled region R_i, it can overwrite this one only if its score is greater than a given threshold and its size is not much larger than R_i. This strategy avoids labeling small objects as their surrounding environment or neighboring large objects.

4.6 Experimental evaluations

The proposed approach is evaluated on the MSRC 21-class dataset [88], which can be considered as the standard evaluation dataset for semantic image segmentation. This dataset contains 591 color images of 21 object classes. Each image has a ground truth segmentation that uses different colors to label each pixel with one of 21 object classes or void (in black). We use the same splitting protocol as previous works [88, 90]: 276 images for training and the rest 315 images for testing.

In order to objectively evaluate the segmentation performance, we adopt two widely used evaluation metrics, i.e., pixel-wise global accuracy and per-class

accuracy. The global accuracy is defined as

$$\overline{g} = \frac{1}{\sum N_i} \sum_i \sum_{p \in T_i} 1(\phi(p) = s(p), s(p) > 0) \tag{4.7}$$

where T_i is a set of image single index, N_i is the number of ground truth labeled pixels in image i; $s(p)$ and $\phi(p)$ are the ground truth and segmentation label of pixel p, respectively. If $s(p) = 0$, the pixel p is ignored to compute the accuracy. The per-class accuracy is defined as

$$\overline{c_l} = \frac{\sum_i \sum_{p \in T_i} 1(\phi(p) = s(p), s(p) = l)}{\sum_i \sum_{p \in T_i} 1(s(p) = l)} \tag{4.8}$$

In the remainder subsections, we first validate the proposed sparse coding region descriptor for semantic image segmentation, and then compare our approach with the state-of-the-art methods.

4.6.1 Validation of sparse coding

To evaluate the sparse coding for region description, we compare it against the bag-of-visual-words (BOV) model in feature reconstruction error and the performance of semantic segmentation by using any one of them.

The reconstruction error is evaluated by the Mean Square Error (MSE) between the local feature descriptors (SSIM/SIFT) and the basis vector(s) representing these local feature descriptors in the visual dictionary. The MSE is defined as

$$MSE = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{a}_n - \sum_{k \in \Gamma} w_k \mathbf{d}_k\|_2^2 \tag{4.9}$$

where Γ denotes a set of basic vectors used to represent the local feature descriptor $\mathbf{a}_n$, w_k denotes nonzero weight of basic vector $\mathbf{d}_k$. Γ only contains a single basic vector for BOV model. In contrast, Γ typically contains $3 \sim 7$ basic vectors for sparse coding.

Figure 4.3 shows the squared errors for randomly selected SIFT descriptors by using BOV model and the sparse coding (SC) method. Both of the BOV and sparse

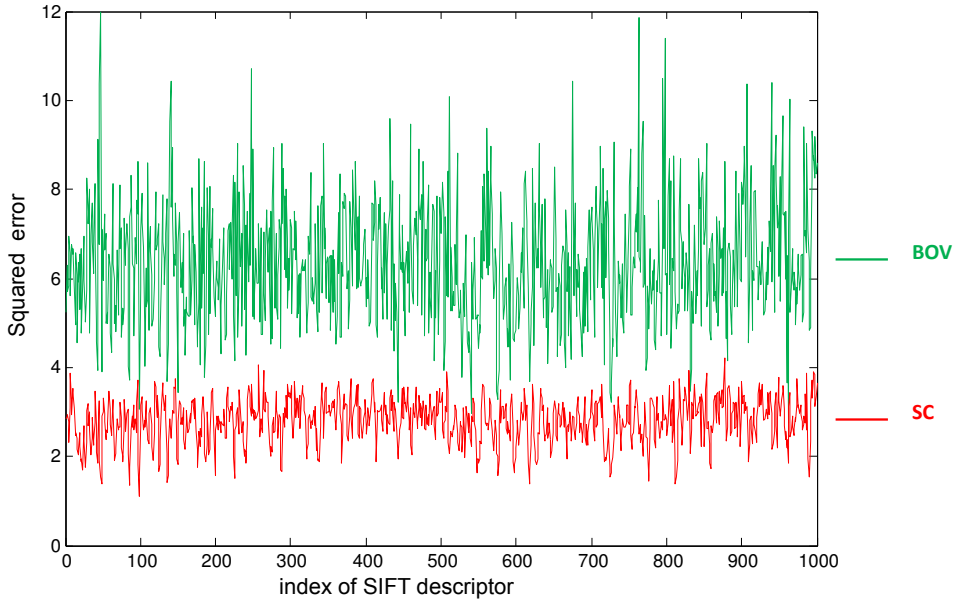


Figure 4.3: Squared errors generated by bag of visual-words (BOV) and sparse coding (SC) for randomly selected SIFT descriptors.

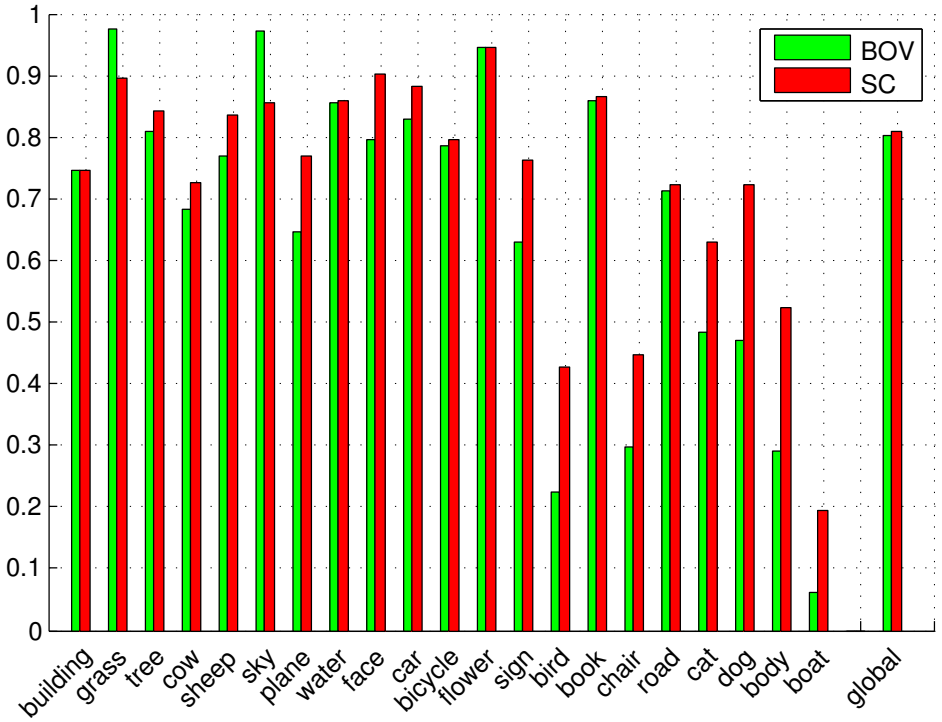


Figure 4.4: Per-class accuracy and global accuracy obtained by using BOV model and sparse coding (SC) method.

coding use a learned visual dictionary containing 2000 basic vectors. Obviously, the sparse coding obtains smaller squared error. We also compute the MSE from the randomly selected samples. The MSE of BOV is 6.4 while it is only 2.6 for sparse coding, i.e., sparse coding decreases 59% MSE than BOV. This suggests that the proposed sparse coding method represents the local descriptors better compared to BOV.

Figure 4.4 compares the semantic segmentation performance of BOV model and sparse coding method in terms of per-class accuracy and global accuracy. Clearly, the sparse coding method substantial outperforms the BOV model. It obtains the better performance for 18 out of 21 object classes, and it increases the global accuracy to 83% and achieves 3% improvement compared to the BOV model. Figure 4.5 presents the confusion matrix of the proposed approach using sparse coding for feature representation.

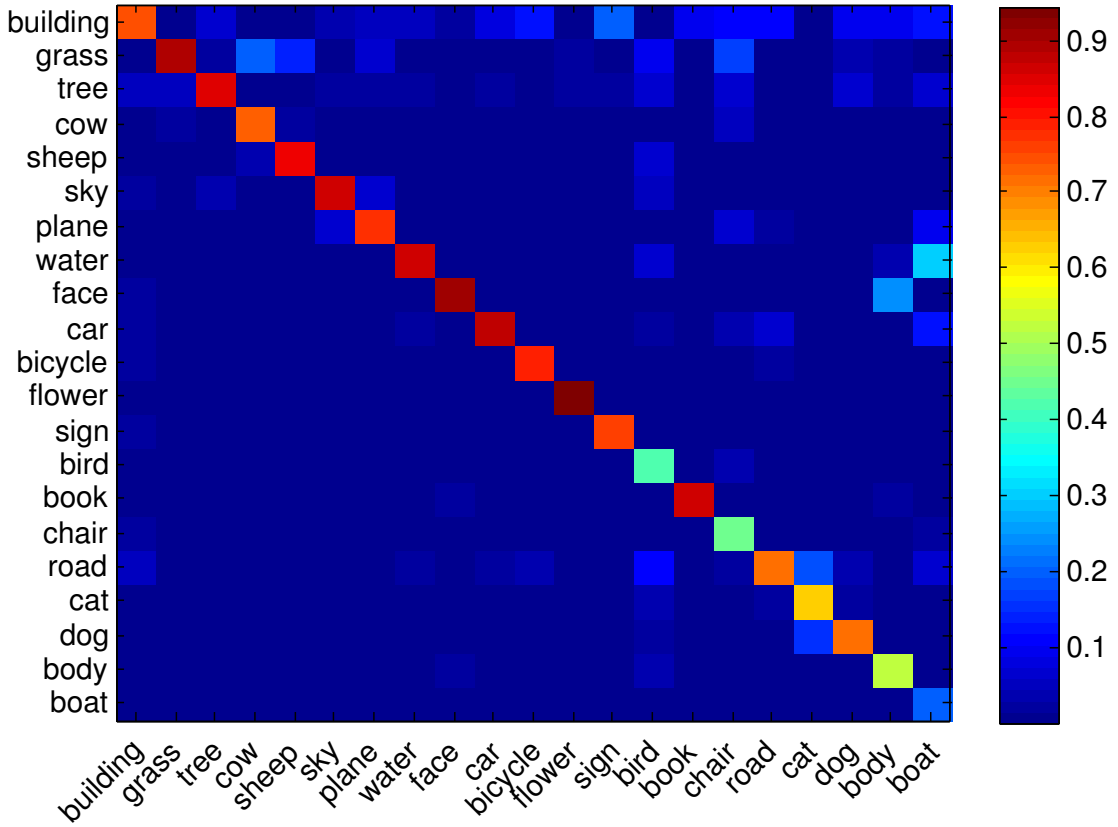


Figure 4.5: Confusion matrix of the proposed sparse-based approach on MSRC-21 dataset.

Table 4.1: Segmentation results (in %) on MSRC-21 dataset.

Class	TB [88]	SF [90]	AC [105]	DAOC [93]	HCRF [106]	Ours
building	49	**84**	30	53	60	74
grass	88	95	71	**97**	78	90
tree	79	81	69	83	77	**84**
cow	**97**	67	68	70	91	72
sheep	**97**	78	64	71	68	83
sky	78	89	84	**98**	88	84
plane	82	72	**88**	75	87	76
water	54	77	58	64	76	**83**
face	87	87	77	74	73	**90**
car	74	71	82	64	77	**89**
bike	72	86	91	88	**93**	80
flower	74	66	90	67	**97**	94
sign	36	59	82	46	73	**76**
bird	24	28	34	32	**57**	43
book	93	85	93	92	**95**	88
chair	51	19	74	61	**81**	46
road	78	68	31	**89**	76	72
cat	75	59	56	59	**81**	63
dog	35	47	54	66	46	**73**
body	**66**	35	56	64	56	53
boat	18	9	**49**	13	46	24
global	72	77	-	78	77	82

4.6.2 Comparison with the sate-of-the-art approaches

We compare the proposed approach with five state-of-the-art methods, i.e.,

- TextonBoost (TB) [88] which incorporates texture, layout, and context information into conditional random field,
- SemanticFisher (SF) [90] which employs Fisher descriptor as an intermediate representation of local features for semantic inference,
- AncestryContext (AC) [105] which models visual context from a hierarchical segmentation tree,
- DAOC [93] which employs a data-assisted output code for semantic classification of object categories,
- HarmonyCRF (HCRF) [106] which integrates a harmony potential representing possible combination of object classes and visual appearance

into conditional random field.

Table 4.1 summarizes the comparison. From this table, we can observe that all approaches consistently achieve good results for those objects showing more uniform appearance, such as the sky and grass, or being regular in appearance, such as books and bikes. However, all approaches tend to fail to segment and recognize birds and boats. The reason for this phenomenon is that there are very few objects of these classes in the dataset, and these objects show a diversity of variations in appearance and scale. For instance, the boat category includes canoe, raft, steamship, sailing ship, yacht, etc., and each of them only have 2-5 examples. Table 4.1 shows that the proposed approach is able to segment and recognize most of objects. It provides more than 70% accuracy for 15 object classes, and more than 80% accuracy for 8 object classes. Moreover, our approach obtains the best performance for 6 out of 21 object classes among the 6 approaches. More importantly, the proposed approach achieves the highest pixel-wise global accuracy which is computed from the whole dataset. Compared to the best one in the reference methods, it obtains 4% improvement in terms of global accuracy and reaches 82%.

Some examples of semantic segmentation results generated by the proposed approach are presented in Figure 4.6 and Figure 4.7 respectively. The proposed approach obtains visually acceptable results for those objects with relatively stable structure, such as the examples in Figure 4.6, and the top five examples in Figure 4.7 . However, it still has difficulties to deal with occlusions and the objects with very small scale in image. For instance, the face in sixth example of Figure 4.7 is segmented and labeled with high correct accuracy; but in last example Figure 4.7 where faces are in small scale and the rightmost one is occluded with a hand, all the faces are not correctly labeled. The main reasons come from two aspects. On one hand, the bottom-up segmentation is unstable (even if using hierarchy might overcome some errors) and its errors might migrate to semantic inference. On the other hand, the extracted features on the small scale objects are not sufficiently representative and lead to a more difficult semantic recognizing.

