Classifier-based approaches for Top-down salient object Detection

HISHAM CHOLAKKAL

School of Computer Science and Engineering

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

2017

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Deepu Rajan for his intelligent guidance and patient nurture during the period of my thesis work. I have learned so much from his ways of critical thinking and his analytic insights into the problems helped greatly in the accomplishment of this research work. His experience in writing skills has been of great help to me in the presentation of my research work.

I want to thank Jubin Johnson for having fruitful collaboration and for his numerous helpful comments and enlightening discussions throughout my research course. I would also like to dedicate this special thanks to the members of my family, friends, my parents and my dearest wife Hasna Shirin who have always been there for me. This research work would not have been possible without their love and encouragements.

Thanks are also dedicated to Multimedia Lab for the research facilities. I also acknowledge Nanyang Technological University for the financial support and this precious opportunity of study. Finally, I pay my tributes to the God Almighty for blessing me in all my endeavors.

Contents

List of Figures

List of Tables

List of Notations

f	SIFT feature
z	Feature code
Z	Set of feature codes
X	Spatial pyramid max-pooled image vector
W	SVM weight for image classification.
v	Patch classifier parameters. SVM weight in Chapter 6, Logistic regression weight in Chapters 3, 5, and CRF weights in Chapter 4.
Y	Image label
l	Patch label
L	Set of patch labels
ρ	Context max-pooled vector
N	Length of max-pooled image vector
$\mathcal{L}$	Contextual saliency
G	Reverse image classifier saliency. R-ScSPM saliency in Chapter 5 and R-ReluSPM saliency in Chapter 6
$\mathcal{H}$	Combined saliency
D	Dictionary
D_n	Object dictionary
d	Length of feature code
T	Number of training images
n_c	Number of object categories
$\hbar$	128
r_D	Number of atoms in D
r	Number of elements in the object dictionary
r_{bg}	Number of elements in the background dictionary
M	Number of image patches

Abstract

Saliency estimation aims to identify visually important regions in an image and to inhibit distractors. It has been used in recent object detectors and image classifiers as a pre-processor to indicate possible object regions in an image. The category-independent object proposals produced by bottom-up saliency approaches include those are irrelevant for tasks like object detection. The precision of the object proposals can be improved through top-down saliency approaches that produce category-specific saliency maps. Although, the prior knowledge about object categories learnt by classifiers are useful for top-down saliency estimation, the relationship between image classifiers and top-down salient object detectors has not been explored substantially. In this thesis we develop classifier-based approaches for top-down salient object detection in which first two are trained in a fully supervised setting and the last two are trained in a weakly supervised setting.

Non-linear feature representations such as sparse coding (SC) or locality constrained linear coding (LLC) cascaded with linear classifiers are proven to be effective in image classification. They are also used for top-down salient object detection to achieve a compact and discriminative representation of SIFT features, which helps to model feature selectivity for saliency map. We analyze the influence of these feature coding approaches in top-down salient object detection and also propose a novel coding strategy for top-down saliency estimation. The proposed coding strategy ensures that similar codes are assigned to the features which are adjacent in spatial, feature and category domains. These Locality constrained contextual sparse codes are max-pooled over a spatial neighborhood and a logistic regression classifier learnt on these max-pooled vectors is used for saliency estimation.

Many practical computer vision systems need to simultaneously identify the presence of an object as well as to segment it. Moreover, image classifiers and top-down salient

object detection often share similar modules such as feature extractor, feature coding and feature classifier. This motivated us to develop our second fully supervised top-down saliency approach, which is a joint framework for saliency estimation and image classification. In this framework, the image classifier is used both to quantify the likelihood of the presence of an object and to update the saliency map using a novel saliency refinement method. A novel saliency-weighted max-pooling is proposed to improve image classification by weighting the max-pooled vector in each block of the spatial pyramid with a weight computed using top-down saliency maps.

Conventional top-down saliency approaches require fully supervised training in which exact object annotation is required. Availability of images from a simple tag-based internet search has made exact annotation for training saliency models unnecessary. This motivated us to develop weakly supervised top-down saliency approaches that are trained with image-level labels indicating the presence or absence of an object of interest. First, the probabilistic contribution of each patch in the image to the confidence score of a sparse coded spatial pyramid max-pooling (ScSPM) image classifier is analyzed to estimate its Reverse-ScSPM (R-ScSPM) saliency. For high-level understanding of the surrounding spatial region, contextual information of the patch is required, which is incorporated using a contextual saliency module. Besides illustrating the accuracy of saliency maps produced by the proposed method, we demonstrate its effectiveness in applications like weakly supervised object annotation, class segmentation and action classification.

Finally, we develop a convolutional neural network (CNN) based, weakly supervised salient object detection approach that has both bottom-up and top-down modules. Here, we modify the backtracking strategy to identify salient regions that make positive contribution to a CNN-based image classifier. From a set of saliency maps of an image produced by fast bottom-up saliency approaches, we propose a novel strategy to select the best saliency map suitable for the top-down task. The selected bottom-up saliency map is combined with the top-down saliency map. Features having high combined saliency are used to train a linear SVM classifier to estimate contextual saliency. This is integrated with combined saliency and further refined through a multi-scale superpixel-averaging of saliency map. Experiments are carried out on seven challenging datasets and quantitative results are compared with 36 closely related approaches across 4 different applications.

Chapter 1

Introduction

Over the past decade, the volume of images and videos captured and shared have increased by several folds. Analyzing the semantic content in these high resolution images or videos is useful for an enriching cyber-social networking experience or for other applications such as surveillance, media retrieval, etc. However, such an analysis is extremely challenging even for current computing systems with large computing power. For images where the object of interest does not standout from the object due to, e.g. clutter, the problem of content understanding becomes all the more difficult. Under these circumstances, it would be helpful to have a fast pre-processor that can quickly select image sub-regions that are relevant for a given application such as object detection or object segmentation. Such region proposal modules are an integral part of many real-time object detectors [15, 16].

In computer vision, saliency estimation is defined as the process of identifying visually important regions in an image that facilitate subsequent processing for object detection or object recognition by reducing the computation time. The saliency of each pixel in an image is indicated as a probability map called *saliency map* that peaks at the most salient region. Fig. 1.1 (b-h) show heat maps corresponding to various saliency maps, where red indicates highest saliency.

The use of saliency estimator as a pre-processor is motivated from the human visual attention system that enables human brain to respond quickly to visual stimuli, by 'looking' at only a small but informative portion of visual data that it actually 'sees'. The notion of importance largely varies between a person looking at a scene without a goal and with a particular goal. The former is independent of any particular task and

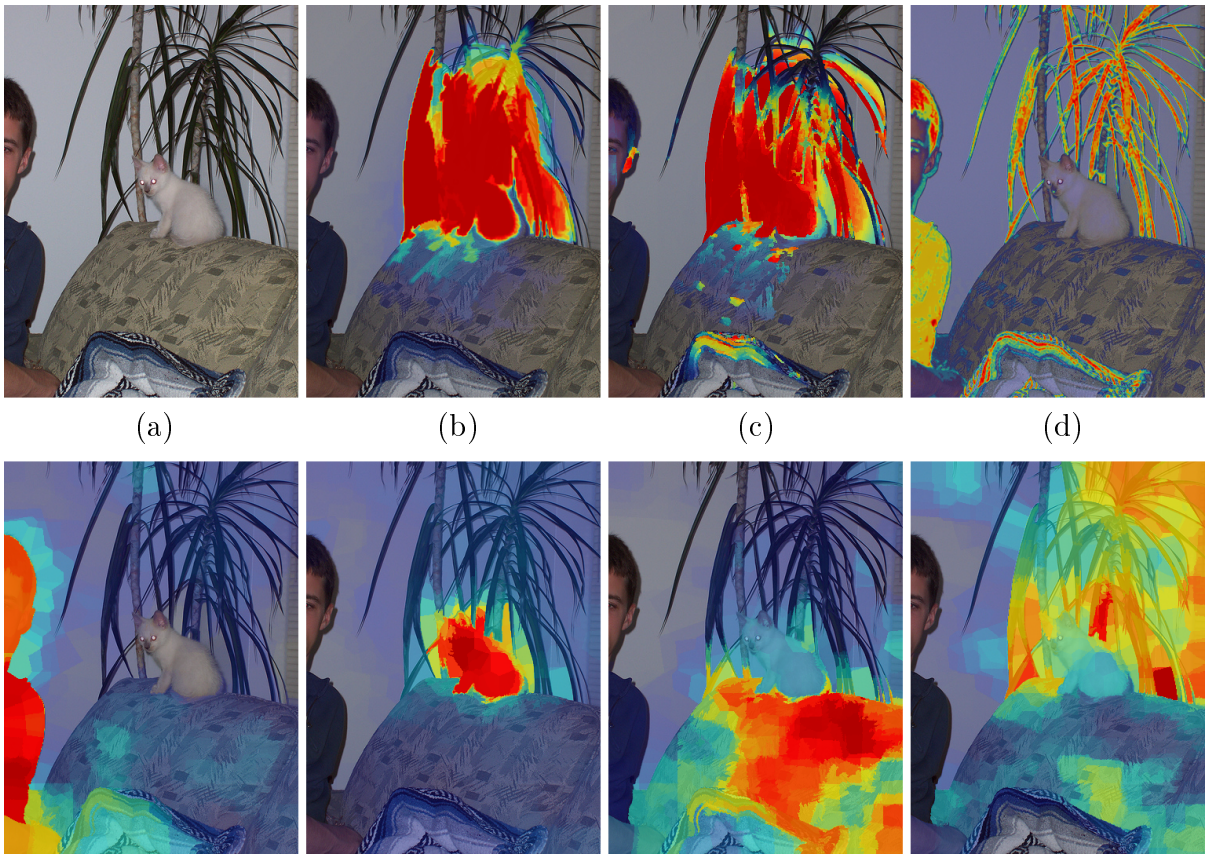


Figure 1.1: Comparison of our top-down salient object detection with bottom-up methods. (a) Input image, bottom-up saliency maps of (b)MB [3], (c) MST [4], and (d) HC [5]; top-down saliency maps for (e) person (f) cat (g) sofa and (h) potted plant categories.

is called bottom-up attention and the latter depends on the task performed by the user and is called top-down attention. Saliency is the distinct subjective perceptual quality which grabs human attention. Bottom-up attention is attributed to image regions that 'pop-out' from its surroundings and the phenomenon is called *bottom-up saliency*. Similarly, top-down attention is attributed to image regions that are salient from a task-driven perspective and the phenomenon is called *top-down saliency*. Typical examples of tasks associated with top-down saliency are searching for a particular object/target or recognizing the semantic contents in a scene.

Bottom-up saliency aims to locate regions in an image that capture human fixations under free-viewing condition. Here, feature contrast at a location plays the central role, with no regard to the semantic contents of the scene, although high-level concepts like

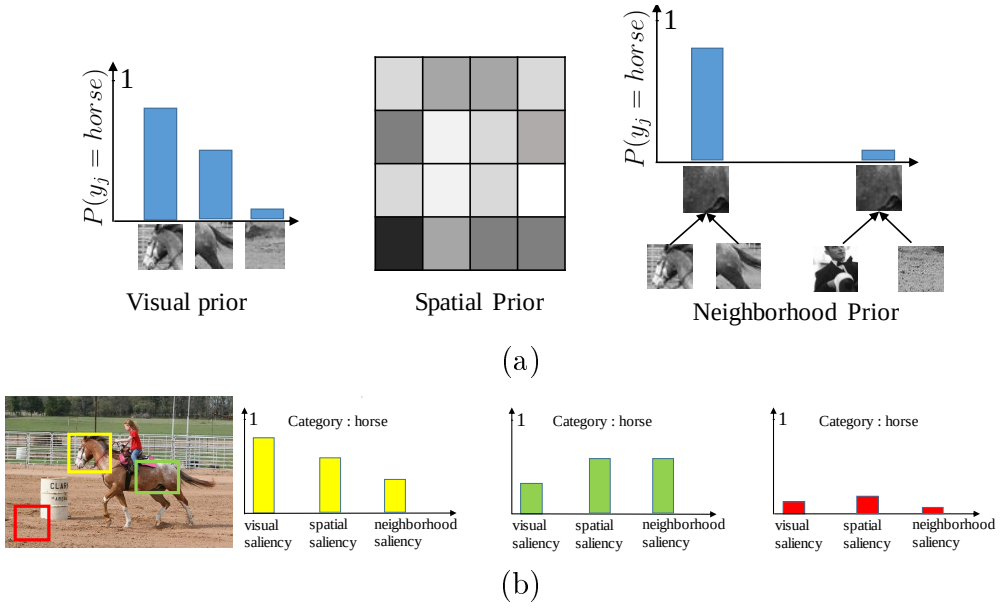


Figure 1.2: Priors for task-specific top-down saliency computation. (a) task-specific priors for horse category learnt from the training images. The visual, spatial and neighborhood saliency values for yellow, green and red colored image boxes in (b) are shown in their respective colors. The spatial prior is a 2-D distribution of horse patches in a 4×4 spatial grid with white indicating high probability and black indicating low probability. The horse's head (yellow box) has high correlation to the horse visual prior, resulting in large visual saliency. Similarly, it is less likely to find a horse patch at the position of the red box, resulting in lower spatial saliency.

face have been used in conjunction with visual cues like color and shape [17]. It often fails in the presence of background clutter due to which the salient object does not 'pop-out'. Moreover, lack of prior knowledge about the target in goal-oriented applications such as object detection and object class segmentation limits its utility. For example, the saliency maps produced by recent bottom-up approaches [3, 4] cannot discriminate between cat and potted plant in Fig. 1.1(b, c).

Top-down attention is largely affected by past experiences of the visual world [18]. This phenomenon includes *contextual cueing* which helps the human brain to respond faster by spending less neural resources at spatial locations where the prior probability of target appearance is low. Mimicking contextual cuing, top-down saliency approaches learn a set of priors about a given task by using a set of training images. Fig. 1.2 illustrates how the priors learnt from training images can be used to compute top-down saliency. Suppose the visual, spatial and neighborhood prior probabilities of features

learnt for horse category are as shown in Fig. 1.2 (a). This prior knowledge is used to estimate visual, spatial and neighborhood saliency separately for an image as shown in Fig. 1.2 (b). The saliency values for yellow, green and red colored boxes are shown as bars with their corresponding colors. If a particular feature in a box has high prior probability of belonging to the horse category, then the box is assigned a high saliency. The yellow colored box is assigned with high visual saliency due to the high visual prior of the horse's head. Thus, the top-down saliency models produce a probability map that peaks at target/object locations [7, 1]. The objective of this thesis is to generate top-down saliency maps like those shown in Fig. 1.1(e, f, g, h). They were generated using the method proposed in Chapter 6 to identify probable image regions that belong to person, cat, sofa and potted plant, separately. It is to be noted that in top-down saliency, the regions that are closer to the pre-learnt priors are assigned higher saliency values even if they are not salient in the 'pop-out' sense.

Salient object detection approaches [1, 3, 19] aim to assign higher saliency values for all pixels of a salient object. In top-down salient object detection, object pixels which are salient for a top-down task are assigned higher saliency values. It finds its use in applications such as object detection [12], object localization [20], object segmentation, semantic segmentation, image classification [9] and action classification [21]. Top-down salient object detection frameworks are closely related to object detection, object localization and object segmentation, but differ in their granularity of representation. Object detection aims to produce a tight rectangular bounding box around all instances of objects belonging to a user-defined category. Here, it is necessary to identify both the location as well as the extent of each object. The process of identifying location of a particular object in an image, without marking the extent of the object, is referred to as object localization [22, 13, 23]. Object segmentation, also referred to as semantic object selection aims to produce a binary mask with '1' indicating all pixels that belongs to a user-defined object category. It differs from the task of semantic segmentation, where the objective is to classify each pixel in the image to one of the predefined classes. In image classification and object recognition, the goal is to identify the objects present in an image. Similarly, action classification aims to classify the action performed by a human in an image.

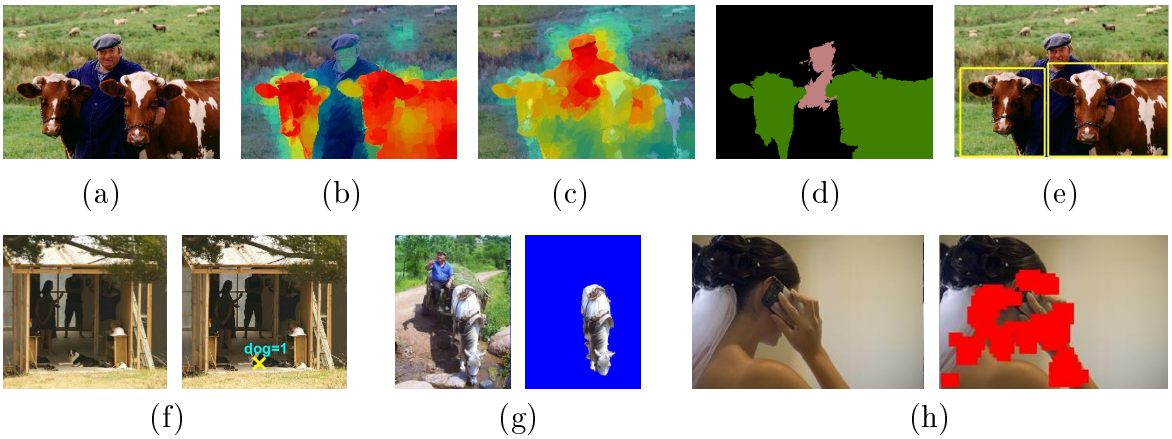


Figure 1.3: Applications of top-down salient object detection. (a) Input image, our top-down saliency map for (b) cow and (c) person categories; (d) semantic segmentation, (e) object detection (cow) , (f) object localization (dog), (g) semantic object selection (horse) and (h) action-specific patch discovery (phoning) results obtained using our weakly supervised top-down saliency approaches.

1.1 Motivation

Being a pre-processor, top-down salient object detection algorithms should be faster in training as well as inference, as compared to dedicated algorithms for object detection or semantic segmentation. Existing top-down salient object detection approaches such as [13, 6] require multiple days to train their saliency model. The iterative learning of feature representation is one of the main reasons for higher training time. For example, [6, 1] use an iterative dictionary learning strategy to improve sparse coding accuracy. Similarly, iteratively fine-tuning of the Convolutional Neural Network (CNN) features in [13] needs nearly 8 days on a GPU. CNN-based object detectors such as [15] and [16] have fast inference times, but require multiple days to train their filter weights through backpropagation [24]. Thus, we develop a set of top-down salient object detection models that do not require iterative dictionary learning (Chapters 3, 5) or fine-tuning of CNN features (Chapter 6). In Chapter 4 we propose an image classifier-based strategy that reduces the training time of [1]. Yang and Yang [1], [25] and [6] require larger time to simultaneously infer multiple object categorizes on an image, due to their use of separate feature representation for each object category. We use common feature representation across all object categories in Chapters 3, 5 and 6.

Image classifiers such as [2, 26] and top-down salient object detection approaches such as [1, 25] share similar modules such as feature extractor, feature coding and feature classifier. The relationship between such image classifiers and top-down salient object detectors has not been explored substantially. We develop a set of top-down salient object detection algorithms based on image classifiers (Chapters 4, 5, 6). Many practical computer vision systems need to simultaneously identify the presence of an object as well as to segment it. We develop a joint framework for image classification and top-down salient object detection (Chapter 4).

Conventional top-down saliency approaches require fully supervised training in which exact object annotation is required [1, 6, 25, 13]. Availability of images from a simple tag-based internet search has made exact annotation for training saliency models unnecessary [27, 28]. This motivated us to develop weakly supervised top-down saliency approaches in Chapters 5 and 6 that are trained with image-level labels indicating the presence or absence of an object of interest. These saliency models can be used as pre-processors for weakly supervised applications of object detection and segmentation. Fig. 1.3(a) shows an input image and Fig. 1.3(b, c) show the top-down saliency maps produced by our weakly supervised algorithm in Chapter 6, for cow and person categories. Fig. 1.3 (d, e, f, g, h) show the use of our top-down saliency maps in weakly supervised applications such as semantic segmentation, object detection (cow), object localization (dog), semantic object selection (horse) and action-specific patch discovery (phoning), respectively.

Conventional sliding window-based object detectors need to sample and classify millions of rectangular boxes with arbitrary shapes, sizes or locations [29]. In order to simplify this computationally intensive process, recent object detection approaches [30] use bottom-up saliency-based object proposals [31, 32] to extract regions of probable objects in an image. Since these region proposals are category-independent, they also result in thousands of rectangular boxes that are irrelevant to the task. This motivates the development of task-specific top-down salient object detectors that can produce category-specific object proposals as shown in Fig. 1.3 (e), where accurate object proposals for cow category are produced using our top-down saliency map shown in Fig. 1.3 (b).

1.2 Objective

The primary objective of this thesis is to develop a set of classifier-based approaches for top-down salient object detection. There are five key issues related to this problem that we address in this thesis.

1. Develop a feature coding strategy specifically for top-down salient object detection.
2. Develop a joint framework for image classification and top-down salient object detection.
3. Develop an image classifier-based, weakly supervised approach for top-down salient object detection, that can be trained with image-level binary labels indicating the presence or absence of object of interest.
4. Develop a convolutional neural network-based, weakly supervised, hybrid approach, that can be utilized for task-specific top-down salient object detection as well as for task-independent salient object detection.

We demonstrate the usefulness of top-down salient object detection for different applications such as image classification (Chapter 4), fully supervised object segmentation (Chapter 4), weakly supervised object segmentation (Chapters 5, 6), weakly supervised semantic segmentation (Chapter 6), weakly supervised category-independent salient object detection (Chapter 6), weakly supervised object detection/object annotation (Chapters 5, 6), weakly supervised object localization (Chapter 6) and weakly supervised action-specific patch discovery (Chapter 5). Moreover, we demonstrate that top-down salient object detection can achieve state-of-the-art results across multiple applications.

1.3 Contributions

We develop four novel frameworks for salient object detection, in which the first two are trained in a fully supervised setting (Chapters 3, 5) and the last two are trained in a weakly supervised setting (Chapters 5, 6). A brief overview of the key contributions are listed below.

- We develop a feature coding mechanism called locality constrained contextual sparse coding (LCCSC). It introduces locality constraints in three different domains - spatial domain, feature domain and category-domain. Even though an image classifier per se is not employed, a similar framework as in [2] is used to produce saliency maps from feature codes. Specifically, the LCCSC codes are max-pooled in its spatial neighborhood followed by a logistic regression classifier to estimate patch-level saliency. A Gaussian weighted interpolation is proposed to map patch-level saliency to pixel-level saliency map (Chapter 3).

- A joint framework is proposed for image classification and top-down salient object detection. The image classifier is used to control the training of top-down saliency model and to refine the saliency map. On the other hand, saliency maps are used to improve image classification through a novel saliency weighted max-pooling. Since the image classifier and the top-down salient object detector share many modules such as SIFT feature extraction and coding, our joint framework reduces the computational cost if these two modules were to be used separately.

- A novel weakly supervised, top-down salient object detection approach is proposed that can be trained with image-level labels indicating the presence or absence of object of interest in training images. The probabilistic contribution of image patches to an ScSPM image classifier is used to estimate so-called R-ScSPM top-down saliency, which is further improved by incorporating contextual saliency (Chapter 5).

- We propose an image classifier-based strategy to select a saliency map from a set of bottom-up saliency maps, which is best suited for a given top-down task (Chapter 6).

- A CNN-image classifier is used in a combined bottom-up top-down framework for salient object detection. This weakly supervised framework can be configured for category-specific top-down salient object detection as well as for category-independent saliency object detection (Chapter 6).

1.4 Organization

This thesis is organized into chapters that discuss different methods for top-down salient object detection. Chapter 2 reviews literature in the area of top-down salient object detection and other related areas. Chapter 3 introduces a novel feature coding strategy for top-down salient object detection. Chapter 4 proposes a joint framework for image classification and top-down salient object detection. Both these frameworks in Chapters 3 and 4 are fully supervised. We introduce an ScSPM image classifier-based weakly supervised top-down salient object detection in Chapter 5 and finally a CNN image classifier-based hybrid salient object detection approach is proposed in Chapter 6. Chapter 7 concludes the thesis and gives directions for the future research.

Chapter 2

Literature review

In the previous chapter, top-down (bottom-up) saliency is defined as the perceptual quality in an image/video that effects top-down (bottom-up) attention. While top-down saliency is largely task driven, there is no such notion in bottom-up saliency. Top-down factors such as the prior knowledge about a task [18, 33] plays an important role in the former one while feature contrast, center-prior etc. play an important role in driving attention in the latter. Thus it is clear that visual saliency approaches can be grouped into two - task-independent bottom-up saliency, and task-specific top-down saliency.

Even under free-viewing condition where no specific task is implied, humans have a tendency to attend to faces, text and objects such as animals, vehicles and persons [18]. Mimicking this influence of prior knowledge in human attention, many bottom-up saliency approaches incorporate top-down knowledge in their saliency model and prioritize such objects in their saliency maps [34]. Similarly many top-down approaches incorporate bottom-up saliency in their pipeline to improve the performance of top-down saliency models on previously unseen tasks [12, 35, 13]. Hence, some approaches can not be grouped clearly into bottom-up and top-down saliency frameworks, especially based on their training pipeline [18]. During inference, the approaches that produce a generic saliency map which is agnostic to the task performed [36, 37] are considered as category-independent bottom-up frameworks, while the saliency map of top-down approaches varies based on the task at hand, i.e., inference of saliency maps are conditioned on the user defined task [7, 12, 1].

2.1 Bottom-up saliency

In this section, we briefly review literature on bottom-up saliency, before moving on to top-down saliency in the next section.

2.1.1 Visual attention under free-viewing condition

The first computational model for visual attention was proposed by Koch *et al.* [38] which computed several features in parallel and integrated attention values computed from each them. This theoretical model was modified and implemented by Clark *et al.* [39]. Following this, several models have been developed to predict human fixation by utilizing various qualities such as surprise, self-information or signal to noise ratio [33, 40].

Borji [34] illustrated that performance of fixation prediction can be largely improved by integrating various top-down detectors such as horizontal line detector, face detector, object detectors [41], with low-level features and saliency maps from bottom-up saliency models of [42, 43, 44]. The process of human fixation prediction is also called as visual saliency detection in some literature [45]. In general, fixation prediction approaches aim to predict the locations which will be attended by a free-viewing person for the first 3-5 seconds [40]. A detailed review of literature in the area of visual attention modeling is available in [35].

2.1.2 Salient object detection

In salient object detection, the goal is not only to detect salient objects but also to segment them out accurately. Hence, the terms 'salient object detection' and 'salient region segmentation' are used interchangeably in literature [19] and throughout this thesis. The key difference between salient object detection and fixation prediction is that the fixation models aim to model human attention system, while salient object detection approaches aim to identify the image regions corresponding to particular task. Moreover, the ground truth for salient object segmentation is a binary map that covers the entire object while that for fixation prediction is a few points on the image corresponding to human attention.

One of the early works on salient object detection was [46, 17], which introduced the binary segmentation of salient objects. A good salient object detection approach should have the following qualities [19] (i) accurate detection and high resolution: it should assign all pixels of the salient object with high saliency values and all background pixels with low saliency values; (ii) computational efficiency: since saliency approaches are often used as a pre-processor to other complex tasks such as object detection or segmentation, they should be faster during training as well as during inference compared to dedicated object detectors or object segmentation algorithms.

The task-independent bottom-up approaches are useful for various applications such as object proposal [31, 32], image segmentation [47] etc. Most of the fast bottom-up saliency approaches are based on intrinsic cues which are extracted from the input image. These approaches utilize heuristics such as center-surround contrast, distinctiveness [48], center bias or rarity of the scene [19]. On the other hand, extrinsic approaches use statistical prior information from similar, user annotated images to estimate saliency. The underlying assumption is that targets and backgrounds share similar attributes across images. Machine learning based saliency approaches that learn saliency models from a set of annotated training images and produce a category-independent saliency map falls under this extrinsic category. Recent deep learning-based approaches learn their saliency model from a larger set of training images which are annotated in a fully supervised setting. In the next subsection, we review the state-of-the-art intrinsic and extrinsic bottom-up saliency approaches. A detailed discussion about bottom-up salient object detection is available in [19].

2.1.2.1 Intrinsic approaches

Intrinsic approaches do not utilize any prior information about the semantic contents of the scene. A two-step strategy is proposed to estimate saliency in [49]. First, a weak saliency map is estimated by using center-bias prior in which the contrast of each region with the image boundaries are utilized. In the second step, a strong classifier is learnt based on the superpixels of an image collected from this weak saliency map. The strong and weak saliency maps are combined through a weighted integration to form the final saliency map. Real-time salient object detection approaches [3, 4] also assume image

boundaries as the background. The minimum barrier distance of each pixel to the image boundary is utilized in [3] for salient object detection. They utilize raster scanning of pixels to speed-up computation resulting in an approximate minimum barrier distance. In [4], an image is represented as a tree and the distance between pixels in a minimum spanning tree is computed to estimate saliency. All the above mentioned approaches that utilize the background-prior or center-bias prior to estimate the saliency fail on images where the salient object appears at image boundaries.

One of the fastest saliency estimation approaches [5] utilizes region contrast to estimate saliency, where global contrast is utilized along with a spatial coherence score. Such contrast-based approaches often fail to detect the uniform regions within the salient object, resulting in a non-contiguous saliency map. Diffusion-based approaches [50, 51] propagate the saliency map to the entire salient object. [45] proposes a generic framework to develop a good diffusion matrix and a seed vector that can be used to improve diffusion-based approaches [50, 51]. The relation spectral graph clustering and modified Hamilton operator from quantum mechanics are explored in [52] and in extended quantum cut (EQCUT) [53].

2.1.2.2 Extrinsic approaches

These approaches learn a saliency model or statistical prior of salient objects using a set of training images in which the salient objects are annotated through a rectangular bounding box or through a pixel accurate segmentation mask. Considering the fact that such a supervised training may incorporate many top-down factors to the saliency model, these learning-based saliency models are also referred to as knowledge-based top-down saliency approaches [18].

In [54], a random forest regressor is trained using annotated training images which is used to map regional saliency features to a saliency score. Unlike the intrinsic approaches which utilize heuristics to handcraft the saliency features, the random forest regressor identifies the discriminative features from a high dimensional saliency feature vector. Recently, convolutional neural network-based approaches achieve state-of-the-art performance [36, 37, 55, 56] in category-independent saliency datasets [57] by learning saliency features from a large number of fully annotated training images [5]. Many of

these deep learning-based approaches [37, 56] initialize their model with the filter weights learnt for image classification on ImageNet [58] dataset. Training or fine-tuning of these deep learning-based saliency approaches takes multiple days. In summary, requirement of large number of fully annotated training data is the main limitation of these CNN-based approaches.

2.2 Top-down saliency

The saliency map of top-down saliency approaches varies based on the task. If the task is to search for a car in an image, the region corresponding to 'car' is considered as salient, even if it does not 'pop-out' in the scene. Discriminative features in a classification task are assumed as salient in [8, 21, 9, 33]. On the other hand, in object search tasks, the regions are salient not just due to the discriminative features, but the entire object is assumed to be salient [1, 6, 13]. These top-down salient object detection approaches have wide range of applications such as category-specific object proposal and semantic segmentation.

Task-specific top-down saliency approaches can be broadly categorized into approaches for (i) object search and for (ii) image classification. Context plays an important role in search task. For example, a person searching for a car is more likely to look at the road than to the sky [59]. Here, road is considered as the context for locating the salient object which is the car. On the other hand, discriminative features are important for classification tasks. For example, if the task is to classify whether a given object is motor bike or car, then the number of tyres of the object is a useful cue that can discriminate between these classes and hence considered as salient in this classification task.

2.2.1 Object search task

Human beings can spot an object much faster if it appears in a previously known context [59]. For example, it is faster to detect a keyboard kept near a computer as compared to a keyboard kept in the kitchen. This is due to the fact that humans utilize contextual information in visual search task. In [60], contextual information is incorporated in a top-down saliency model for visual search task. At first, bottom-up saliency is computed

based on the rarity of visual features and then the neighborhood and strategies that were associated with previous successful search tasks are incorporated as context to the model.

SUN [61] computes the probability of appearance of the object of interest at each location in an image, by combining location and appearance priors of the target along with bottom-up saliency through a Bayesian framework . Here location prior and appearance prior are the top-down saliency components and are computed from the training images. The statistics of occurrence of target at each locations in an image is utilized to compute the location prior and the likely-hood of image features with the target is computed for the appearance prior. Its accuracy reduces considerably if the target appears at a location different from its location prior.

In [62], bag-of-words representation is improved by spatial weighting of features using shape masks, causing foreground features to be boosted, thereby decreasing the influence of background clutter. The high dimensional hypothesis clustering of shape mask, used in this approach, requires separate annotation for each object within an image. i.e., if multiple objects from same category are present in an image, each of them needs to be labeled separately. Additionally, training the model requires images to be marked as *difficult* or *truncated*. Our methods produce better saliency maps compared to [62] without the added requirement of such annotations.

A fully supervised top-down saliency model is proposed by Yang and Yang [1] that jointly learns a conditional random field (CRF) and dictionary using sparse codes of SIFT features as latent variables. Although they show good results in distinguishing between object and background, there are numerous false detections due to their inability to discriminate similar objects (e.g. dog and cat) and background patches that are visually similar to the object. Even though CRF incorporates the immediate neighborhood prior to the saliency model, the lack of larger contextual information limits its performance if the object is too big relative to patch size. Apart from the requirement of exact object annotation, training of this model is computationally expensive as well. It takes 10-20 iterations of dictionary learning for each category, and the sparse codes of all training images need to be recomputed in each iteration.

Kocak *et al*. [6] improves upon [1] by considering the first and second order statistics of color, edge orientation and pixel location within a superpixel, along with objectness [32]

instead of SIFT features. Although this improved the accuracy of distinguishing objects from background, its ability to discriminate between object categories did not improve, causing large number of false detections if the test image contained irrelevant objects. Blocking artifacts are observed in the saliency map at the super-pixel boundaries, since the super-pixels are extracted in only one scale. Khan and Tappen [63] use label and location-dependent smoothness constraint in a sparse code formulation to produce a smooth saliency map compared to the conventional sparse coding, but with additional computation cost.

Top-down saliency via contextual pooling is proposed in [25] which makes use of neighborhood information (spatial context) for saliency estimation. They follow a framework that is very similar to that of image classification [26, 2] in which SIFT [64] features are extracted and encoded using locality constrained linear coding (LLC). For each patch, a set of contextual patches are evaluated at multiple scales, orientation and neighborhood using block-level max-pooling. These max-pooled vectors are mapped to the saliency maps using logistic regression. Since the max-pooling is limited to a local neighborhood of few patches, overall idea about the image and the spatial priors of the category are not available, which limits the performance of this model. Moreover, to estimate top-down saliency maps corresponding to each category on same test image, the feature codes need to be recomputed for each category as in [6, 1]. Except for the saliency module of Chapter 4, the feature codes are category-independent in all the frameworks proposed in this thesis and hence does not require re-computation.

Recently, a fully supervised, CNN-based top-down saliency is proposed that utilizes visual association of query images with multiple object exemplars [13]. They use a two stage deep model in which the first stage is to learn object-to-object association and the second stage is to learn object to background discrimination. They follow a sliding window setting to extract image patches and each of these patches are resized to 224×224 and fed separately to the CNN to extract deep features. There are around 500 patches on an image of size 500×400, resulting in 500 forward passes through the network to extract CNN features of an image. Due to this huge computational requirement, this approach needs more than a week on a GPU to train their saliency model.

2.2.2 Image classification task

In a classification task, top-down saliency is related to the ability of an image region to discriminate among different categories [8]. Discriminative models [29, 21] often represent a few patches on the object as salient and not the entire object. Hence, such models end up with low recall rates as compared to [6, 1].

In [9, 65], an object classifier is learned using randomly selected, random sized sub-windows from an image, which is then used to build and update a saliency map in a weakly supervised setting. Using this saliency map, the classifier samples more sub-windows in the salient region. The drawback of this joint framework is that if the randomly initialized windows do not contain the object, it will lead to an error in the initial saliency map. This error in the initial estimate of the saliency map gets propagated to consecutive iterations resulting in the failure of both classifier and saliency estimation.

The appearance statistics of discriminative features from a pre-defined filter bank (e.g., DCT) are used in DSD [8] to distinguish different object classes. The discriminative features that maximize the mutual information to the category label in an image classification task are considered as top-down salient features. These discriminative features are learned in a weakly supervised setting and are combined with bottom-up features to estimate the saliency. However, by only considering the image-level statistics of a feature and not its neighborhood information, they are unable to remove background patches leading to poor performance on images with heavy background clutter or viewpoint variations.

Apart from these dedicated top-down saliency approaches, saliency is shown to improve image classification in [21, 66, 67, 29, 68]. However, the accuracy of saliency maps are not reported. In [21], visual features are weighed with discriminative spatial saliency to improve image classification. Saliency of a local region is calculated by considering its appearance along with its spatial location. Integrated max-margin learning is used to learn saliency and classifier. In [29], a random forest with discriminative decision tree is used to mine out the discriminative patches for fine-grained image classification. Category-specific color features are used in [66, 68] to modulate SIFT features. Shape features from the regions with higher color-attention are given more weight than those from lower attention.

2.3 Hybrid approaches

Primates are capable of handling both free-viewing scenarios as well as searching for a particular object when needed [69]. In this section, we review hybrid approaches that can be utilized in both these scenarios. These approaches are useful in robotic navigation for top-down detection of traffic lights and landmarks along with unpredicted accidents or obstacles. Similarly, hybrid models can be used in surveillance systems where pre-defined search for suspects (top-down) as well as unexpected events such as intrusions (bottom-up) need to be handled simultaneously. The top-down approaches perform well on previously known categories but they do not generalize well on unseen categories. On the other hand, since bottom-up approaches are too generic, they are outperformed by the top-down approaches for pre-defined tasks by a huge margin.

Navalpakkam and Itti [69] proposed a hybrid model to speed-up the visual search task. First, bottom-up saliency is computed using [70], followed by weighting with pre-computed top-down weights corresponding to the user-defined task. Here, the top-down weights are computed based on signal-to-noise-ratio (SNR) and the ratio of bottom-up saliency of the target to the saliency of the distracting background. Similarly, in VOCUS [71], bottom-up saliency map of [48] is combined with a top-down part to improve the search task. Yang and Yang [12] extended their top-down saliency approach [1] for fixation prediction by combining with bottom-up saliency maps like GBVS [42] or AWS [72]. This enabled [12] to be configured for object search task as well as fixation prediction.

2.4 Other related approaches

In this thesis we demonstrate the efficacy of top-down saliency for object segmentation, object localization, object detection, discriminative patch discovery, and to improve image classification. We compare our results with other relevant approaches for these tasks. Hence, in this section, we briefly review some of the state-of-the-art approaches for these applications.

2.4.1 Image classification

Spatial Pyramid Matching (SPM) [73] has been widely used in image classification. In sparse coded SPM (ScSPM) [2], features were sparse coded and then max-pooled over a multi-scale spatial pyramid (SPM) [73]. This reduced the classifier complexity to $O(T)$ compared to $O(T^2 \; or \; T^3)$ in SPM for T training images. Since max-pooled vectors from the spatial pyramid blocks that contain an object and those that do not contain any object are considered equally, ScSPM often performs poorly in the presence of high background clutter. The computational complexity of ScSPM is further reduced in LLC [26] by imposing feature locality constraint, and in [74] by compromising on the accuracy through division of the dictionary into cartesian product of sub-dictionaries. In [75], feature extraction from an automatically selected bounding box around objects is shown to improve image classification accuracy. However, this method needs iterative expansion of latent parameter space for effective localization of the object, which is computationally expensive.

2.4.2 Object localization

Recently, CNN is used in a number of weakly supervised object localization approaches [76, 22, 77, 78]. In [76] image regions are masked out to identify the regions causing maximal activation. Multiple-instance learning is combined with CNN features in [77] for object localization. The output of CNN on multiple overlapping patches are evaluated in [78] which is then utilized for object localization. All these approaches need multiple forward passes on a network to localize objects, which makes them computationally less efficient. Our CNN-based approach (Chapter 6) needs just one single forward pass on the network to extract CNN features.

Oquab *et al.* [22] applied global max-pooling to localize a point on objects, due to which this approach fails to identify the full extent of the object. Global max-pooling is replaced by average pooling in [20] to overcome this drawback. The underlying assumption is that loss for average pooling enables the network to identify all discriminative object regions. However, the spatial information is lost, whereas it is retained in our framework (Chapter 6) via spatial pyramid pooling in the image classifier. The image

classifier weights are reused for localization in [20]. We learn an additional contextual saliency to better estimate saliency at object regions.

Internal representations learned by CNN is visualized in [79, 80, 81, 82] for better understanding of its properties. [79, 82] analyzes the convolution layers using techniques such as deconvolutional networks [79]. In [80, 81], deep features are inverted at different layers including the fully connected layers to analyze the visual encoding of CNN. Even-though, these approaches are capable of visualizing the information preserved in the deep features, their relative importance are not analyzed to identify the discriminative image regions. In [83], CNN back-propagation is used to obtain the saliency map of images. Here the CNN is back-propagated from fully connected layers till image pixel-level, to identify the image regions responsible for activations corresponding to an object class (salient regions). Our approaches in Chapters 5 and 6 backtrack the spatial pyramid only upto feature-level, not upto pixel-level.

2.4.3 Object detection

A weakly supervised, end-to-end CNN architecture is proposed in [84] for simultaneous object detection and image classification. Object detection requires classification of a large number of category-independent object proposals [31], [85] which are less precise. On a test image, the CNN features are extracted on the original and flipped image at five scales totaling to 10 feature extraction iterations, which is computationally expensive. The proposed frameworks in Chapters 5 and 6 do not require thousands of category-independent object proposals. Instead, less than 5 category-specific object proposals are produced by the proposed approach for each category.

A curriculum learning strategy is used in [86], where easy images obtained through Google image search are used for initial CNN training. Google search results are normally with clean background and object placed at the image center. Hard examples from Flickr images is used to fine-tune the model which helps the model to generalize well. We do not use such a two -step strategy, instead all training images are used at a time, irrespective of their complexity.

2.4.4 Discriminative patch discovery

In [87], an iterative learning strategy is used to identify discriminative patches. The approach switches between clustering and SVM training. Given an initial set of clusters, a linear SVM is trained for for each cluster, using patches within the cluster as positive examples. The proposed weakly-supervised approach in Chapter 5 does not require such iterative strategy to identify task-specific image patches.

[88] use an extension of the classic mean-shift algorithm for discriminative patch discovery. Here, the discriminative patches are detected based on individual patch features, without considering position of patches within the image. But images often contain global structures beyond patches, hence we use spatial positioning of patches within an image to identify action-specific patches in Chapter 5.

2.4.5 Object segmentation

Bag-of-feature based classification of local image regions having a fixed size is used in [89]. Absence of spatial consistency and context-type constraints causes many false detections leading to less accurate localization of objects. The class segmentation approach of [90] uses superpixels as the basic unit to build a classifier using histogram of local features within each superpixel. Histograms in a neighborhood are aggregated and used to regularize the classifier. A CRF built on superpixel graph further improved the segmentation accuracy. The performance of this model depends on the neighborhood size whose optimum size varies across object categories.

A fast and robust object segmentation method proposed in [91] computes multi-class pixel-level object segmentation of an image through an integral linear classifier built on bag-of-words representation of local feature descriptors. Here, a large dictionary of 500,000 words is required and they use a cascaded classifier containing 2 or 3 linear classifiers. [92] uses bag-of-features for joint categorization and segmentation. The interaction between pixels and superpixels is modeled using random fields and global representation for categorization is achieved through a bag-of-features representation. The number of parameters to be learned for classification increases with increasing number of training images and number of categories. So, it is computationally infeasible to use this

approach for large datasets like PASCAL VOC-07 [93]. The joint categorization and segmentation model proposed in [94] simultaneously learns segmentation, categorization, and dictionary learning parameters. Simplicity of bag-of-features representation limits the performance of both. Co-segmentation approaches [10, 95, 11, 96, 97] segment out the common objects among a set of input images. Semantic object selection [28] collects images with white background from the internet using tag-based image retrieval.

From the literature review, it is evident that many computer vision applications such as object localization, object detection object segmentation can gain from employing efficient top-down saliency detection methods as pre-processors. In Chapter 6, we demonstrate how these applications use a single top-down salient object detection framework.

Chapter 3

Top-down Saliency with Locality-constrained Contextual Sparse Coding

3.1 Introduction

Non-linear feature representations such as sparse coding (SC) or locality constrained linear coding (LLC) cascaded with linear classifiers are proven to be effective in image classification [2, 26]. They are used for top-down salient object detection in [1] and [25] to achieve a compact and discriminative representation of SIFT features, which helps to model feature selectivity for saliency map. In this chapter, we analyze the influence of these feature coding approaches in top-down salient object detection and also propose a novel coding strategy.

A set of basis vectors learned from training image features is called a dictionary and each basis vector d_i is called a dictionary atom. Sparse coding aims to represent a given data f as a linear combination of a minimum number of atoms in an overcomplete dictionary D. A dictionary having larger number of atoms compared to the feature dimension is called overcomplete dictionary. In other way, a sparse code z should minimize the data reconstruction error $\|f - Dz\|_2$ by using a limited number of non-zero elements in the code (l_0-norm), i.e., $\arg\min_z \|f - Dz\|_2 + \lambda \|z\|_0$. Conventional sparse coding approaches [98] replaces the non-convex l_0-norm with its convex relaxation, which is l_1-norm and hence the optimization function for the sparse coding is

$$\arg\min_z \|f - Dz\|_2 + \lambda \|z\|_1 . \tag{3.1}$$

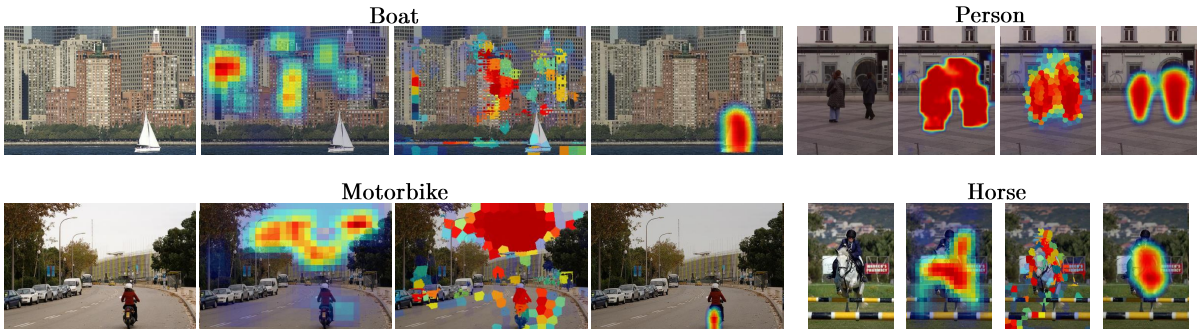


Figure 3.1: Illustration of proposed saliency model on challenging test images from PASCAL VOC-07 and Graz-02 datasets (best viewed in color). (a, e) input image, (b, f) Saliency maps of Yang and Yang [1], (c, g) Kocak *et al.* [6] and (d, h) the proposed method.

In Locality-constrained linear coding (LLC) [26], Wang *et al.* replaced the sparsity constraint $\|z\|_1$ of eq. (3.1) with a feature domain locality constraint to ensure that similar features are assigned with similar feature codes. In image classification frameworks, LLC improved the speed of feature coding by multiple folds compared to sparse coding, without compromising on classification accuracy.

While sparse codes guarantee minimum reconstruction error by using less number of dictionary atoms, LLC ensures that similar features are coded similarly by limiting the coding of each feature to its nearest neighbor atoms in the dictionary. Sparse codes often fail to maintain the locality of features while LLC codes fail to ensure minimum reconstruction error for the feature. Both sparse coding and LLC coding do not consider contextual information in the neighborhood due to which features in spatially adjacent smooth regions are coded differently.

In the case of classification, it has been suggested [99] that using a discriminative dictionary in which features common to various classes are removed through supervised learning, results in increased accuracy with fewer number of dictionary atoms. However, removing common features limits performance in top-down salient object detection, where the all pixels of a salient object need be marked as salient, not just the discriminative regions. In this chapter, we propose a coding scheme that overcomes the above mentioned drawbacks of SC and LLC coding. The chapter has two major contributions: (i) we propose locality-constrained contextual sparse coding (LCCSC) for a feature in which

three forms of locality constraints are imposed on the code: category locality in which κ-nearest neighbor atoms of a feature are chosen from sub-dictionaries of each category, spatial locality in which context is incorporated by selecting the nearest neighbor atoms for spatially nearby features and finally, feature locality as in [26]; (ii) we modify the contextual max-pooling of [25] by pooling the locality-constrained contextual sparse codes instead of category-specific LLC codes (note that context in LCCSC refers to the context arising from the spatial locality constraint and not to the contextual max-pooling operation). A logistic regression classifier trained on context max-pooled vectors estimates the top-down saliency of image patches. The advantages of the proposed method are (i) unlike [1, 6, 25], the sparse codes for different saliency models (classes) are not recomputed since the dictionary is common for all categories and (ii) faster training time since there is no iterative dictionary learning as in [1, 6]. We also propose a Gaussian-weighted interpolation step to generate pixel-level saliency maps from patch-level maps. These contributions result in improved top-down saliency maps on Graz-02 [100] and Pascal VOC-07 datasets [93].

Fig. 3.1 shows three images from PASCAL VOC -07 dataset (boat, motorbike and horse) and one from person category of Graz-02 dataset along with their corresponding saliency maps. Our boat model assigned boat pixels with highest saliency values (Fig. 3.1 (d)), while boat models of other top-down saliency approaches [1, 6] fail (Fig. 3.1 (b, c)) due to cluttered background. Similarly, our person model could separate two persons (Fig. 3.1 (h)) in the person image of Graz-02 dataset (Fig. 3.1 (e)) while [1, 6] (Fig. 3.1(f, g)) failed to do so. Motorbike was successfully assigned highest saliency values by the proposed motorbike model (Fig. 3.1 (d)) even though the image contains person, car and bus categories. Similarly, the proposed horse model produces a better saliency map (Fig. 3.1 (h)) as compared to others.

3.2 System overview

Fig. 3.2 shows the pipeline of the method proposed in this chapter, which is similar to the widely used sparse coded spatial pyramid matching (ScSPM) image classifier, i.e., feature extraction, feature coding, pooling and feature classifier. To reduce computations, dense

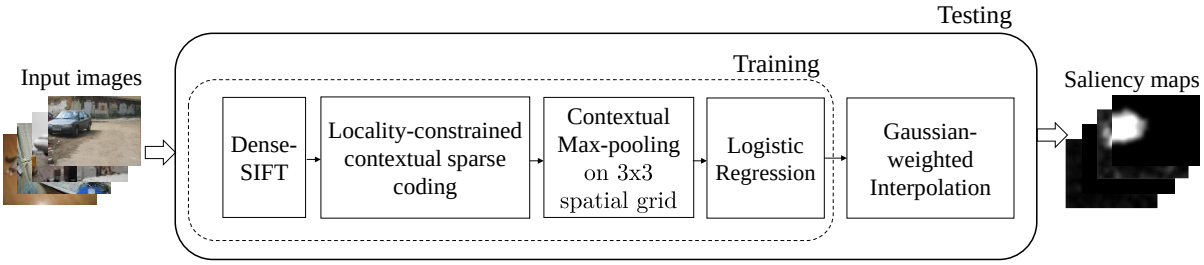


Figure 3.2: Training and testing of the proposed top-down saliency approach.

SIFT features are extracted using only a single patch size. A novel locality-constrained contextual sparse coding strategy is proposed for feature coding. Spatial neighborhood of a patch is divided into a regular grid and the codes in each cell of the grid are max-pooled individually. These max-pooled vectors are vertically concatenated to form a context max-pooled vector representing the patch. Logistic regression-based feature classifier is learnt using these context max-pooled vectors. For saliency inference on a test image, the class-conditional probability of context max-pooled vectors is estimated from the learnt logistic regression model. This probability is the saliency value of a patch in the image. The pixel-level saliency map is obtained using a novel Gaussian-weighted interpolation of the patch-level saliency map.

3.3 Locality-constrained contextual sparse coding (LCCSC)

3.3.1 Formulation

Various coding schemes aim for specific objectives like sparsity [98], feature-domain locality [26] and spatial-domain locality [101, 102]. Here, all these desired properties are integrated into a single objective function. Also, feature coding using a *discriminative* dictionary is not the goal in top-down saliency; rather feature codes should be agnostic to object categories as long as the features contribute to locating the salient object (this will be elaborated soon). LCCSC ensures that features representative of salient regions are not ignored even if they are not discriminative.

Given a feature vector f and dictionary D with elements $D = [d_1, d_2, ., d_{r_D}]$, LCCSC

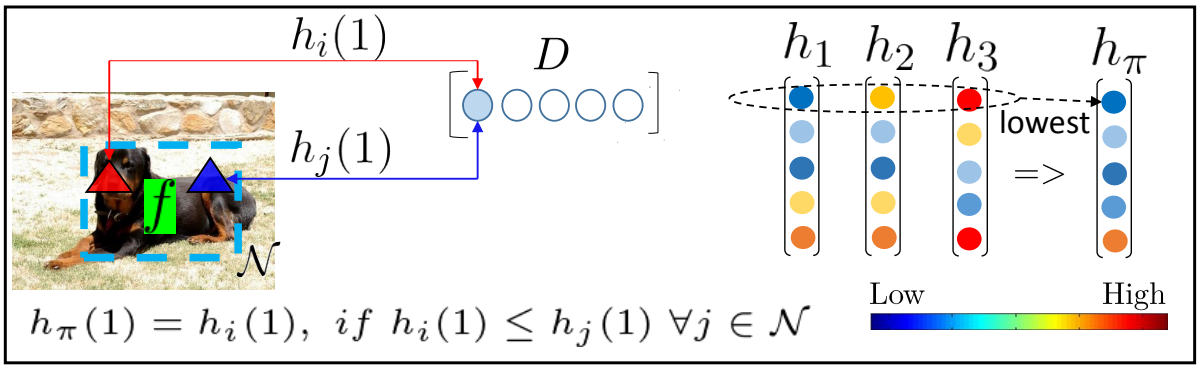


Figure 3.3: Illustration of computation of h_π.

coding searches for the codeword z that satisfies the following criteria:

$$\arg\min_z \|f - Dz\|_2 + \lambda_1 \|z\|_1 + \lambda_2 \|z \odot h_\pi\|_2 + \lambda_3 \sum_{j=1}^{n_c} (|\frac{\|z\|_0}{n_c} - \|z \odot col_j[\alpha]\|_0|); \quad (3.2)$$

where n_c is the number of object categories, $col_j[\alpha]$ is the j^{th} column of a binary matrix α (to be described later) and $\odot$ is element-wise multiplication. The first two terms are the conventional sparse coding of feature f with l_1 constraint [98]. The third term imposes locality constraint in the feature domain as well as in the spatial domain. It is motivated from LLC [26] in which the feature domain locality constraint is denoted by $||z \odot h||_2$, where $h = exp(\frac{dist(f,D)}{\sigma})$ and $dist(f, D)$ is a vector representation of the Euclidean distance between feature f and each atom (dictionary entry) of D. σ adjusts the rate of decay of the locality weight. In our formulation, the difference from [26] is in ensuring spatial-domain locality constraint also by considering not only a single feature f, but a set of features $f_\mathcal{N} : \mathcal{N} = 1, \ ... \ , s_\mathcal{N}$ in the spatial neighborhood of f. The *context* in LCCSC refers to the spatial neighborhood. Each vector f_i in $f_\mathcal{N}$ will have a corresponding vector h_i, which is a function of its Euclidean distance to the dictionary atoms. The minimum of this distance for all vectors in $f_\mathcal{N}$ constitutes h_π in eq. (3.2). i.e, n^{th} element of h_π is the minimum distance to the n^{th} atom in the dictionary among $f_\mathcal{N}$ which is computed by $h_\pi(n) = h_i(n),\ if\ h_i(n) \leq h_j(n)\ \forall j \in \mathcal{N}$ as illustrated in Fig. 3.3. Thus, h_π ensures that the third term draws lower penalty when the non-zero terms in the code z corresponds to the dictionary atoms for which the distance from the feature or its neighbors is minimum.

As stated earlier, in top-down saliency, we are interested in how useful a feature is in identifying the salient object rather than in its discriminative ability. For example, a wheel which is common to the two classes of motorbike and car may not appear in a discriminative dictionary learnt for image classification [103]. If the dictionary is formed by unsupervised k-means clustering of features from all categories [2], atoms corresponding to wheel may be an averaged version of motorbike and car wheels due to the possibility of both types of wheels falling into the same cluster. Both these scenarios are not suitable for top-down saliency. In top-down saliency since the goal is to assign a probability to a feature f based on its representativeness in each category, its code z should be such that the number of atoms that contribute to the non-zero values of z are distributed among all object categories. Based on the association of the feature to each class their values can differ. This corresponds to $||z||_0/n_c$ in the fourth term of eq. (3.2). The underlying assumption is that the dictionary is partitioned so that there is the same number of atoms from each category in a partition (see Fig. 3.4). In a practical situation an equal distribution of z values ($||z||_0/n_c$) is desired but not always possible and thus, the third term penalizes any deviation from the desired case. To this end, we define a binary matrix α of size $r_D \times n_c$ (r_D is the number of dictionary atoms) whose element α_{ij} is set to 1 if the i^{th} atom in the dictionary belongs to the j^{th} category. Now, consider two cases–one in which a code has two non-zero elements, both of which correspond to the same category (e.g., the first two elements are non-zero) and the other in which the two non-zero elements correspond to different categories. The penalty for deviation from desired, as represented by the fourth term term in eq. (3.2), is higher in the former case. This illustrates the third aspect of locality, viz., category-specific locality, where atoms *local* to each object category participate in the sparse coding process as shown by the filled circles of O_1, O_2 and O_3 in Fig. 3.4.

3.3.2 Approximate solution

A closed-form solution of eq. (3.2) is clearly not possible. Our approach is to ensure that the dictionary used in the sparse coding process satisfies the three locality constraints–feature domain, spatial domain (context) and category domain as formulated in the third and fourth terms of the objective function. The dictionary formation and subsequent

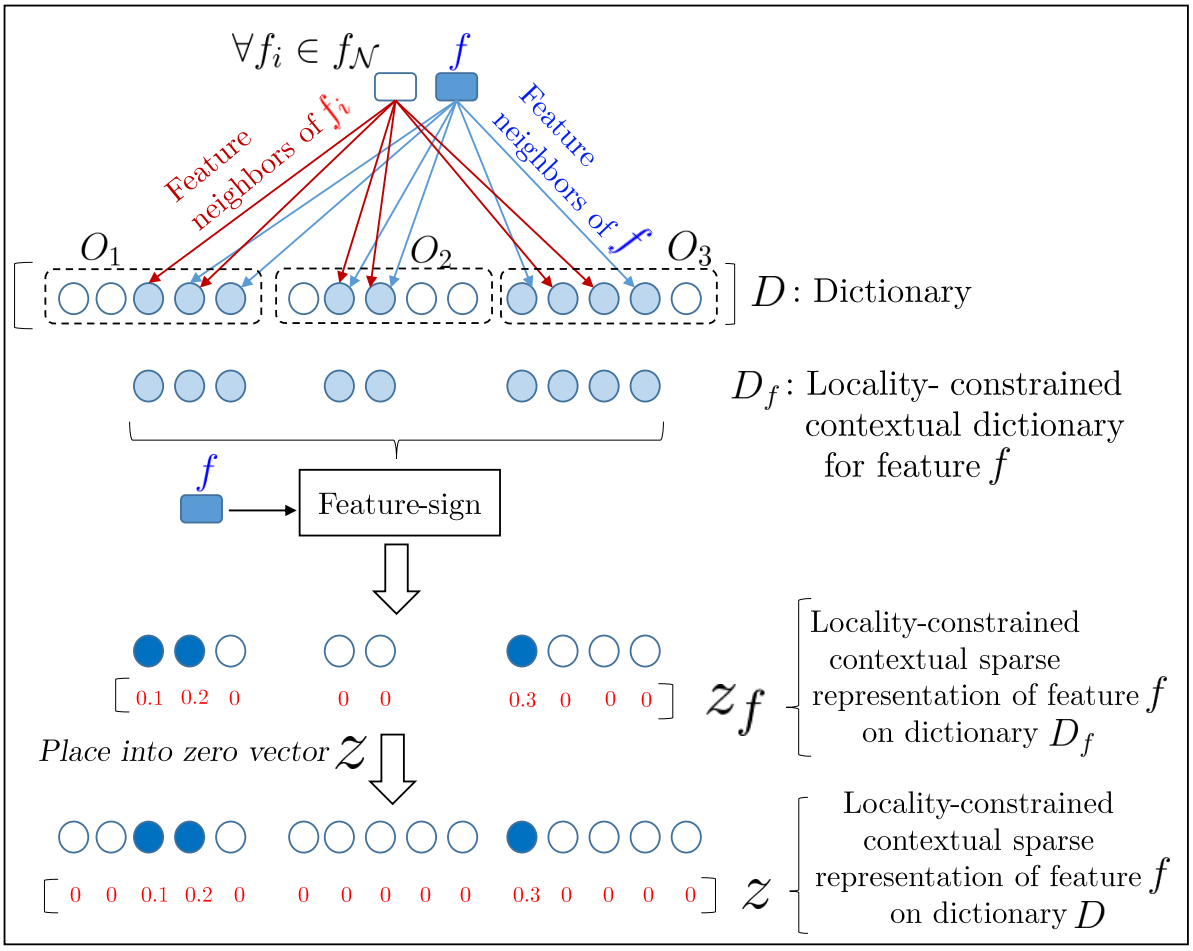


Figure 3.4: Illustration of approximate solution for locality-constrained contextual sparse coding for a feature f. Here f_i indicates a neighboring patch (best viewed in color).

sparse coding is shown in Fig. 3.4. As noted in the previous section, an initial dictionary is formed by concatenating separate sub-dictionaries formed by k-means clustering of SIFT features from each category. In Fig. 3.4, O_1, O_2 and O_3 are three such sub-dictionaries that make up the dictionary D. Category domain locality along with feature domain locality is enforced by picking k-nearest neighbors of a feature f from each sub-dictionary O_i using hierarchical k-nearest neighbor search. Next the spatial locality or contextual constraint involves consideration of the set of features $f_{\mathcal{N}}$ in the spatial neighbourhood of f and imposing category domain locality on each element of $f_{\mathcal{N}}$; this results in another set of atoms picked from D. $f_{\mathcal{N}}$ is made up of patches that overlap the patch containing f. These patches constitute the *context* for LCCSC.

Having picked atoms from D based on the three aspects of locality constraints, they

form a smaller locality constrained contextual dictionary D_f, which is used to encode feature f (Fig. 3.4) into z_f for minimizing the first two terms in eq. (3.2) through the feature-sign solver [98] with $\lambda_1 = 0.15$. The final sparse code z is formed by placing values in z_f in their respective positions in a vector z initialized to 0. This is to ensure that the vectors passed on for subsequent max-pooling and logistic regression are of the same size irrespective of the number of atoms that are picked to form D_f.

Computational complexity: Let r be the number of atoms per category and r_D be the number of atoms in dictionary D. Let r_f be the size of D_f and q_f be the number of non-zero terms in the final sparse code z. Traditional sparse coding on D using feature-sign solver has a computational complexity of $O(\hbar\, r_D) + O(r_D\, q)$. Here q is the number of non-zero terms in the code and $\hbar$ is the length of a dictionary atom ($\hbar = 128$). LLC reduces these computations to $O(r_D + \kappa^2)$ for κ-nearest neighbors (typically $\kappa = 5$). Computational complexity of our approach is in between those of sparse coding and LLC. Hierarchical k-nearest neighbor computations on each category dictionary separately results in $O(r)$ per category. These k-nearest neighbor computations for each category can be implemented in parallel to achieve multifold speed-up compared to nearest neighbor computations on D in LLC which has a complexity of $O(r_D)$. Feature-sign solver requires additional computations of $O(\hbar\, r_f) + O(q_f\, r_f)$. So, in a parallel implementation, the computational complexity of proposed LCCSC can be reduced to $O(r) + O(\hbar\, r_f) + O(q_f\, r_f)$. Due to the inherent parallelism in the framework, and much smaller size of feature-specific dictionary D_f as compared to the full dictionary D, a parallel implementation of the proposed LCCSC is faster than a parallel implementation of conventional feature-sign solver.