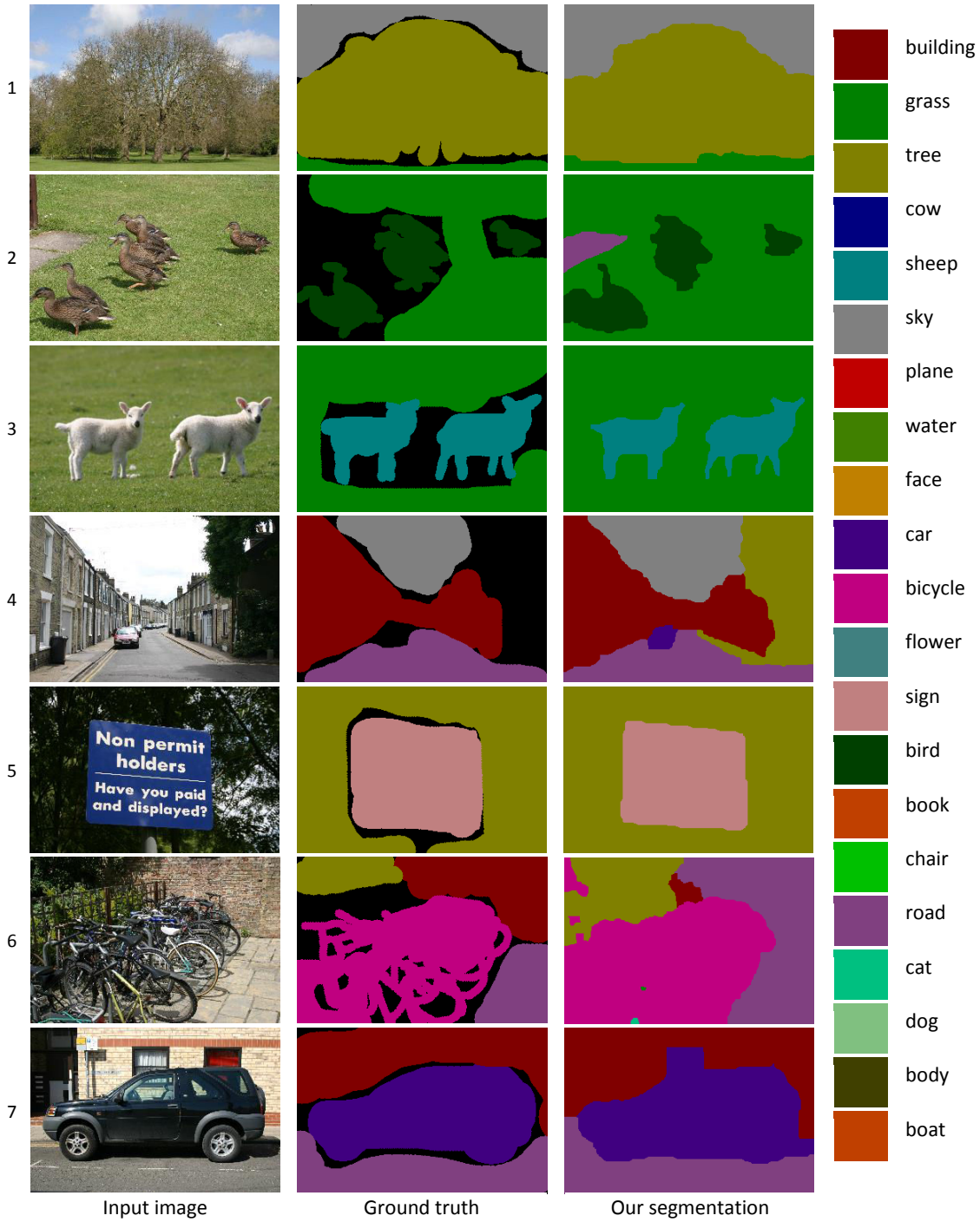


Figure 4.6: Some examples of semantic segmentation results on MSRC-21 dataset.

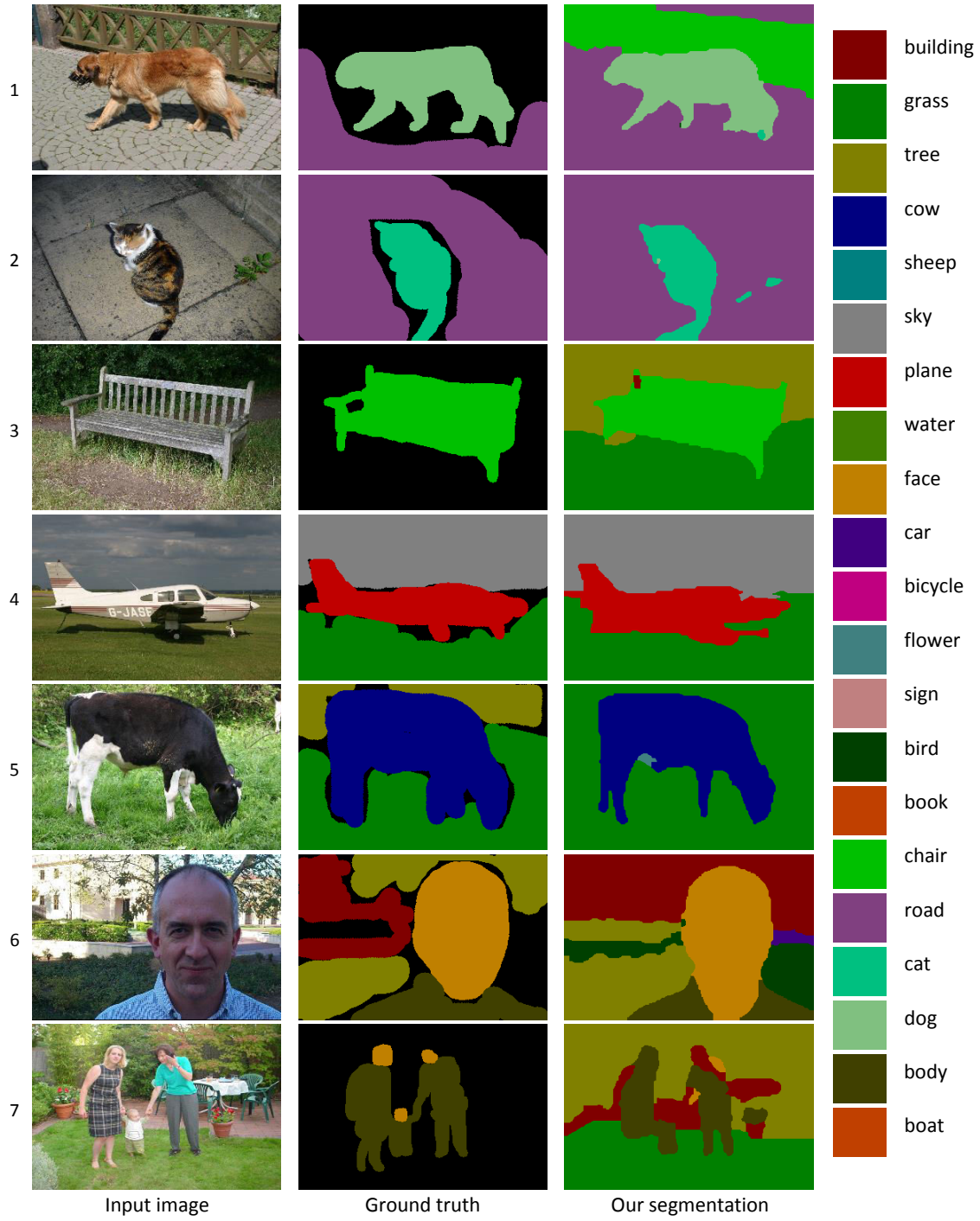


Figure 4.7: Some examples of semantic segmentation results on MSRC-21 dataset.

4.7 Conclusion

In this chapter, a novel approach for semantic image segmentation which aims to assign each pixel in the image with a predefined semantic label. This approach is based on a training region bank (TRB) and a query region bank (QRB), which are generated by a hierarchical segmentation on a set of training images and on the testing image, respectively. For robust region description, we proposed the sparse coding method, which softly represents a local feature descriptor in a region with several basic vectors of the learned visual dictionary and describes all local feature descriptors within the region by a single histogram. Support vector machine with multiple kernel learning is employed for region semantic inference.

The proposed approach is evaluated on the standard dataset for semantic segmentation, which is MSRC dataset consisting of 21 object classes. Experiments demonstrate that, i) compared to the standard bag-of-visual-words model, the sparse coding provides a more accurate representation of local features and leads to higher performance for semantic segmentation, ii) the proposed approach is comparable to the state-of-the-art methods.

5 Conclusion and perspective

This thesis focused on the problems of object segmentation and semantic segmentation which aim at separating objects from background or assigning a specific semantic label to each pixel in an image. We proposed two approaches for the object segmentation and one approach for semantic segmentation.

The first one for object segmentation is based on saliency detection. This approach concentrates on separating salient objects from background. Motivated by our ultimate goal for segmentation, a novel salient object detection model is proposed, which is formulated in the low-rank matrix recovery model by taking the information of image structure derived from a bottom-up segmentation as an important constraint. For the purpose of the performance evaluation of saliency detection, a new dataset consisting of 1500 images with ground-truths is collected also. The segmentation is built within an iterative and interactive optimization framework, which simultaneous performs object segmentation based on the saliency map resulting from saliency detection, and saliency quality boosting based the segmentation. Optimal saliency map and segmentation result are achieved after several iterations. We compared our saliency model and segmentation approach with the state-of-the-art saliency models and saliency-based segmentation algorithms, respectively. Experiments demonstrated that both of them obtain significant improvement over the state-of-the-art approaches.

The second proposed approach for object segmentation is based on exemplar images. This approach aims at segmenting all foreground objects from the background by leveraging a set of available segmented exemplar images. For the

purpose of finding the most matching exemplar images for segmentation, we proposed a novel high-level image representation method called object-oriented descriptor (OOD). OOD captures both global and local information of image, thus it can implicitly encode the objects in the image and represent image geometric structure. Then a foreground/background predictor is learned on-the-fly using the exemplar images retrieved by OOD. Such a predictor assigns a probabilistic score of foreground to each region of the over-segmented input image. After that, the predicted scores are integrated into the segmentation framework of Markov random field (MRF) optimization. Iteratively finding minimum energy of MRF leads the final segmentation. Extensive evaluation across several datasets, including Pascal VOC 2010, Pascal VOC 2011 and iCoseg, demonstrated that, i) the proposed segmentation framework using the typical PHOG for image retrieval already outperforms the state-of-the-art methods, ii) using the proposed OOD representation improves the segmentation performance further, iii) the proposed approach is able to segment large-scale images, e.g. internet images, by only using a small set of segmented exemplar images.

For semantic segmentation, we proposed a new approach which is based on region bank and sparse coding. Region bank is a set of regions generated by multi-level segmentations. This is motivated by the observation that a single-level bottom-up segmentation is hardly to separate objects from background, however, objects might be captured at certain levels in hierarchical segmentation. Therefore, combining multi-level segmentations together may help to improve the performance of semantic segmentation. Once generated the region bank for the input image, we proposed sparse coding method for region description. The sparse coding method represents each local feature descriptor with several basic vectors in the learned visual dictionary, and describes all local feature descriptors within a region by a single histogram. With the sparse coded region bank, support vector machine with multiple kernel learning was employed for semantic inference. We have carried out evaluations on the standard dataset MSRC-21. Experiments demonstrated that, i) the sparse coding produces less quantization errors, compared to the typical bag-of-visual-word model which represents a local feature only by one basic vector

in the dictionary, and this sparse coding yields higher semantic segmentation performance, ii) the proposed approach achieves the state-of-the-art performance.

Some reflections of future works can be derived from the previous summary.

First, it is interesting to validate if the bottom-up saliency detection can be integrated with the exemplar-based object segmentation. Although the proposed OOD image representation method is shown to find more relevant exemplar images for segmentation, it certainly retrieves some failure exemplar images also. When most exemplars are irrelevant to the input image, the performance might dramatically drop down. However, the saliency detection may provide complementary information to localize objects in the image, therefore combining the saliency detection with exemplar-based object segmentation may yield a more robust segmentation model.

Second, it is valuable to verify if the nearest saliency maps can help to improve the quality of saliency map of the input image. Similar images generally share similar object locations, thus, saliency may be boosted by exploiting its nearest neighbors. The proposed OOD image representation method can be considered as the first choice for retrieving a set of nearest neighbors, as it is able to represent both the local objects and the global image structure.

Third, it is worthwhile to further investigate models of Markov random field (MRF) or conditional random field (CRF) for semantic inferring in the proposed semantic segmentation. Context information, which can be partly obtained from the proposed saliency model and the exemplar-based segmentation method, can be considered as an important cue and integrated to the MRF/CRF scheme. As more object-level cues are seamlessly combined, image semantics may be extracted better.

A

Appendix : Résumé étendu français

A.1 Chapitre 1 : Introduction

La segmentation d'image, l'un des problèmes fondamentaux en vision par ordinateur et en traitement d'image, consiste à regrouper des pixels de l'image en différentes partitions, de telle sorte que les pixels au sein d'une même partition ont les mêmes caractéristiques visuelles. Elle a de nombreuses applications, comme le web sémantique, le codage vidéo intelligent, la robotique, l'imagerie médicale et la surveillance militaire.

Selon ses objectifs, les approches existantes peuvent être classées en trois catégories : segmentation basée *région*, segmentation basée *objet* et segmentation *sémantique*. Comme le montre la Figure A.1, la segmentation basée région partitionne l'image en un ensemble de régions homogènes ; la segmentation basée objet, également dénommée segmentation *figure-fond*, vise à séparer du fond les objets, enfin la segmentation sémantique propose d'affecter un label (ou une étiquette) à chaque chaque pixel de l'image décrivant ainsi une catégorie d'objets (chien, visage, eau,...). La segmentation basée région a été largement étudiée depuis plusieurs décennies et un certain nombre d'approches ont été proposées, par exemple : la ligne de partage des eaux [2], les contours actifs [3], les approches basées sur les mean-shift sauts [4] et la segmentation basée sur les graphes [5, 6]. La segmentation basée objet et la segmentation sémantique sont plus difficiles que celle basée région et elles ont été moindrement étudiées. Cette thèse se concentre principalement sur ces deux segmentations soit basée objet, soit sémantique .

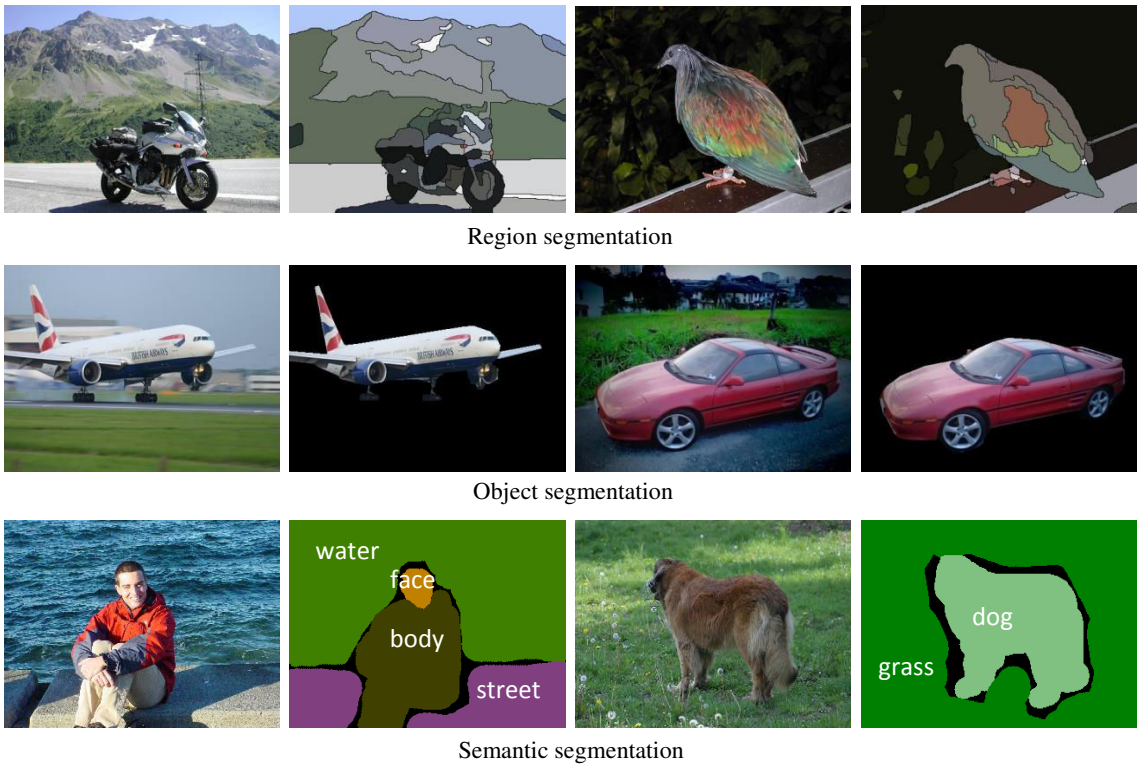


FIGURE A.1 – *Exemples de catégories de segmentation.* En haut : la segmentation basée région fusionne les pixels en régions homogènes. En milieu : la segmentation basée objet extrait du fond les objets de premier plan. En bas : la segmentation sémantique attribue un label à chaque pixel de l'image.

Selon l'apprentissage, sur des images manuellement étiquetées, ou en requérant une intervention humaine ou non, les méthodes de segmentation d'images peuvent également se classer comme segmentation supervisée ou segmentation non supervisée.

En pratique, il est très difficile de réaliser une segmentation totalement non supervisée des objets, puisque la notion d'objet dépend du contexte et de l'application spécifique Par conséquent, nous nous concentrons uniquement sur la segmentation non supervisée des objets saillants, c'est-à-direles objets qui se détachent relativement nettement du fond en modélisant les données de bas niveau de l'image elle-même sans utiliser d'autres indices d'une analyse descendante. Par ailleurs, nous abordons un cas plus difficile dont l'objectif est d'extraire tous les objets de premier-plan dans une image en en tirant profit de l'ensemble des images exemples segmentées manuellement. Comme les objets à segmenter peuvent ne

jamais apparaître dans les images exemples, cette approche peut être considérée comme une approche de segmentation faiblement supervisée. Les deux approches mentionnées ci-dessus produisent un masque de segmentation binaire, où une étiquette indique les objets et l'autre représente l'étiquette de fond.

En outre, nous abordons également le problème de l'affactation d'une étiquette significative (comme chat, chien, voiture ou route) à chaque pixel de l'image, ce qui s'appelle une segmentation sémantique. Dans ce contexte, nous proposons une méthode de représentation des attributs pour établir une passerelle entre caractéristiques locales et sémantique. La segmentation sémantique nécessite un ensemble d'images étiquetées sémantiquement pixel par pixel pour l'apprentissage de l'ensemble des prédicteurs et ainsi chaque pixel d'une image test ne peut se voir attribuer que l'une seule des catégories pré-définies. Une telle approche se classée dans la catégorie de la segmentation supervisée.

Ce résumé en français présente les contenus principaux de la thèse. Il s'organise comme suit : Deux approches proposées pour la segmentation d'objet sont brièvement présentées en Section A.2 et Section A.3, respectivement. Ensuite, notre segmentation sémantique proposée est dans Section A.4. Enfin, les conclusions et perspectives sont présentées dans Section A.5.

A.2 Chapitre 2 : segmentation d'objets basée saillance

Dans ce chapitre, nous nous intéressons à traiter conjointement les problèmes de détection de saillance et de segmentation d'objets saillants en exploitant les indices bénéfiques à chacun d'eux. Pour atteindre cet objectif, nous proposons un système composé de deux éléments clés correspondant également à nos deux contributions principales, à savoir, un modèle de détection de saillance, appelé *segmentation driven low-rank matrix recovery (SLR)* et un système unifié améliorant conjointement la qualité de la carte de saillance et la segmentation des objets du fond.

A.2.1 Modèle de détection de saillance

Le modèle low-rank matrix recovery (LRMR), visant à décomposer une matrice en une matrice de faible rang et une matrice creuse, a montré son potentiel pour résoudre le problème de la détection de saillance, où la matrice de faible rang décomposée correspond naturellement au fond, et la matrice creuse aux objets saillants. Cependant, ceci n'est que dans l'hypothèse d'un fond uniforme et d'objets évidemment distincts. Malheureusement, dans images réelles, le fond peut présenter différents objets de façon éparse et présenter un faible contraste avec les objets. Ainsi, l'application directe du modèle LRMR pour la détection de saillance s'avère d'une robustesse limitée. En conséquence, nous proposons une nouvelle approche qui exploite une segmentation ascendante pour guider la récupération de la matrice.

Un élément clé et distinctif de ce modèle est l'utilisation de la segmentation a priori proposée s'intégrant dans la récupération de la matrice de faible rang. Tout d'abord, si l'on observe les images et leurs segmentations à grains grossiers (coarse-grained, i.e. CG) dans la Figure A.2. Les objets saillants se localisent à différentes positions : centre, bas, gauche, droite ou coin. Les arrière-plans et les objets sont généralement segmentés en plusieurs régions, et ainsi, on n'espère pas de la segmentation ascendante de séparer totalement des objets du fond.

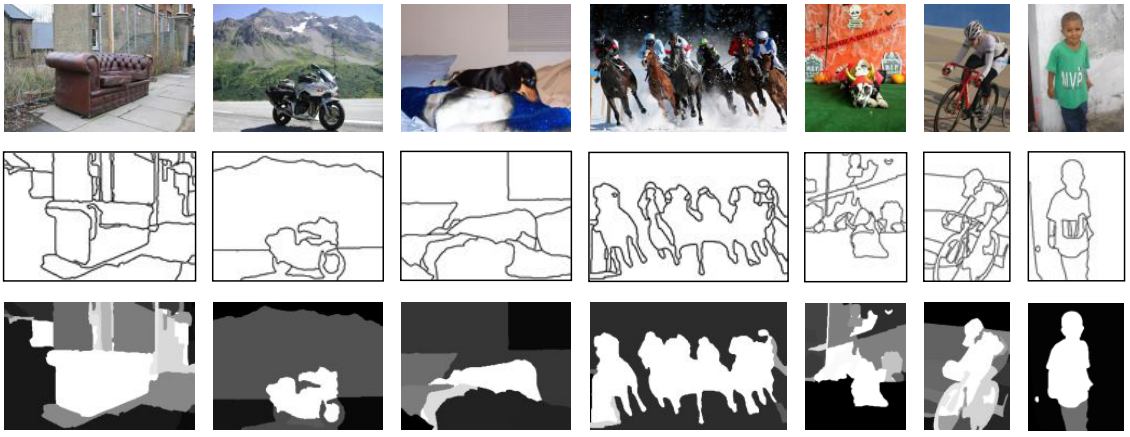


FIGURE A.2 – Exemples de segmentation a priori. Première ligne : images d'entrée ; deuxième ligne : Résultats de segmentation ; dernière ligne : segmentation a priori où un niveau blanc indique un poids plus élevé d'appartenance à un objet et le noir représente un poids inférieur.

Cependant, les régions appartenant au fond et segmentées ont une très forte probabilité d'intersecter le bord de l'image, alors que très peu de régions appartenant aux objets intersectent le bord de l'image. Même si un objet est tronqué sur le bord, comme le vélo et l'enfant dans les deux images les plus à droite, les régions frontalières de l'objet sont petites par rapport à la totalité de l'objet dans l'image. En revanche, les régions frontalières du fond sont généralement de grande taille, car le fond reste plus uniforme, comme le ciel, la route, l'arbre, le mur, etc. Cette observation implique que les objets peuvent être plus ou moins séparés du fond par la segmentation ascendante. En conséquence, nous proposons une segmentation a priori selon la connectivité entre chaque région et le bord de l'image. Soit r_m une région segmentée de l'image I, la segmentation a priori de la région r_m est définie comme

$$h_m = \exp(-\frac{|r_m \cap C|}{\sigma \psi_m}) \tag{A.1}$$

où $|\cdot|$ indique la longueur d'intersection, C est le bord de l'image $\mathbf{I}$, ψ_m est le périmètre extérieur de la région r_m, et σ un paramètre d'ajustement réglé à 0.3 dans nos expériences. Il est clair que, si une région est en contact avec le bord de l'image, sa valeur a priori est dans la plage de (0, 1), sinon elle est égale à 1. En d'autres

termes, la segmentation a priori donne un poids faible à la région en contact avec le bord de l'image. En utilisant l'Eq (A.1), les a priori de segmentation de toutes les régions peuvent être calculés, et ils forment la segmentation a priori de l'image d'entrée.

Nous utilisons la segmentation a priori générée comme guide du modèle LRMR pour la détection de saillance. Supposons qu'une image $\mathbf{I}$ d'entrée soit segmentée en N superpixels, et représentée par une matrice caractéristique $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N]$, et notons $\mathbf{H}_c = [h_1^c, h_2^c, \cdots, h_N^c]$ un ensemble de valeurs de segmentation a priori des superpixels dans la segmentation CG. Afin de récupérer les objets avec le même modèle LRMR, la matrice caractéristique $\mathbf{A}$ est d'une part modulée par la segmentation a priori $\mathbf{H}_c$

$$\mathbf{B} = [h_1^c\mathbf{a}_1, h_2^c\mathbf{a}_2, \cdots, h_N^c\mathbf{a}_N]. \tag{A.2}$$

Ensuite, la matrice caractéristique B modulée est utilisée comme entrée du modèle LRMR

$$\begin{aligned} \min \quad & \|\mathbf{U}\|_* + \lambda\|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{B} = \mathbf{U} + \mathbf{E} \end{aligned} \tag{A.3}$$

où λ est un coefficient de pondération, $\|\cdot\|_*$ indique la norme du noyau de la matrice $\mathbf{U}$ (la somme des valeurs singulières de U), et $\|\cdot\|_1$ la norme l_1 qui assure la production d' une matrice creuse $\mathbf{E}$. Avec la matrice creuse optimale $\mathbf{E}$, la saillance d'un superpixel est donnée par l'énergie selon la norme l_1 des vecteurs correspondants dans $\mathbf{E}$.

Le succès du modèle proposé, c'est-à-dire, le modèle de récupération de matrice de rang faibe guidée par une segmentation (SLR), tient pour deux raisons. D'une part, les images naturelles possèdent généralement une grande redondance dans l'espace des caractéristiques, et les pixels d'objets ont tendance à être saillants par rapport à l'arrière-plan, ce qui permet de trouver les objets de la matrice creuse $\mathbf{E}$. D'autre part, comme la de segmentation a priori attribue un petit poids à la plupart des vecteurs de caractéristiques dans la matrice $\mathbf{B}$ de l'arrière-plan, les énergies, selon la

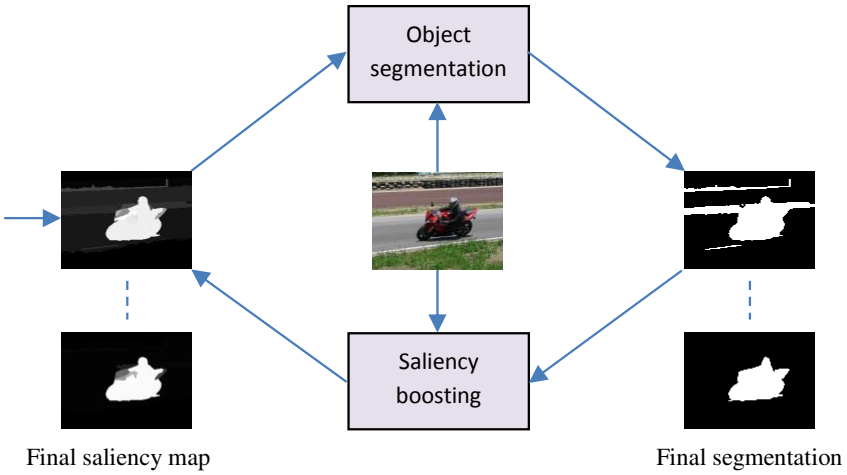


FIGURE A.3 – Schéma unifié pour obtenir conjointement segmentation d'objets et rehaussement de saillance.

norme l_1, des vecteurs correspondants dans la matrice **E** récupérée sont enclines à être faibles. En conséquence, les objets peuvent être récupérés de façon plus efficace à partir de la matrice **E**.