3.4 Contextual max-pooling for top-down saliency estimation

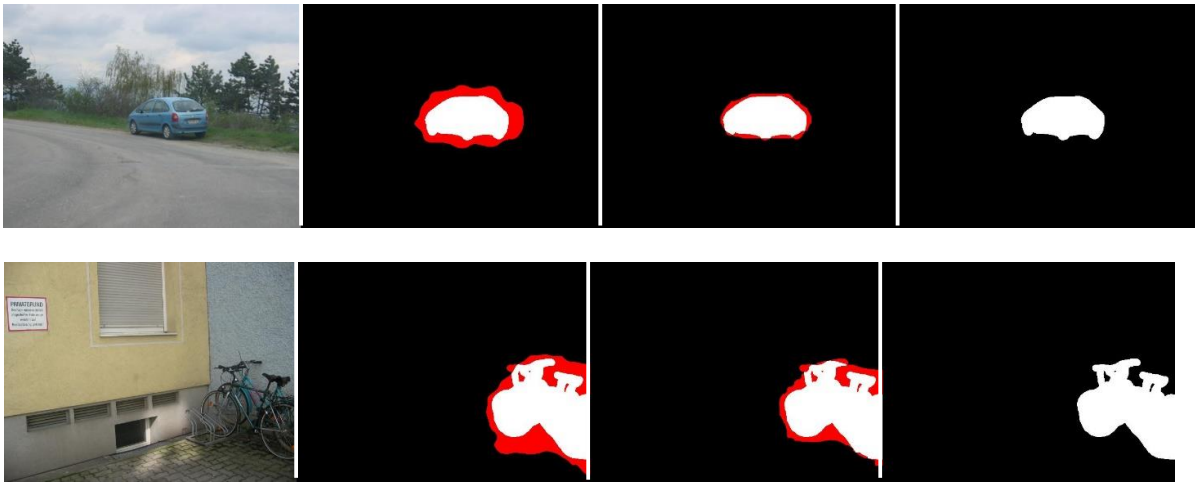
Context has an important role in deciding whether an image patch belongs to a particular object [25]. Contextual max-pooling refers to the representation of each patch by a max-pooled vector computed over its spatial neighborhood. The contextual max-pooling is done for LCCSC code vectors. The contextual neighborhood scale for max-pooling is

empirically set to 6. i.e., 6 patches surrounding the current patch in each direction are considered for max-pooling. To preserve the spatial layout, feature codes of these 169 $((2 \times 6 + 1) \times (2 \times 6 + 1))$ patches in the context are equally divided into a 3×3 spatial grid [25]. Separate max-pooling on each of these 9 regions followed by vertical concatenation of these max-pooled vectors forms the contextual max-pooled vector which represents the patch containing feature f. Contextual max-pooled vectors from object patches and from an equal number of negative patches are collected from all training images and a logistic regression is learnt for each object. These logistic regression models are the top-down saliency models for the object, and are used to estimate the saliency map. The proposed saliency inference on a test image is simple and fast. SIFT feature extraction followed by locality-constrained contextual sparse coding, contextual max-pooling and prediction of class conditional probability by logistic regression gives the saliency for a patch. Pixel-level saliency maps are computed from patch-level saliency values through our Gaussian-weighted interpolation.

3.5 Gaussian-weighted interpolation for pixel-level saliency map generation

We propose a simple Gaussian-weighted approach to compute pixel-level saliency from patch-level saliency. Upsampling by bicubic interpolation used by Yang & Yang [1] reduces the pixel-level precision rates at EER by 10% compared to its patch-level accuracy [101]. In order to estimate the saliency value at a given pixel, we consider those patches that contain that pixel. Let $[p(1),\ p(2), ... \ p(M)]$ be the saliency values for M image patches having centers at locations $(\mathsf{x}_1, \mathsf{y}_1), (\mathsf{x}_2, \mathsf{y}_2),\ ...., (\mathsf{x}_M, \mathsf{y}_M)$. Let g be the grid spacing i.e., the distance between adjacent patches. Since B is the width of the image patch, all image patches whose centers are located within $\frac{B}{2}$ radius from $(\mathsf{x}_l, \mathsf{y}_l)$ contain the pixel at $(\mathsf{x}_l, \mathsf{y}_l)$. Let $\Omega = [p_1, p_2, ...p_j]$ be the patches containing $(\mathsf{x}_l, \mathsf{y}_l)$ and $G(\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_i, \mathsf{y}_i) = exp^{-\frac{(\mathsf{x}_i - \mathsf{x}_l)^2 + (\mathsf{y}_i - \mathsf{y}_l)^2}{2g^2}}$ be the Gaussian weight of patch i having center at $(\mathsf{x}_i, \mathsf{y}_i)$ on this pixel. The saliency value at $(\mathsf{x}_l, \mathsf{y}_l)$ is computed as

$$p(\mathsf{x}_l, \mathsf{y}_l) = \frac{\sum_{i\ \epsilon\ \Omega} p(i) G(\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_i, \mathsf{y}_i)}{\sum_{i\ \epsilon\ \Omega} G(\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_i, \mathsf{y}_i)}. \tag{3.3}$$



(a) (b) (c) (d)

Figure 3.5: Illustration of Gaussian-weighted interpolation (best viewed in color). (a) Input image, (b) bicubic interpolation, (c) proposed Gaussian-weighted interpolation and (d) ground truth. White indicates true positive pixels, black true negatives, and red indicates false positves and false negatives.

Fig. 3.5 shows car and bike input images. From the same patch-level saliency values generated by proposed method, pixel level saliency maps are generated by bicubic interpolation and proposed Gaussian weighted interpolation. The saliency maps generated in both cases are binarized using a common threshold and misclassified pixels are shown in red (false positive+false negative). The proposed interpolation step results in better pixel-level saliency map with lesser false detections (Fig. 3.5(c)) compared to bicubic interpolation used by [1] (Fig. 3.5(b)) which spreads to the background pixels.

3.6 Experimental results

We test our method on two challenging datasets - Graz-02 and PASCAL VOC-07. The same parameters are maintained in both datasets. For fair comparison, the same experimental set-up as [1] is followed, i.e., dense SIFT features are extracted from 64×64 image patches with a grid spacing of 16 pixels. For each object category, a sub-dictionary of 512 atoms is formed through k-means clustering of features extracted from positive

training patches. A patch is considered as positive if at least 25% of the pixels belong to the object category.

3.6.1 Graz-02 dataset

Graz-02 has 3 object categories- bike, car and person, each having 300 images with pixel-level object annotations and an additional 65, 120 and 11 images without object annotations in each category respectively. Apart from this 380 background images are also present. As in [1, 6], from each category, 150 odd numbered images are used for training and remaining 150 for testing. Saliency models for each object category are tested on 300 test images (150 test images of the object and 150 background test images) and precision rate at Equal Error Rate (EER) is determined. The saliency maps are thresholded at 100 levels between 0 and 1 and a precision recall graph is drawn. The EER refers to the point at which the precision is equal to the recall.

Table 3.1(a) compares the patch-level results of our method with other top-down saliency models. The proposed method is called as LCCSC-pooled, where pooled indicates contextual max pooling of the code. LLC-pooled and SC-pooled refer to our framework except that LCCSC is replaced by LLC and SC respectively. DSD and SUN results are reported in [1]. In all the 3 classes we achieve state-of-the-art results. As illustrated in Fig. 3.6, when LCCSC in the proposed framework is replaced by LLC (LLC-pooled) or SC (SC-pooled), the performance deteriorates, most notably in the car category because of background features that are similar to car features. Since LLC and SC do not consider context while coding, these features resulted in false detection in this category (Fig. 3.6(b, c)). The smooth regions within the person are not detected by LLC and SC (Fig. 3.6(b, c)), but using context along with the other two locality constraints in LCCSC helped generate better saliency maps (Fig. 3.6(d)). For fair comparison, LLC-pooled and SC-pooled are implemented on a dictionary of 2048 atoms formed by k-means clustering. Matlab simulations using parallel processing toolbox shows that LCCSC achieved 27% speed-up compared to conventional feature-sign based sparse coding on D (0.54 sec for LCCSC versus 0.74 sec for SC to encode 1000 features).

Table 3.1(b) compares pixel-level results of the proposed model with recent top-down saliency approaches, a bottom-up approach [32], and with two results in object segmentation [91, 104]. Even-though the saliency values are estimated at patch level, the Gaussian

Table 3.1: Precision rates (%) at EER on Graz-02. (a) Patch level results and (b) pixel level results.

(a) Patch-level

	Bicycle	Car	Person	*Mean*
DSD [8]	62.5	37.6	48.2	49.43
SUN [61]	61.9	45.7	52.2	53.27
Yang and Yang [1]	80.1	68.6	72.4	73.7
LLC-pooled	81.91	71.3	70.9	74.71
SC-pooled	83.15	72.81	72.06	76.01
LCCSC-pooled	**83.46**	**75.97**	**73.13**	**77.52**

(b) Pixel-level

	Bicycle	Car	Person	*Mean*
Objectness [32]	53.5	48.3	43.5	48.43
Aldavert *et al.* [91]	71.9	64.9	58.6	65.13
Khan and Tappen [101]	72.1	-	-	-
Marszalek and Schmid [104]	61.8	53.8	44.1	53.23
Yang and Yang [1]	62.1	60.0	62.0	61.46
Kocak *et al.* [6]	73.9	68.4	**68.2**	70.16
LCCSC-pooled (*upsampling of [1]*)	73.41	68.6	61.25	67.75
LCCSC-pooled (*Gaussian-weighted interpolation*)	**76.19**	**71.2**	64.13	**70.49**

weighted interpolation yields the best reported results at pixel-level, which is better than models that estimate saliency directly at pixel-level [6]. The seventh row shows the EER of pixel-level saliency map using upsampling of [1] and the last row shows the results for Gaussian-weighted interpolation. The improvement by about 3% indicates the benefits of the proposed interpolation scheme. It is to be noted that we achieve this performance by using simpler feature coding and contextual max-pooling in comparison with computationally complex dictionary learning and graph-based approaches of [1, 6]. Since conditional random field (CRF) used in these models are built on sparse codes, increasing the dictionary size will drastically increase the computational complexity by many times which is practically not feasible. Contextual max-pooling of LLC codes [26] are used by [25] to generate the saliency map. They use separate dictionaries of 1024 atoms for each category as opposed to a common dictionary for all categories. Instead of precision at EER, they report average precision at pixel-level. Our mean average precision is 75.5 (bike- 83.1, cars-75.7, person-68.3) which is much higher compared to their 62.1 (bike-69.1, cars-58.0, person-59.2).

3.6.2 PASCAL VOC-07

PASCAL VOC 2007 is a challenging dataset consisting of 20 different categories with some images having objects from multiple object categories. As in [1], all models are evaluated on the entire 210 segmentation test images irrespective of the presence or absence of target. Since there are only 422 segmentation images to train 20 categories,

(a) (b) (c) (d)

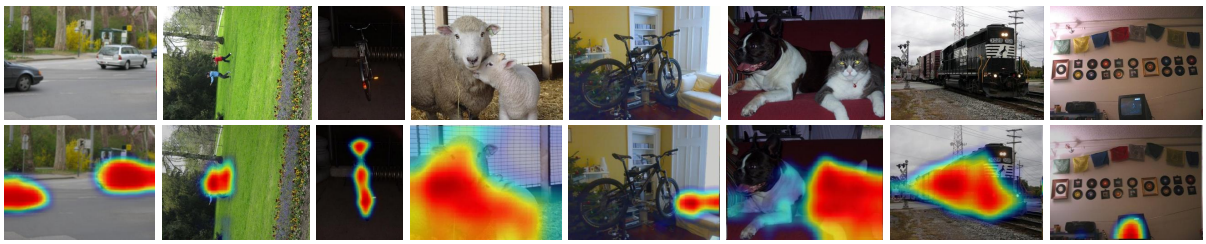
Figure 3.6: Comparison of saliency maps generated using LCCSC against LLC and SC coding. (a) input image, (b) LLC-pooling, (c) SC-pooling and (d) LCCSC-pooling.

Table 3.2: Patch-level precision rates at EER on PASCAL VOC-07

	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	
Yang and Yang [1]	**15.2**	**39**	9.4	5.7	3.4	22.0	30.5	**15.8**	**5.7**	8.0	
Proposed	13.3	33.2	**22.1**	**11.2**	**8.6**	**33.5**	**37.2**	14.3	3.9	**22.3**	
	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	T.V Monitor	Mean
Yang and Yang [1]	11.1	12.8	10.9	23.7	**42.0**	2	20.2	10.4	24.7	10.5	16.15
Proposed	**23.0**	**14.9**	**25.0**	**30.6**	38.9	**16.4**	**36.3**	**18.3**	**29.2**	**36.3**	23.4

we use additional images from the object detection challenge as in [1] (nearly 150 images per category). We outperform [1] in 15 out of 20 classes as shown in table 3.2. On averaging across all classes, in patch-level, we achieve a mean precision rate at EER of 23.4% which is better than 16.15% of [1]. With the help of proposed Gaussian weighted interpolation, we achieve a mean precision rate at EER of 17.65% in pixel-level. Khan and Tappen [101] report precision of 8.5% only for cow category for which our method gives 22.66%. We do not compare with [6] since they manually assign an all zero saliency map, if the object of interest is not present in the test image.

Fig. 3.7 shows qualitative results on challenging test images from Graz-02 (Cars, Person and Bike) and PASCAL VOC-07 (Sheep, Sofa, Cat, Train and TV monitor) datasets. Proposed method could perform well even on a rotated image (Person). The sofa was correctly detected even though the image is dominated by bicycle which is another category in the dataset. Similarly, cat is assigned with higher saliency, in spite of the presence of dog (another category) in the image. TV monitor is correctly identified



Cars Person Bicycle Sheep Sofa Cat Train TV monitor

Figure 3.7: Top row: Input images from Graz-02 and PASCAL VOC-07 datasets. Bottom row: Our LCCSC-pooled results.

in spite of the presence of visually similar structures within the image.

To compare with pixel classification accuracy of [91], our pixel-level saliency maps are threshold at 0.5, so that pixels having a saliency value above 0.5 is treated as belonging to that object category otherwise background category. A pixel is assigned to the class having highest saliency value in the cases where more than one category produces saliency value above threshold. This simple thesholding of saliency map gives an average pixel classification accuracy of 32.33% which is far superior as compared to 23% of dedicated class segmentation approach [91].

3.7 Conclusion

In this chapter we propose a simple and efficient feature coding strategy, named LCCSC, specifically for top-down salient object detection. The spatial, feature and category-domain locality constraints of LCCSC ensure that the features which are nearby in the spatial domain or in the feature domain are assigned with similar codes. The contextual max-pooled vectors of positive and negative image patches are used to train a logistic regression classifier that estimates patch saliency. The proposed Gaussian-weighted interpolation produces better pixel-level saliency map from patch-level saliency values. The top-down salient object detection framework proposed in this chapter has the following limitations: (i) the length of the LCCSC feature code is dataset dependent and increases proportional to the total number of object categories in the dataset. (ii) A fixed patch size is used to extracted SIFT features which introduces false positives when the

patch size is too small relative to the object size. Image classifiers used in the remaining chapters address the latter issue by indicating the object presence in an image.

Chapter 4

A Joint Framework for Top-down Salient Object Detection and Image Classification

4.1 Introduction

The salient object detection approaches like [1, 6, 25] use neighborhood and visual priors but they lack spatial prior indicating the image regions where the object is likely to appear (Fig. 1.2). On the other hand, the image classification approaches like [2, 26] have multi-scale spatial information, but lack neighborhood prior. This chapter is motivated by a need for top-down salient object detection and image classification joint framework that uses spatial, neighborhood and visual priors. Just as a sparse coded spatial pyramid max-pooling (ScSPM) image classifier can be improved by improving the discriminative quality of the dictionary [1], saliency maps can be improved using the image classifier by leveraging information about presence of the object in an image. Thus, we propose an interconnected and mutually benefiting saliency-classification framework based on [1] and [2], that also helps reduce the computational cost. The conditional random field (CRF) in the saliency module of [1] is built on sparse codes whose computation cost increases proportional to the length of feature code. In order to reduce computations, the LCCSC proposed in the previous chapter is not used in our saliency module, instead, conventional sparse coding is used as in [1].

The main processes in our joint framework are:

Category-aware sparse coding. ScSPM-based image classifiers are known to perform better [105] if the sparse codes are computed on a global dictionary whose atoms are representative features from all categories. Since the saliency models are developed based on sparse codes using individual object dictionaries, it would be computationally expensive to form another global dictionary using dictionary learning [2] or by k-means clustering of thousands of patches from all categories and recompute sparse codes with respect to that dictionary. In this chapter, we propose a category-aware sparse coding that utilizes the discriminative dictionaries already learned during saliency modeling, thereby enabling tight coupling between saliency models and the image classifier. This strategy helps in reducing the computational cost of feature coding for the classifier, while improving its classification accuracy.

Classifier-guided saliency model training. The image classifier used to update the saliency model is trained using a training set and validated using a validation set. The misclassified images during validation indicate that the features of those images are not represented adequately in the discriminative dictionary. The top-down saliency module is updated using such images. This classifier-guided saliency model training helps to improve not only the top-down saliency component, but also the image classification accuracy.

Saliency-weighted max-pooling. Conventional ScSPM image classifier [2] is blind to max-pooled vectors from the spatial pyramid blocks that contain an object and from those that do not contain any object. Hence, ScSPM often performs poorly in the presence of high background clutter. A novel saliency-weighted max-pooling proposed in this chapter helps improve the performance. When a saliency model of a particular class is applied to an image, it highlights those patches that are likely to contain object parts belonging to that class. High saliency values appearing in a negative training image indicate false positive patches. Weighting corresponding max-pooled sparse codes with this high value and training the SVM [106] with a negative label will help the image classifier to label similar test images as negative.

Saliency refinement. The top-down saliency approaches of [1, 6] compute saliency of an image region without ascertaining whether the target object is present or not. Due to

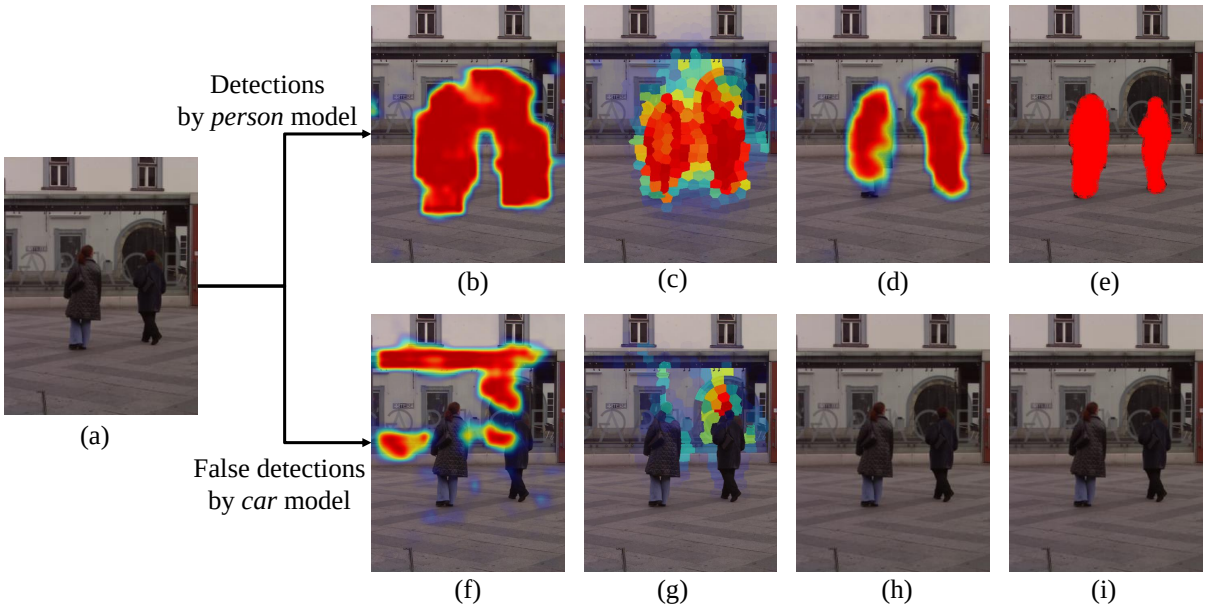


Figure 4.1: Comparison of saliency maps for an (a) input image by (b) Yang and Yang [1], (c) Kocak *et al.* [6], (d) proposed method and (e) ground truth on *person* model. False detections on applying *car* model of (f) Yang and Yang [1] and (g) Kocak *et al.* [6] are eliminated by (h) the proposed method resulting in a saliency map that is visually similar to (i) the ground truth.

this, they often end up in producing false detection on negative images, which reduces the accuracy. In our framework, the image classifier is used to quantify the likelihood of the presence of an object and to refine the saliency map using a novel saliency refinement.

A preview of the effectiveness of our joint framework for top down saliency and image classification for the task of salient object detection is shown in Fig. 4.1. Fig. 4.1(a) shows an image containing two persons in a cluttered background. The person model of [1] and [6] are unable to distinguish between the persons due to their proximity in the image as seen in Fig. 4.1(b) and (c) respectively. Our framework produces a saliency map (Fig. 4.1(d)) closer to that of ground truth (Fig. 4.1(e)). Fig. 4.1(f) and (g) show the false detections by [1] and [6] respectively when their car saliency models are applied on the input image. As seen in Fig. 4.1(h), with the help of the classifier in our framework, we avoid these false detections.

In summary, the key idea of the proposed method is to add the image classification module both to update the saliency models, and to refine the saliency map for a given test image. On the other hand, the saliency inferred from the saliency models can also

be used to improve the accuracy of the image classifier. The major contributions of this chapter are the following:

1. A novel category-aware sparse coding strategy, which is computationally more efficient as compared to conventional sparse coding.

2. A novel framework to train saliency models with the help of a classifier.

3. A novel saliency-weighted image classification framework.

4. Refinement of saliency maps using image classifier confidence.

4.1.1 Brief review on top-down saliency of [1]

In [1], dense SIFT features are extracted from regular, rectangular gray-scale image patches and their sparse representations are initially computed with a dictionary D_n formed by k-means clustering. The dictionary is formed from cluster centroids and it represents the most representative patches of an object category. The use of sparse coding of SIFT features results in a more compact and discriminative representation which helps to model feature selectivity for saliency map. Using these sparse codes as latent variables, a conditional random field is learned. The CRF node weight v_n is initialized with a binary SVM classifier weight learned on these sparse codes and pairwise energy is set to 1. The dictionary and CRF weights are jointly learned in 20 iterations using max-margin framework. Finally, loopy belief propagation is used to infer saliency values on test images. In the proposed method, the number of iterations is set to 10 to save computations because the improvement in accuracy of the model after 10 iterations is insignificant compared to its computation cost.

One major drawback of [1] is that its focus is on distinguishing objects from background and not from other objects, resulting in a large number of false positives as shown in Fig. 4.1(f). Our method overcomes this limitation by integrating a classifier that is trained on novel category-aware sparse codes, which also results in considerable savings in computation, especially when dealing with large datasets.

4.1.2 Brief review on ScSPM image classification [2]

In a ScSPM-based classifier [2], dense SIFT features are extracted from gray-scale image patches. SIFT features from training images are used to form a global dictionary. The SIFT features of an image are sparse coded using this dictionary. The spatial distribution of the features in the image is encoded in the max-pooled image vector through a multi-scale max-pooling operation of the sparse codes on a 3-level spatial pyramid [73]. The max pooled feature is more robust to local transformations than mean statistics in histogram. Biophysical evidence in visual cortex (V1) [107] also establishes the use of max-pooling. Image-label pairs of training images are used to train a linear binary SVM classifier.

4.2 The proposed joint framework

Fig. 4.2 shows an overview of the proposed joint framework for salient object detection and image classification. Similar to the original framework of [1] and [2], we only use SIFT features extracted from gray-scale images. From every image I^i, image patches with a fixed size and grid spacing are extracted. For each patch j, its dense SIFT feature $f^{i,j}$ and ground-truth $l_n^{i,j} \in \{\text{-}1, 1\}$ are computed, where -1 and 1 denote the absence or presence, respectively, of object n in patch j. We split the training images (Train1+Train2) into Train1 (training set) and Train2 (validation set), to save computations by avoiding the update of saliency model with a training image whose features are already well represented in the saliency model. The discriminative dictionary D_n for each object category n is initialized with the centroids of the clusters formed by k-means clustering applied on positive SIFT features.

Similar to [1], the sparse codes of SIFT features with D_n are used as latent variables in our saliency models. The sparse codes of positive and negative patches from Train1 images are used to learn a linear SVM and the SVM weights are used to initialize the CRF node weight v_n. The pair-wise energy of the CRF in a four connected graph is set to 1 as in [1]. Following [1], for each category n, the dictionary D_n and the CRF weight v_n are jointly updated using Train1 images for 10 iterations. The updated dictionary D_n represents the most representative patches of the n^{th} object category, and the CRF weights learnt on sparse codes represent our saliency model. Since sparse code is used as a latent

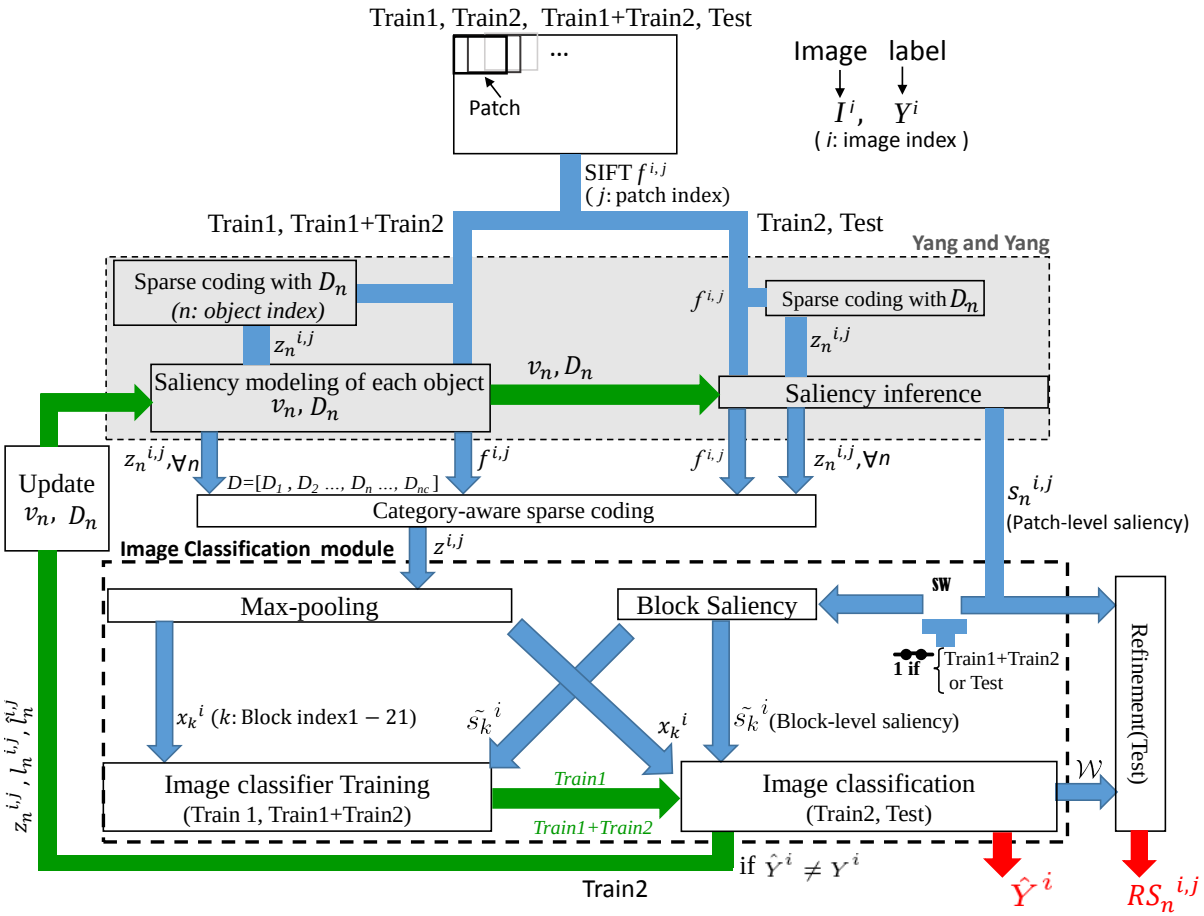


Figure 4.2: Overview of the proposed framework for classifier-guided salient object detection. The shaded region represents the framework of [1]. **SW** is connected only during training and testing of saliency-weighted classifier. **SW**=1 and **SW**=0 indicate that the switch is connected and disconnected, respectively. Green arrows indicate output of training stages and red arrows indicate the final saliency map and classifier output on a test image.

variable for saliency modeling, the sparse code $z_n^{i,j}$ of $f^{i,j}$ with D_n, $n \in \{1, 2, ..., n_c\}$, is recomputed in each of these 10 iterations.

Following 10 iterations, the *classifier-guided saliency model update* process is iterated 2 times (the green colored feed-back arrows in Fig. 4.2). An image classifier that uses D_n and $z_n^{i,j}$ of all n_c object categories, and trained on Train1 images is used to choose images from Train2 to update saliency models. The image classifier performs better if the sparse representations are on a global dictionary D than n_c separate sparse representations on n_c discriminative dictionaries D_n[105]. So, we propose a *category-aware*

sparse coding strategy that reuses each D_n and $z_n^{i,j}$ computed during saliency estimation for n_c categories to produce a global representation $z^{i,j}$ of $f^{i,j}$. This category-aware sparse representation helps to reduce the additional computational requirement for image classification compared to the conventional sparse coding with D [2].

The category-aware representation of all patches in image i is max-pooled over a multi-scale spatial pyramid and the max-pooled vector in each block k of the spatial pyramid is represented as ${x_k}^i$. The max-pooled vectors from all 21 blocks are vertically concatenated followed by l_2-normalization to form the max-pooled image vector X^i. The X^i from all images in Train1 and their corresponding image labels Y^i are used to train a linear SVM with weight W and bias $bias$ ($\mathbf{SW} = 0$ in Fig.4.2). The classifier is used to predict the label $\hat{Y}^i$ of Train2 images. A misclassification $\hat{Y}^i \neq Y^i$ is an indication that the corresponding object model has not been learned by the CRF comprehensively enough to include its appearance as in the misclassified image. Thus, the classifier selects those images with which the saliency model v_n needs to be updated. Consequently, the corresponding object dictionary D_n is also updated, which, in turn, refines the global dictionary D formed by concatenating the object dictionaries. We use max-margin approach to identify the most violated constraints for the misclassified image and to update D_n and v_n accordingly (classifier-guided saliency update, $\mathbf{SW} = 0$ in Fig.4.2).

After the above mentioned *classifier-guided saliency update*, we improve the image classifier using saliency maps. The proposed saliency-weighted max-pooling operation ($\mathbf{SW}$=1 in Fig.4.2) weights the max-pooled vectors ${x_k}^i$ of each block k with its block saliency ${\tilde{s_k}}^i$. We infer the saliency maps from all training images Train1+Train2 and compute their saliency-weighted max-pooled $\tilde{X}^i$ vectors, which are used to train a linear SVM $(\tilde{W}, \tilde{bias})$. This *saliency-weighted image classifier* $(\tilde{W}, \tilde{bias})$ is applied on a test image to refine the saliency map, and to classify it. At the end of the training stage, we get (i) a saliency model with updated CRF weight v_n and dictionary D_n and (ii) a saliency-weighted image classifier $(\tilde{W}, \tilde{bias})$.

For test images, the saliency of each category, $s_n^{i,j}$ is estimated by CRF inference using loopy belief propagation as in [1]. These saliency maps are refined using the posterior probability $\mathcal{W}$ estimated from the saliency-weighted classifier as explained in section 4.5. This *refined saliency* $RS_n^{i,j}$ reduces the false detections in the saliency as shown

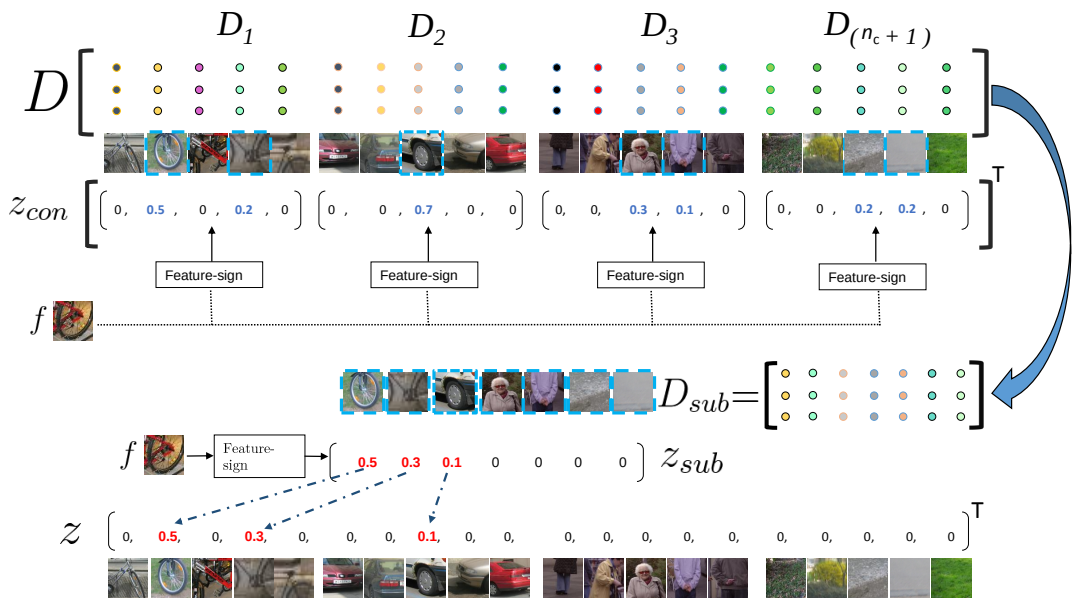


Figure 4.3: Illustration of category-aware sparse coding for classification (best viewed in color).

in Fig. 4.1(h) and supported quantitatively in our experimental results. In summary, the proposed approach is able to simultaneously identify and locate the object categories present in a test image.