A.2.2 Exploitation conjointe de la segmentation d'objets et saillance rehaussement

La sortie du modèle de détection de saillance est une carte de saillance dans laquelle une valeur réelle de l'intervalle [0, 1] est attribué à chaque pixel pour indiquer sa probabilité d'appartenance à un objet saillant. Ainsi, cette carte de saillance fournit des informations utiles pour la segmentation d'objet. A l'opposé, la segmentation de l'objet peut être utile aussi pour identifier la saillance dans l'image. Si le modèle de segmentation est suffisamment robuste, la saillance peut être rehaussée par une mise en relief des régions des objets segmentés. Par conséquent, nous proposons un schéma unifié pour adresser conjointement rehaussement de saillance et segmentation d'objets.

Comme illustré à la Figure A.3, nous gérons conjointement à la fois la segmentation d'objets et l'optimisation de la saillance comme suit :

1. Proposition d'une solution candidate comme carte de saillance **S** de l'image.

2. A partir de cette saillance **S**, segmentation des objets du fond en utilisant le modèle de segmentation d'objet.
3. Sur la base du résultat de la segmentation, optimisation de la carte de saillance **S** en utilisant le modèle de rehaussement de saillance.
4. Répétition du traitement de l'étape 2 jusqu'à ce que la convergence ou le nombre maximal des itérations soit atteint.

De toute évidence, le modèle de segmentation d'objet et le modèle de rehaussement de saillance dans ce schéma fonctionnent de manière itérative et mutuelle. La segmentation et la carte optimale de saillance sont obtenues lorsque la convergence est atteinte.

Auparavant, existaient aussi des approches exploitant la carte de saillance pour la segmentation d'objets. Cependant, elles ne réutilisaient pas les informations du résultat de segmentation pour re-évaluer la carte de saillance. Aussi introduisons-nous un nouveau modèle de rehaussement de la saillance dans le paragraphe suivant. Pour le modèle de segmentation d'objet, on peut se référer au chapitre 2 du document principal de cette thèse.

Modèle de rehaussement de saillance

Nous supposons que les objets sont au moins en partie extraits par le modèle de segmentation d'objets. Les pixels spatialement à proximité des régions marquées comme saillantesainsi que les pixels similaires aux régions marquées saillantes en doivent être assignés à une valeur de saillance plus élevée, et inversement. Sur la base de cette hypothèse, le modèle de renforcement de la saillance est défini comme

$$\mathbf{S}^* = \mathbf{S} \odot (\mathbf{M} + \mathbf{C}) \tag{A.4}$$

où $\odot$ indique une opération de multiplication élément par élémment, $\mathbf{M}$ est la matrice spatiale a priori et $\mathbf{C}$ la matrice d'apparence a priori.

Étant donné que la carte de saillance **S**, générée par le modèle SLR de détection de saillance, est calculée sur la base d'une segmentation de région, nous calculons aussi l'a priori spatial et l'a priori d'apparence pour chaque région dans la carte

de saillance $\mathbf{S}$, et montons les a priori spatial/apparence de toutes les régions pour former l'a priori spatial/apparence de l'image entière.

Soit $\mathbf{R} = \{r_1, r_2, \cdots, r_K\}$ un ensemble de régions de la segmentation basée région de l'image $\mathbf{X}$, et $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_P\}$ un ensemble d' objets de premier plan séparés et enfin $\mathcal{B}$ l'arrière-plan dans la segmentation résultat $\mathbf{L}$, avec K le nombre de régions de la segmentation basée région, et P le nombre d'objets segmentés. Nous voulons calculer un ensemble d'a priori spatiaux $\mathbf{M} = \{m_1, m_2, \cdots, m_K\}$ et un ensemble d'a priori d'apparence $\mathbf{A} = \{a_1, a_2, \cdots, a_K\}$.

A priori spatiale

L'priori spatial de la région r_k est défini comme

$$m_k = \frac{1}{P} \sum_{p}^{P} \exp\Big\{ - \alpha \cdot \rho \cdot \eta \cdot \mathcal{D}(r_k, \mathcal{O}_p) \Big\} \tag{A.5}$$

où

α est un paramètre d'ajustement constant, et réglée sur 10 dans nos expériences,

$\rho = \frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}} (s_n)$ est la moyenne des valeurs de saillance des régions d'arrière-plan, où $|\cdot|$ indique le nombre d'éléments,

$\eta = \frac{|\mathcal{B}| + \sum_{p=1}^{P} |\mathcal{O}_p|}{\sum_{p=1}^{P} |\mathcal{O}_p|}$ est le rapport de la taille de l'image à la surface totale de tous les objets saillants

$\mathcal{D}(\cdot)$ est une fonction de la distance spatiale.

Nous pouvons observer de l'Eq. (A.5) que l'priori spatial est adaptatif à la qualité de la carte de saillance et à la taille de l'objet. D'une part, la qualité de la carte de saillance est mesurée par ρ. Une valeur faible de ρ signifie une meilleure qualité de carte de saillance comme la plupart des pixels du fond sont affectés avec de faibles valeurs de saillance, ce qui conduit à une plus forte valeur de l'a priori spatial. D'autre part, l'information de taille d'objet est représentée par η. Un forte valeur de η indique de petits objets dans l'image, et la fonction de distance spatiale $\mathcal{D}(\cdot)$ est multipliée par un poids élevé. Ainsi, l'a priori spatial est plus sensible à la distance entre région et centre de gravité objet.

A priori d'apparence

L'a priori d'apparence calcule la similarité entre les régions et les objets segmentés. Pour la représentation de l'apparence, nous utilisons les histogrammes des dans l'espace CIE L* a* b* et de teinte de couleur, où les canaux L*, a*, b* et la teinte sont quantifiés sur 8, 16, 16 et 4 bins, respectivement. Ainsi, chaque région/objet est représenté par un histogramme $(8 \times 16 \times 16 \times 4)$-dimensionnel qui est normalisé à l'unité. Soit $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_P\}$ qui représentent les histogrammes de couleur des objets segmentés $\{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_P\}$, et $\mathbf{h}_k^t$ qui représente l'histogramme de couleur de la région r_k, l'a priori d'apparence non normalisé de r_k est défini comme

$$a'_k = \sum_{p=1}^{P} |\mathcal{O}_p| \cdot \mathcal{K}(\mathbf{h}_k^t, \mathbf{h}_p) \tag{A.6}$$

où $\mathcal{K}(\cdot)$ est une fonction noyau de similarité. Dans notre expérience, le noyau d'intersection des histogrammes est adopté, ainsi

$$\mathcal{K}(\mathbf{h}_k^t, \mathbf{h}_p) = \sum_{i=1}^{T} \min\left\{\mathbf{h}_k^t(i), \mathbf{h}_p(i)\right\} \tag{A.7}$$

où T est la dimension de l'histogramme. En utilisant Eq. (A.6), les a priori d'apparence non-normalisés de toutes les régions sont calculés. Ensuite, l'a priori d'apparence final de la région r_k est calculé par une fonction de normalisation, c'est à dire,

$$a_k = \frac{a'_k}{\max\{a'_1, a'_2, \cdots, a'_K\}}. \tag{A.8}$$

Notons que, dans Eq. (A.6), le noyau $\mathcal{K}(\cdot)$ de similarité est pondéré par la taille de l'objet $|\mathcal{O}_p|$. Cela implique que les objets les plus grands contribuent plus fortement à l'apriori d'apparence quand il y a plusieurs objets. En outre, ceci diminue également considérablement l'impact des erreurs de segmentation dans lesquelles très peu de pixels forment une région et qui sont étiquetées comme objet. Dès lors, la prise en considération de la taille de l'objet améliore la robustesse de l'a priori d'apparence.

A.2.3 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche pour assurer conjointement le problème de la détection de saillance et de la segmentation

d'objets.

Comme première contribution, un nouveau modèle de détection de saillance, appelé segmentation driven low-rank matrix recovery model (SLR), est proposé. L'idée principale de ce modèle est une décomposition d'une matrice de caractéristiques de l'image en une matrice de faible ranget une matrice creuse, et dans laquelle la matrice de faible rang décomposée correspond naturellement à l'arrière-plan, et la matrice creuse aux objets saillants. Pour une amélioration de la robustesse, une segmentation ascendante, appelée segmentation a priori, est définie sur la base de la connectivité des régions avec la frontière de l'image. Cette segmentation est proposée comme unindice de contrainte important pour la décomposition de la matrice et elle a montre une amélioration sensible des performances de détection de saillance.

En second, un schéma unifié est proposé pour conjointement extraireles objets et rehausser la carte de saillance générée par le modèle SLR. D'une part, le modèle de segmentation est basé sur le schéma MRF qui se compose d'un terme de données et d'un terme de lissage. Nous avons proposé un terme de données robuste grâce à l'utilisation optimale de l'information de saillance. D'autre part, le modèle de rehaussement de saillance (saliency boosting, i.e. SB), améliore la qualité de la carte de saillance en tirant efficacement partie de l'emplacement de l'objet et de l'information de l'apparence du résultat de la segmentation. Mutuellement, la segmentation d'objets et l'optimisation de saillance favorisent un meilleur résultat de segmentation et une carte de saillance s de qualité supérieure.

Pour valider la performance de la détection de saillance et la segmentation d'objets, une évaluation approfondie a été menée sur deux ensembles de données d'images, d'une part la base d'images MSRA-B contenant 5000 images et d'autre part la base d'images PASCAL-1500 que nous avons introduit (soit 6500 images au total) Les expériences montrent que : i) le modèle SLR surpasse les modèles de l'état-de-l'-art pour la saillance, ii) le modèle SB amélioreles performances de détection de saillance, iii) l'approche de segmentation proposée est supérieure aux méthodes de l'état-de-l'-art pour la segmentation d'objetsi.

.

A.3 Chapitre 3 : Segmentation d'objet basée sur l'exemple

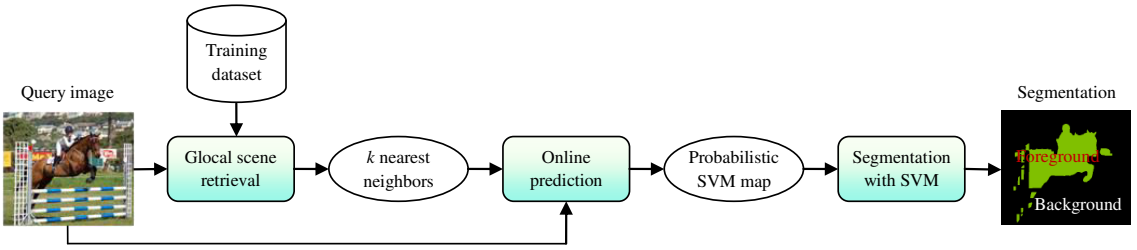


FIGURE A.4 – Schéma générique de transfert glocal en ligne composé de trois principaux modules algorithmiques : récupération glocal de scène, prédiction en ligne et segmentation avec SVM a priori.

Dans ce chapitre, nous nous concentrons sur l'extraction de du fond *tous* les objets de premier plan, ce qui est habituellement désignée segmentation figure-fond. Pour ce faire, nous proposons une nouvelle approche basée sur l'exemple, dénommé exemplaire, nommée transfert glocal en ligne (*online glocal transfer*). L'idée sous-jacente est de transférer, comme similaires à l'image requête, les étiquettes des exemples aux niveaux globalement et localement (glocalement). La Figure A.4 montre ce schéma de l'approche proposée. Il comporte trois principaux modules algorithmiques :

1. *Récupération glocale de scène*, en utilisant un descripteur de haut niveau d'image proposé nommé descripteur orienté objet (OOD), qui permet de récupérer un ensemble d'images voisines les k plus proches glocalement de l'image requête. Dans ces voisins, l'apparence des objets et le contexte spatial de la scène sont tous les deux similaires à ceux de l'image requête.
2. *Prédiction en ligne*, elle prédit la probabilité pour un pixel d'appartenir au premier plan pour l'image de requête. Les images voisines les plus proches récupérés et l'image requête sont sur-segmentées en régions. Un classifieur discriminant machine à vecteurs de support (SVM), ayant appris en ligne à partir des régions récupérées des exemplaires, prédit la probabilité initiale de l'avant plan pour chaque région de l'image requête d'appartenir au premier

plan.

3. *Segmentation avec SVM a priori* produisant la segmentation optimale en combinant la carte des probabilités des SVM, créé par la prédiction en ligne, et le champ de Markov (MRF) d'optimisation de l'énergie.

Le schéma proposé est générique, puisque n'importe quel algorithme s'adaptant aux modules ci-dessus peut être intégré directement substitué dans le schéma. Dans les trois paragraphes suivants, nous présentons les trois modules algorithmiques clés, puis résumons l'approche proposée.

A.3.1 Récupération glocal de scène

Pour récupérer un ensemble de voisins les plus proches glocalement à l'image requêtent, nous nous appuyons sur le nouveau descripteur haut niveau d'image proposé et nommé descripteur orientée objet (OOD).