4.3 Category-aware sparse coding

The category-aware sparse coding reuses the object dictionary D_n and corresponding spare codes $z_n^{i,j}$ computed by the saliency component. The dictionary update in the saliency component improves the discriminative quality of the object dictionary D_n, and a global dictionary D formed by concatenating updated object dictionaries of all categories helps improve the image classification performance. Moreover, this approach reduces the additional computations incurred by forming a global dictionary D using k-means clustering of thousands of patches from all categories followed by recomputation of sparse codes with respect to that dictionary. Spatial pyramid max-pooling of the category-aware sparse codes (ScSPM) is used in our classification module. Tight coupling between saliency modeling and image classification is obtained through the proposed category-aware sparse coding. It is to be noted that the following discussion

pertains to computation of category-aware sparse code $z^{i,j}$ of a feature $f^{i,j}$. For simplicity, we drop the superscripts i and j in this section. Conventional sparse representation of a given feature f aims to achieve the minimum reconstruction error while using sparse number of atoms from D, i.e.,

$$z = \arg\min_{z} \|f - Dz\|_2 + \lambda \|z\|_1 . \tag{4.1}$$

Since the objective of our image classifier is to improve saliency estimation, we introduce a category-aware constraint to the feature coding for classification, resulting in a novel category-aware sparse code z. i.e., the category-aware sparse code z aims to achieve minimum reconstruction error while representing the feature f with respect to each category dictionary D_n as well as to the global dictionary D.

4.3.1 Formulation

Let D be the global dictionary formed by concatenating the individual object dictionaries, i.e. D has a structure $[D_1,\ D_2,\ \ldots D_{(n_c+1)}]$, where D_n represents each object dictionary and $(n_c + 1)$ includes the n_c object categories and the background class. The saliency model for an object category n and its corresponding dictionary D_n is learned as described in [1]. The dictionary for the background class is formed by k-means clustering of background patches; thus, background features also have a sparse representation. The size of D is $\hbar \times r_D$ where $r_D = (n_c \cdot r + r_{bg})$, $\hbar$ is the dimension of the feature vector, r and r_{bg} are the number of atoms in the dictionary for each object class and the background, respectively.

The objective function for category-aware sparse coding is

$$z = \arg\min_{z} \|f - Dz\|_2 + \lambda_1 \|z\|_1 + \lambda_2 \sum_{n=1}^{n_c} (\|f - DC_n z\|_2 + \lambda_3 \|C_n z\|_1), \tag{4.2}$$

where the first two terms are the conventional sparse coding of feature f with l_1 constraint and the third term imposes our category-aware constraint. C_n is a selection matrix that selects the atoms of a particular object category n from D. C_n is derived from a zero matrix of size $r_D \times r_D$ by replacing its m^{th} diagonal element with 1, if the m^{th} atom in

the dictionary D belongs to the category n. For example if D contains 6 atoms ($r_D = 6$) with the third and fourth atoms of D belonging to category n, then

$$C_n = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $C_n z$ selects the elements of z that belongs to category n. $\|f - DC_n z\|_2$ represents the reconstruction error for the sparse representation of feature f with category dictionary D_n (non-zero elements of DC_n= D_n) and the l_1-norm constraint $\|C_n z\|_1$ ensures that only few atoms of D_n is used in this sparse representation. Imposing a constrain that 'the reconstruction error of feature code should be minimum with each object category dictionary' will help to identify the dictionary atoms from each sub-dictionary, which can better reconstruct the image patch.

4.3.2 Approximate solution

Since a closed-form solution of eq. (4.2) is not possible, we propose a computationally efficient approximate solution. Since the sparse codes z_n of each feature f with respect to object dictionaries D_n are already available from the saliency module, we develop a strategy to reuse these codes which is computationally more efficient than conventional sparse coding of the feature f with global dictionary D.

The dictionary formation process and the category-aware sparse code vector z are shown in Fig. 4.3. The proposed solution minimizes eq. (4.2) in two steps. First, the sparse codes z_n, $\forall n \in \{1, ...n_c\}$ of each feature f with all category dictionaries D_n $\forall n \in \{1, ...n_c\}$ are computed using feature-sign solver [98], which reduces the third term in eq. (4.2), i.e $\lambda_2 \sum_{n=1}^{n_c} (\|f - DC_n z\|_2 + \lambda_3 \|C_n z\|_1)$.

Let z_n be the sparse code for feature f evaluated with respect to object dictionary D_n. We form a vector z_{con} by vertically concatenating the sparse codes of f obtained for each D_n as shown in Fig. 4.3. The non-zero terms in z_{con} point to atoms in D that contribute to minimizing the reconstruction error of f and hence, can effectively represent it. We

pick these atoms from D to form the classifier dictionary D_{sub} and generate sparse code vector z_{sub} for f on D_{sub} to minimize the overall objective function.

Since the number of non-zero terms in z_{con} is small, the number of atoms in D_{sub} is much lesser than in D resulting in lesser computation for generating z_{sub}. Since the number of atoms in D_{sub} is different for every feature and the ScSPM classifier needs a dictionary of the same size as D, we need to represent z_{sub} in a vector with the same length as z_{con}. Since D_{sub} is formed by picking atoms from D, the elements of z_{sub} can be placed in their respective locations of the category-aware sparse code z having same length as the number of atoms in D, initialized with a zero vector. Let a sparse code element z_{sub_p} in the p^{th} position in the code vector z_{sub} correspond to atom d_p in D_{sub}. If d_p is the m^{th} atom in D, then the code z_{sub_p} is placed in the m^{th} position of category-aware sparse code z. The number of atoms in D_{sub} for PASCAL VOC 2007 dataset [93] ranges from 300-400, which is much smaller compared to a typical classifier dictionary size for the same dataset, which ranges between 8000 and 12000 [75].

4.3.3 Computational complexity

We use the Feature Sign sparse solver (FS) [98] whose time complexity to encode feature f is $O(\hbar\, r_D) + O(T_f\, r_D)$ when the dictionary size is $\hbar \times r_D$ and T_f is the sparsity of the code (number of non-zero elements in the code). Using FS solver on the global dictionary D (obtained by concatenating category dictionaries) will result in a computational complexity of $O(\hbar\, r_D) + O(T_f\, r_D)$, where $r_D = n_c \cdot r + r_{bg}$, which is very large for datasets with a large number of classes (n_c). One possible approach to reduce the computation is to reduce r_D by forming a smaller global dictionary by clustering of features from all categories (instead of concatenating category dictionaries). However, it may result in loss of fine-grained information that helps in distinguishing similar classes.

In the proposed framework, for every feature, FS solver is used in two stages - first, during saliency estimation, with respect to dictionaries D_n and then with respect to sub-dictionary D_{sub}. Since there is no dependency between the first stage solvers, a parallel implementation can effectively result in a time complexity of $O(rk) + O(rc_n)$ per feature for each object class where c_n is the sparsity of the sparse code with D_n. Since r_D is nearly $(n_c + 1)$ times larger as compared to r, parallel implementation of proposed framework

is $(n_c + 1)$ times less complex as compared to the time complexity of conventional sparse coding on D. For the same sparse penalty λ, we have observed that c_n in each category code is less than T_f, the sparsity with the global dictionary D. For sparse coding using D_{sub}, each feature requires an additional time complexity of $O(r_{sub}\hbar)+O(r_{sub}s_{sub})$, where s_{sub} is the sparsity of z_{sub}. Since sparse codes are already available from the saliency estimation stage, the second round of sparse coding computations are the only additional computations required for sparse coding of classification, resulting in significant savings in computation.

4.4 The image classification module

Classifiers are used for saliency model training, saliency map refinement and for image classification. The proposed framework tightly couples these processes.

4.4.1 Classifier to train saliency model (classifier feedback)

The classifier uses 3-level spatial pyramid max-pooling [2] of category-aware sparse codes. i.e, an image i is divided into 21 blocks. Each block k is represented by a single max-pooled vector x_k^i of dimension $(n_c \cdot r + r_{bg}) \times 1$ formed by element-wise maximum of the category-aware sparse code vectors $z^{i,j}$ in that block. Each image is represented with a max-pooled vector X^i of dimension $21(n_c \cdot r + r_{bg}) \times 1$, formed by vertical concatenation of the max-pooled vectors from each block.

Let $\{\mathbf{X}^i, Y^i\}$, $i \in$ Train1 be the training data where X^i is the l_2-normalized vector from image i and $Y^i \in \{1, -1\}$ indicates the presence or absence of the target object in that image i. A one-vs-rest (binary) linear SVM classifier with SVM weight W and bias *bias* is trained on half of the training images Train1 by minimizing following objective function [106]

$$\arg\min_{W} \|W\|^2 + C \sum_{i \in \text{Train1}} max\,(0, 1 - Y^i(W^\top X^i + bias)\ ), \tag{4.3}$$

where C is the cost of constraints violation.

The max-pooled vector of each image i from the other half of the training set, Train2, are used to validate the classifier using $f(X^i) = W^\top X^i + bias$. Correct classification of

image i is indicated by $(f(X^i) \cdot Y^i) > 0$. The saliency model corresponding to the misclassified object is updated through refinement of CRF weights and dictionary (classifier feedback). For example, if a bike image (n=bike) is misclassified by the bike classifier, the bike saliency model is updated using this image. Let $Z_n^i = [z_n^{i,1},\ z_n^{i,2} \dots z_n^{i,t}]$ be the set of all sparse codes of a misclassified image i using D_n. and $L_n^i = [l_n^{i,1},\ l_n^{i,2}, \dots l_n^{i,t}],\ l_n^{i,t} \in \{-1, 1\}$ be the ground truth label for target n presence in each patch. Here t is the total number of patches in the image. If $\hat{L}_n^i$ are the labels predicted using category n CRF model, the loss function for the image is [1]

$$\beta(v_n, D_n) = E(\hat{L}_n^i, Z_n^i, v_n) - E(L_n^i, Z_n^i, v_n), \tag{4.4}$$

where E is the energy function of CRF built on a four connected graph conditioned on sparse codes Z_n^i (for details please refer to [1]). The ground truth energy $E(L_n^i, Z_n^i, v_n)$ is less than any other energies $E(L, Z_n^i, v_n)$ by a large margin $\Delta(L, L_n^i)$. i.e, $E(L_n^i,\ Z_n^i,\ v_n) \leq E(L,\ Z_n^i,\ v_n) - \Delta(L,\ L_n^i)$ [1]. Here L represents set of binary labels assigned to patches in the the image i and $\Delta(L, L_n^i) = \sum_{j=1}^{t} \mathbb{I}(l^j, l_n^{i,j})$ indicates the margin function, where $\mathbb{I}$ is an indicator function which is 1 when $l^j \neq l_n^{i,j}$ and 0 otherwise. The most violated constraints are identified by solving

$$\hat{L}_n^i = \arg\min_{L} E(L,\ Z_n^i,\ v_n) - \Delta(L,\ L_n^i), \tag{4.5}$$

which is used to update the CRF weights v for object O as [1],

$$v_n = v_n - \rho_0 \frac{\partial \beta}{\partial v_n}, \tag{4.6}$$

where $\rho_0 = 10^{-3}$ is the learning rate.

Following [1], the updated CRF weight is used to update the object dictionary D_n for a given object n (e.g. bike) as

$$D_n = D_n + \rho_0 \frac{\partial \beta}{\partial D_n}. \tag{4.7}$$

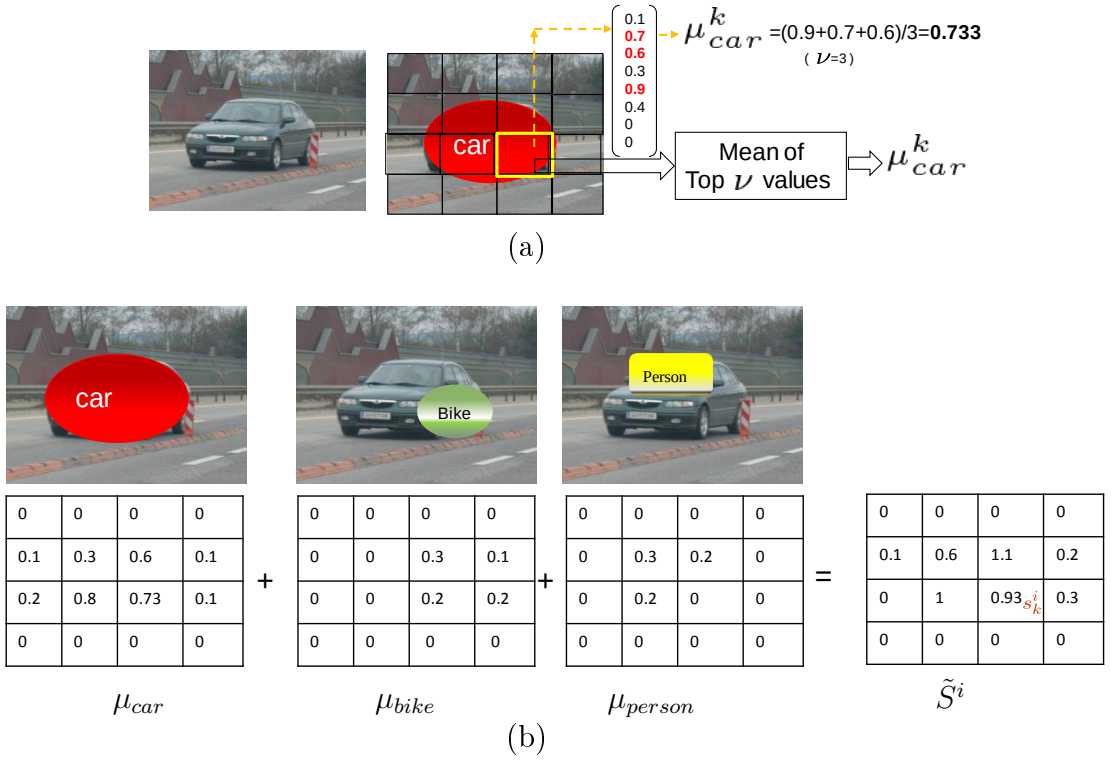


Figure 4.4: Illustration of block saliency computation for blocks of spatial pyramid (best viewed in color).

4.4.2 Saliency-weighted classifier

Once the object dictionary refinement and saliency model learning is complete, we use saliency maps to improve classifier accuracy. Fig. 4.5 illustrates the saliency-weighted classifier pipeline. The max-pooled vector at block k, x^i_k is weighted with the corresponding block-saliency value $\tilde{s}^i_k$ to form the saliency-weighted max-pooled vector $\tilde{x_k}^i$ for that block. The saliency-weighted max-pooled vectors from all 21 blocks of the spatial pyramid are vertically concatenated to form the saliency-weighted image vector $\tilde{X}^i$. An image classifier is learned to indicate the presence or absence of target object ($Y^i \in \{1, -1\}$) using $\tilde{X}^i$ from all the training images, i.e (Train1+Train2).

4.4.2.1 Block saliency computation

The saliency model of a particular class highlights those patches that are likely to contain object parts belonging to that class. High saliency values indicate either object regions in

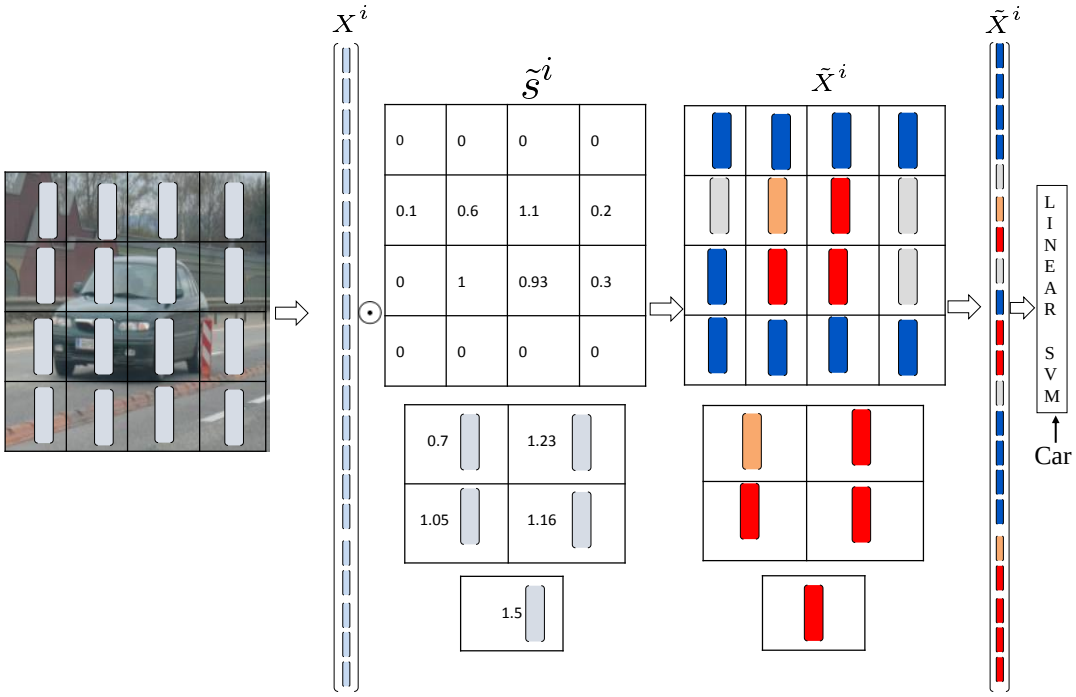


Figure 4.5: Illustration of saliency-weighted max-pooling (best viewed in color).

a positive image or possible false positive patches in a negative training image. Weighting corresponding max-pooled sparse codes with this high value and training the SVM with its corresponding image label will help the image classifier to reduce false detections and improve its performance against background clutter. To this end, for each image, our objective is to determine a saliency weight for the max-pooled vector in each of the 21 spatial blocks of the pyramid, using the n_c saliency maps computed for the n_c categories.

The first step involves finding the saliency value for each category in each block of the spatial pyramid. Choosing the maximum saliency for each block would lead to a poor representation caused by outliers in the saliency map. Using the mean saliency of an object within a block may result in loss of saliency information at higher levels of the pyramid. This is attributed to the presence of many background pixels with low saliency values that are present in the larger image areas covered by these levels. Hence, we use the mean, μ, of the top ν, ($\nu = 2$) saliency values of an object category in a block as shown in Fig. 4.4(a).

The block saliency $\tilde{s_k}^i$, which is the weight of the max-pooled sparse code vector for

the k^{th} block is obtained as $\tilde{s_k}^i = \sum_{n=1}^{n_c} \mu_n^k$ (Fig. 4.4(b)). Since we use mean of top saliency values in a block corresponding to all object models, background patches are suppressed compared to the object patches.

4.4.2.2 Saliency-weighted max-pooling

Our weighting criteria is simple and computationally efficient enabling its use in larger datasets having multiple objects in an image. Every element of the max-pooled vector from a block k is multiplied with its corresponding block saliency $\tilde{s}_k^i$ to form the saliency-weighted max-pooled vector $\tilde{x}_k^i$ for that block. i.e

$$\tilde{x}_k^i = \tilde{s}_k^i \cdot {x^i}_k, \quad \forall k \in \{1, 2, \dots 21\} \tag{4.8}$$

The saliency-weighted max-pooled vectors from all 21 blocks of the image i are concatenated to form the saliency weighted max-pool vector of the image, $\tilde{X}^i$. The l_2-normalized $\tilde{X}^i$ from entire training set (Train1+Train2) are used to learn a one-vs-rest (binary) linear SVM classifier $(\tilde{W}, \tilde{bias})$ for image classification as in eq. (4.3) [106].

$$\arg\min_{\tilde{W}} \|\tilde{W}\|^2 + C \sum_{i \in (\text{Train1+Train2})} max\ (0,\ 1 - Y^i(\tilde{W}^\top \tilde{X}^i + \tilde{bias})\ ), \tag{4.9}$$

4.5 Saliency inference and refinement

The final saliency map generated by the proposed method is a weighted version of the one proposed in [1] where the weights are based on the output of the saliency-weighted classifier explained in section 4.4.2. In [1], a four connected graph having Markov property is formed on image patches based on their spatial adjacency. The probability of label $l_n^{i,j} \in \{-1, 1\}$ indicating the absence or presence of object n respectively at a patch j is computed from its neighbours using marginal probability [1]

$$P(l_n^{i,j} \mid z_n^{i,j},\ v_n) = \sum_{l_n^{i,\mathcal{N}(j)}} p(l_n^{i,j},\ l_n^{i,\mathcal{N}(j)} \mid z_n^{i,j},\ v_n)\,, \tag{4.10}$$

where $z_n^{i,j}$ is the sparse code of a patch j, v_n is the CRF weight vector and $\mathcal{N}(j)$ is the neighbourhood of node j with label $l_n^{i,\mathcal{N}(j)}$.

Saliency of a patch j is given by

$$s_n{}^{i,j} = P(l_n{}^{i,j} = 1 | z_n^{i,j}, v_n), \tag{4.11}$$

which is inferred using loopy belief propagation [1]. The neighborhood and visual priors of SIFT features are used in this saliency computation. The refined saliency $RS_n^{i,j}$ for patch j is

$$RS_n^{i,j} = s_n{}^{i,j} \cdot \mathcal{W}, \tag{4.12}$$

where $\mathcal{W}$ is derived from SVM confidence $(\tilde{W}^\top \tilde{X}^i + \tilde{bias})$ of the saliency-weighted classifier. Since the weight is the same for all patches, it is equivalent to multiplying the saliency map of the image with the weight. The spatial prior is used by the saliency-weighted classifier for the computation of $\mathcal{W}$ through its spatial pyramid max-pooling operation. $\mathcal{W}$ is assigned a high value if the location of the object of interest in a test image matches its spatial prior (object location in the training data). Objects appearing at spatial locations which are significantly different from their training data are assigned lower $\mathcal{W}$ values.

The training images of a dataset indicate the maximum number of object categories in an image. If there is only 1 object category per image and the class with highest SVM confidence matches the model being tested, $\mathcal{W} = 1$ and the generated map becomes the final saliency map. If there are multiple object categories per image, we evaluate the classifier confidence for each of the object categories within a test image and sort them in descending order. If the confidence of an desired object category is ranked below the maximum number of objects per image in the training set, it is highly unlikely that object is present in the image and hence the refined saliency map will have no salient patches in that test image. If the confidence of the object category is ranked within the highest number number of objects per image, a rescaled classifier confidence serves as $\mathcal{W}$. In Graz-02 dataset [100], which has only one object class per image, the saliency map corresponding to the object class predicted by the saliency-weighted classifier is weighted by 1; the saliency maps of the other categories are weighted by 0. While this strategy of weighting the saliency map might appear to be totally dependent on the classifier performance, it must be noted that the classifier uses saliency-weighted pooling so that an accurate saliency map will contribute to lowering the error in classification.

In PASCAL VOC-07 which has 20 classes, there is no training image with more than 5 object classes. If the classifier confidence for a particular class is ranked more than 5, $\mathcal{W}$=0. Since most of the training images have 1 or 2 object classes, $\mathcal{W}$=1 if the classifier confidence is ranked first or second. For the remaining positions, $\mathcal{W}$ is taken as the linearly scaled classifier confidence.

4.6 Experimental results

We evaluated the performance of our joint framework on Graz-02 and PASCAL VOC 2007 datasets. SIFT features are extracted from 64×64 patches with a grid spacing of 16 pixels as in [1]. We choose λ in the sparse code formulation $z_n = \arg\min_{z_n} \|f - D_n z_n\|_2 + \lambda \|z_n\|_1$ to be 0.15 for both classifier and saliency modules. We evaluate top-down saliency using precision rates at equal error rate (EER).

4.6.1 Training and testing image selection

Graz-02 dataset

In this chapter, we conduct the experiments on different sets of training and test image combinations of Graz-02 dataset. For comparison of the proposed top-down saliency with recent top-down saliency approaches, we only consider object annotated images for training and testing. Following [1, 6], odd numbered images are used for training the proposed saliency model and even numbered images for testing.
Secondly, to compare classifier performance with related image classifiers, the same training and test images selection procedure in Bilen *et al.* [75] is used. All (1096) object images are used for classification. For each category, 150 training images are selected at random, and remaining are used for testing. The average results of 10 such experiments are reported.

PASCAL VOC-07 dataset

Similar to previous chapter , we evaluated the performance of our saliency model on 210 segmentation annotated test images. For comparison with related classifier approaches, the training, validation, and test image combinations in the *classification challenge* of the dataset is used.

4.6.2 Top-down saliency

4.6.2.1 Graz-02 dataset

For each object category, an object dictionary with 512 atoms and corresponding CRF parameters are learned for 10 iterations. From $3 \times 512 = 1536$ object atoms and 512 background atoms, a global dictionary of 2048 atoms is formed by concatenation. The ground truth label of a patch is 1 if at least 25% of its pixel belongs to object of interest, and -1 otherwise. In [1], training for each category is done using 150 positive images and 150 background images (column T1 in Table 4.1). To improve performance against the false positive detection, training of proposed framework uses negative images from other categories as well (T2 in table. 4.1). Since T2 has 150 positive images and 450 negative images, the training set to re-train [1] is balanced by randomly selecting 150 negative images from 450 negative images available in T2.

Since the proposed framework requires training and validation cycles, T2 is divided into T2a (training set) and T2b (validation set). T2a contains 70 positive images for each category and 80 negative images, 30 of which are from background and 25 each from the other 2 categories. Since 70 positive images from each category are used to form T2a, the remaining 80 images per category form 320 images in T2b. Misclassified images from T2b are used to update the top-down saliency model.

Table 4.2(a) compares the patch-level precision rates at EER of the proposed saliency detection with [1] on the Graz-02 dataset. In [1], the authors tested each saliency model on object annotated images from its respective category and background. An ideal top-down saliency estimator should be able to distinguish the object from background clutter as well as from other objects. So, we tested [1] on images from all categories in the dataset, i.e, we use their model trained on T1 and evaluated on all 600 images. The average EER of 54.9% is less than 73.7% reported in [1]. Since our training mechanism involves negative images from other categories in addition to the background category, we used T2 to train [1]. The increase of about 5% in EER illustrates the utility of negative images for training.

Effect of saliency refinement. Our saliency modeling process has 2 stages - in the first 10 iterations, the models are learned using T2a and in the second stage the

Table 4.1: Graz-02: Saliency training sets used in our experiments.

Image set	T1	T2	T2a	T2b
Number of positive training images	150	150	70	80
Number of background images	150	150	30	80
Number of negative images except background	0	2x150	2x25	2x80
Total number of training images per category	300	600	150	320

Table 4.2: Precision rates at EER (%) of proposed method against other top-down saliency approaches on all (600) test images of Graz-02 dataset.

(a) **Patch-level**

Algorithm	Yang and Yang [1]	Yang and Yang [1]	Proposed method	Proposed method	Proposed method on a classifier of 100% accuracy
Training Set	T1	T2	T2a	T2	T2
Number of trg. iter.	20	20	10	10+2	10+2
Bike	62.5	69.4	**75.6**	**75.6**	79
Car	53.6	53.2	53.8	**58.3**	65.8
Person	48.6	57.2	62.8	**64.5**	71
Mean	54.9	59.93	64.06	**66.13**	71.9

(b) **Pixel-level**

Algorithm	Yang and Yang [1]	Kocak *et al.* [6]	Proposed method
Training Set	T2	T2	T2
Number of trg. iter.	20	20	10+2
Bike	59.43	59.92	**64.4**
Car	47.36	45.18	**50.9**
Person	49.82	51.52	**56.4**
Mean	52.2	52.21	**57.23**

classifier feedback improves the models through misclassified images. In this experiment, we consider only the first stage and demonstrate the utility of refining the saliency map in table 4.2(a) (column 3). The saliency models trained using T2a for 10 iterations resulted in a mean precision rate at EER of 54.1% without saliency refinement and without classifier feedback. Using saliency model and dictionaries obtained at this 10^{th} iteration of saliency modeling, we train a saliency-weighted image classifier (section 4.4.2) that uses multi-class SVM [108] on the entire training images. On test images, the saliency maps are weighted using classifier confidence output for that image as described in section 4.5. There is a gain of 10% in precision rates at EER due to the saliency

Table 4.3: Precision rates at EER (%) of proposed method on 150 test images of Graz-02 dataset.

Algorithm	Fulkerson *et al.* [89]	Marszalek and Schmid [62]	Proposed method
Bike	66.4	61.8	**67.3**
Car	54.7	53.8	**59.8**
Person	47.1	44.1	**57.1**
Mean	56.07	53.23	**61.4**

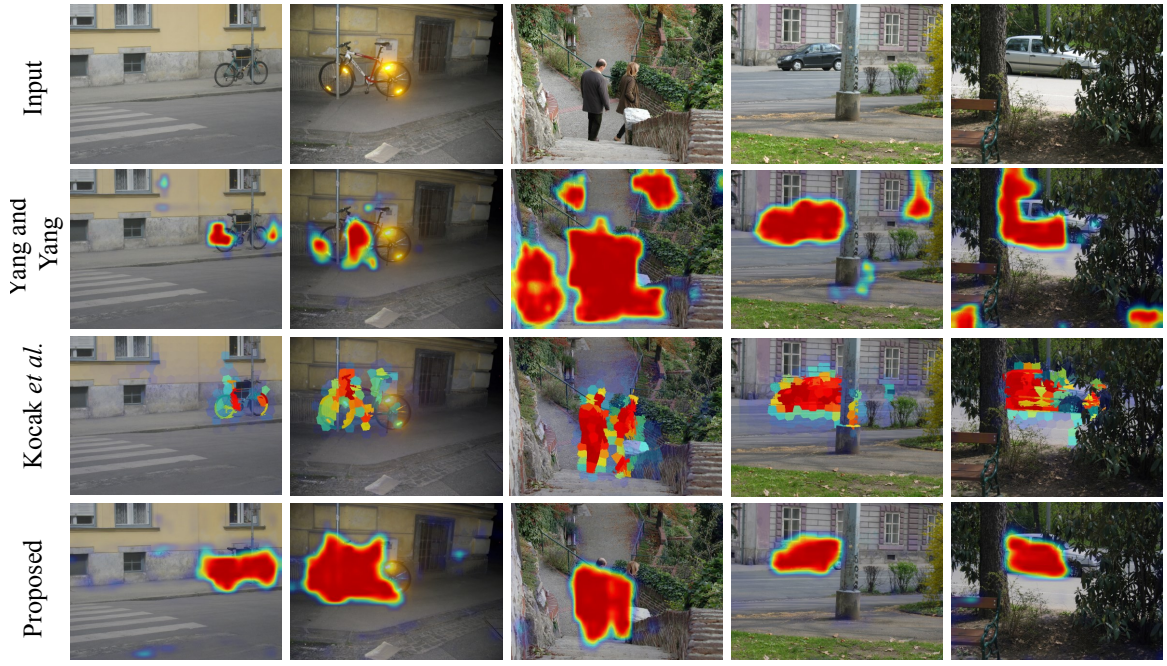
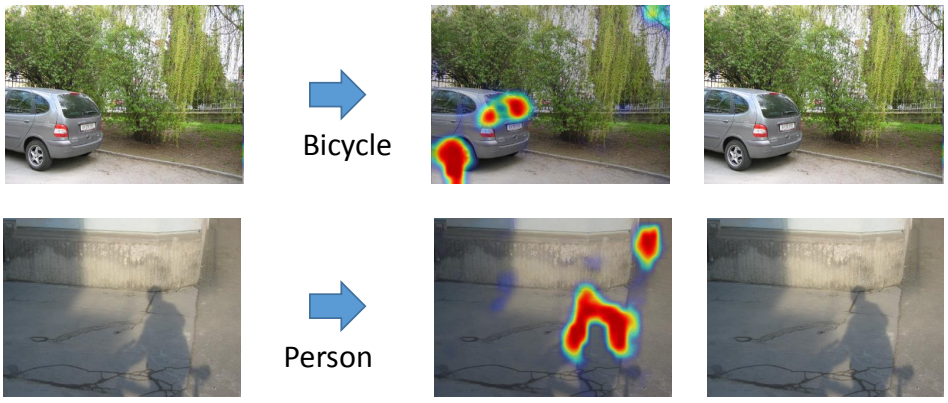


Figure 4.6: Qualitiative comparison of saliency maps of Yang and Yang [1] and Kocak *et al.*[6] with the proposed method.

refinement and the resulting model outperforms [1] with a 5% gain in precision rates at EER. Moreover, we use only 150 training images and 10 iterations when compared to 300 training images and 20 iterations in [1] for saliency modeling. The selection of relevant atoms from the dictionary that can represent the feature for its sparse representation and a saliency-weighted classifier trained on the category-aware sparse codes jointly contribute to improve the performance.

The bicycle model causes few false detections on the *car* image of Fig. 4.7, due to



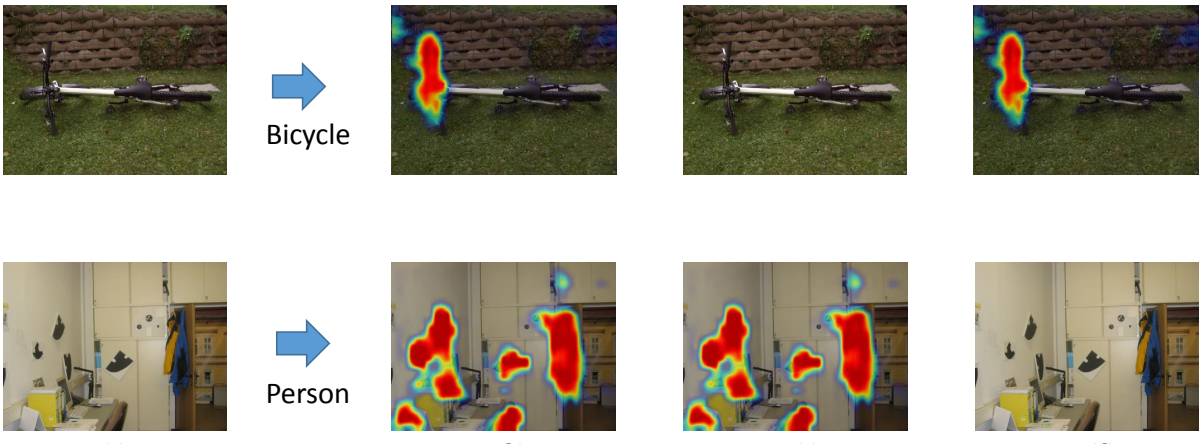
(a) (b) (c)

Figure 4.7: Removal of false detections on negative images by saliency refinement.(a) Input image, (b) false detections of bicycle model (row 1) and person model(row 2) before saliency refinement, (c) saliency-refined image with no false detections

the similarity in the structure. For example, the round shaped tyre patches in *car* is detected due to the similarity with the bicycle tyre. Since the image classifier uses high-level information in the image, it could better predict that bicycle is absent in the given image and it removed the false detections as shown in Fig. 4.7 (c). Similarly, the shadow of human in the *background* image introduced false detections while inferring a person model. Again, classifier-based saliency refinement removed those false detections.

Failure cases of the saliency refinement. Saliency refinement largely depends on the accuracy of the image classifier. It may introduce two types of errors in the saliency map, (i) false negative, when the image classifier wrongly predicts the absence ($\mathcal{W}$= 0) of the object in a positive image Fig.4.8 (top row) (ii) false positive, when the image classifier wrongly predicts the object presence in a negative image Fig.4.8 (bottom row). Due to viewpoint changes, the image classifier fails to predict bicycle presence in the first case and in the second image, shirt hanging on the door matches with the position and features of person class. However, there are very few such errors in our model.

Performance on 100% accurate image classifier. A 100% accurate image classifier will assign $\mathcal{W} = 1$ for positive test images and $\mathcal{W} = 0$ for negative test images. To evaluate the performance under 100% accuracy, we manually assigned these values to $\mathcal{W}$, based on the ground-truth of the test image. Such an image classifier with 100% accuracy can



(a) (b) (c) (d)

Figure 4.8: Failure cases of saliency refinement. (a) Input image, (b) true detections of bicycle model (top row) and false detections of person model (bottom row) before saliency refinement. The misclassifications of the classifier leads to errors in the final saliency map (c) with false negative in the top row and false positive in the bottom row for the bike and person models respectively. (d) An image classifier with 100% accuracy could avoid both of these errors

remove both these errors and produce a better saliency map as shown in Fig.4.8 (c). Such a classifier improves our saliency accuracy on 600 test images to 71.9% (last column in table 4.2(a)), which is closer to the 73.7% reported in [1] for 300 test images. The slight degradation in accuracy is due to the use of less number of training cycles in our model, to save training time. To achieve a faster training of the model, we used only 10 initial iterations of the saliency model training compared to 20 in [1]. Moreover, in these 10 iterations, we used only 150 training images in Train1 compared to 300 training images of [1]. With these two changes, our initial training time reduced to 30% of the time in [1] (10 iterations using 150 images vs 20 iterations using 300 images).

Effect of classifier feedback. In this experiment, we study the second stage of saliency modeling. The dictionaries obtained at the end of the first stage of saliency modeling are used to train the multi-class classifier using T2a training set and validated using T2b. Misclassified images from T2b are used to train the corresponding saliency model. Another iteration of this feedback is carried out by interchanging T2a and T2b. There is an improvement of 2% in precision (column 4 of table 4.2(a)) due to classifier

feedback, which is attributed to the car and person classes. The bike class is, by far, the easiest to model among the three as seen in [1] also, and hence the initial saliency models are able to capture the variability in most of the images. However, using the failed images to provide feedback from the classifier was not sufficient to improve the bike models. Compared to [1], there is an improvement of about 7% with only 10 iterations for saliency modeling and 2 iterations for classifier feedback. The results are supported qualitatively too, as shown by the saliency maps in Fig. 4.6.

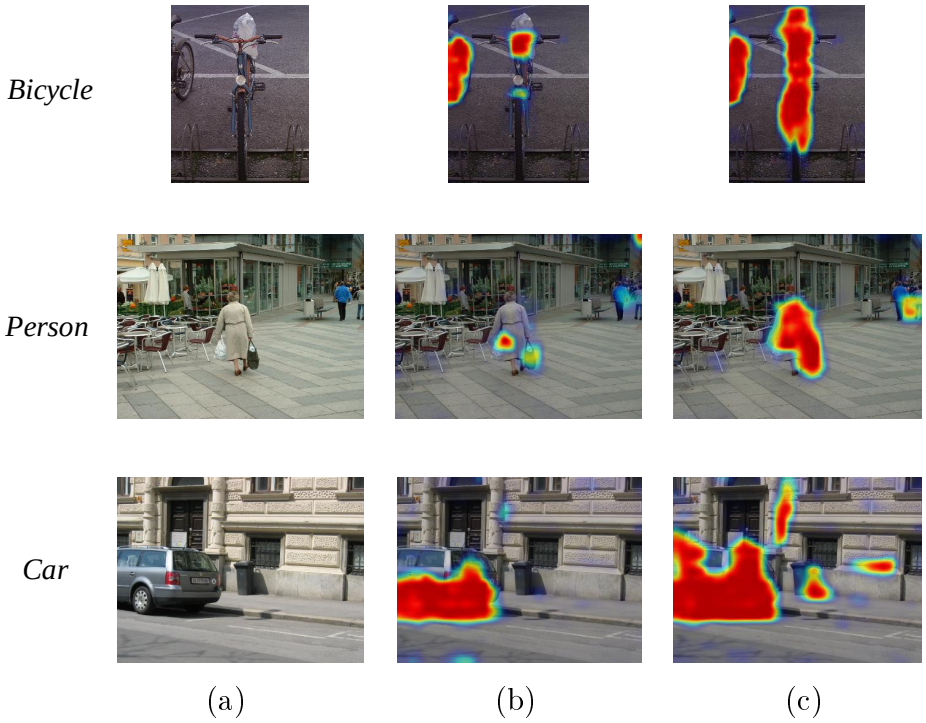


Figure 4.9: Effectiveness of classifier feedback. (a) Input image, (b) saliency map of a model before classifier feedback and (c) saliency map of a model trained with classifier feedback. The *bicycle* and *person* images (c) show improvement in the saliency map due to feedback, while the *car* image (c) shows a failure case, where the classifier feedback introduced few false positives.

In Fig. 4.9, the saliency maps of *bicycle* and *person* images with distinct pose/features are improved by the classifier feedback, while on *car* image, even though the refinement improved the true positive detection of the saliency map, it also introduced some false detections along straight edges of the background clutter. Inspite of few exceptional cases, the classifier feedback helps to improve the saliency map especially on the images

which has a rare pose or view point as shown in 4.9 (c) (*bicycle* , *person*). This is justified quantitatively too in Table 4.2.

Pixel-level result comparison. For comparison with a recent top-down saliency approach [6], we used their publicly available code and re-trained their models using T2 and evaluated on entire 600 test images. Pixel-level precision rates at EER (table 4.2(b)) show that the proposed patch-based approach achieves state-of-the-art result at pixel-level as well. Note that pixel-level saliency maps are generated from patch-level saliency maps as described in [1]. Despite using computationally intensive characteristics like 'objectness', and extracting color-based features from every superpixel, the performance of [6] drops when evaluated on the entire 600 test images, instead of evaluating on 150 same category and 150 background test images. This can be attributed to the inability to discriminate between object categories.

Comparison on 150 test images. The proposed saliency models are evaluated on the test image set-up of shape mask [62], i.e, each model is evaluated only on 150 test images from their respective category. Table 4.3 shows the effectiveness of proposed saliency model by outperforming [89] and [62] in all the three categories. It is to be noted that [62] needs an additional level of supervision by manually labeling training images as *truncated* or *difficult*. Object class segmentation [90] extends [89] using superpixels as the basic unit for computation and the segmentation results are refined using a CRF operating on the superpixel graph. By maintaining identical parameters as our approach (without aggregating the histograms of a superpixel with its neighboring superpixels), [90] achieves a mean precision at EER of 54.56% which is lower than the proposed method operating on regular rectangular patches (61.4%).

4.6.2.2 PASCAL VOC 2007 dataset

We use all the available positive training (P1) and validation (P2) images to train the initial saliency models for 10 iterations. Since multiple classes are present in some images, we train one-vs-rest binary classifiers for each object using P1 images and validate on P2. Set of positive images with lower confidence are used to train respective saliency model incrementally.

Fig. 4.10 shows results for patch-level saliency estimation in which we achieve state-of-the-art performance in 15 out of 20 classes. Averaging over the entire 20 classes, we

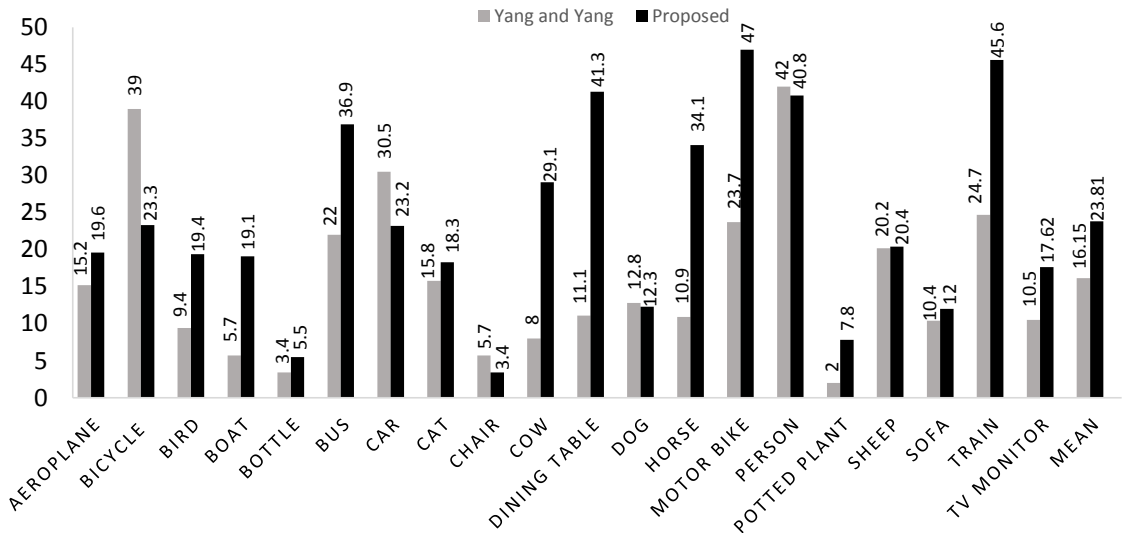


Figure 4.10: Patch-level precision rates (%) at EER on PASCAL VOC-07 compared to Yang and Yang [1].

achieve a mean precision rate at EER (%) of 23.81 compared to 16.15 of [1], which is an improvement of 47%. We maintained the same dictionary size (512) and number of CRF weights as in [1]. The proposed saliency model trained for less number of cycles performs much better than [1] trained for 20 iterations. This is attributed to [1] failing in images having large dominant objects like bus or horse, as local patches of these contain limited relevant information. The additional classifier-guided dictionary training in the proposed model is able to capture this information leading to a significantly higher precison rate at EER for such classes.

Objects that are not visible clearly as in the *Train* image and those in the presence of clutter as in *TV Monitor* are correctly detected in Fig. 4.11. Even when a textureless object such as Sofa, is large and occupies almost the entire image, our method shows good accuracy in marking the area as salient.

Images with multiple objects

Fig. 4.12 shows the the ability of the proposed approach to discriminate among object categories on test images in which objects from multiple categories are simultaneously present. These qualitative results are supported by the improved performance in

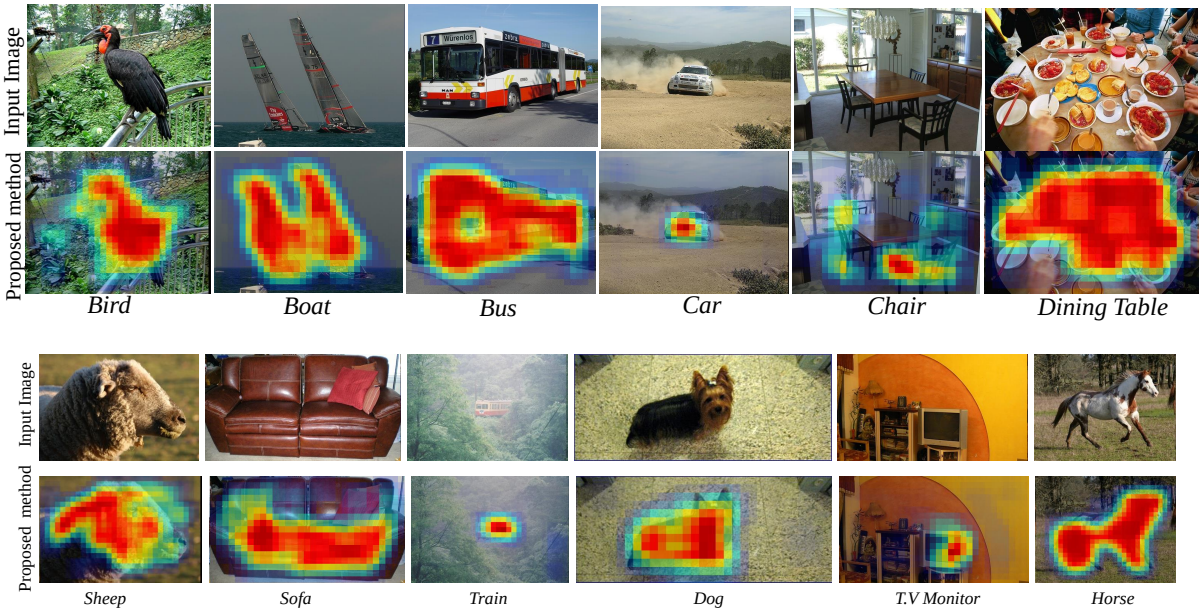


Figure 4.11: Saliency detection by the proposed method on PASCAL VOC-07.

Fig. 4.10. Even when a boat is present in the *Person* image, the method marks only the person as salient. Similarly, bottles and bicycle are correctly detected even though person is dominating in the *bottle*, *Bicycle* images respectively. The presence of dominant TV monitor, makes cat detection a challenging task.Multiple instances of an object in *Cow* and *Motorbike* have been successfully marked as salient in Fig. 4.13.

Qualitative comparison

Fig. 4.14 compares the saliency results of Yang and Yang [1] and Kocak *et al.* [6] with our method. With the help of classifier-guided training, proposed method outperform [1, 6], especially on test images in which object size is too big compared to the patch size as can be observed from Fig. 4.14 (*Train*, *Cow*). In accordance with the precision rates at EER results, Dog and Person saliency maps of [1] are slightly better than proposed method (see *Dog* and *Person* in Fig. 4.14). Improved Performance of proposed method on less textured object classes like aeroplane, bottle and sofa is clearly visible in Fig. 4.14.

4.6.3 Image segmentation

Although object segmentation in cluttered background is a challenging problem in itself, we investigate the effectiveness of using the saliency maps obtained by our method in

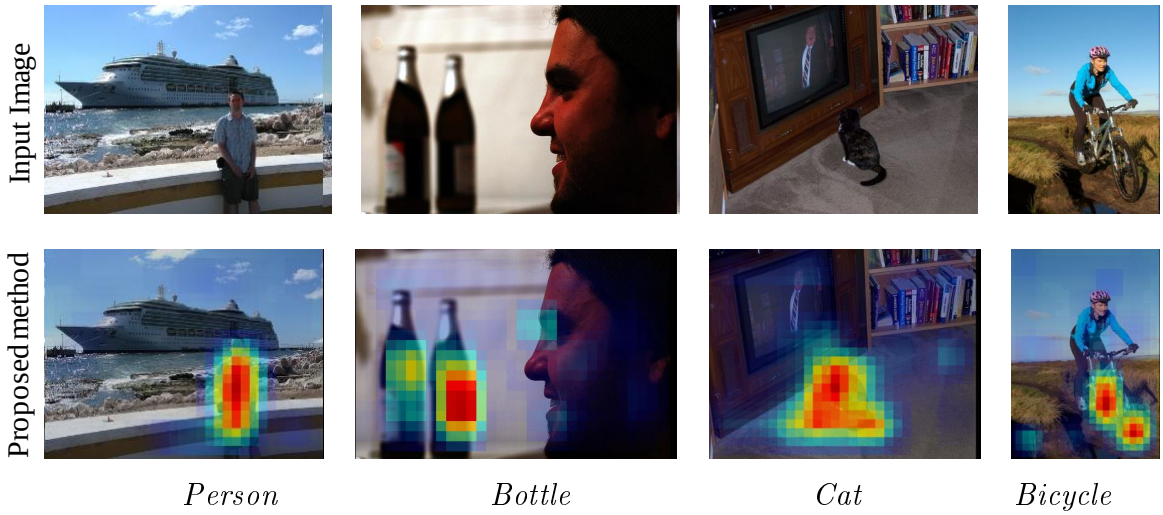


Figure 4.12: Images with multiple objects from different categories.

Figure 4.13: Images with multiple objects from same category.

segmentation. The saliency map generated is a probability map with 1 indicating the presence of object of interest and 0 indicating absence of object. Here non-object includes background as well as object pixels of negative classes. The saliency models of all categories are inferred on a test image and each pixel is assigned to the category of the saliency model that produces highest saliency value at that pixel. If the highest saliency at a pixel is below 0.5, those pixels are assigned to the background class. Fig. 4.15 shows segmentation results achieved by this simple thresholding . Even though our saliency models are trained and inferred at patch-level, it is capable of producing a segmentation that follows object boundaries. An improved pixel accurate segmentaion can be achieved by applying dedicated segmentation approaches such as Grabcut [109] on salient regions.

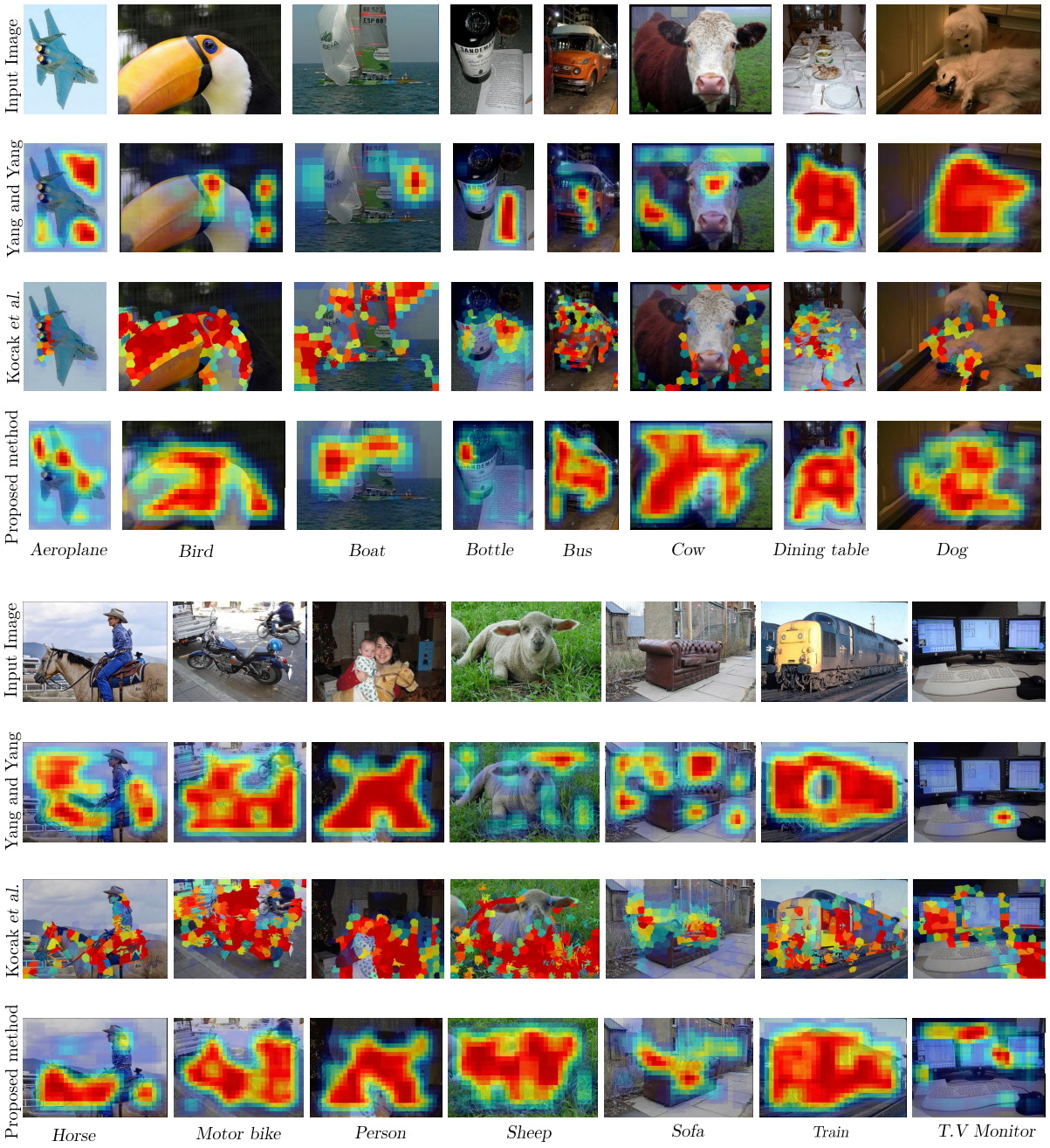


Figure 4.14: Qualitative comparison with Yang and Yang [1] and Kocak *et al.*[6] on PASCAL VOC-07 dataset

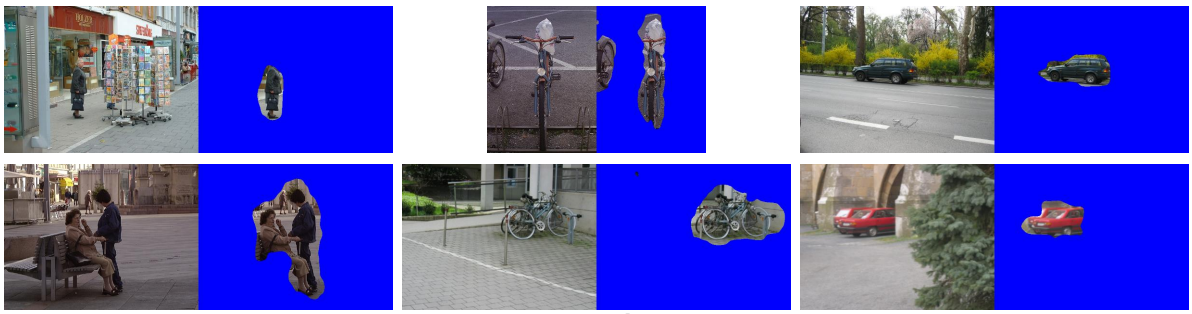


Figure 4.15: Segmentation from the saliency map by simple thresholding.

Table 4.4: Comparison with state-of-the-art semantic segmentation tasks on 450 test images of Graz-02 dataset using intersection over union metric.

Algorithm	Singaraju and Vidal [92]	Jain *et al.* [94]	Proposed
Background	82.32	77.97	**83.46**
Bike	46.18	**55.6**	50.02
Car	36.49	**41.51**	40.81
Person	**38.99**	37.26	38.8
Mean	50.99	53.08	**53.27**

Graz-02 dataset

Intersection over union (IOU) is used as a metric to evaluate segmentation performance, computed as $IOU = TP/(TP + FP + FN)$, where TP is number of true positive pixels, FP is the number of false positive pixels and FN is number of false negative pixels. Table 4.4 compares the performance of the proposed method with [92] and [94] on the Graz02 dataset. Following [94], the proposed saliency models are evaluated on 150 test images from each object category (3×150).

Even with a simple thresholding of the saliency map, the segmentations produced with our method outperforms both the methods. Better performance in background pixel classification illustrates the effect of classifier-weighted saliency inference, which reduced false positive detection.

PASCAL VOC-07 dataset

Image segmentation approach [91] evaluates their model using pixel classification accuracy; i.e percentage of pixels correctly classified for each category. To compare with this

Table 4.5: Classification accuracy for Graz-02.

Algorithm	LLC [26]	Bilen *et al.* [75]	ScSPM [2]	ScSPM using proposed category-aware sparse coding	Proposed saliency-weighted classifier
Accuracy	87.8 ± 0.9	91.18 ± 1.4	88.5 ± 1	89.5 ± 1.2	$\mathbf{91.21 \pm 1.2}$

approach, the proposed model is also evaluated using this metric. The average classification accuracy (average accuracy of 21 categories- 20 object and one background category) obtained by thresholding saliency maps at 0.5 is better (29.66%) than image segmentation approach [91] (23%). Although our saliency maps are estimated using patch-level computations, the accuracy is comparable to that of the superpixel based segmentation approach of [90] (27% using 4 superpixel neighbors)

4.6.4 Image classification

This chapter proposes a novel and effective method for salient object detection in which the classifier plays an important role in improving results because of its ability to identify those objects whose models needs to be retrained by the CRF in order to remove false positives. Since there is a tight coupling between the saliency and classifier modules, it would be pertinent to investigate whether image classification performance gains with improved object models.

Graz-02 dataset

Table 4.5 compares the image classification accuracy of the classifier in the proposed framework with LLC [26], ScSPM [2] and [75]. In our implementation of ScSPM, we randomly sample 100 features from each training image and form a dictionary of size 2048. As suggested in [26], 5 local dictionary atoms are used to code a feature in LLC. Although LLC is faster than ScSPM, its classification accuracy is less. When ScSPM is used in conjunction with our category-aware sparse coding, there is an improvement of about 1%. However, when the saliency information is incorporated in the form of a weight derived from the classifier, the classification accuracy is 91.21%, which is better than LLC and [75]. It may be noted that in [75], the results are obtained using SIFT features extracted from multiple patch sizes for training the classifier using the AUC criterion. In [110], the authors achieve an accuracy of 92.23% using multiple features

such as color, shape and SIFT. Thus, although the classifier falls short of state-of-the-art performance by about 1%, the top-down saliency framework provides a reasonably good classifier as an accessory.

PASCAL VOC-07 dataset

In PASCAL VOC-07 dataset, objects from multiple categories are present in an image. So we train a binary SVM classifier for each object category as in [26] instead of one-vs-rest SVM used for Graz-02 dataset. For example, in an image that contains both cat and TV monitor, the cat binary classifier will estimate the presence of cat in the image and the TV monitor classifier will estimate the presence of TV monitor in the image. If both classifier respond positively, the image will be marked as one containing both cat and TV monitor. On the other hand if the cat classifier wrongly estimated the absence of cat, then it is considered as a false negative for cat category. Average precision of each category is evaluated seperately on all test images of PASCAL VOC-07 image classification dataset and the mean across 20 categories is evaluated. We achieved better mean average precision of 50.84% as compared to 50.65% by ScSPM and better classification in 12 out of 20 object classes. Although the improvement is not significant, most of the errors are due to inter-class similarity, e.g. between bike and motor bike classes. The joint saliency-classifier framework needs D_{sub} and a background dictionary in addition to the dictionaries for saliency modeling. These dictionaries are much smaller than the large dictionary (12,000 atoms) used by ScSPM.

4.6.5 Computation time

The training of joint framework for saliency estimation and image classification is faster as compared to the saliency estimation approaches of [6, 1], due to the reduced number of iterations and images used by proposed classifier-guided training. MATLAB implementations of all approaches were evaluated on a PC running on Intel Xeon 2.4GHz processor. Our training of all three object categories in Graz-02 and image classifier took just 3 hours and 34 minutes, while training of saliency models alone took 4 hours and 49 minutes by [1] and 30 hours and 10 minutes by [6].

The saliency estimation for 3 categories as well classifying the image took just 9.89 seconds. The SIFT feature extraction took 2.83 seconds, saliency inference per category took 1.72 seconds and category-aware sparse coding, saliency-weighted max-pooling, image classification and saliency refinement took additional 1.9 seconds, totaling to 9.89 seconds ($3 \times 1.72 + 2.83 + 1.9 = 9.9$) seconds. The saliency estimation of 3 object categories alone took 7.99 seconds in [1]. The inference time is much larger in [6], which took 52 seconds. The ScSPM image classfication of [2] alone need 6.9 seconds, due to conventional sparse coding on D, while with the help of category-aware sparse coding, the additional computational time for classification reduces to 1.9 seconds only . For comparison, cascaded saliency and classification modules of [1] and [2] requires 14.89 seconds per image, while our joint approach gives better accuracy in 9.89 seconds.

4.7 Conclusion

In this chapter we propose a joint framework for top-down salient object detection and image classification. Since the pipeline of image classification [2] and top-down saliency [1] contains many common stages, our interconnected and mutually benefiting saliency-classification framework reduces the computational cost when compared to their independent implementations. The image classifier is trained on novel category-aware sparse codes computed on object dictionaries used for saliency modeling. A novel saliency-weighted max-pooling is proposed to improve image classification by weighting the max-pooled vector in each block of the spatial pyramid with a weight derived from net saliency of that block. Similarly, saliency maps are improved by using the image classifier that leverages information about presence of the object in an image. The salient object detection algorithms proposed in this chapter and in the previous chapter are trained in a fully supervised setting. In the next two chapters we explain our weakly supervised salient object detection algorithms.

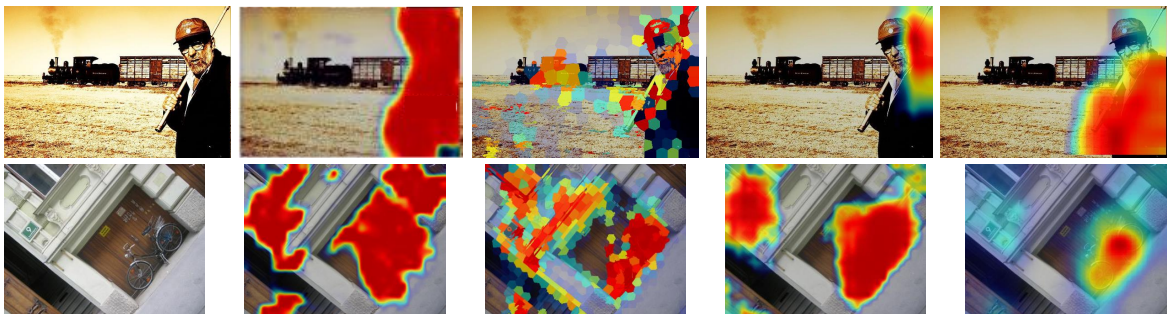
Chapter 5

Backtracking ScSPM Image Classifier for Weakly Supervised Top-down Saliency

5.1 Introduction

Most methods for top-down salient object detection, including the approaches discussed in the previous two chapters, need to be trained in a fully supervised manner, where an exact object annotation is available. Weakly supervised learning (WSL) alleviates the need for such user-intensive annotation by providing only class labels for an image during learning. Previously, iterative training strategies such as multiple-instance learning (MIL) [111] were found to be effective in training object detectors [112] and object segmentation [113] models in a weakly supervised setting. To the best of our knowledge, there are only two approaches [9, 8] that use weakly supervised training for top-down salient object detection. The quality of saliency maps produced by these approaches are poor and hence can not be used for practical applications such as object segmentation or detection. [9] employs iterative refinement of object hypothesis on a training image. In this chapter we propose a weakly supervised top-down saliency approach based on ScSPM image classifier, that does not require these iterative steps.

In the proposed method, first, an ScSPM-based image classifier [2] is trained for an object category. On a validation/test image, the classifier gives a confidence score indicating the presence of the object. The probabilistic contribution of each patch in the image to this confidence score is analyzed to estimate its Reverse-ScSPM (R-ScSPM) saliency.