Pour construire le descripteur OOD, nous générons d'abord un ensemble de pseudo-catégories, dans lequel chaque pseudo-catégorie regroupe des objets partageant ensemble une apparence similaire sans se restreindre à d'une même catégorie réelle. Les pseudo-catégories sont construites à l'aide les images d'apprentissage segmentées manuellement. Plus précisément, les objets dans les images d'apprentissage sont extraits, et chacun d'eux est représenté par un vecteur sac-de-mots visuels- (bag-of-visual-words, i.e. BOV). Les vecteurs BOV de tous ces objets sont rassemblés et classifiés en N sous-ensembles. De toute évidence, les objets dans le même sous-ensemble ont une apparence similaire. Toutefois, cela ne signifie pas qu'ils appartiennent à la même catégorie réelle, puisque des objets intra-catégorie peuvent montrer une forte variation (par exemple, les chaises), et des objets de différentes catégories peuvent être similaires en apparence (par exemple, les chevaux et les vaches). Nous appelons donc ce sous-ensemble pseudo-catégorie. Pour classer les objets, nous utilisons une classification ascendante hiérarchique, dans laquelle chaque objet forme un regroupement (cluster) et des paires de regroupement clusters sont réunies pour former une nouvelle tendance cluster à travers une structure d'arbre hiérarchie. Pour la mesure

de similarité entre les regroupements, nous utilisons la distance χ^2 définie comme

$$\chi^2(\mathbf{f}_i, \mathbf{f}_j) = \sum_{d=1}^{D} \frac{(\mathbf{f}_i(d) - \mathbf{f}_j(d))^2}{\mathbf{f}_i(d) + \mathbf{f}_j(d)} \tag{A.9}$$

où $\mathbf{f}_i$ et $\mathbf{f}_j$ sont des vecteurs BOV d'une paire d'objets, D est la dimension du vecteur BOV. La raison principale du choix de la classification ascendante hiérarchique plutôt que des K-means est le fait que K-means ne supporte pas la distance métrique χ^2 qui convient bien puissante pour regrouper des histogrammes.

Ensuite, N classifieurs SVM des pseudo-catégories font un apprentissage pour calculer les similarités d'une image avec chaque pseudo-catégorie. Les classifieurs SVM sont construits en mettant comme des exemples positifs les en déclarant positifs les exemples d'images contenant des objets d'une pseudo-catégorie spécifique, et les autres comme négatifs. Notons que, puisqu'une image peut contenir des objets de différentes pseudo-catégories, une image peut appartenir à l'exemple positif de plusieurs classifieurs. Avec les classificateurs SVM appris, une image est représentée par un vecteur score $\mathbf{v}_i$ constitué de N scores de classification SVM qui sont généralement dans la plage de $[-3, 3]$. Ces scores de classification expriment naturellement les probabilités pour une image d'appartenir à chacune des N pseudo-catégories, i.e. un score plus élevé indique une plus forte probabilité, et vice versa.

Enfin, le descripteur de haut niveau de l'image est créé en normalisant le vecteur score SVM. Nous normalisons chaque vecteur $\mathbf{v}_i$ en exploitant la distribution des vecteurs de score extraits de toutes les images de l'apprentissage. . Soit $\mathbf{V}_t = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_P\}$ qui un ensemble de vecteurs de score de P image de l'ensemble d'apprentissage. La normalisation est effectuée de la manière suivante

$$\mathbf{h}_i = \frac{\mathbf{v}_i^{'}}{\|\mathbf{v}_i^{'}\|_2} \tag{A.10}$$

où $\|\cdot\|_2$ indique la norme l_2, $\mathbf{v}_i^{'}$ est le vecteur différence entre $\mathbf{v}_i$ et le vecteur moyenne

de tous les vecteurs de scores calculés à partir des images d'apprentissage

$$\mathbf{v}_i^{'} = \mathbf{v}_i - \frac{1}{P}\sum_{p=1}^{P}\mathbf{v}_p \tag{A.11}$$

ici, le vecteur de score normalisé $\mathbf{h}_i$ est nommé descripteur orienté objet (OOD). Le nombre N de classificateurs SVM est déterminé par les distributions d'apparition des objets dans les images d'apprentissage. Si les objets présentent de fortes variations dans l'espace d'apparence, N doit être réglé relativement à une plus grande valeur grande. Dans nos expériences, N est modérément fixé à 40.

Certaines propriétés peuvent être observées sur le descripteur OOD d'OOD :

- Comme l'hyperplan de séparation de SVM est généralement très éparpillé, les classifieurs SVM effectuent simultanément la sélection de caractéristiques et la classification.
- la sélection de caractéristique avec l'agrégation de descripteurs locaux en pyramide spatiale permet au descripteur OOD de capturer l'arrangement géométrique global et aussi de mettre en évidence des objets locaux dans une image.
- avec les classificateurs SVM ayant appris, il est simple de calculer le descripteur OOD de BOV, étant donné que seule une multiplication et une opération de normalisation sont nécessaires.

A.3.2 Prédiction en ligne

L'objectif de ce module est d'initialiser la probabilité de premier plan pour l'image de requête. Puisque les images similaires partagent généralement une segmentation similaire, nous utilisons les k plus proches voisins comme échantillons de référence pour prédire la probabilité d'appartenir au premier plan.

Le classifieur figure-fond fait un apprentissage en ligne en utilisant un ensemble de régions segmentées à partir des k plus proches voisins. Pour l'apprentissage du classifieur, nous employons la machine à vecteurs support à noyaux d'apprentissage multiple (SVM-MKL) [81]. Des exemples positifs pour l'apprentissage sont les régions exemples qui appartiennent principalement aux objets, et des exemples

négatifs sont le reste de ces régions exemples correspondant au fond. Soient $\mathbf{f}_Q = \{\mathbf{f}_q^1, \mathbf{f}_q^2, \cdots, \mathbf{f}_q^U\}$ qui un ensemble de vecteurs BOV d'une région test et $\mathbf{f}_T = \{\mathbf{f}_t^1, \mathbf{f}_t^2, \cdots, \mathbf{f}_t^U\}$ qui une collection de vecteurs BOV d'une région d'apprentissage, où U est le nombre de descripteurs d'apparence. La fonction de classification d'un SVM dans la formulation du noyau est exprimée comme

$$C(\mathbf{f}_Q) = \sum_{n=1}^{N} y_n a_n K(\mathbf{f}_Q, \mathbf{f}_T^n) + b \tag{A.12}$$

où $y_n \in \{+1, -1\}$ indique l'étiquette premier plan/arrière-plan de la région d'apprentissage, N étant le nombre de régions d'apprentissage, et $K(\cdot, \cdot)$ le noyau défini positif, calculé comme une combinaison linéaire de noyaux de caractéristiques

$$K(\mathbf{f}_Q, \mathbf{f}_T^n) = \sum_{u=1}^{U} w_u \Psi(\mathbf{f}_q^u, \mathbf{f}_t^u) \tag{A.13}$$

Les noyaux $\Psi(\cdot, \cdot)$ sont généralement choisis sur la base des expériences. Les noyaux d'histogrammes typiques sont de trois types : linéaires, quasi-linéaires et non-linéaires.

La SVM-MKL apprend une série de coefficients $\mathbf{a} = \{a_1, a_2, \cdots, a_N\}$, un seuil b et un ensemble de poids non négatifs de caractéristiques $\mathbf{w} = \{w_1, w_2, \cdots, w_U\}$. Avec les paramètres appris, chaque région de l'image de requête peut obtenir un score de classification SVM à partir de la fonction de classification (A.12), qui est typiquement dans la plage de $[-3, 3]$. Un tel score de classification SVM lie naturellement à la probabilité d'appartenance d'une région à l'avant-plan. Pour le post-traitement, les scores de classification SVM de toutes les régions sont convertis en valeurs probabiliste en leur lui appliquant une fonction sigmoïde [82]. Ainsi, une carte SVM de l'image requête est générée en affectant les valeurs probabilistes des régions à leurs pixels correspondants.

A.3.3 Segmentation avec SVM a priori

Pour la segmentation, nous utilisons le modèle MRF [39], qui définit un champ de Markov sur les pixels de l'image avec un système de voisinage. Dans un tel modèle,

chaque pixel est associé à une variable aléatoire, qui correspond à l'étiquette de segmentation. La segmentation optimale est obtenue en trouvant le maximum en trouvant la de vraisemblance maximum a posteriori (MAP) dans une MRF et elle est réalisée en minimisant la fonction de l'énergie d'une paire de MRF

$$E(\mathbf{L}) = \sum_{n \in \mathcal{P}} \Lambda_n(l_n) + \sum_{\{n,j\} \in \mathcal{N}} \Theta_{n,j}(l_n, , l_j) \tag{A.14}$$

où $\mathcal{P}$ indique l'ensemble de tous les pixels de l'image, $\mathcal{N}$ correspond au système de voisinage défini sur les pixels, et est choisi en un voisinage à quatre ou à huit connectivités, $\mathbf{L} = \{l_1, l_2, \cdots, l_N\}$ est un ensemble d'étiquettes (variables aléatoires) des) aux pixels, n est indice de l'image, $l_n = \{0, 1\}$ avec 0 indiquant le fond et 1 les objets de premier plan, Λ_n est le terme de données et $\Theta_{n,j}$ est le terme de lissage.

Terme de données

Le terme de données mesure la cohérence entre le pixel et son étiquette, et est généralement défini comme le logarithme négatif de la probabilité d'une étiquette de premier plan/arrière-plan assignée à un pixel, à savoir

$$\Lambda_n(l_n) = -\log(\Omega(\mathbf{x}_n|l_n)) \tag{A.15}$$

où $\mathbf{x}_n \in \mathbb{R}^3$ est le vecteur de caractéristique de couleur, Ω est un modèle d'apparence pour prédire la probabilité de premier plan ou d'arrière-plan en modélisant les distributions de couleurs dans l'image. Cependant, la caractéristique de la couleur n'est pas très discriminante et peut conduire à une segmentation inexacte. Pour surmonter ce problème, nous proposons un nouveau terme de données qui intègre un SVM a priori et un modèle d'apparence

$$\Lambda_n(l_n) = -\log(\Phi(l_n) \cdot \Omega(\mathbf{x}_n|l_n)) \tag{A.16}$$

où le SVM a priori $\Phi(l_n)$ est calculé à partir des scores de classification SVM figure-fond. Compte tenu de la carte probabiliste de SVM $\mathbf{S} = \{s_1, s_2, \cdots, s_N\}, s_n \in \mathbb{R}^1$ de l'image, qui est normalisée à [0, 1], le SVM a priori d'un pixel n pour le modèle de

premier plan est défini comme étant

$$\Phi(l_n = 1) = s_n. \tag{A.17}$$

De la même façon, le SVM a priori du pixel n pour un modèle de fond est défini comme

$$\Phi(l_n = 0) = 1 - s_n. \tag{A.18}$$

Notons que, la SVM de figure-fond est apprise en ligne sur un ensemble des images les plus similaires, le SVM a priori $\Phi(l_n)$ est relie naturellement à chaque pixel au premier plan/arrière-plan de ses voisins les plus proches. Ceci suggère que l'Eq (3.8) favorise les pixels les plus similaires aux objets de premier plan dans les images exemples à pour être étiquetés comme appartenant à l'avant-plan, et par contre incite les autres pixels plus semblables à l'arrière-plan de ces images pour être étiquetés comme de l'arrière-plan.

Le modèle d'apparence est défini par deux modèles de mélange de gaussiennes (GMMs), où l'un est attribué à la modélisation du premier plan et l'autre à la modélisation du fond. Le GMM est une fonction de densité de probabilité paramétrique représentée comme une somme pondérée de densités gaussiennes

$$\Omega(\mathbf{x}_n|\vartheta) = \sum_{i=1}^{Q} w_i g(\mathbf{x}_n|\mu_i, \Sigma_i) \tag{A.19}$$

où Q est le nombre de composantes gaussiennes (typiquement $Q = 5$), w_i est le poids de la composante dans le mélange, avec la contrainte que la somme de tous les poids des composantes soit égale à 1, et $g(\mathbf{x}_n|\mu_i, \Sigma_i)$ est une fonction de densité de probabilité gaussienne

$$g(x_n|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^3|\Sigma_i|}} \exp(\frac{-(x_n - \mu_i)'\Sigma_i^{-1}(x_n - \mu_i)}{2}) \tag{A.20}$$

où $\mu_i \in \mathbb{R}^3$ est le vecteur moyen des vecteurs de données dans la même composante gaussienne, et $\Sigma_i \in \mathbb{R}^{3\times3}$ est la matrice de covariance.

Terme de lissage

Le terme de lissage est défini dans un système de voisinage qui consiste en toutes les paires de pixels adjacents. Son objectif est d'assurer le lissage global de l'étiquette en pénalisant les pixels voisins affectés avec des étiquettes différentes. Comme dans [39, 41], le terme de lissage est défini en fonction de la distance spatiale et du contraste de couleur entre les voisins des pixels

$$\Theta_{n,j}(l_{n,}, l_j) = \frac{\varphi}{\mathrm{dis}(n,j)}[l_n \neq l_j] \exp\{-\beta \|\mathbf{x}_n - \mathbf{x}_j\|_2^2\} \tag{A.21}$$

où $\mathrm{dis}(\cdot)$ est la distance euclidienne spatiale des pixels voisins, $\|\cdot\|_2$ indique la norme l_2. Le paramètre d'ajustement φ est réglé sur 50 ce qui s'est avéré adéquat pour convenir à la plupart des images réelles [83]. La constante β est un poids pour le contraste. Lorsque β est 0, tous les pixels voisins sont lissés avec un degré fixe déterminé par φ. Pour rendre le lissage adaptatif au contraste global des pixels voisins, β est choisi pour être

$$\beta = \frac{1}{2 \cdot \mathrm{mean}((\|\mathbf{x}_n - \mathbf{x}_j\|_2^2) \cdot \mathrm{dis}(n,j))}. \tag{A.22}$$

A.3.4 Conclusion

Nous avons proposé une approche automatique de segmentation automatique d'objet en transférant des étiquettes de segmentation d'exemples similaires glocalement à l'image de requête. Tout d'abord, le descripteur orienté objet (OOD) est proposé pour une représentation de haut niveau de d'image qui code implicitement l'information géométrique et met en évidence les objets dans une image. Ce descripteur permet de trouver efficacement les meilleurs exemplaires pour le transfert de segmentation et conduit à une plus grande précision de la segmentation par rapport à l'utilisation de combinaison les descripteurs GIST et PHOG. Deuxièmement, un nouveau schéma qui combine la prédiction en ligne et l'optimisation de l'énergie d'un champ de Markov est proposé pour améliorer la robustesse des modèles de segmentation et réaliser une segmentation optimale.