(a) (b) (c) (d) (e)

Figure 5.1: Our weakly supervised top-down saliency map in comparison with fully supervised methods. (a) Input images, person and bicycle saliency maps of (b) [1], (c) [6], (d) [7] and (e) proposed method are shown in row 1 and row 2 respectively.

The patches having high R-ScSPM saliency are generally from object regions, but they lack contextual information. For high-level understanding of the surrounding spatial region, contextual information of the patch is required. Hence, we incorporate a contextual saliency module that computes the probability of object presence in a patch using logistic regression trained on contextual max-pooled vectors [7]. The training of contextual saliency needs a set of positive patches from the object region and a set of random negative patches from images that do not contain the object. Since a patch-level annotation is not available, we use patches from the positive training images having high R-ScSPM saliency to train the contextual saliency. The contextual saliency inferred on a test image is combined with the R-ScSPM saliency to form the final saliency map. R-ScSPM saliency considers the spatial location of patch through backtracking max-pooled vector whereas contextual saliency considers its spatial neighborhood information, thereby complementing one another. We also propose a classifier confidence-based refinement to the saliency map. Besides illustrating the accuracy of saliency maps produced by the proposed method, we demonstrate its effectiveness in applications like weakly supervised object annotation and class segmentation. Although the proposed approach does not use an iterative training strategy, it produces a saliency map (Fig. 5.1 (e)) that is comparable to fully supervised approaches as shown in Fig. 5.1(b, c, d) .

In section 5.3, we present the weakly supervised R-ScSPM framework (Fig. 5.2) to obtain R-ScSPM saliency. We then introduce contextual saliency (section 5.4) that estimates object presence in a patch by considering its neighborhood information. Training of

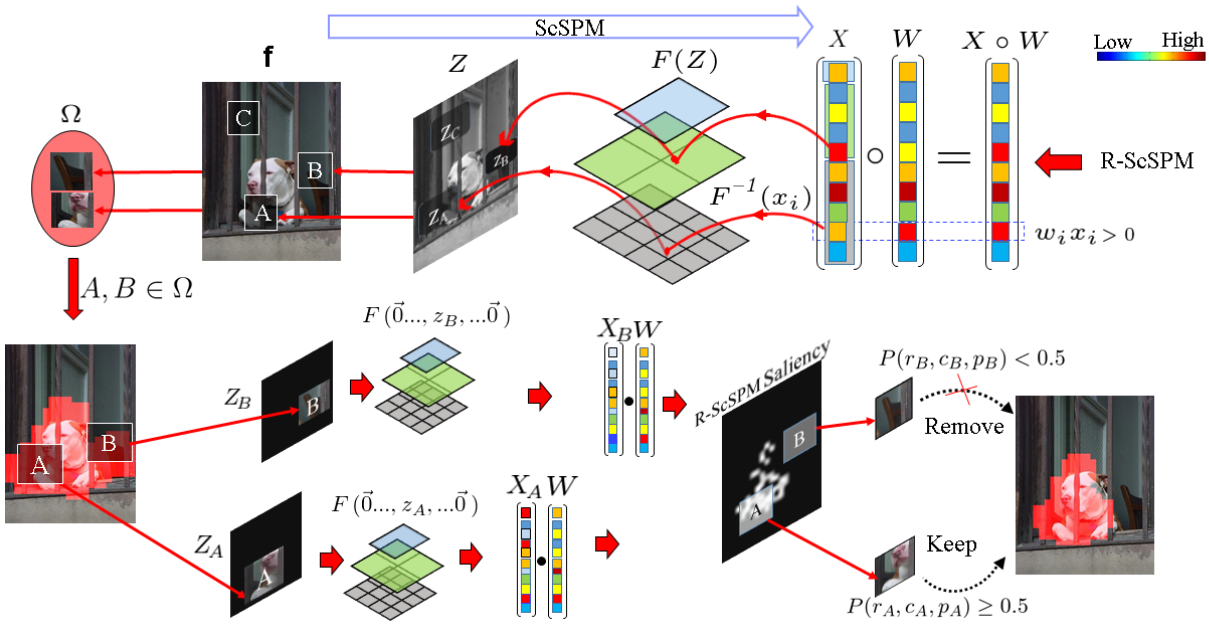


Figure 5.2: Illustration of our R-ScSPM saliency estimation and patch selection for *dog* category. Red arrows indicate the proposed R-ScSPM framework. The elements x_i of X having $(w_i x_i > 0)$ are traced back to the image patches A, B and are added to Ω. The patch $C \notin \Omega$ as it does not contribute positively to classifier confidence. For a patch $A \in \Omega$, R-ScSPM saliency $P(r_A, c_A, p_A)$ is evaluated by setting the sparse codes of all patches except A to $\vec{0}$ forming Z_A followed by a scalar product $(\cdot)$ with the classifier weight W. Similar procedure is followed for all patches in Ω. The patch A is selected as object patch since $P(r_A, c_A, p_A) \geq 0.5$.

contextual saliency requires object patches that are selected using the R-ScSPM saliency map. Finally, during inference (section 5.5), our framework combines both the saliency maps to generate the final saliency map.

5.2 Notations

In our ScSPM-based classifier, dense-SIFT features are extracted from gray-scale image patches. K-means clustering of the SIFT features from training images are used to form a dictionary D of d elements (atoms). The SIFT features $\mathbf{f} = [f_1, f_2...f_M]$ from M patches of an image are sparse coded using D to $Z = [z_1, z_2...z_m, ...z_M]$. Here, z_m is a d-dimensional vector representing the sparse code of a feature f_m from the m^{th} image patch. The spatial

distribution of the features in the image is encoded in the max-pooled image vector X through a multi-scale max-pooling operation $F(z_1, z_2, ..., z_M)$ of the sparse codes on a 3-level spatial pyramid [73] as shown in Fig. 5.2. The i^{th} element x_i of X is a max-pooled value derived using maximum operation on j^{th} elements of all patches in a spatial region $\mathcal{R}$ defined by i and $j = 1 + (i-1)\ mod\ d$. It is represented as

$$x_i = max\{|z_{1j}|, |z_{2j}|, ...|z_{qj}|\}, \quad s.t. \quad 1, 2...q \in \mathcal{R}. \tag{5.1}$$

Let the label $Y_k \in \{1, -1\}$ indicate the presence or absence of an object O in the k^{th} image. If $Y_k = 1$ it is a positive image, else it is a negative image. Image-label pairs (X_k, Y_k) of T training images are used to train a linear binary SVM classifier by minimizing following objective function [106]

$$\arg\min_{W} \|W\|^2 + \mathcal{C} \sum_{k=1}^{T} max(0, 1 - Y_k(W^\top X_k + bias)), \tag{5.2}$$

where $W = [w_1, w_2....w_N]^\top$ and $bias$ are the SVM weight vector and bias respectively. W is learnt separately for each object category. N is the length of the max-pooled image vector X_k and $\mathcal{C}$ is a constant. Given a validation/test image with max-pooled vector X, the classifier score $W^\top X + bias$ indicates the confidence of the presence of object O in it.

5.3 R-ScSPM saliency

In an ScSPM image classifier, both the linear-SVM and multi-scale max-pooling operations can be traced back to the patch-level. This enables us to analyze the contribution of each patch towards the final classifier score which is then utilized to generate the R-ScSPM saliency map for an object. Since a common dictionary D is learned for all objects by unsupervised clustering of random SIFT features from training images, the correspondence of a particular dictionary atom to an object or background is unknown. ScSPM stipulates that a patch is representative of its image if its sparse code makes the largest contribution (max-pooling) to a particular dictionary atom among other patches in the same spatial region $\mathcal{R}$. The representativeness, r_m, of a patch m for an image is indicated by the number of times the elements of that patch's sparse code made it to the max-pooled vector. Representative patches may either contribute positively or negatively

to the classifier score with higher contribution indicating more relevance of the patch to an object O. The relevance of the patch to the object is denoted c_m.

It is possible that among the elements of the sparse code of a patch that contributes positively to the classifier confidence, there are other elements that may contribute negatively. For example, let $[z_{m1}, 0, ... \ z_{mj} \ ...0, z_{md}]^\top$ be the sparse code of a patch m with its j^{th} element z_{mj} being a local maximum in its spatial pyramid region. Although z_{mj} contributes positively to the classifier confidence $W^\top X + bias$, the other non-zero elements z_{m1} or z_{md} may contribute negatively, indicating absence of the object in that patch. So, the relevance of a patch to the object requires its contribution to be computed in the absence of other patches; this relevance is denoted p_m. The probability of a patch m belonging to an object, which in turn indicates the saliency G of the object, depends on the three parameters–r_m, c_m and p_m as

$$G = P(r_m, c_m, p_m) = P(p_m|r_m, c_m)P(c_m|r_m)P(r_m). \tag{5.3}$$

The representative elements of the sparse code z_m is identified by

$$\Psi_m = \{i\delta(F^{-1}(x_i), z_{mj})\}, \quad \forall i \in \{1, 2, ..N\}, \tag{5.4}$$

where δ is the Kronecker delta function and F^{-1} is the inverse operation of spatial pyramid max-pooling and the location of x_i in X identifies the region $\mathcal{R}$ in the spatial pyramid and its position j in the sparse code z_m. The probability of representativeness of the m^{th} patch to the image is then defined as

$$P(r_m) = \mathbf{card}(\Psi_m)/N, \tag{5.5}$$

where $\mathbf{card}(.)$ represents cardinality and N is the length of X.

The classifier confidence is a score indicating the presence of the object in the image, which proportionally increases from a definite absence ($score \leq -1$) to definite presence ($score \geq 1$). Normalizing the confidence scores between 0 and 1 using parameters $\beta = 0.5$

and $b = \beta(bias + 1)$, we can represent it as the probability

$$\begin{aligned} & P(Y = 1|F(z_1, z_2, ..., z_M)) \\ & = \beta W^\top F(z_1, z_2, ..., z_M) + b, \\ & = \beta W^\top X + b = \beta \sum_{\forall i \in \{1,..N\}} w_i x_i + b, \\ & = \beta \sum_{\forall i \in \Psi_m} w_i x_i \quad + \quad \beta \sum_{\forall i \in \{1,..N\} \setminus \Psi_m} w_i x_i + b, \\ & = \theta(c_m|r_m) \quad + \quad \beta \sum_{\forall i \in \{1,..N\} \setminus \Psi_m} w_i x_i + b, \end{aligned}$$

where $\theta(c_m|r_m)$ is the contribution of the patch m to the image classifier confidence.

Given that the patch is representative of the image, the probability of it belonging to the object is

$$P(c_m|r_m) = \begin{cases} \theta(c_m|r_m), & \text{if} \quad \theta(c_m|r_m) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.6}$$

Using the above probabilities, we select a set Ω of all patches that contribute positively to the classifier confidence as

$$\Omega = \{P(c_t|r_t)P(r_t) > 0\}, \qquad \forall t = 1, 2, ..., \ M. \tag{5.7}$$

The net contribution of a patch $m \in \Omega$ in the absence of other patches is

$$P(p_m|r_m, c_m) = \beta W^\top F(\vec{0}.., z_m, ..., \vec{0}) + b, \tag{5.8}$$

where $F(\vec{0}.., z_m, ..., \vec{0})$ is the max-pooling operation performed by replacing the sparse codes of all other patches with a zero vector $\vec{0}$ of size d to form a max-pooled vector X_m.

Implementation details

Fig. 5.2 illustrates three patches A, B and C on an image and their corresponding sparse codes. The classifier score $W^\top X + bias$ indicates the confidence of the presence of object as mentioned earlier. Each element x_i of X has a corresponding weight w_i. The elements from the Hadamard product $W \circ X$ with $w_i x_i > 0$ mark the patches A and B that contribute positively to the classifier confidence through a $F^{-1}(.)$ operation, i.e the set Ω. The contribution of patch A in the absence of other patches is evaluated using max-pooling operations $F(\vec{0}.., z_A, .., \vec{0})$ on sparse code vectors Z_A in which sparse codes of

all other patches except z_A are replaced with $\vec{0}$ forming max-pooled vector X_A. The R-ScSPM saliency of a patch A is given by

$$P(r_A, c_A, p_A) = \begin{cases} \beta W^\top F(\vec{0}.., z_A, .., \vec{0}) + b & \text{if } A \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

5.4 Contextual saliency training

The purpose of contextual saliency is to include neighborhood information of a patch. In Chapter 3, a fully supervised setting is used to select object patches to train our contextual saliency module. In this chapter, we remove this requirement by using object patches that are extracted by R-ScSPM saliency. From positive training images, patches with R-ScSPM saliency $G > 0.5$ are selected as positive patches with label $l = +1$, while random patches are selected from negative images with patch label $l = -1$. The selected patches are indicative of belongingness to an object category. In fig 5.2, patch A having $P(r_A, c_A, p_A) \geq 0.5$ is selected for contextual model training while patch B having $P(r_B, c_B, p_B) < 0.5$ is removed.

For the selected patches, a 13×13 neighborhood of surrounding patches are divided into a 3×3 spatial grid followed by max pooling of sparse codes over each grid and concatenated to form a context max pooled vector ρ. A logistic regression model with weight v and bias b_v is learned using positive and negative patches from the training images to form the contextual saliency model [7]. Since the sparse codes for every patch are already computed for R-ScSPM, max pooling over the context of a patch followed by logistic regression learning is the only additional computation required for this contextual saliency model.

5.5 Saliency inference

On a test image, the contextual saliency $\mathcal{L}$ is inferred using the logistic regression by

$$P(l = 1 \mid \rho, v) = \frac{1}{1 + exp(-(v^T \rho + b_v)} \ , \tag{5.10}$$

where $P(l = 1 \mid \rho, v)$ indicates the probability of presence of an object in a patch and ρ is the contextual max-pooled vector for a test patch. For each patch, the contextual and

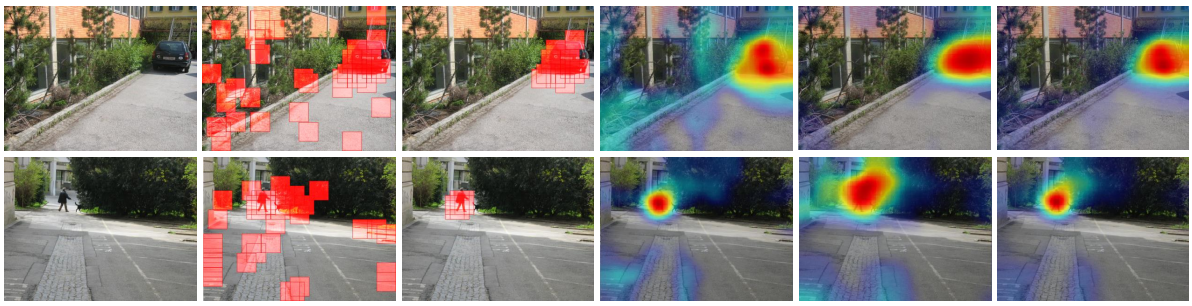
R-ScSPM saliency values are combined as $G\mathcal{L} + 0.5(G + \mathcal{L})$ and normalized to values in $[0, 1]$ to form the saliency S. We choose this combination criteria instead of a product between the two, since the R-ScPSM saliency values are non-zero only for R-ScSPM selected patches.

Classifier-based refinement. The saliency map is refined using the same image classifier used for R-ScSPM saliency having SVM parameters $(W, bias)$. Given a test image, we compute its classifier confidence $W^\top X + bias$. The test image could either contain a single class or multiple classes. For the former, as in the Graz-02 dataset, the classifier estimates the presence or absence of an object. However, for multiple classes, thresholding of classifier confidence determines the presence or absence of an object. During training, we compute the average classifier confidences $W^\top X_j$ for all positive training images j in each run of K-fold learning (K=15). The mean of K such values is used as the final threshold th_O to indicate the presence of object O on a test image. To avoid situations where false negative values drastically reduce the threshold, we maintain the lowest possible confidence value as -0.5. If $W^\top X < th_O$, it is less probable that the object O is present in that image, and therefore, there will be no salient object marked in the image. However, if $W^\top X > th_O$, the saliency of the patch is retained as S. Pixel-level saliency maps are generated from patch-level saliency S using Gaussian-weighted interpolation as in [7].

5.6 Experimental evaluation

We evaluate the performance of our weakly supervised top-down saliency model on 5 challenging datasets across three applications. We compare with other top-down saliency approaches using Graz-02 [100] and PASCAL VOC-07 [93] segmentation datasets. The top-down saliency map is applied to the tasks of class segmentation, object annotation and action-specific patch discovery on Object Discovery dataset [10], PASCAL VOC-07 detection dataset and PASCAL VOC-2010 action dataset [114] respectively. All these datasets are challenging, especially from a weakly supervised training perspective, due to heavy background clutter, occlusion and viewpoint variation.

We maintain the same parameters across the datasets. Following [1], SIFT features are extracted from 64×64 patches on a grayscale image with grid spacing of 16 pixels. The



(a) (b) (c) (d) (e) (f)

Figure 5.3: Illustration of individual stages of the proposed model. (a) Input image, (b) patches in Ω and (c) patches selected by thresholding (d) R-ScSPM saliency map. (e) contextual saliency map and (f) final saliency map.

Table 5.1: Components analysis: Pixel-level precision rates at EER (%).

	Bike	Car	Person	Mean
Random trained contextual saliency	51.2	27.3	38.3	39
R-ScSPM trained contextual saliency	66.9	55.3	54.5	58.9
R-ScSPM saliency	61.6	46.6	54.8	54.3
Complete (R-ScSPM trained contextual saliency + R-ScSPM saliency)	67.5	56.48	57.56	60.52

dictionary size for sparse coding is set to 1536 disregarding individual object categories whereas in [1, 6] separate dictionaries of size 512, corresponding to each object category are iteratively learned. The size of the context-pooled vector is $9 \times 1536 = 13824$.

5.6.1 Analysis of individual components

Fig. 5.3 shows a visual comparison of the effect of each stage in our proposed method. For a test image, the patches in Ω (refer Fig. 5.2) from the R-ScSPM pipeline are shown in Fig. 5.3(b) and the R-ScSPM saliency map is shown in Fig. 5.3(d). This saliency map is thresholded at 0.5 to obtain the most relevant patches weeding out the false detection in Ω as shown in Fig. 5.3(c). Fig. 5.3(f) shows the final saliency map formed by combining the contextual (Fig. 5.3(e)) and R-ScSPM saliency maps. At non-textured patches of the car (top), the lower R-ScSPM saliency is boosted by high contextual saliency in the final saliency map. The smearing of saliency in the contextual saliency map for small objects (bottom) is removed when combined with the R-ScSPM saliency map.

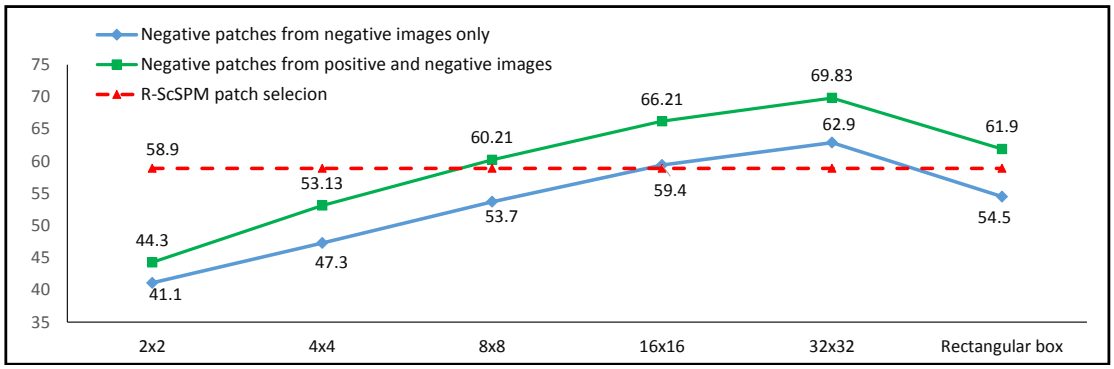


Figure 5.4: Effect of patch selection strategy for training. X-axis specifies the supervision settings, Y-axis denotes the mean of precision at EER (%) across 3 categories.

Table 5.1 analyzes the contribution of each component of the proposed saliency model on Graz-02 dataset. The effectiveness of the proposed method in selecting positive patches is demonstrated by comparing its performance to that using random selection of patches from positive images in training contextual saliency. The mean precision rate at EER (%) of 39% is much lower than 60.52% obtained using the complete framework. This can be attributed to poor model learning in categories like car where the object size could be much smaller relative to the image, whereby random selections are more probable to pick out patches from the background. By training the contextual model with the R-ScSPM selected patches, the result improved to 58.9%, which shows that the R-ScSPM patch selection is effective in identifying object patches in an unannotated positive image. The contribution of R-ScSPM saliency is studied by removing the contextual saliency component from the framework. The results are poorer compared to 'R-ScSPM trained contextual saliency' due to lack of contextual information. Our complete framework gives 60.52% which shows that both the contextual and R-ScSPM saliency maps complement one another. R-ScSPM utilizes the patch location and contextual saliency utilizes its neighborhood information and hence they complement each other.

5.6.2 Comparison with various levels of supervision

Previous WSL localization and top-down saliency works [115, 9] select initial negative patches from either the boundaries or at random locations from the postive training images. They need to iteratively refine their model in order to remove potentially erroneous negative patches. Since the training of the proposed method is not iterative, we need

to select negative patches only from negative images. We analyze the influence of negative patches extracted from positive images on the performance of contextual saliency using different supervision settings in Graz-02. Each positive training image is divided into regular sized grids varying from 2×2 to 32×32 and each grid is manually labeled to indicate if an object is present or not. We also consider the case of a rectangular bounding box around the object. The contextual saliency model is learned using the additional label information. We maintain the same number of positive and negative patches throughout the experiment. Each category's model is evaluated on its respective test images. Pixel-level precision rates at EER (%) is averaged over all categories and shown in Fig. 5.4. The model trained using negative patches from both positive and negative images (green) outperforms the result when only negative images are used (blue), with the performance increasing with increasing scale of supervision. It indicates that if an iterative learning is used in our method, the results can be improved considerably. The proposed weakly supervised method (red) matches the performance of the 16×16 supervised setting (a label for every 40×40 pixels) learned using negative patches from negative images despite having the label at the image-level. We outperform the results of a labeled bounding box using the same learning settings.

5.6.3 Graz-02 dataset

Similar to previous chapters, we split the images into training and testing sets following [1]. We report our results on 3 test set configurations. First, pixel-level results of the proposed saliency model and recent top-down saliency models [1, 6, 7] are evaluated on all 600 test images of the dataset. Second, for comparison with related approaches [62, 90], each object category is evaluated on test images from its respective category and the pixel-level results are reported. Finally, to compare with [8, 61], the patch-level results on 300 test images [1] are evaluated, where 150 test images are from a single category and the remaining 150 are from the background class.

Table 5.2 compares our pixel-level results with top-down saliency approaches [1, 6, 62]. SV indicates supervision level with TF, WS, FS referring to training-free, weakly supervised and fully supervised respectively. [1] and [6] are fully supervised (FS), needing 20 iterations of CRF learning with sparse codes relearned at each iteration. Separate

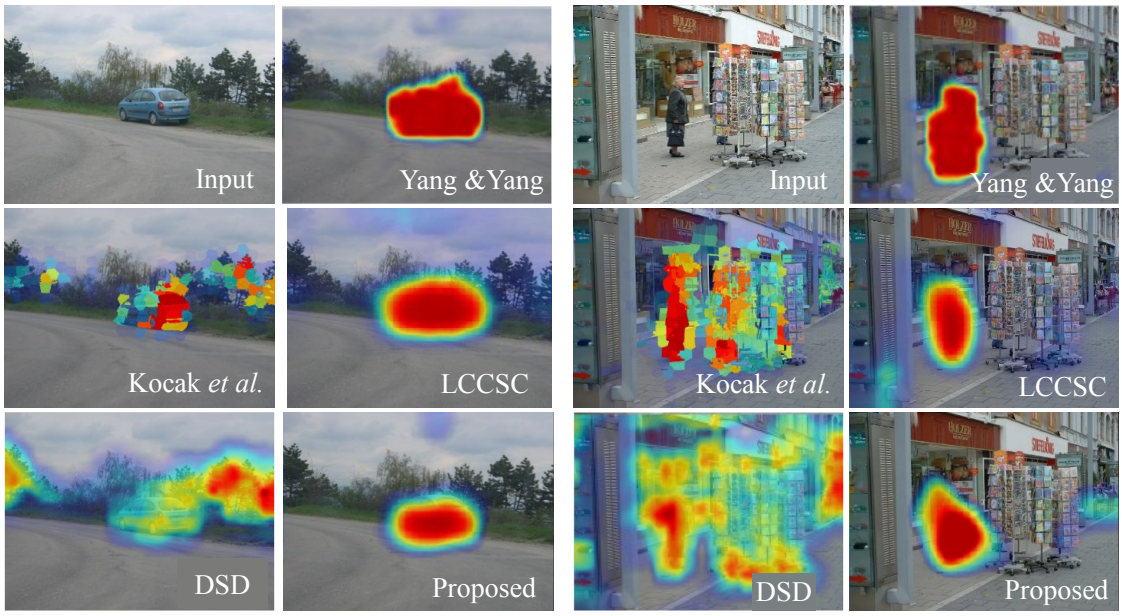


Figure 5.5: Comparison of the proposed weakly supervised method with other fully supervised (Yang&Yang [1] ,Kocak *et al.* [6], LCCSC [7]) and weakly supervised (DSD [8]) top-down saliency approaches on car and person images.

dictionaries are used for each object category. On the contrary, our weakly supervised method does not require any iterative learning and sparse codes are computed just once on a single dictionary. [7] uses a larger dictionary of 2048 atoms for Graz-02 as compared to 1536 atoms in our approach. When their model is evaluated on the entire 600 test images, the mean precision at EER is 52.2% and 52.21% respectively. The discriminative capability of [6] does not improve by incorporating objectness [32] and superpixel features to [1]. The proposed method achieves 54.76% with better discrimination against objects of other categories in a weakly supervised setting. [62, 91] reports results in which each model is tested on images from its own category. Pixel-level results of our proposed model is evaluated using same setting. It is seen that [6] is better (row 13) than proposed weakly supervised approach (row 15), however is inferior to our model (row 5) in removing false positives (row 3). [91] uses 500,000 dictionary atoms in their fully supervised framework to produce 65.13% accuracy as compared to 60.52% in our weakly supervised approach that uses only 1536 atoms. Our results are far superior compared to the fully supervised shape mask [62]. As expected, recent bottom-up saliency model [3] produces

Table 5.2: Pixel-level precision rates at EER (%) on Graz-02.

Method	SV	Test set	Bike	Car	Person	Mean
1 - Zhang *et al.* [3]	TF	All test images	31.77	18.66	30.71	27.1
2 - Yang and Yang [1]	FS		59.4	47.4	49.8	52.2
3 - Kocak *et al.* [6]	FS		59.92	45.18	51.52	52.21
4 - LCCSC [7]	FS		69.07	**58.39**	58.22	61.89
5 - Proposed WS	WS		63.96	45.11	55.21	54.76
6 - FS version	FS		**71.5**	56.6	**62.3**	**63.51**
7 - Zhang *et al.* [3]	TF	Test images from respective category	54.67	39.03	52.04	48.58
8 - Aldavert *et al.* [91]	FS		71.9	64.9	58.6	65.13
9 - Fulkerson *et al.* [90]	FS		72.2	**72.2**	66.1	70.16
10 - Shape mask [62]	FS		61.8	53.8	44.1	53.23
11 - Yang and Yang [1]	FS		62.4	60	62	61.33
12 - Khan and Tappen [63]	FS		72.1	-	-	-
13 - Kocak *et al.* [6]	FS		73.9	68.4	**68.2**	70.16
14 - LCCSC [7]	FS		76.19	71.2	64.13	70.49
15 - Proposed WS	WS		67.5	56.48	57.56	60.52
16 - FS version	FS		**77.61**	71.91	66.95	**72.16**

Table 5.3: Patch-level precision rates at EER (%) on 300 test images.

	Bike	Car	Person	Mean
DSD [8]	62.5	37.6	48.2	49.4
SUN [61]	61.9	45.7	52.2	53.3
Proposed WS	**76.0**	**53.7**	**66.7**	**65.4**

poor performance when compared to our result.

For fair comparison with fully supervised approaches, we report the results of our model in a fully supervised setting as well (FS version), i.e. the contextual saliency model is trained on object patches from training images using patch-level object annotations as in [1], instead of R-ScSPM. With this supervised setting, our model achieves state-of-the art results in top-down saliency.

Table 5.3 compares the patch-level precision at EER of the proposed saliency model on 300 test images with other representative patch-level methods. As evident from Fig. 5.5, DSD [8] has limited capability to remove background clutter, resulting in poor performance of their model. Feature learning using independent component analysis helped SUN [61] to perform better than DSD, but substantially poorer than the proposed method.

Qualitative comparison with weakly supervised top-down saliency [9]

Being an image categorization approach, [9] does not report quantitative results of saliency estimation. Instead saliency maps corresponding to 6 images from Graz-02

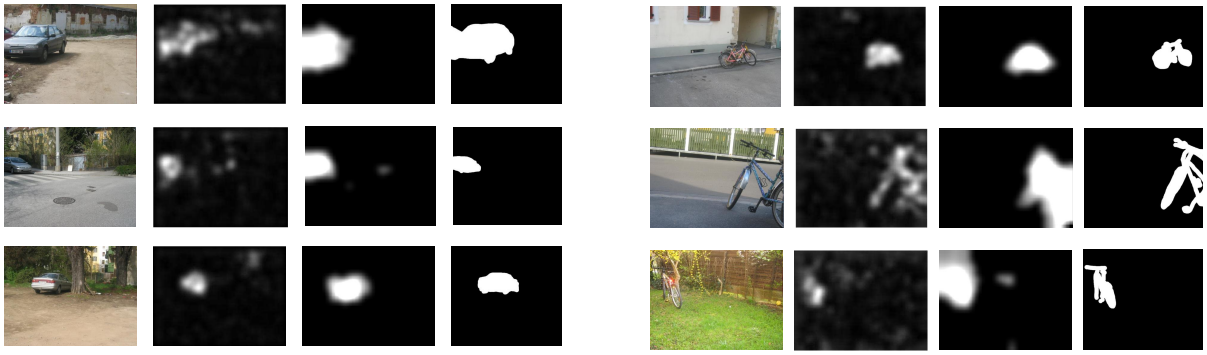


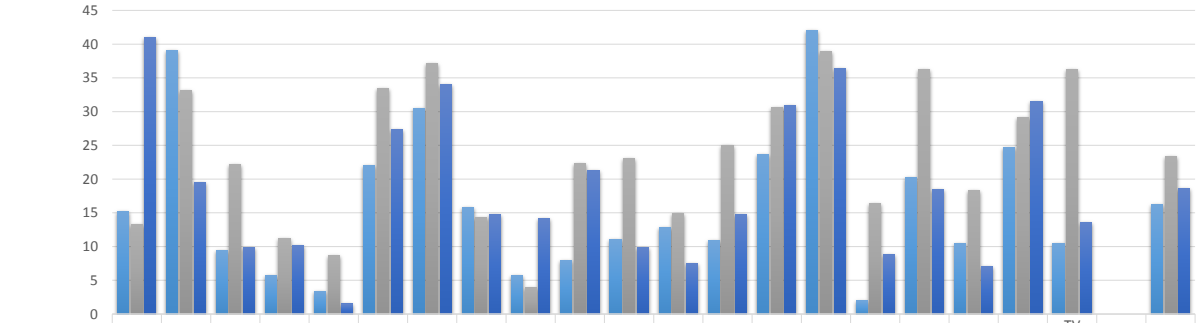
Figure 5.6: Qualitative comparison with [9]. The proposed weakly supervised method produces saliency maps which are similar to the ground truth (GT).

dataset are given in the paper. In contrast to traditional training-test set split, they used 150 even numbered images per category for training and 150 odd numbered images per category for testing. We also evaluated our model using this train-test split and saliency maps corresponding to these 6 test images are produced using our approach. Fig. 5.6 shows that the proposed method clearly outperforms [9] (they evaluate only on Graz-02 dataset and not on PASCAL VOC-07).

5.6.4 PASCAL VOC-07 segmentation dataset

Similar to previous chapters, training of PASCAL VOC-07 models uses object detection images and testing is performed on 210 segmentation test images. Also, to reduce the computational complexity of sparse coding, a common dictionary of 1536 atoms is used for all object classes, which is much smaller than (20×512) atoms of [7] . For each object category, separate sparse codes are computed in [1, 6].

Fig. 5.7 shows the patch-level performance comparison between the proposed WS method and FS top-down saliency approach [1]. Our fully supervised approach in Chapter 3 is shown as LCCSC [7]. Knowledge about object presence inferred from the classifier refinement helped the saliency map to outperform [1, 7] in classes like *aeroplane* and *train*. However, the use of a fixed context neighborhood size and lack of object annotation limits the performance in smaller objects like *bottle* and *bird*. Our method outperforms the fully supervised approach [1] and achieves a higher mean precision rates at EER (%) computed over all 20 classes.



	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV Monitor		Mean
Yang and Yang	15.2	39	9.4	5.7	3.4	22	30.5	15.8	5.7	8	11.1	12.8	10.9	23.7	42	2	20.2	10.4	24.7	10.5		16.2
LCCSC	13.3	33.2	22.1	11.2	8.6	33.5	37.2	14.3	3.9	22.3	23	14.9	25	30.6	38.9	16.4	36.3	18.3	29.2	36.3		23.4
Proposed	41	19.5	9.9	10.2	1.5	27.3	34	14.7	14.1	21.2	9.9	7.5	14.8	30.9	36.4	8.8	18.5	7.1	31.5	13.6		18.6

Figure 5.7: Quantitaive Comparison of proposed weakly supervised approach vs fully supervised top-down saliency approaches on PASCAL VOC-07 (patch-level precision rates at EER (%)

Table 5.4: Percentage of correctly labeled pixels on PASCAL VOC-07 dataset.

Background	Aero plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
73.05	17.9	35.3	5.8	9.1	0.25	15.8	40.0	7.4	18.9	3.12

Dining table	Dog	Horse	Motor bike	Person	Potted Plant	Sheep	Sofa	Train	TV Monitor	Average Accuracy	Overall % of pixels correctly labeled
1.3	6.9	15.13	6.1	37.2	13.8	7.0	9.3	20.7	16.3	17.07	57.31

Khan and Tappen [63] report their pixel-level precision rates at EER only for cow category (8.5%) which is lower than the proposed weakly supervised approach (9.7%). We did not compare with [6] since they manually assign an all zero map if the object of interest is absent. The presence of multiple, visually similar object classes in a single image is challenging for a weakly supervised approach, yet we achieve patch-level precision rate at EER comparable to the fully supervised approaches proposed in the previous chapters.

To compare with related object segmentation approaches, on each of the 210 test images, the saliency maps corresponding to each object models are applied, and the

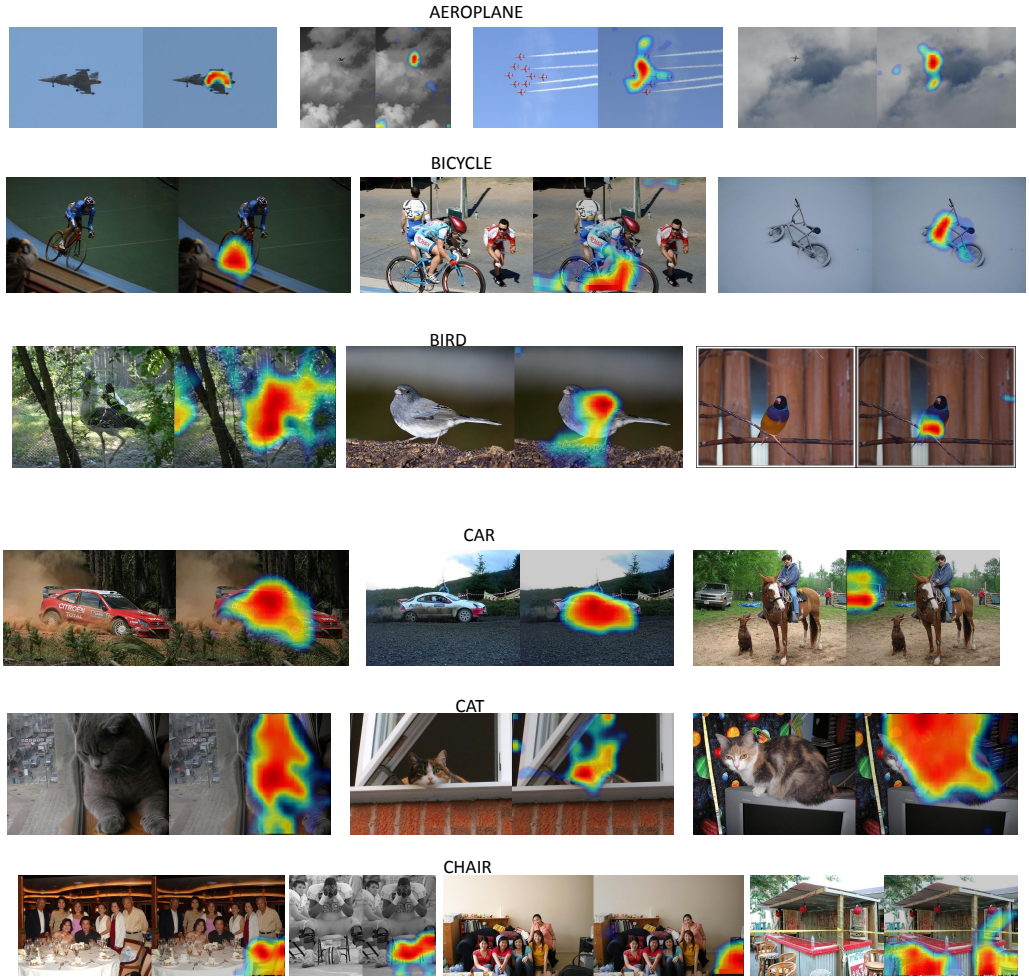


Figure 5.8: Qualitative results of proposed weakly supervised approach on PASCAL VOC-07.

overlapping pixels are assigned with a class corresponding to highest saliency. On this combined image, each category saliency maps are threshold at their EER, so that pixels below threshold are classified as background. Following such an approach, 57.3% of the total pixels are correctly classified, which is better than 57% in [90], which uses fully supervised learning of CRF on superpixel based image features. The percentage of correctly labeled pixels for each class is evaluated and averaged across 21 classes(20 object classes and background class). We obtain a mean of 17.07%, which is comparable to 16% of [91] using similar parameters (SIFT like IHOG feature, hierarchical K-means

Table 5.5: Comparison with segmentation approaches on Object Discovery dataset.

Method	Airplane	Car	Horse
Joulin *et al.* [11]	15.36	37.15	30.16
Joulin *et al.* [96]	11.72	35.15	29.53
Kim *et al.* [97]	7.9	0.04	6.43
Object Discovey [10]	55.81	64.42	51.65
Proposed	57.27	67.42	50.51

based dictionary, using spatial bins and linear classifier). Pixel classification accuracy of individual classes are presented in table. 5.4.

Saliency maps generated on PASCAL VOC-07 segmentation and detection test images are shown in Fig. 5.8. It is hard to identify the presence of bird (column 1) in between trees. Car (column 3) is successfully identified even with the presence of other 3 object categories-dog,horse and person.

5.6.5 Computation time

Training of the proposed framework is significantly faster compared to [1, 6], since we do not use iterative dictionary learning. MATLAB implementations of all approaches were evaluated on a PC running on Intel Xeon 2.4GHz processor. Our unoptimized implementation needs only 3.5 seconds for inference on a test image, which is faster when compared to 5.5 seconds for [1] and 28 seconds for [6]. In our framework, all the saliency models share a common sparse code and contextual max-pooled vector. Inferring another model on same image needs an additional 1 second only. However, [1, 6] needs to calculate sparse codes for each model separately. [7] uses a larger dictionary of different sizes in both datasets. It took 3.85 seconds for inference in Graz-02 dataset and 17 seconds in PASCAL VOC-07.

5.6.6 Applications

5.6.6.1 Object class segmentation

The saliency maps obtained for a particular class are thresholded as in [95] followed by GrabCut [109]. Co-segmentation aims to segment the common object from a given set

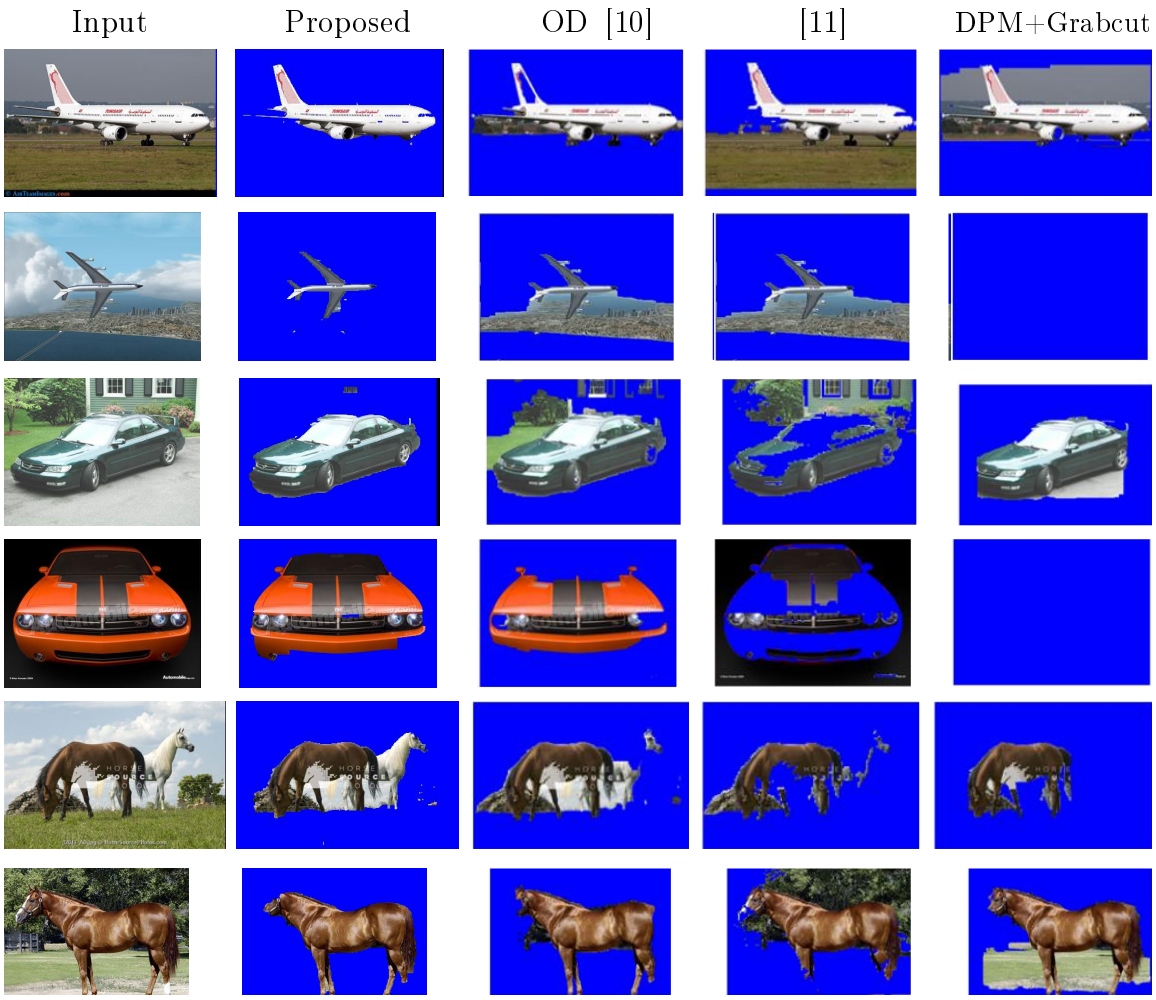


Figure 5.9: Comparison with co-segmentation approaches: object discovery [10], Joulin *et al.* [11] and grabcut applied on DPM detection output on Object Discovery dataset.

of images, which is similar to the image-level label provided in our weakly supervised training which enables a fair comparison with our approach. We train airplane, car and horse models using 130 images per category from PASCAL VOC 2010 detection dataset and evaluated on object discovery dataset [10]. Quantitative comparisons with co-segmentation approaches are shown in table 5.5 and qualitative comparisons are shown in Fig. 5.9.The jaccard similiarity, i.e, intersection over union (IOU) with the ground-truth is evaluated as in [10]. Although [10] performs better than our method on the horse class, we achieve better precision (84.09% vs 82.81%) which indicates that the proposed method can remove false detections on negative images.

Segmentation results obtained using our saliency maps are compared with OD (object

Table 5.6: Comparison with weakly supervised object annotation approaches on PASCAL -07 detection dataset

Method	Nguyen *et al.* [116]	Siva and Xing [117]	Siva *et al.* [118]	Proposed
Annotation accuracy (Avg. of 20 Classes)	22.4	30.4	32.0	36.22

discovery) [10], Joulin *et al.* [11], and DPM [41]+Grabcut implementation given in [28]. We did not compare with semantic object selection [28] since their training requires an additional level of supervision to select training images having white background.

5.6.6.2 Object annotation

We generated rectangular boxes from our saliency maps using coherent sampling [118] to annotate PASCAL VOC-07 detection images. As in [118] we select the first object location proposal in each image as the annotation of the object of interest. If $IOU > 0.5$ it is labeled as a correct annotation. Table 5.6 shows average annotation accuracy across 20 object categories. It illustrates that proposed approach is better than a deformable part-based model [41] trained on saliency maps of [118], which in turn uses several iterations of weakly supervised training to produce the reported result. It can be observed from Fig. 5.10 that the proposed method can successfully annotate objects (yellow boxes) even on images having low contrast with the background (bird and cat images). Green colored rectangular box indicates the ground truth.

5.6.6.3 Action-specific patch discovery

We aim to automatically identify patches that help to describe the action. Qualitative evaluation of our R-ScSPM patch selection strategy on PASCAL VOC-2010 action dataset indicates that it is effective in identifying the most representative patches of different action categories as shown in Fig. 5.11. The representative patches of an action category include class-specific objects as well as the action-specific orientation of human body parts. The patches corresponding to the body part performing the action, namely the hand, and the objects with which the hand interacts, namely the phone, camera and instrument have been extracted correctly. It can be observed that for *phoning* category, hand near the ear together with phone are identified as the class-specific patches.

Figure 5.10: Object annotation obtained on PASCAL VOC-07 detection training images using proposed approach. Green rectangular boxes show the ground truth and yellow boxes indicate the annotation boxes obtained using the proposed approach.

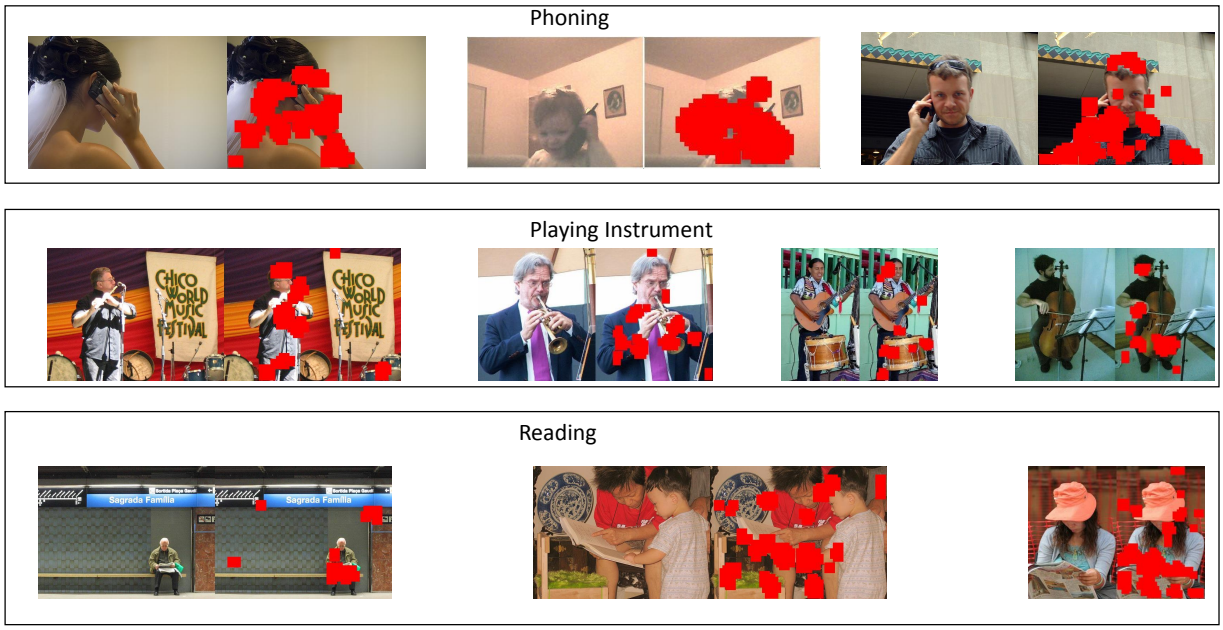


Figure 5.11: Action category-specific patches identified on PASCAL VOC 2010 action dataset training images by the proposed patch selection strategy; i.e, by thresholding R-ScSPM saliency at *0.5*).

The instruments in *playing instrument*, books and newspaper in *reading* are detected as category-specific patches.

5.7 Conclusion

In this chapter, a weakly supervised top-down saliency approach is presented that requires just a binary label indicating the presence/absence of the object in an image for training. A novel R-ScSPM framework produces a saliency map based on the contribution of image patches to an ScSPM image classifier. In a cross validation setting, the image patches having high R-ScSPM saliency on validation images are selected to train a contextual saliency model. A logistic regression classifier learnt on contextual max-pooled vectors of image patches estimates the contextual saliency, which is integrated with R-ScSPM saliency to form the final saliency map. Extensive experimental evaluations show that the proposed method performs comparably with that of fully-supervised top-down saliency

approaches. In the next chapter we further improve the quality of saliency maps by replacing sparse codes used in this chapter with CNN features.

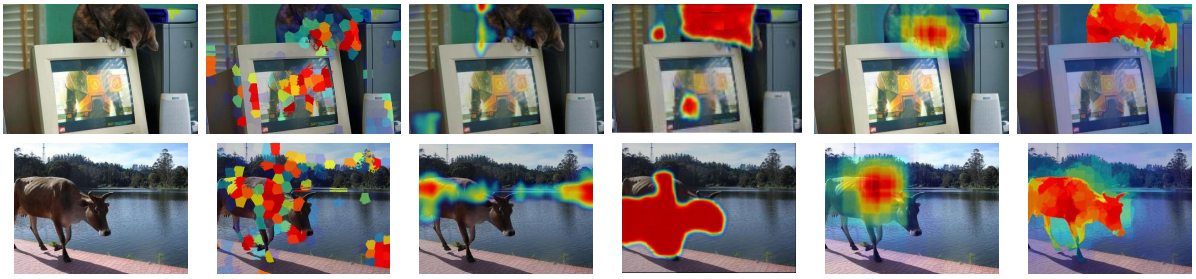
Chapter 6

Weakly Supervised Salient Object Detection using CNN Features

6.1 Introduction

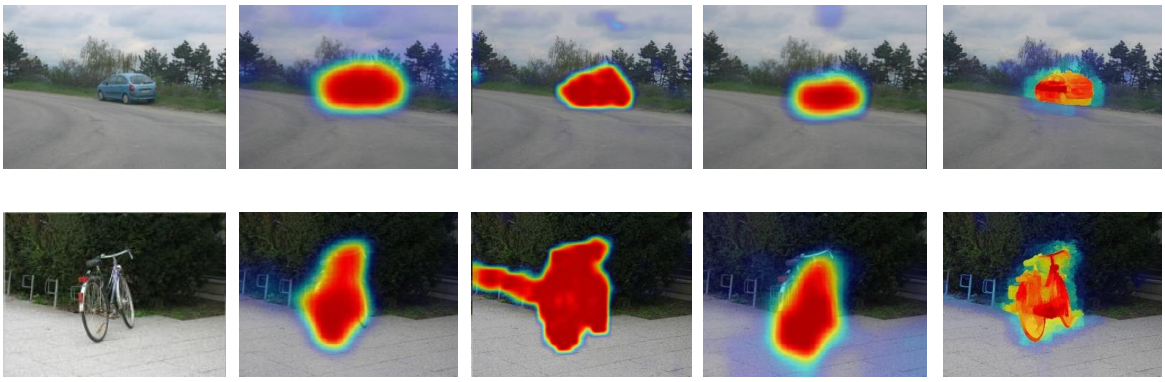
In this chapter we extend the sparse coding based weakly supervised top-down saliency model of the previous chapter using CNN features. Compared with Chapter 5, the framework proposed here has the following modifications: (i) sparse codes of SIFT features are replaced with CNN features. (ii) For a given task, a saliency-weighted max-pooling strategy is proposed to select a bottom-up saliency map among several candidates, which is combined with top-down saliency map to form a hybrid map. (iii) Considering the fact that CNN features span larger spatial context compared to SIFT features, we removed the contextual max-pooling step from the contextual saliency. (iv) Multi-scale averaging of saliency values within each superpixel is carried out to improve accuracy along object boundaries. These modifications lead not only to better performance than the previous version, but also with recent fully supervised top-down saliency approaches.

We first train a convolutional neural network (CNN) image classifier using image-level representation of CNN features, that gives a confidence score on the presence of an object in an image. The probabilistic contribution of each discriminative feature to this confidence score is represented in a top-down saliency map, which is combined with a bottom-up saliency map that is selected from several candidate bottom-up maps through a novel selection strategy. Next, the contextual saliency of each feature is separately evaluated using a dedicated feature classifier.Saliency inference at a pixel involves combining the image classifier-based saliency map and the contextual saliency map.



(a) (b) (c) (d) (e) (f)

Figure 6.1: Visual comparison of the proposed weakly supervised approach with fully supervised top-down saliency approaches. (a) Input image, top-down saliency maps of (b) Kocak *et al*. [6], (c) LCCSC (Chapter 3), (d) Yang and Yang [12], (e) Exemplar [13] and (f) the proposed method for cat (top row) and cow (bottom row) categories.



(a) (b) (c) (d) (e)

Figure 6.2: Comparison of the proposed CNN-based weakly supervised method with fully supervised and weakly supervised algorithms proposed in the previous chapters. (a) Input image, (b) LCCSC proposed in Chapter 3, (c) CG-TD proposed in Chapter 4, (d) weakly supervised WS-SC proposed in Chapter 5 and (e) CNN-based weakly supervised approach proposed in this chapter.

Extensive experiments on 7 challenging datasets across 4 different application shows that the proposed method achieves state-of-the-art performance compared to fully supervised CNN based top-down saliency [13] as demonstrated in Fig. 6.1. It produces better quality saliency maps compared all algorithms proposed in the previous chapters as shown in Fig. 6.2. We achieve state-of-the-art performance in weakly supervised semantic segmentation, weakly supervised semantic object selection and weakly supervised object localization applications. Our performance is comparable with CNN based, state-of-the-art co-saliency, co-segmentation, weakly supervised object detection, and

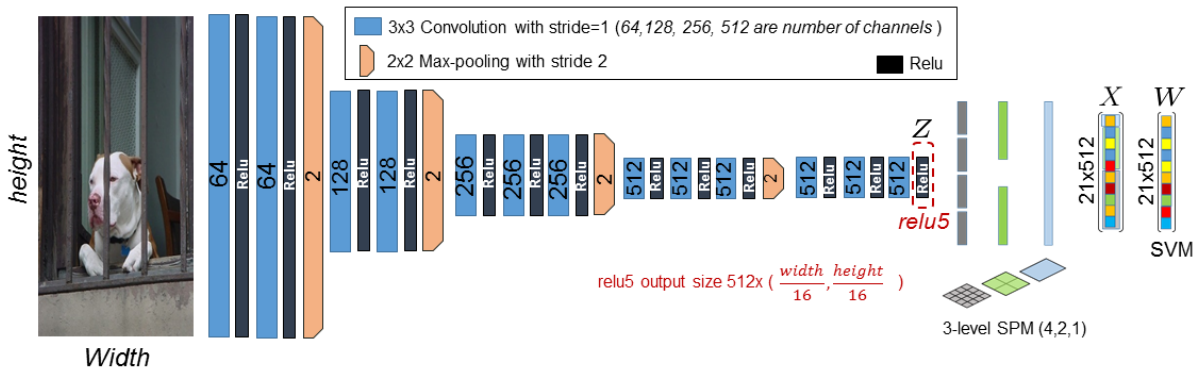


Figure 6.3: Architecture of our CNN image classifier. Pool5 in VGG-16 [14] is replaced with a 3-level SPM and the fully connected layers are replaced with a binary linear SVM.

category-independent salient object segmentation approaches.

The proposed top-down selection of bottom-up saliency maps is described in section 6.4, then we present weakly supervised R-ReluSPM framework to obtain R-ReluSPM saliency (section 6.5.1) and computation of C-ReluSPM saliency (Fig. 6.4). We then introduce contextual saliency (6.6) that estimates object presence in a region. Finally, during inference (section 6.7), C-ReluSPM saliency map and contextual saliency map are combined, which is then re-sized to image resolution. The pixel-level accuracy of the final saliency map is improved using multi-scale superpixel-averaging (section 6.7.1).

6.2 Structure of VGG-16 CNN and modifications for our ReluSPM

In Chapter 5, we extracted dense SIFT features from equally spaced image patches and these features are represented as a 1536 dimensional sparse code. Here, our objective is to replace sparse codes of image patches with its CNN-features, so that the same backtracking framework can be utilized for saliency estimation. The CNN features of an image are extracted using VGG-16 [14] CNN that has been pre-trained on the ImageNet ILSVRC 2012 data with only image-level supervision (no bounding box annotation). We directly use the convolution weights which are trained for image classification, without fine-tuning. However, unlike [14, 119], we do not crop/resize the input image for feature extraction, but use the original image at its full resolution. A fixed-length, image-level

representation of CNN features is obtained through a multi-scale spatial pyramid max-pooling as in Spatial Pyramid Pooling (SPP-net) [120] image classifier. We use a binary linear SVM after the spatial pyramid pooling layer, instead of fully connected layers in [120]. In order to reduce computations, we extract CNN features at a single image scale instead of multiple scales. Our CNN image classifier is referred as ReluSPM in the following discussions.

In VGG-16, there are 16 weight layers in which 13 are convolution layers and 3 are fully connected layers. Every convolution filter has a uniform receptive field of 3×3 and each convolution layer is followed by a rectification non-linearity ($relu$) layer which replaces all negative values in the convolution output with zeros. The width of the convolution layer, which is also called the number of channels is 64 in the first layer and increases by a factor of 2 until 512. Spatial pooling is carried out by five max-pooling layers over 2×2 pixel window, with a stride of 2 placed between convolution layers of different widths (see Fig. 6.3). The three convolution layers and three $relu$ layers before fifth max-pooling (pool5) are denoted as $conv5\_1$, $conv5\_2$, $conv5\_3$, $relu5\_1$, $relu5\_2$ and $relu5\_3$ respectively. Other layers are denoted similarly. The spatial dimensions of an input image are down sampled by a factor of 16 at the $relu5\_3$ feature map due to spatial max-pooling in the first four layers. There are 512 filter channels in $conv5\_3$ (third convolution in the fifth layer), and hence each spatial location in $relu5\_3$ feature map can be represented using a 512 dimensional feature vector, which we refer to as relu5 feature. The relu5 feature represents the overall response of multiple pixels from its receptive field in the original image.

6.3 Implementation of ReluSPM image classifier

Let $Z = [z_1, z_2 ... z_m, ... z_M]$ denote M relu5 features each of dimension d. The spatial distribution of the features in the image is encoded in the spatial pyramid max-pooled image vector X through a multi-scale max-pooling operation $F(z_1, z_2, ..., z_M)$ of the relu5 features on a 3-level spatial pyramid [73] as shown in Fig. 6.4. Similar to previous chapter, the i^{th} element x_i of X is a max-pooled value derived using maximum operation on j^{th} elements of all relu5 features in a spatial pyramid region $\mathcal{R}$ defined by i, and

$j = 1 + (i-1)\ mod\ d$. It is represented as

$$x_i = max\{z_{1j}, z_{2j}, ...z_{qj}\}, \quad \forall\ 1, 2...q \in \mathcal{R}. \tag{6.1}$$

Let the label $Y_k \in \{1, -1\}$ indicate the presence or absence of an object O in the k^{th} image. If $Y_k = 1$, it is a positive image, else it is a negative image. Image-label pairs (X_k, Y_k) of T training images are used to train a binary linear SVM classifier [121, 122] with weight vector $W = [w_1, w_2....w_N]^\top$ and bias b. W is learnt separately for each object category. N is the length of the max-pooled image vector X_k.

Given a validation/test image with max-pooled vector X, the classifier score $W^\top X + b$ indicates the confidence of the presence of object O in it, which is normalized to $[0, 1]$ using the *sigmoid* function.

6.4 Top-down selection of bottom-up saliency map

As mentioned in section 2.3, a hybrid saliency framework that contains both top-down and bottom-up components can produce a better quality saliency map. Moreover, such hybrid approaches can be configured for both top-down salient object detection as well as for task-independent salient object detection. In this section, we propose a novel strategy to select the best saliency map from a set of bottom-up maps based on saliency-weighted max pooling [123].

State-of-the-art bottom-up saliency approaches [3, 4] can produce a category- independent saliency map for an image within 40 milliseconds. They assume image boundaries as the background while approaches such as [5] focus on feature contrast to estimate saliency. These approaches do not require any training and give reasonably good results. Since bottom-up saliency maps are task-independent from a user's perspective, the definition of 'good saliency map' varies based on the application. For example, consider the cat image in Fig. 6.1, where two different objects are present. If a user searches for a 'cat' in the image, bottom-up approaches [3, 4] that assume image boundary as the background fail to produce a 'good saliency map'. In such scenarios, an approach [5] that does not use such assumptions can produce better results. Thus, our objective is to develop a strategy to select a bottom-up saliency method for a particular image that is best suited for the task at hand.

Our ReluSPM image classifier (W, b) which was trained to estimate the presence of object O in an image is employed to select a bottom-up saliency map suitable for the task of identifying image regions that belong to object O. To achieve a one-to-one correspondence between pixels in the bottom-up saliency map and the relu5 features, we downsample the saliency maps to the spatial resolution of feature map at relu5_3 , i.e., by a factor of 16. From n_ρ bottom-up saliency maps, we need to select one for which features that belong to an object are assigned high saliency and those that do not belong to an object are assigned low saliency. For a max-pooled vector X of an image, the SVM predicts a confidence score $W^\top X + b$ which is proportional to the confidence of object presence in that image. i.e.,

$$\Theta\ (Y = 1 \,|\, X) = W^\top X + b = \sum_{\forall i \in \{1,..N\}} w_i x_i + b, \tag{6.2}$$

$$= \sum_{\forall i \in \mathcal{I}^+} w_i x_i + \sum_{\forall i \in \mathcal{I}^-} w_i x_i + b. \tag{6.3}$$

$$\Theta\ (Y = 1 \,|\, X) = \sum_{\forall i \in \mathcal{I}^+} w_i x_i - \sum_{\forall i \in \mathcal{I}^-} |w_i| x_i + b \tag{6.4}$$

where

$$\mathcal{I}^+ = \{i \mid w_i > 0\}, \quad \forall i \in \{1, 2, ..N\},$$

$$\mathcal{I}^- = \{i \mid w_i < 0\}, \quad \forall i \in \{1, 2, ..N\}.$$

Ideally, features belonging to object O contribute positively to the classifier confidence and hence they correspond to elements in X whose indices belong to $\mathcal{I}^+$, while the background features result in $\mathcal{I}^-$ indices. It is to be noted that x_i is non-negative since it is derived from relu5 through max-pooling operation.

First, the m^{th} feature z_m is weighted with ρ_m^t, the bottom-up saliency value for that feature estimated by t^{th} approach. i.e, $\hat{z}_m = z_m \times \rho_m^t$. The saliency-weighted relu5 features $\hat{Z} = [\hat{z}_1, \hat{z}_2 ... \hat{z}_m, ... \hat{z}_M]$ are used to estimate the saliency-weighted max-pooled vector $\hat{X}$ and similar to eq. (6.4), the modified confidence score $\hat{B}(t) = \Theta\ (Y = 1 \,|\, \hat{X})$ due to the t^{th} bottom-up map is computed as,

$$\hat{B}(t) = \sum_{\forall i \in \mathcal{I}^+} w_i \hat{x}_i - \sum_{\forall i \in \mathcal{I}^-} |w_i| \hat{x}_i + b. \tag{6.5}$$

If higher values in the saliency map produced by algorithm t falls exactly on the object regions, the second summation will be largely reduced, due to weighting background indices with low saliency values and hence $\hat{B}(t)$ will be high. If some of the background also garners high saliency, then $\hat{B}$ will be relatively low. In order to reinforce the above assertion, we invert the saliency map (by subtracting saliency values from the maximum saliency value in the image), and recompute the saliency-weighted max-pooled vector $\tilde{X}$ and $\tilde{B}(t)$ using same procedure. i.e,

$$\tilde{B}(t) = \sum_{\forall i \in \mathcal{I}^+} w_i \tilde{z}_i - \sum_{\forall i \in \mathcal{I}^-} |w_i| \tilde{z}_i + b. \tag{6.6}$$

If all object regions are assigned with higher saliency values in eq. (6.6), higher weights are assigned to the background regions and lower weights to the salient regions, leading to a lower score of $\tilde{B}(t)$. Combining the above two observations, an ideal saliency map should maximize

$$\hat{B}(t) - \tilde{B}(t) = \sum_{\forall i \in \mathcal{I}^+} w_i (\hat{x}_i - \tilde{z}_i) - \sum_{\forall i \in \mathcal{I}^-} |w_i| \, (\hat{x}_i - \tilde{z}_i). \tag{6.7}$$

In order to prevent the selection of a map that assigns high saliency to the entire image, we impose a penalty of $1 - \mu_t$ on saliency map t with a mean saliency μ_t. Combining the above observations, the final objective function to select a bottom-up saliency map is

$$\mathcal{B}(t) = \{ \sum_{\forall i \in \mathcal{I}^+} w_i (\hat{x}_i - \tilde{z}_i) - \sum_{\forall i \in \mathcal{I}^-} |w_i| \, (\hat{x}_i - \tilde{z}_i) \} \times (1 - \mu_t). \tag{6.8}$$

If the saliency map of t^{th} algorithm is not aligned with the object, then the false positives will increase $\hat{x}_i$ and decrease $\tilde{z}$ in $\mathcal{I}^-$, thus increasing the second term of eq. (6.8). False negatives will reduce $\hat{x}_i$ and increase $\tilde{z}$ reducing the first term. Hence an inaccurate bottom-up saliency map will result in low $\mathcal{B}(t)$. The saliency map that maximizes eq. (6.8) is selected.