Une évaluation approfondie a été réalisée sur trois bases de données, à savoir Pascal VOC 2010, VOC 2011 segmentation et iCoseg. Les expériences montrent que :

(i) l'utilisation du schéma de transfert glocal en ligne avec le descripteur PHOG pour la recherche d'images peut surpasser les techniques de l'état de l'art, (ii) le transfert glocal en ligne avec OOD améliore encore plus la performance, par exemple, par rapport aux meilleurs résultats par transfert de masques de fenêtre [68], la précision de la segmentation en termes des critères F-score augmente de 63,0% à 68,7% sur Pascal VOC 2011 ; (iii) l'approche proposée possède le potentiel pour segmenter sur une grande échelle des images contenant des objets inconnus n'ayant jamais apparu dans les images exemples.

A.4 Chapitre 4 : Segmentation sémantique d'image

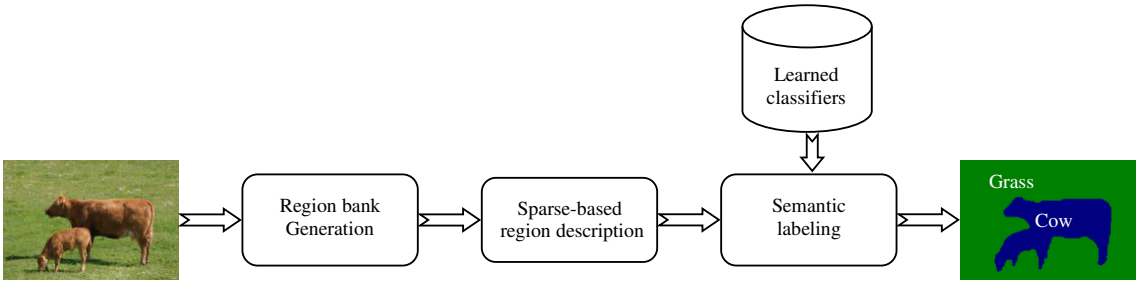


FIGURE A.5 – Schéma proposé pour la segmentation sémantique d'image.

Dans ce chapitre, nous nous intéressons au problème de la segmentation sémantique d'image, qui vise à attribuer une étiquette sémantique, par exemple, "voiture"ou "bâtiment", à chaque pixel dans une image. La Figure A.5 montre le schéma de l'approche proposée, qui se compose de trois modules algorithmiques clés, à savoir, la génération de la banque des régions, la description basée sur la représentation parcimonieuse de la région et l'étiquetage sémantique.

La génération de la banque des régions génère un ensemble de régions à niveaux multiples pour une image d'entrée. La motivation de l'utilisation de régions multi-niveaux est basée sur l'observation que les algorithmes de l'état de l'art de la segmentation sur un seul niveau ont encore de la difficulté à séparer les objets du fond, alors que les objets peuvent être capturés sur certains niveaux.

La description basée sur la représentation parcimonieuse extrait les caractéristiques invariantes locales de chaque région dans la banque des régions, et représente les caractéristiques locales extraites par la représentation parcimonieuse. Alors que de nombreux descripteurs de caractéristiques locales sont disponibles, nous axons notre travail sur une représentation compacte et robuste des descripteurs de caractéristiques locales par un codage parcimonieux, qui représente chaque descripteur de caractéristique locale avec plusieurs vecteurs de base et décrit tous les descripteurs de caractéristiques locales de la même région avec un seul histogramme.

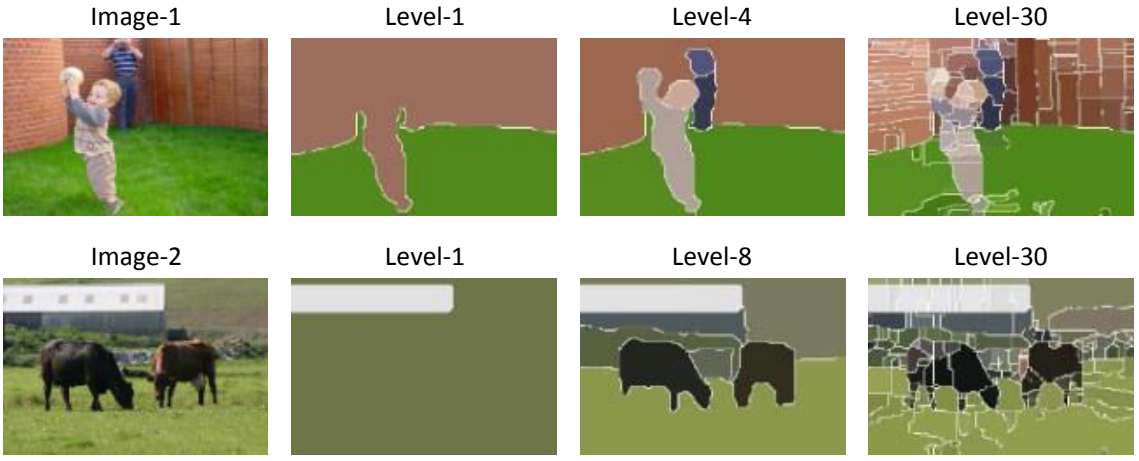


FIGURE A.6 – Deux exemples de segmentations multi-niveaux.

L'étiquetage sémantique attribue, à chaque région dans la banque de régions, une étiquette sémantique prédéfinie et fusionne toutes les régions marquées dans une seule carte d'étiquettes de la même taille que l"image originale. Nous considérons le problème de l'étiquetage sémantique comme une classification en région, qui associe une région avec un ensemble de scores de classification, la fusion sémantique de décisions étant basée sur ces scores et la taille de la région.

A.4.1 Génération de la banque des régions

La banque de régions est un ensemble de régions multi-niveaux. Il y a principalement deux raisons d'utiliser la banque de régions pour la segmentation sémantique. D'une part, la segmentation sur un seul niveau ou sur une sur-segmentation est instable et loin de séparer précisément les objets. Dans la plupart des cas, les objets sont segmentés en de nombreuses régions. D'autre part, la segmentation hiérarchique peut capturer des objets à divers niveaux, mais le niveau de segmentation optimal est imprévisible pour les objets et peut changer selon les composantes de l'image. Comme le montre la Figure A.6, la meilleure segmentation de l'image-1 est au niveau-4, où le visage, les corps, l'herbe et le bâtiment sont quasi parfaitement séparés, tandis que pour l'image-2 le meilleur résultat est au niveau-8, où les vaches, l'herbe et le bâtiment sont segmentés avec très peu de pixels ambigus. Basé sur cette observation, nous tirons parti des

régions multi-niveaux pour une segmentation sémantique.

Pour générer la banque de régions, nous avons choisi la méthode de segmentation hiérarchique basée contour GPB [48] qui génère une sortie comme une carte ultra-métrique estimée (UCM) de contours, dont les valeurs reflètent le contraste entre régions voisines. Les régions hiérarchiques sont créées par seuillage de l'UCM avec un ensemble de seuils. Nous proposons une approche auto-adaptative pour définir la plage des seuils : les seuils minimum et maximum sont calculés en multipliant la valeur maximale de l'UCM par des paramètres prédéfinis α et β. Dans nos expériences, α et β sont fixés à 0,25 et 0,8 respectivement. Les valeurs de l'UCM dans cette plage sont alors utilisées comme des seuils pour générer les régions hiérarchiques. Typiquement, on obtient 5 à 20 seuils par image. L'ensemble des régions généré par GPB pour une image de requête est appelé banque de régions de la requête (query region bank, i.e. QRB) et celui générée à partir de la segmentation GPB et de la vérité terrain pour les images de l'apprentissage est appelé banque de régions d'apprentissage (training region bank, i.e., TRB).

A.4.2 Description de la région basée représentation parcimonieuse

Une fois obtenues les banques de régions, nous visons à décrire chaque région à l'aide d'une représentation compacte et robuste. A cet effet, nous extrayons des caractéristiques locales SIFT et SSIM à partir de pixels pour chaque région, et nous représentons les caractéristiques locales extraites par le codage parcimonieux proposé.

Étant donné qu'une région peut contenir une grande quantité de descripteurs locaux (SIFT/SSIM), le problème restant est de savoir comment représenter ces descripteurs demanière compacte sans perte d'information représentative. Généralement, cela se fait à l'aide d'un modèle de sac-de-mots visuels- (bag-of-visual-words, i.e. BOV), qui apprend d'abord un dictionnaire visuel et représente chaque descripteur de caractéristique locale avec le vecteur le plus

proche dans le dictionnaire en termes d'une mesure de distance prédéfinie. Cependant, le modèle BOV conduit à des erreurs de quantification, car un seul vecteur dans le dictionnaire est utilisé pour représenter un vecteur de caractéristique locale. Pour résoudre ce problème, nous introduisons le codage parcimonieux pour la description de la région.

Soit un ensemble de vecteurs de caractéristiques locales $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ dans $\mathbb{R}^{M\times N}$, nous visons à construire un dictionnaire $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K]$ dans $\mathbb{R}^{M\times K}$, où chaque colonne représente un vecteur de base, et à décrire chaque vecteur de caractéristique locale approximativement comme une combinaison linéaire pondérée de quelques vecteurs de base

$$\begin{aligned} \mathbf{x}_n &\cong \mathbf{D}\mathbf{a}_n \\ \text{such that } \mathbf{a}_n &\trianglerighteq 0, \forall n = 1, 2, \cdots, N \end{aligned} \tag{A.23}$$

où $\mathbf{a}_n$ dans $\mathbb{R}^{K\times 1}$, est un vecteur de pondérations, dans lequel la plupart des entrées sont nulles, $\mathbf{a}_n \trianglerighteq 0$ indique que tous les éléments de $\mathbf{a}_n$ sont non-négatifs. La résolution de ce problème est équivalente à l'optimisation de la fonction de coût

$$\begin{aligned} f(\mathbf{D}, \mathbf{A}) &= \min_{\mathbf{D},\mathbf{A}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{D}\mathbf{a}_n\|_2^2 \\ \text{such that } \mathbf{a}_n &\trianglerighteq 0, \forall n = 1, 2, \cdots, N \end{aligned} \tag{A.24}$$

où $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_K]$ dans $\mathbb{R}^{K\times N}$, $\|\cdot\|_2$ est la norme l_2. Pour ce faire, nous appliquons le codage parcimonieuxavec contrainte positivité [103] à l'Eq (A.24).

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{A}} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{D}\mathbf{a}_n\|_2^2 + \lambda\|\mathbf{a}_n\|_1 \\ \text{such that } &\|d_k\|_2 \leq 1, \forall k = 1, \cdots, K, \mathbf{a}_n \trianglerighteq 0, \forall n = 1, 2, \cdots, N \end{aligned} \tag{A.25}$$

où λ est un paramètre de régularisation. La régularisation l_1 assure de générer des coefficients parcimonieuxpour a_n [104]. La contrainte de la norme l_2 du vecteur $\mathbf{d}_k$ inférieure ou égale à l'unité vise à prévenir $\mathbf{D}$ de prendre des valeurs arbitrairement grandes qui entraîneraient arbitrairement de petites valeurs dans $\mathbf{A}$. Le dictionnaire $\mathbf{D}$ est obtenu en minimisant l'Eq (A.25) par rapport à $\mathbf{D}$ et $\mathbf{A}$ (c'est à dire en minimisant alternativement selon une matrice tout en maintenant l'autre

fixe). Une fois le dictionnaire **D** construit, le vecteur parcimonieuxde coefficients peut être calculé en minimisant l'Eq (A.25) par rapport à **A**. En conséquence, chaque descripteur de caractéristique x_n peut être approximé en multipliant le dictionnaire **D** et un vecteur parcimonieux de coefficients a_n. En d'autres termes, le codage parcimonieux représente un vecteur de caractéristique locale avec une combinaison linéaire de quelques vecteurs de base. Nous avons comparé les performances de reconstruction des méthodes BOV et du codage parcimonieux. Le codage parcimonieux diminue l'erreur quadratique moyenne (MSE) de 6,4 à 2,6, ce qui correspond à une réduction de 59%, en cas de reconstruction de la caractéristique SIFT avec un dictionnaire contenant 2000 vecteurs de base (voir la Section 4.6).

Pour la représentation de caractéristiques compactes, un sous-ensemble de vecteurs de caractéristiques locales est choisi au hasard pour former les dictionnaires parcimonieux de SIFT et SSIM avec 2000 et 800 vecteurs de base respectivement (ces valeurs sont déterminées expérimentalement). Ensuite, les dictionnaires sont utilisés pour calculer les vecteurs parcimonieux des régions.

A.4.3 Etiquetage sémantique

Pour générer une carte de segmentation sémantique dans laquelle chaque pixel est affecté d'une étiquette sémantique prédéfinie, nous associons d'abord chaque région avec un score de similarité des catégories sémantiques prédéfinies, puis générons la carte de segmentation sémantique en fusionnant les régions marquées.

Marquer de la Région

Nous classons maintenant les régions codées en classes d'objets pertinents. Théoriquement, n'importe quel classifieur discriminant peut être appliqué pour cette tâche. Dans cette étude, nous préférons la machine à support vecteur (SVM) avec le multiplenoyau d'apprentissage (multiple kernel learning MKL) [81], comme il est facile de former des classifieurs intégrant plusieurs types de caractéristiques, même si ces caractéristiques sont adressées par différents noyaux.

Pour la classification, nous calculons tout d'abord l'histogramme normalisé de

vecteurs parcimonieux pour chaque région

$$h_i = \frac{1}{J_i} \sum_{j=1}^{J_i} a_j \tag{A.26}$$

où a_j dénote les vecteurs parcimonieux dans la région R_i, J_i désigne la dimension du vecteur parcimonieux.

En utilisant l'Eq (A.26), nous pouvons calculer l'histogramme des vecteurs parcimonieux SIFT qui est noté $\mathbf{h}_i^t$, et celui des vecteurs parcimonieux de SSIM qui est noté $\mathbf{h}_i^m$. Soit $\mathbf{h}_i^c = \{\mathbf{h}_i^t, \mathbf{h}_i^m\}$ défini comme la combinaison d'histogrammes des vecteurs parcimonieux. De cette façon la fonction de classification d'une SVM dans la formulation du noyau est exprimée comme suit :

$$SVM(h^c) = \sum_{i=1}^{I} y_i a_i K(h^c, h_i^c) + b \tag{A.27}$$

où h_c est l'histogramme des vecteurs parcimonieux dans une région de test ; $\{h_i^c \; \forall i = 1, \cdots, I\}$ sont des histogrammes de vecteurs parcimonieux dans les régions d'apprentissage ; $y_i \in \{+1, -1\}$ indique l'étiquette de la classe, et K est le noyau défini positif, qui est calculé comme une combinaison linéaire des noyaux d'histogramme

$$K(h^c, h_i^c) = d_t K(h^t, h_i^t) + d_m K(h^m, h_i^m) \tag{A.28}$$

où d_t et d_m représentent les poids non négatifs de noyaux. De nombreux noyaux peuvent être appliqués pour la classification basée sur un histogramme, comme le noyau d'intersection, noyaux Chi2 et RBF. Dans nos expériences, le noyau Chi2 est utilisé pour les deux histogrammes des EIPD et SSIM. MKL apprend les poids du noyau, d_t et d_m et les paramètres a_i, et b pour chaque classe. En utilisant l'Eq (A.27), une région test peut obtenir un score de SVM, indiquant la vraisemblance de la classe de l'objet, à partir de chaque classificateur. Ces scores sont ensuite utilisés pour l'étiquetage des régions.

Etiquetage des région

L'approche la plus directe pour l'étiquetage des régions marquées d'un score d'une image de test est d'affecter ces régions avec les étiquettes de classe les plus

probables. Cependant,celal ne peut pas être directement appliqué à notre méthode, car les régions hiérarchiques sont superposées ou croisées entre elles, de plus, ces régions générées par un seuillage grossier peuvent couvrir plusieurs objets. Notre solution est de combiner l'effet des scores SVM avec celui des tailles de régions.

Le procédé d'étiquetage est principalement constitué de trois étapes. Tout d'abord, les classes d'objets les plus probables qui ont les scores de SVM maximales sont utilisées pour pré-étiqueter chaque région. Deuxièmement, ces régions sont triées selon leurs scores croissants. Enfin, les régions sont fusionnées progressivement, à partir des scores les plus faibles, pour former une image complètement étiquetée par l'observation de leur taille etdes scores SVM. Ainsi, quand une région candidate R_j, ou une partie, se localise à la même position que la région marquée R_i, elle ne peut remplacer celle-ci que si son score est supérieur à un seuil donné et si sa taille n'est pas beaucoup plus grande que R_i. Cette stratégie permet d'éviter l'étiquetage de petits objets par leur environnement ou par de gros objets voisins.

A.4.4 Conclusion

Dans ce chapitre, une nouvelle approche pour la segmentation sémantique de l'image qui vise à attribuer à chaque pixel une étiquette sémantique prédéfinie. Cette approche est basée sur une banque de régions d'apprentissage (TRB) et une banque de régions de la requête (QRB), qui sont respectivement générées par une segmentation hiérarchique sur un ensemble d'images d'apprentissage et sur l'image de test. Pour une robuste description de région, nous avons proposé la méthode de codage parcimonieux, ce qui représente pour un traitement un descripteur de caractéristique locale dans une région avec plusieurs vecteurs de base du dictionnaire visuel appris et décrit tous les descripteurs de caractéristiques locales intérieurs à la région par un seul histogramme. La machine à support de vecteurs avec apprentissage multiple des noyaux est utilisé pour l'inférence sémantique.

L'approche proposée est évaluée sur une base de données standard pour la segmentation sémantique, et qui est la base MSRC composée de 21 classes d'objets. Les expériences montrent que, i) par rapport au modèle de sac de mots visuels, le

codage parcimonieux fournit une représentation plus précise des caractéristiques locales et conduit à de meilleures performances pour la segmentation sémantique, ii) l'approche proposée est comparable aux méthodes de l'état-de-l '-art.

A.5 Chapitre 5 : Conclusion et perspective

Cette thèse a porté sur les problèmes de segmentation d'objets et la segmentation sémantique qui visent soit à séparer des objets du fond, soit à l'attribution d'une étiquette sémantique spécifique à chaque pixel de l'image. Nous proposons deux approches pour la segmentation d'objets, et une approche pour la segmentation sémantique.

Concernant la segmentation d'objets, la première approche est basée sur la détection de saillance. Cette approche se concentre sur la séparation des objets saillants de fond. Motivés par notre but ultime pour la segmentation, un nouveau modèle de détection de saillance est proposé, qui est formulée dans le modèle de récupération de la matrice de rang faible (low-rank matrix recovery) en prenant les informations de structure d'image provenant d'une segmentation ascendante comme une contrainte importante. Aux fins de l'évaluation de la performance de la détection de saillance, un nouvel ensemble de données constitué de 1500 images avec des vérités-terrain ont été recueillies également. La segmentation est construite au sein d'un schéma d'optimisation itératif et mutuelle, qui effectue simultanément segmentation d'objets basée sur la carte de saillance résultant de la détection de saillance, et l'amélioration de la qualité de saillance base de la segmentation. La carte de saillance optimale et le résultat de la segmentation finalement sont obtenus après plusieurs itérations. Nous avons comparé notre modèle de saillance et l'approche de segmentation avec les modèles de l'état de l'art sur la saillance et des algorithmes de segmentation basée saillance, respectivement. Des expériences ont montré que tous les deux obtiennent une amélioration significative par rapport aux approches de l'état de l'art.

La deuxième approche proposée pour la segmentation d'objets est basée sur des images exemples. Cette approche met l'accent sur la segmentation de tous les objets au premier plan de l'arrière-plan en s'appuyant sur un ensemble d'images exemples segmentées. Dans le but de trouver les exemples les plus assortis pour l'image de requête, nous avons proposé une nouvelle méthode de représentation d'image de haut niveau qui est appelé descripteur orientée objet (OOD). OOD

capture des informations à la fois globale et locale de l'image ; il peut donc implicitement décrire les objets dans l'image et représenter la structure géométrique de l'image. Puis un prédicteur de premier plan/arrière-plan est appris en ligne en utilisant les exemples récupérés par OOD. Ce prédicteur attribue un score probabiliste de premier plan à chaque région de l'image d'entrée. Après cela, les scores prédits sont intégrés dans le schéma de segmentation du champ de Markov (MRF) d'optimisation. Trouver itérativement l'énergie minimum de MRF mène la segmentation finale. Une évaluation approfondie à travers plusieurs ensembles de données, y compris Pascal VOC 2010, Pascal VOC 2011 et iCoseg, a démontré que, i) le schéma de segmentation proposé, en utilisant la PHOG typique pour la récupération d'images surpasse déjà les méthodes de l'étatde l'art, ii) en utilisant le descripteur OOD proposé améliore encore les performances de segmentation, iii) l'approche proposée est capable de segmenter les images à grande échelle, par exemple les images sur Internet, en utilisant seulement un petit ensemble d'exemples segmentés.

Pour la segmentation sémantique, nous avons proposé une nouvelle approche qui se fonde sur la banque de régions et la représentation parcimonieuse. La banque des régions est un ensemble de régions générées par segmentations multi-niveaux. Ceci est motivé par l'observation que la segmentation à un seul niveau éprouve des difficultés à séparer les objets distincts de fond ; cependant, les objets peuvent être capturés à certains niveaux dans la segmentation hiérarchique. Par conséquent, la combinaison des segmentations multi-niveaux peut aider à améliorer la performance de la segmentation sémantique. Après avoir générer la banque des régions de l'image d'entrée, nous avons proposé la méthode de codage parcimonieux pour la description de région. Le codage parcimonieux représente chaque descripteur de caractéristique locale avec plusieurs vecteurs de base dans le dictionnaire visuel appris, et décrit tous les descripteurs de caractéristiques locales dans une région à l'aide d'un seul histogramme. La machine à support de vecteurs (SVM) avec l'apprentissage de noyaux multiple est utilisée pour l'inférence sémantique. Nous avons effectué des évaluations sur l'ensemble de données norme MSRC-21. Des expériences ont démontré que, i) le codage parcimonieux produit

moins d'erreurs de quantification, par rapport au modèle sac-de-mots-visuels- qui ne représente une caractéristique locale que par un seul vecteur de base dans le dictionnaire, et ce codage parcimonieux donne des performances supérieures de segmentation sémantique, ii) l'approche proposée permet d'atteindre les performances de l'état de l'art.

Quelques réflexions de travaux futurs peuvent être dérivées du résumé ci-dessus.

Tout d'abord, il est intéressant de valider si la détection de saillance peut être intégrée à la segmentation d'objets basé sur l'exemple. Bien que le descripteur OOD proposé se soit montré capable de trouver des images exemples les plus pertinentes pour la segmentation, il récupère certainement des images exemples erronés aussi. Quand la plupart des exemples sont sans rapport avec l'image d'entrée, la performance pourrait considérablement glisser vers le bas. Cependant, la détection de saillance peut fournir des informations complémentaires pour localiser des objets dans l'image, en combinant par conséquent la détection de saillance avec la segmentation d'objets basée sur l'exemple peut produire un modèle de segmentation plus robuste.

Deuxièmement, il est précieux de vérifier si les cartes de saillance les plus proches peuvent aider à améliorer la qualité de la carte de saillance de l'image d'entrée. Les images similaires partagent généralement des endroits d'objets similaires, ainsi, la saillance peut être améliorée en exploitant ses voisins les plus proches. La méthode de représentation de l'image par OOD proposé peut être considérée comme le premier choix pour la récupération d'un ensemble de voisins les plus proches, comme il est capable de représenter à la fois les objets locaux et la structure globale de l'image.

Troisièmement, il est utile d'étudier davantage les modèles des champs de Markov aléatoire (MRF) ou champ aléatoire conditionnel (CRF) pour l'inférence sémantique dans la segmentation sémantique proposée. Les informations de contexte, qui peut être en partie obtenues à partir du modèle de saillance proposé et la méthode de segmentation basée exemple, peut être considérée comme une indication importante et intégrée au schéma MRF/CRF. Puisque les indices de niveau objet peuvent être combinés positivement, les sémantiques de l'image pourront être mieux extraites.

List of Figures

List of Tables

List of Publications

Journal Paper

[1] **Wenbin Zou**, Cong Bai, Kidiyo Kpalma and Joseph Ronsin, "Online glocal transfer for automatic figure-ground segmentation,"*IEEE Trans. on Image Processing*, [http://dx.doi.org/10.1109/TIP.2014.2312287].

[2] Zhi Liu*, **Wenbin Zou*** and Olivier Le Meur "Saliency tree: a novel saliency detection framework,"*IEEE Trans. on Image Processing*, [http://dx.doi.org/10.1109/TIP.2014.2307434]. *Contributed equally to this paper.

[3] **Wenbin Zou**, Zhi Liu, Kidiyo Kpalma, Joseph Ronsin and Yong Zhao, "Unsupervised joint saliency detection and object segmentation,"*IEEE Trans. on Image Processing*, under review.

[4] Cong Bai, Weizhi Lu, **Wenbin Zou**, Kidiyo Kpalma and Joseph Ronsin, "Color texture retrieval using sparse representation based histogram,"*IEEE Trans. on Multimedia*, to be submitted.

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

[6] **Wenbin Zou**, Kidiyo Kpalma and Joseph Ronsin, "Automatic foreground extraction via joint CRF and online learning,"*Electronics Letters*, 49(18), 1140-1142, 2013.

[7] Cong Bai, **Wenbin Zou**, Kidiyo Kpalma and Joseph Ronsin, "Efficient colour texture image retrieval by combination of colour and texture features in wavelet domain,"*Electronics Letters*, 48(23), 1463-1465, 2012.

Conference Paper

[1] **Wenbin Zou**, Kidiyo Kpalma, Zhi Liu and Joseph Ronsin, "Segmentation driven low-rank matrix recovery for saliency detection", in *Proc. British Machine Vision Conference (BMVC)*, Bristol, UK, Sep. 2013.

[2] **Wenbin Zou**, Kidiyo Kpalma and Joseph Ronsin, "Semantic image segmentation using region bank", in *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, Tsukuba, Japan, Nov. 2012.

[3] **Wenbin Zou**, Kidiyo Kpalma and Joseph Ronsin, "Semantic segmentation via sparse coding over hierarchical regions", in *Proc. IEEE International Conference on Image Processing (ICIP)*, Orlando, USA, Sep. 2012.

Bibliography

[1] InfoTrends, "U.s. image sharing market forecast: 2010-2015," 2011.

[2] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[3] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, Feb. 1997.

[4] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603 –619, may 2002.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888 –905, aug 2000.

[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[7] W. Zou, K. Kpalma, Z. Liu, and J. Ronsin, "Segmentation driven low-rank matrix recovery for saliency detection," in *Proc. British Machine Vision Conference (BMVC)*, 2013.

[8] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of Intelligence*, pp. 115–141, Springer, 1987.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254 –1259, 1998.

[10] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[11] D. Gao, V. Mahadevan, and N. Vasconcelos, "The discriminant center-surround hypothesis for bottom-up saliency," *Advances in neural information processing systems*, vol. 20, pp. 1–8, 2007.

[12] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] M.-M. Cheng, G.-X. Zhang, N. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

[14] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010.

[15] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior," in *Proc. British Machine Vision Conference (BMVC)*, 2011.

[16] Z. Liu, O. Le Meur, S. Luo, and L. Shen, "Saliency detection using regional histograms," *Optics letters*, vol. 38, no. 5, pp. 700–702, 2013.

[17] N. Bruce and J. Tsotsos, "Saliency based on information maximization," *Advances in neural information processing systems*, vol. 18, p. 155, 2006.

[18] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics," *Journal of Vision*, vol. 8, no. 7, 2008.

[19] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems 19*, pp. 545–552, MIT Press, 2007.

[20] T. Avraham and M. Lindenbaum, "Esaliency (extended saliency): Meaningful attention using stochastic image modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 693–708, 2010.

[21] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[22] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," 2013.

[23] Z. Liu, R. Shi, L. Shen, Y. Xue, K. N. Ngan, and Z. Zhang, "Unsupervised salient object segmentation based on kernel density estimation and two-phase graph cut," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1275–1289, 2012.

[24] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Salient region detection by modeling distributions of color and orientation," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 892–905, 2009.

[25] W. Zhang, Q. J. Wu, G. Wang, and H. Yin, "An adaptive computational model for salient object detection," *IEEE Transactions on Multimedia*, vol. 12, no. 4, pp. 300–316, 2010.

[26] Y. Xie, H. Lu, and M. Yang, "Bayesian saliency via low and mid level cues," *IEEE Transactions on Image Processing*, vol. 34, no. 11, pp. 1689–1698, 2013.

[27] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007.

[28] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008.

[29] C. Jung and C. Kim, "A unified spectral-domain approach for saliency detection and its application to automatic object segmentation," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1272–1283, 2012.

[30] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 733–740, 2012.

[31] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[32] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 29–42, Springer, 2012.

[33] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 914–921, 2011.

[34] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *Proc. European Conference on Computer Vision (ECCV)*, 2012.

[35] J. Yan, M. Zhu, H. Liu, and Y. Liu, "Visual saliency detection via sparsity pursuit," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 739–742, 2010.

[36] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] B. C. Ko and J.-Y. Nam, "Object-of-interest image segmentation based on human attention and semantic region clustering," *JOSA A*, vol. 23, no. 10, pp. 2462–2470, 2006.

[38] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 141–145, 2006.

[39] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[40] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 366–379, 2010.

[41] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics*, vol. 23, pp. 309–314, 2004.

[42] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *arXiv preprint arXiv:0912.3599*, 2009.

[43] C. Lang, G. Liu, J. Yu, and S. Yan, "Saliency detection by multitask sparsity pursuit," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1327 –1338, 2012.

[44] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 277–284, 2009.

[45] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[46] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: a flexible architecture for multi-scale derivative computation," in *Proc. IEEE International Conference Image Processing*, 1995.

[47] H. G. Feichtinger, *Gabor Analysis and Algorithms: Theory and Applications*. Birkhauser Boston, 1997.

[48] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[49] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[50] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 73–80, 2010.

[51] R. Achanta and S. Susstrunk, "Saliency detection using maximum symmetric surround," in *Proc. IEEE international Conference on Image Processing (ICIP)*, pp. 2653–2656, IEEE, 2010.

[52] K. E. A. Van de Sande, J. R. R. Uijlings, T. Gevers, and A. Smeulders, "Segmentation as selective search for object recognition," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1879–1886, 2011.

[53] J.-J. Chen, C.-R. Su, W. L. Grimson, J.-L. Liu, and D.-H. Shiue, "Object segmentation of database images by dual multiscale morphological reconstructions and retrieval applications," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 828–843, 2012.

[54] Q. Zhang and K. N. Ngan, "Segmentation and tracking multiple objects under occlusion from multiview video," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3308–3313, 2011.

[55] P.-F. Chen, H. Krim, and O. L. Mendoza, "Multiphase joint segmentation-registration and object tracking for layered images," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1706–1719, 2010.

[56] J. Xue, C. Li, and N. Zheng, "Proto-object based rate control for jpeg2000: an approach to content-based scalability," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 1177–1184, 2011.

[57] L. Bertelli, T. Yu, D. Vu, and B. Gokturk, "Kernelized structural svm learning for supervised object segmentation," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 2153–2160, 2011.

[58] B. Alexe, T. Deselaers, and V. Ferrari, "Classcut for unsupervised class segmentation," in *Proc. European Conference on Computer Vision (ECCV)*, 2010.

[59] N. Jojic, A. Perina, M. Cristani, V. Murino, and B. Frey, "Stel component analysis: Modeling spatial correlations in image class structure," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 2044–2051, 2009.

[60] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detection and image interpretation," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 43–56, 2008.

[61] J. Winn and N. Jojic, "Locus: learning object classes with unsupervised segmentation," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 756 – 763, oct. 2005.

[62] E. Borenstein and S. Ullman, "Class-specific, top-down segmentation," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 639–641, 2002.

[63] D. Kuettel, M. Guillaumin, and V. Ferrari, "Segmentation propagation in imagenet," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 459–473, 2012.

[64] J. Carreira and C. Sminchisescu, "Cpmc: Automatic object segmentation using constrained parametric min-cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1312–1328, 2012.

[65] I. Endres and D. Hoiem, "Category independent object proposals," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 575–588, 2010.

[66] W. Zou, K. Kpalma, and J. Ronsin, "Semantic segmentation via sparse coding over hierarchical regions," in *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 2577–2580, 2012.

[67] A. Rosenfeld and D. Weinshall, "Extracting foreground masks towards object recognition," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1371–1378, 2011.

[68] D. Kuettel and V. Ferrari, "Figure-ground segmentation by transferring window masks," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 558–565, 2012.

[69] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[70] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in *Proc. ACM International Conference on Image and Video Retrieval*, 2009.

[71] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 2126–2136, 2006.

[72] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[73] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176, 2010.

[74] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1–8, 2007.

[75] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1150–1157, 1999.

[76] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[77] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 2169–2178, 2006.

[78] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE international Conference on Vision and Pattern Recognition (CVPR)*, pp. 1778–1785, 2009.

[79] D. Parikh and K. Grauman, "Relative attributes," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 503–510, 2011.

[80] S. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, pp. 241–254, 1967.

[81] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proc. International Conference on Computer Vision (ICCV)*, pp. 1–8, 2007.

[82] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[83] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *Proc. European Conference on Computer Vision (ECCV)*, vol. 3021, pp. 428–441, 2004.

[84] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[85] B. Catanzaro, B.-Y. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer, "Efficient, high-quality image contour detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 2381–2388, 2009.

[86] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade, "Distributed Cosegmentation via Submodular Optimization on Anisotropic Diffusion," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011.

[87] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proc. IEEE the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[88] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, pp. 2–23, 2009.

[89] L. Yang, P. Meer, and D. J. Foran, "Multiple class segmentation using a unified framework over mean-shift patches," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[90] G. Csurka and F. Perronnin, "An efficient approach to semantic segmentation," *International Journal of Computer Vision*, vol. 95, pp. 198–212, 2011.

[91] L.-J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2036–2043, 2009.

[92] J. Verbeek and B. Triggs, "Region classification with markov field aspect models," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 1 –8, june 2007.

[93] J. Jiang and Z. Tu, "Efficient scale space auto-context for image segmentation and labeling," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1810–1817, 2009.

[94] X. He, R. S. Zemel, and M. A. Carreira-Perpinán, "Multiscale conditional random fields for image labeling," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–695, 2004.

[95] D. Batra, R. Sukthankar, and T. Chen, "Learning class-specific affinities for image labelling," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[96] P. Kohli, P. H. Torr, *et al.*, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[97] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *International Journal of Computer Vision*, vol. 80, no. 3, pp. 300–316, 2008.

[98] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1284–1291, 2005.

[99] D. Parikh, C. L. Zitnick, and T. Chen, "From appearance to context-based recognition: Dense labeling in small images," in *Proc. IEEE international*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8, 2008.

[100] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1–8, 2007.

[101] W. Zou, K. Kpalma, and J. Ronsin, "Semantic image segmentation using region bank," in *21st International Conference on Pattern Recognition (ICPR),*, pp. 922–925, IEEE, 2012.

[102] J. Yang, J. Wang, and T. Huang, "Learning the sparse representation for classification," in *Proc. IEEE Multimedia and Expo (ICME)*, pp. 1–6, 2011.

[103] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[104] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*, p. 78, ACM, 2004.

[105] J. J. Lim, P. Arbeláez, C. Gu, and J. Malik, "Context by region ancestry," in *Proc. IEEE international Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1978–1985, 2009.

[106] J. Gonfaus, X. Boix, J. van de Weijer, A. Bagdanov, J. Serrat, and J. González, "Harmony potentials for joint classification and segmentation," in *Proc. IEEE International Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 3280 –3287, june 2010.

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Semantic-oriented object segmentation

Nom Prénom de l'auteur : ZOU WENBIN

Membres du jury :
- Monsieur CARRE Philippe
- Monsieur LIU Zhi
- Monsieur RONSIN JOSEPH
- Monsieur SENHADJI Lotfi
- Monsieur JURIE Frédéric
- Monsieur KPALMA Kidiyo

Président du jury : SENHADJI Lotfi

Date de la soutenance : 13 Mars 2014

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 13 Mars 2014

Signature du président de jury

L. SENHADJI

Le Directeur,

M'hamed DRISSI

Résumé

Cette thèse porte sur les problèmes de segmentation d'objets et la segmentation sémantique qui visent soit à séparer des objets du fond, soit à l'attribution d'une étiquette sémantique spécifique à chaque pixel de l'image. Nous proposons deux approches pour la segmentation d'objets, et une approche pour la segmentation sémantique.

La première approche est basée sur la détection de saillance. Motivés par notre but de segmentation d'objets, un nouveau modèle de détection de saillance est proposé. Cette approche se formule dans le modèle de récupération de la matrice de faible rang en exploitant les informations de structure de l'image provenant d'une segmentation ascendante comme contrainte importante. La segmentation construite à l'aide d'un schéma d'optimisation itératif et conjoint, effectue simultanément, d'une part, une segmentation d'objets basée sur la carte de saillance résultant de sa détection et, d'autre part, une amélioration de la qualité de la saillance à l'aide de la segmentation. Une carte de saillance optimale et la segmentation finale sont obtenues après plusieurs itérations.

La deuxième approche proposée pour la segmentation d'objets se fonde sur des images exemples. L'idée sous-jacente est de transférer les étiquettes de segmentation d'exemples similaires, globalement et localement, à l'image requête. Pour l'obtention des exemples les mieux assortis, nous proposons une représentation nouvelle de haut niveau de l'image, à savoir le descripteur orienté objet, qui reflète à la fois l'information globale et locale de l'image. Ensuite, un prédicteur discriminant apprend en ligne à l'aide les exemples récupérés pour attribuer à chaque région de l'image requête un score d'appartenance au premier plan. Ensuite, ces scores sont intégrés dans un schéma de segmentation du champ de Markov (MRF) itératif qui minimise l'énergie.

La segmentation sémantique se fonde sur une banque de régions et la représentation parcimonieuse. La banque des régions est un ensemble de régions générées par segmentations multi-niveaux. Ceci est motivé par l'observation que certains objets peuvent être capturés à certains niveaux dans une segmentation hiérarchique. Pour la description de la région, nous proposons la méthode de codage parcimonieux qui représente chaque caractéristique locale avec plusieurs vecteurs de base du dictionnaire visuel appris, et décrit toutes les caractéristiques locales d'une région par un seul histogramme parcimonieux. Une machine à support de vecteurs (SVM) avec apprentissage de noyaux multiple est utilisée pour l'inférence sémantique.

Les approches proposées sont largement évaluées sur plusieurs ensembles de données. Des expériences montrent que les approches proposées surpassent les méthodes de l'état de l'art. Ainsi, par rapport au meilleur résultat de la littérature, l'approche proposée de segmentation d'objets améliore la mesure d F-score de 63% à 68,7% sur l'ensemble de données Pascal VOC 2011.

Mots-clés : segmentation d'objets, segmentation sémantique, détection de saillance

Abstract

This thesis focuses on the problems of object segmentation and semantic segmentation which aim at separating objects from background or assigning a specific semantic label to each pixel in an image. We propose two approaches for the object segmentation and one approach for semantic segmentation.

The first proposed approach for object segmentation is based on saliency detection. Motivated by our ultimate goal for object segmentation, a novel saliency detection model is proposed. This model is formulated in the low-rank matrix recovery model by taking the information of image structure derived from bottom-up segmentation as an important constraint. The object segmentation is built in an iterative and mutual optimization framework, which simultaneously performs object segmentation based on the saliency map resulting from saliency detection, and saliency quality boosting based on the segmentation. The optimal saliency map and the final segmentation are achieved after several iterations.

The second proposed approach for object segmentation is based on exemplar images. The underlying idea is to transfer segmentation labels of globally and locally similar exemplar images to the query image. For the purpose of finding the most matching exemplars, we propose a novel high-level image representation method called object-oriented descriptor, which captures both global and local information of image. Then, a discriminative predictor is learned online by using the retrieved exemplars. This predictor assigns a probabilistic score of foreground to each region of the query image. After that, the predicted scores are integrated into the segmentation scheme of Markov random field (MRF) energy optimization. Iteratively finding minimum energy of MRF leads the final segmentation.

For semantic segmentation, we propose an approach based on region bank and sparse coding. Region bank is a set of regions generated by multi-level segmentations. This is motivated by the observation that some objects might be captured at certain levels in a hierarchical segmentation. For region description, we propose sparse coding method which represents each local feature descriptor with several basic vectors in the learned visual dictionary, and describes all local feature descriptors within a region by a single sparse histogram. With the sparse representation, support vector machine with multiple kernel learning is employed for semantic inference.

The proposed approaches have been extensively evaluated on several challenging and widely used datasets. Experiments demonstrated the proposed approaches outperform the stateof-the-art methods. Such as, compared to the best result in the literature, the proposed object segmentation approach based on exemplar images improves the F-score from 63% to 68.7% on Pascal VOC 2011 dataset.

Keywords: Object segmentation, semantic segmentation, saliency detection

N° d'ordre : 14 ISAR 05 / D 14-05

Institut National des Sciences Appliquées de Rennes

20, Avenue des Buttes de Coësmes • CS 70839 • F-35708 Rennes Cedex 7

Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96