In addition to choosing individual bottom-up saliency maps, we also analyze whether a combination of these maps has an effect on improving top-down saliency. To this end, we combine saliency maps by picking the maximum saliency for each pixel and use eq. (6.8) to select the best map from a set of saliency maps that includes the maximum

map. In this section, we have assumed that the SVM weights learnt for an object is accurate and that the object appears only at locations where w_i are positive. Although this may not be always true, we retain this assumption since object locations are not available in a weakly supervised setting.

6.5 C-ReluSPM saliency: Combining bottom-up and top-down saliency

The R-ReluSPM saliency map and the selected bottom up saliency map are combined through a simple multiplication to generate a combined saliency map as shown in Fig. 6.4. We denote this hybrid saliency map as $\mathcal{H}$.

6.5.1 R-ReluSPM saliency

The ReluSPM image classifier follows an identical framework with the ScSPM image classifier of Chapter 5, except that the sparse codes are replaced with $relu5$ features. So a strategy identical to that in section 5.3 is used to backtrack linear-SVM and multi-scale max-pooling operations to the $relu5$ feature locations. Eq. (5.3), eq. (5.5), eq. (5.6) and eq. (5.8) are applicable to this chapter as well.

The confidence of the presence of object O in an image is indicated by the classifier score $W^\top X + b$ as mentioned in the previous section and the element x_i of X has a corresponding weight w_i. Fig. 6.4 illustrates three relu5 features z_A, z_B and z_C. The elements from the Hadamard product $W \circ X$ with $w_i x_i > 0$ mark the features z_A and z_B that contribute positively to the classifier confidence through a $F^{-1}(.)$ operation, i.e the set Ω. The feature $z_C \notin \Omega$ as it does not contribute positively to classifier confidence. The contribution of feature z_A in the absence of other features is evaluated using max-pooling operations $F(\vec{0}.., z_A, .., \vec{0})$ in which all features except z_A are replaced with $\vec{0}$ forming max-pooled vector X_A. Finally, the saliency G of a feature m is evaluated from X_m using a scalar product $(\cdot)$ with the classifier weight W as,

$$G = \begin{cases} \beta\ (W^\top F(\vec{0}.., z_m, .., \vec{0}) + b) & \text{if } m \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{6.9}$$

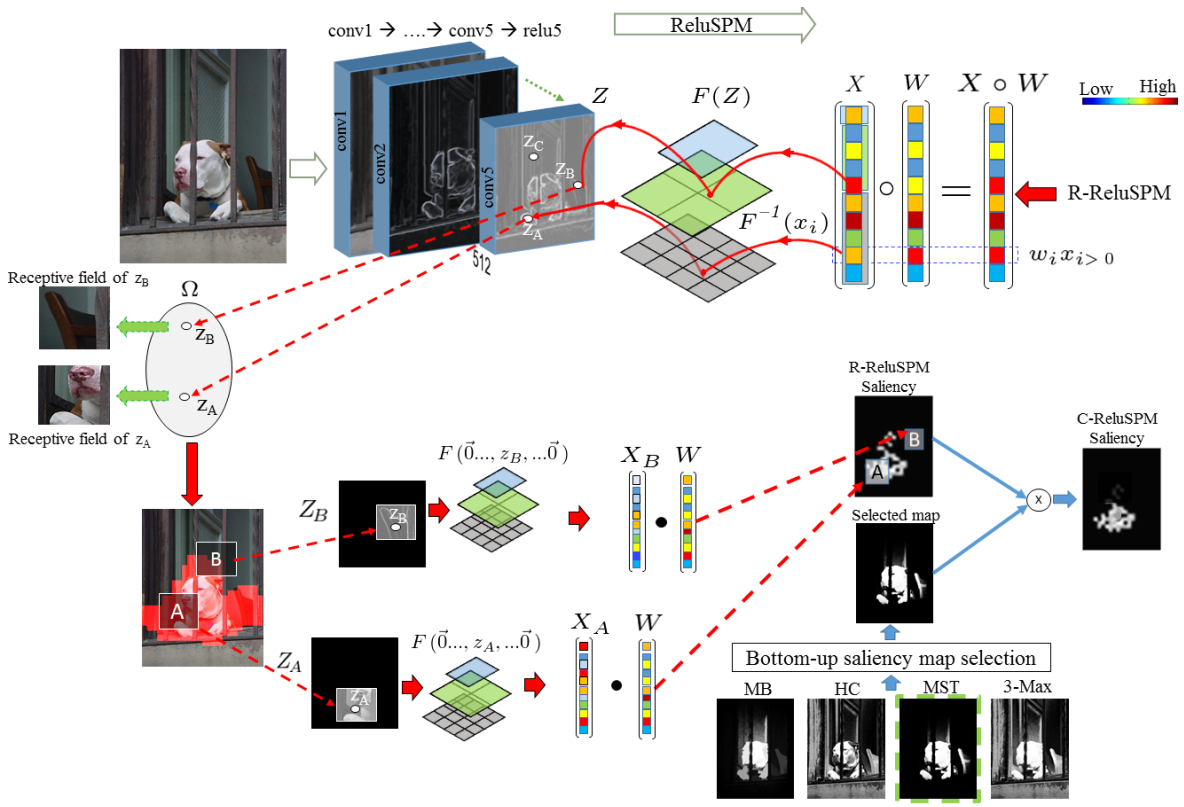


Figure 6.4: Illustration of combined (C-ReluSPM) saliency estimation for *dog* category. Red arrows indicate the proposed backtracking strategy for top-down saliency (R-ReluSPM). From a set of bottom-up saliency maps, the best one is selected and is integrated with R-ReluSPM saliency to produce C-ReluSPM saliency. 3-Max is the saliency map obtained by taking the maximum saliency at each pixel across the 3 bottom-up saliency maps.

Here β is the sigmoid function which we used to map the confidence score to a probabilistic value between 0 and 1. Similar procedure is followed for all patches in Ω. Since this top-down saliency of a feature is arrived at by reverse tracing the ReluSPM classifier, we call it *R-ReluSPM saliency* and the corresponding saliency map as *R-ReluSPM saliency map*. The feature z_A from the object (dog) region is assigned high R-ReluSPM saliency while z_B from background is assigned zero R-ReluSPM saliency.

6.6 Contextual saliency training

Image classifiers trained on image-level representation of features have shown to be effective in discriminative top-down saliency estimation [124, 21, 8]. The combined saliency

map $\mathcal{H}$ takes non-zero values only at discriminative image regions whose features make positive contribution to the image classifier confidence. The assumption is that the object appears only at grids in the spatial pyramid where w_i are positive, which may not be true across all images. Our objective is not limited to identifying the discriminative image regions, but to assign higher saliency values to all pixels belonging to the salient object. In order to independently estimate the saliency value of each relu5 feature, we also learn a top-down *contextual saliency* model that uses a linear SVM learnt on positive and negative relu5 features from the training images. Since feature-level annotation is not available, we use object features extracted using the C-ReluSPM saliency map $\mathcal{H}$ to train the model.

From positive training images of object O, relu5 features with $\mathcal{H}$ saliency greater than 0.5 are selected as positive features with label $l = +1$. In order to prevent training features from non-discriminative object regions of positive images with negative label, only those features at which both R-ReluSPM and bottom-up saliency are selected as negative features with label $l = -1$. Additionally, random features are selected from negative images with label $l = -1$. A linear SVM model with weight v and bias b_v is learned. Since the relu5 features are already computed for R-ReluSPM, learning of linear SVM is the only additional computation required to train this top-down model. The saliency map obtained from contextual saliency is denoted $\mathcal{L}$.

6.7 Saliency inference

For inference on a test image, the C-ReluSPM saliency $\mathcal{H}$ and contextual saliency are first integrated followed by multi-scale superpixel averaging and finally associated with the confidence of the image classifier to obtain the saliency at a pixel. While the combined saliency is obtained as described in section 6.5, the contextual saliency for a feature z_m is the probability of the feature belonging to an object computed by applying a sigmod function β to the linear SVM score,

$$P(l = 1 \mid z_m, v) = \beta\ (v^T z_m + b_v). \tag{6.10}$$

The contextual saliency and combined saliency values are integrated using a mean operation to form the saliency map, $S_p = \frac{\mathcal{H}+\mathcal{L}}{2}$.

6.7.1 Multi-scale superpixel-averaging of saliency map

The low resolution saliency map S_p is upsampled to the original image size using bicubic interpolation. As a consequence, saliency values may not be uniform within a superpixel. Also, the saliency map will not be edge-aware with object regions spreading to the background. Hence, a multi-scale superpixel-averaging strategy is employed. The mean saliency at a superpixel (obtained by SLIC segmentation [125]) is assigned to every pixel in it. This process is repeated at multiple scales by varying the SLIC parameters. The resulting maps are averaged to produce a smooth, pixel-level saliency map S_{pix} that uniformly highlights the salient object and also produces a sharp transition at object boundaries.

6.7.2 Integrating with image classifier confidence

For a given image, the top-down saliency map S_{pix} indicates the probable pixels that belong to object O. Since the presence of a specific object in a test image is not known apriori for applications such as semantic segmentation and object detection, the saliency map needs to be estimated for both positive and negative images. Hence, it is beneficial to integrate S_{pix} with a confidence score that indicates the presence of object O in at least one pixel in the image. For this, we use the same ReluSPM image classifiers learnt earlier for each category. The SVM associated with the ReluSPM image classifier gives a confidence score $\Phi(O)$ for a particular object O as $\Theta\ (Y = 1 \,|\, X)$. These scores are scaled between 0 and 1 as

$$\hat{\Phi}(O) = \frac{exp(\Phi(O))}{\max\limits_{1 \leq j \leq n_c} \{exp(\Phi(j))\}}\ , \tag{6.11}$$

where, n_c is the total number of categories. Unlike soft-max that sums to 1, we normalize the score with the maximum because multiple categories can simultaneously appear in an image such as in PASCAL VOC-2012 [126]. In such scenarios, softmax will end up assigning a lower value to all positive categories. However, our objective is to identify the relative confidence across categories, and assign 1 to the most probable category. To reduce false detections from less probable categories, we assume values of $\hat{\Phi}(O)$ that are less than 0.5 as less important, and replace it with 0. This limits the number of

probable object categories per image to less than 5 categories in most images, and hence the category-specific saliency map S_{pix} needs to be computed only for these few probable object categories. We compute the classifier-weighted, category-specific score for each object O,

$$S_{categ}(O) = S_{pix}(O) \cdot \hat{\Phi}(O). \tag{6.12}$$

6.7.3 Category-independent salient object detection

The proposed category-specific top-down saliency map S_{categ} in eq. (6.12) can be used to compute the category-independent saliency value S_{ind}, by computing the maximum saliency value at each pixel (x,y) as

$$S_{ind}(x,\, y) = \max_{1 \le j \le n_c} \{S_{categ}(j)(x,\, y)\}. \tag{6.13}$$

Since the bottom-up information is integrated to S_{categ} through the combined saliency map $\mathcal{H}$, the $S_{ind}(x, y)$ gives an accurate estimate of saliency maps under free-viewing condition and hence enables our framework to be used as a hybrid salient object detection model.

6.8 Applications of category-specific saliency map

Top-down saliency [12, 6, 124, 13] and tasks like object detection, localization and segmentation mainly differ in their granularity of representation. Object detection produces a tight rectangular bounding box around all instances of objects belonging to user-defined categories. It is necessary to identify both the location as well as the extent of each object. The process of identifying the location of a particular object in an image, without marking the extent of the object, is referred to as object localization [22]. Object segmentation, also referred to as semantic object selection produces a binary mask with '1' indicating all pixels that belong to a user-defined object category. It differs from the task of semantic segmentation, where the objective is to classify each pixel in the image to one of predefined classes. In this section, we detail the use of our top-down saliency framework for the above applications in a weakly supervised setting.

Weakly supervised semantic segmentation

The category-specific saliency maps in the proposed framework can be easily adapted for semantic segmentation. In the saliency map, a pixel with $S_{categ}(O) < 0.5$ is less likely to belong to an object O. The pixels at which the maximum saliency across all categories is less than 0.5 is more likely to be background. Hence, the additional map corresponding to the background category is generated as a uniform map with $S_{categ} = 0.5$. We assign to each pixel the category for which its saliency is the maximum.

Weakly supervised object segmentation

Conventional object segmentation approaches use scribbles or rectangular boxes to indicate the object of interest, while in our approach, only the semantic label of the object of interest is input to the system, similar to the semantic object selection [28]. We threshold our top-down saliency map to identify definite foreground and background regions in an image, followed by Grab-cut [109] to accurately segment out the object of interest. Being a weakly supervised approach, framework is comparable to co-segmentation approaches that segment out a common object from a given set of images. We learn a model for the common object, which helps to achieve faster inference for a newly added test image, whereas co-segmentation approaches need to re-segment every image in the set upon encountering a new image.

Weakly supervised object localization

Object localization deals with locating object O within a positive image. Here, only the location of the object needs to be identified, not its extent. The peaks of our saliency map, S_{pix} indicates the location of object O,

$$Loc\ (O) = \underset{(x,\ y)}{\text{argmax}}\ \{S_{Pix}(O)(x,\ y)\}. \tag{6.14}$$

Weakly supervised object detection

In object detection, multiple instances of the same object category need to be identified separately. This is more challenging than localization and especially so in a weakly

supervised setting. Conventional object detectors such as R-CNN [30, 84] need to classify thousands of category-independent object proposals generated using selective search [31], [85]. This incurs a huge computational cost. The proposed top-down saliency framework simplifies object detection by generating less than 5 proposals for an object category per image. First, the category-specific saliency $S_{categ}(O)$ is binarized by applying a threshold at 0.5. The smallest rectangular box enclosing each disconnected region is the detection box for object O. With this simple strategy, we achieve a performance which is comparable to dedicated weakly supervised object detectors [23].

6.9 Experimental evaluation

We evaluate our weakly supervised saliency model on PASCAL VOC-2012 [126] and PASCAL-S datasets [57] in addition to the PASCAL VOC-07 and Graz-02 datasets used in the previous chapters. Additionally, we use PASCAL VOC-2012 segmentation test set and validation set to evaluate weakly supervised semantic segmentation, Object Discovery dataset [10] to compare with semantic object selection and co-segmentation approaches, and validation set of PASCAL VOC-2012 detection challenge to evaluate object localization and object detection performance. In all these applications, we achieve a performance comparable with dedicated weakly supervised approaches.

PASCAL VOC-2012 is a challenging dataset with category-specific annotations for 20 object categories. It has 5717 training images and 5823 validation images for image classification/object detection challenge. There are 1464 training images, 1449 validation images and 1456 test images for segmentation challenge. Our PASCAL VOC-2012 saliency models are trained using 5717 training images for image classification task. There are 210 test images in the segmentation challenge of PASCAL VOC-2007.

PASCAL-S is a widely used dataset to evaluate category-independent saliency models. It has 850 images picked from the validation set of PASCAL VOC-2010 [114] segmentation images. Given the segmented objects in an image, the ground truth salient objects are marked by twelve subjects under free-viewing condition. We use Object Discovery dataset [10] to evaluate object segmentation. The dataset has three object categories, namely airplane, car and horse. Apart from 100 test images per category, there are 461, 1206 and 779 additional images for airplane, car and horse, respectively.

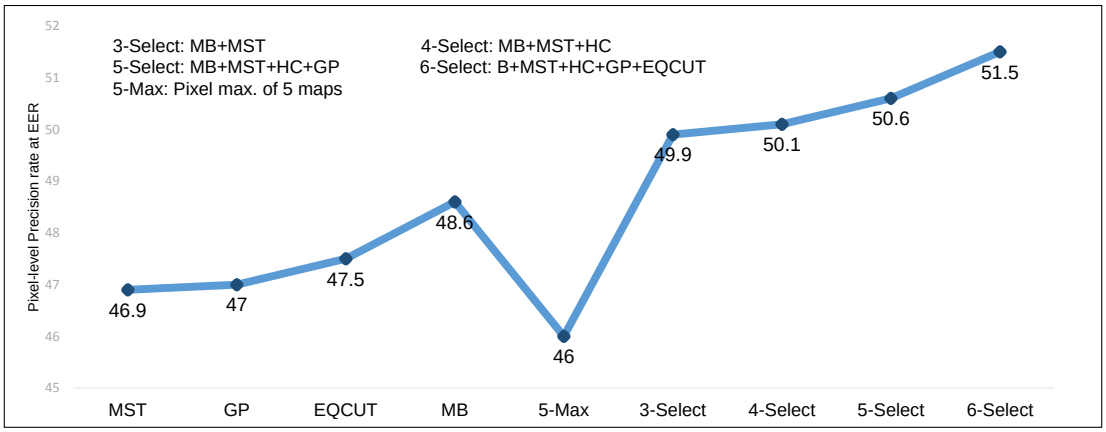


Figure 6.5: Evaluation of selection of bottom-up approaches using pixel-level precision rate at EER (%) across 3 categories of Graz-02 dataset. The proposed selection strategy achieves better performance than the individual algorithms

.

6.9.1 Top-down selection of bottom-up saliency map

Fig. 6.5 illustrates the performance of the proposed strategy for selection of bottom-up saliency map on positive test images, evaluated using mean of pixel-level precision rates at EER across all 3 object categories of Graz-02 dataset. Comparing the individual performances of 5 recent training-free algorithms HC [5], GP [45], EQCUT [53], MST [4] and MB [3] showed that MB outperforms others while HC has the lowest precision rate at EER. Since the Y-axis of the graph in Fig. 6.5 is limited to a range between 45 and 53, HC with mean precision rate at EER of 29.84% is not shown. Across individual categories, MB gives the best performance in bike category while EQCUT outperforms others in car and person categories.

First, we evaluate the performance of a maximum map formed by pixel-level maximum operation across the saliency maps of these 5 algorithms. Since the false positives from all the maps accumulate due to maximum operation, the mean precision rate at EER of this maximum map drops to 46% and denoted 5-Max in Fig. 6.5. Thus, combining bottom-up maps without top-down information about the task can deteriorate the quality of the map.

Next, the proposed strategy to select the best saliency map among MB, MST and their maximum map is evaluated and shown as 3-Select in Fig. 6.5. Although MB outperforms MST in all the 3 categories, a performance boost of 1.3% is observed as a result of the

selection of saliency map from MB, MST or maximum map for those images on which it outperforms others. The same procedure is repeated for the maps of MB, MST, HC and their maximum map and denoted 4-Select. In all the three categories, the performance of newly added HC algorithm is much lower than other approaches (less than 35%). We still observe an improvement of 0.2% in the mean precision rate at EER of 4-Select. Our approach automatically selects the best performing MB for 50.7% of the total images, 33.5% from MST, and only 8.5% from least performing HC. The remaining 7.3% are selected from the maximum map.

Similarly, addition of GP improved the accuracy by 0.5% in 5-Select. It is to be noted that MST, HC and GP are not the best performing algorithms in any of the individual categories, but their addition resulted in a gradual increase in the average accuracy. This shows that even though these algorithms have inferior performance in majority of the images in all 3 categories, they give better quality saliency maps for few images and the proposed selection strategy is able to accurately select those saliency maps.

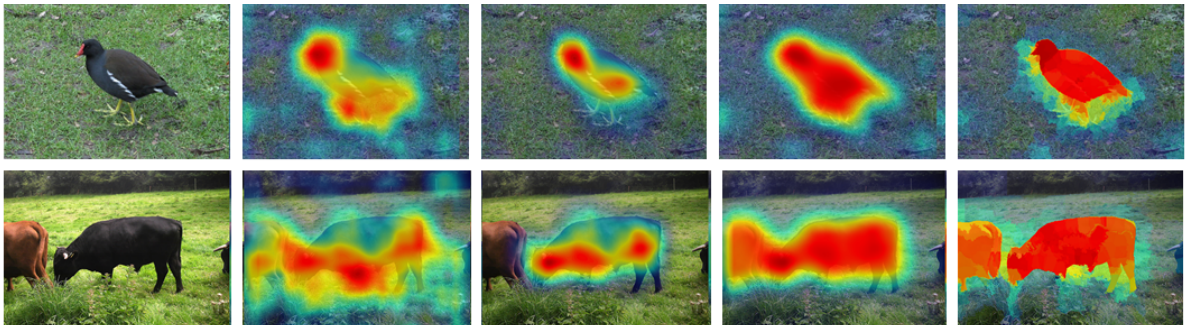
Finally, 6-Select uses saliency maps of MB, MST, HC, GP, EQCUT, and the maximum map. In bike category, the largest number of maps are selected from MB (28% of bike images), which is the best performing algorithm for that category. Similarly, the largest number of car maps (23.3%) are selected from EQCUT, the best performing algorithm for car category. This shows the ability of the proposed strategy to carefully select the best algorithm for a given category.

Computation wise, GP and EQCUT take approximately 10 seconds per image to estimate saliency, while MB, MST and HC need less than 40 milliseconds. We use the latter three algorithms in our final framework (4-Select) to improve the runtime performance of the proposed method. The contribution of the selected bottom-up saliency maps towards the final accuracy in PASCAL VOC-2012 dataset is analyzed in the next section.

6.9.2 Analysis of proposed framework

6.9.2.1 Contribution of individual modules

Fig. 6.6 shows the visual comparison of the effect of each stage in the proposed method. For the input images in Fig. 6.6(a), image regions containing bird's head and cow's legs make positive contribution to their image classifiers and are, therefore, assigned



(a) (b) (c) (d) (e)

Figure 6.6: Qualitative results at individual stages of the proposed method. (a) Input image, (b) R-ReluSPM saliency map, (c) (b) + bottom-up saliency, (d) (c) + contextual saliency, (e) (d) + superpixel averaging.

high R-ReluSPM saliency in Fig. 6.6(b). Combining R-ReluSPM saliency with bottom-up saliency removed false detections in R-ReluSPM saliency as shown in Fig. 6.6(c). Integration of contextual saliency assigns higher saliency value to the non-discriminative object regions (Fig. 6.6(d)). Finally, the addition of the multi-scale superpixel-averaging improved the accuracy along object boundaries as shown in Fig. 6.6(e).

We evaluate the improvement in the mean precision rate (%) at EER at each stage of our framework. The evaluation is done across 20 object categories of PASCAL VOC-2012 segmentation-validation set. The contribution of each component in the proposed saliency model to the final accuracy is shown shaded in Fig. 6.7. The accuracy of R-ReluSPM saliency is 34.1%. On adding the bottom-up map to yield combined saliency, the accuracy increased to 43.6%, demonstrating the effectiveness of the proposed bottom-up selection strategy.

In Chapter 5, we demonstrated that training the contextual saliency model using negative patches from positive images can improve the accuracy up to 5%. Since the training of the proposed method is not iterative, in [124], we selected negative patches only from negative images to remove potentially erroneous negative patches. In the current framework, the accuracy of the combined saliency map $\mathcal{H}$ is improved by weighting R-ReluSPM with the selected bottom-up map, which enabled us to train the contextual saliency using negative patches from positive images. This resulted in an additional improvement of 3% in accuracy, totaling to 16% with the addition of contextual saliency.

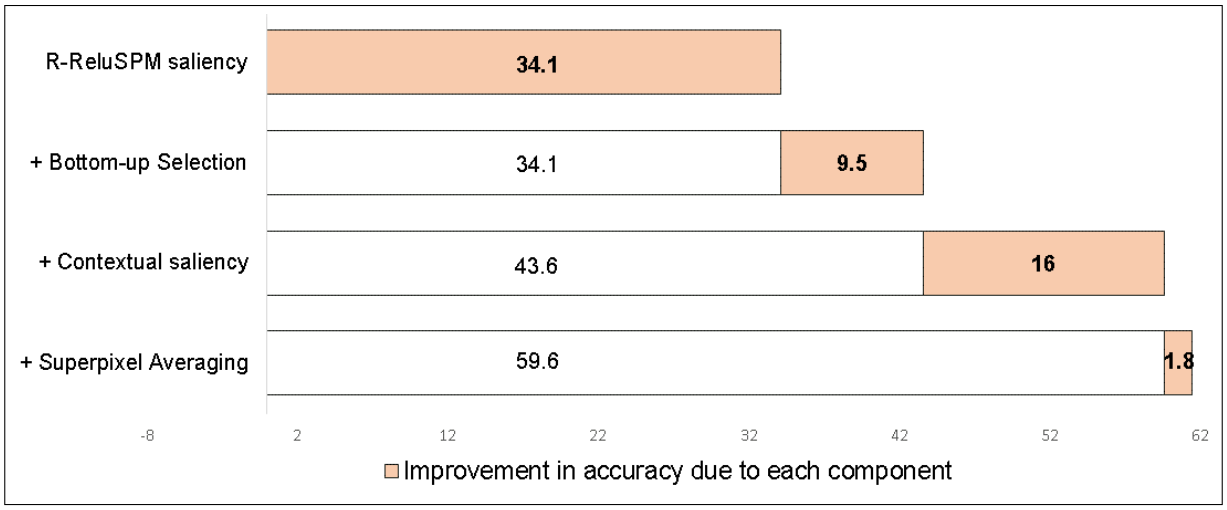


Figure 6.7: Evaluation of individual stages of the proposed framework across 20 categories of PASCAL VOC-2012 using pixel-level precision rate at EER. The improvement in accuracy by the addition of each module is shaded.

This demonstrates that (i) contextual saliency with a dedicated feature classifier plays an important role for top-down saliency and (ii) combined saliency map $\mathcal{H}$ and contextual saliency map complement each other. A similar trend in improvement was observed in [124], where a feature classifier is learnt using contextual max-pooled sparse codes. Since relu5 features span larger spatial context compared to SIFT features computed on 64×64 patches, contextual max-pooling on relu5 features is not required.

The contextual saliency map $\mathcal{L}$ and combined saliency map $\mathcal{H}$ are integrated as $mean\ (\mathcal{H},\ \mathcal{L})$. Other combinations such as $max\ (\mathcal{H},\ \mathcal{L})$ and $\mathcal{HL} + 0.5(\mathcal{H} + \mathcal{L})$ gave similar results with less than 1% variation in accuracy. Taking the product of both saliency maps reduced the accuracy by 6% as the combined saliency $\mathcal{H}$ is often 0 in non-discriminative object regions and multiplication causes 0 values in such locations of the integrated map, disregarding contextual saliency.

Finally, superpixel-averaging is applied at 6 different scales, by extracting 8, 16, 32, 64, 128 and 256 superpixels from an image. The saliency values at each pixel are further averaged across these 6 scales to get the saliency map S_{pix}. The quality of the saliency map at object boundaries is improved leading to 1.8% improvement in the accuracy, to obtain an accuracy of 61.4%. Since superpixel computation at multiple scales is time consuming relative to other modules, inference speed can be largely improved in

applications such as object localization that do not require exact object boundaries by removing this step.

6.9.2.2 Performance comparison on different CNN architectures

We evaluated the performance of the proposed model on 5 different CNN architectures, namely VGG-F, VGG-M, VGG-S [119], VGG-16 and VGG-19 [14]. In all these architectures, layers from 'pool5' to 'prob' are removed to extract relu5 features. The performance is evaluated for all categories in PASCAL VOC 2012 dataset and mean of their pixel-level precision rate at EER is used to compare the architectures. The accuracy of saliency estimation varies by 10% across these architectures. The faster VGG-F gives the lowest accuracy (52%) and the deepest VGG-19 performs the best with 61.9%. The accuracy of VGG-M is 52.1%, followed by VGG-S (55%) and VGG-16 (61.4%). We do not extract relu5 features at multiple scales, nor do we crop or zero pad the input to a fixed size. Since there is no significant performance difference between VGG-16 and the deeper VGG-19, we use VGG-16 in our final framework across all datasets.

6.9.3 Comparison with other approaches

6.9.3.1 Graz-02 dataset

We report our pixel-level results on different test set configurations of Graz-02. First, the proposed saliency model is compared with other top-down saliency algorithms [1, 6] on all 600 test images. The algorithms proposed in this thesis are indicated as LCCSC (Chapter 3), CG-TD (Chapter 4) and WS-SC (Chapter 5). Second, for comparison with related approaches [62, 90], each object category is evaluated on test images from its respective category. Finally, to compare with [8, 61], results on 300 test images are evaluated, where 150 test images are from a single category and the remaining 150 are from the background.

The pixel-level comparisons in the first two test set configurations are shown in Table 6.1, where SV indicates supervision level with TF, WS, FS referring to training-free, weakly supervised and fully supervised training, respectively. [1, 6] and [123] are fully supervised (FS), needing multiple iterations of CRF learning with sparse codes relearned

Table 6.1: Pixel-level precision rates at EER (%) on Graz-02.

Method	SV	Test set	Bike	Car	Person	Mean
1 - Yang and Yang [1]	FS	All test images	59.4	47.4	49.8	52.2
2 - Kocak *et al.* [6]	FS		59.9	45.2	51.5	52.2
3 - LCCSC [7]	FS		69.1	58.4	58.2	61.9
4 - CG-TD [123]	FS		64.4	50.9	56.4	57.2
5 - WS-SC[124]	WS		64.0	45.1	55.2	54.8
6 - Proposed	WS		80.5	61.4	75 .0	72.3
7 - MB [3]	TF	Test images from respective category	54.67	39.03	52.04	48.58
8 - Aldavert *et al.* [91]	FS		71.9	64.9	58.6	65.13
9 - Fulkerson *et al.* [90]	FS		72.2	72.2	66.1	70.16
10 - Shape mask [62]	FS		61.8	53.8	44.1	53.23
11 - Yang and Yang [1]	FS		62.4	60	62	61.33
12 - Khan and Tappen [63]	FS		72.1	-	-	-
13 - CG-TD [123]	FS		67.3	59.8	57.1	61.4
14 - WS-SC[124]	WS		67.5	56.5	57.56	60.5
15 - Proposed	WS		84.1	81.5	81.8	82.5

at each iteration. Separate dictionaries are used for each object category. On the contrary, the proposed weakly supervised method does not require any iterative learning and the relu5 features are extracted with a single forward pass on the CNN. [124] does not require any iterative learning and uses a smaller dictionary of 1536 atoms, compared to 2048 atoms used in [7]. Despite incorporating objectness [32] and superpixel features to [1], the discriminative capability of [6] did not improve (row 2 vs row 1). The proposed weakly supervised method (row 6 and row 15) outperforms all other fully supervised top-down saliency approaches [1, 6, 123] .

With respect to the second test configuration, [62] requires images to be marked as *difficult* or *truncated* in addition to the object annotation for training of shape mask. [91] uses 500,000 dictionary atoms in their fully supervised framework to obtain 65.13% (row 8), whereas the dimension of our relu5 feature is only 512. In this test setting, the proposed method achieves a mean accuracy of 82.5% outperforming the weakly supervised approach in Chapter 5 [124] by 22%.

DSD [8] and SUN [61] did not evaluate their model on Graz-02 dataset, but Yang and Yang [1] reported their patch-level precision rates at EER on 300 test images as 49.4% and 53.3%, respectively. Feature learning using independent component analysis helped SUN to perform better than DSD, but substantially poorer than [124] (65.4%) and the proposed method. It is to be noted that the performance of [8, 61, 1, 124] deteriorates while converting their patch-level results to pixel-level. The proposed weakly supervised

Table 6.2: Pixel-level precision rates at EER on validation set of PASCAL VOC-2012 segmentation dataset.The proposed weakly supervised approach outperforms all fully supervised approaches including [13], which is based on CNN, in 14 out of 20 classes and in mean accuracy.

Method	SV	plane	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	Mean
Yang [1]	FS	14.7	28.1	9.8	6.1	2.2	24.1	30.2	17.3	6.2	7.6	10.3	11.5	12.5	24.1	36.7	2.2	20.4	12.3	26.1	10.2	15.6
Kocak [6]	FS	46.5	**45.0**	33.1	**60.2**	25.8	48.4	31.4	64.4	19.8	32.2	44.7	30.1	41.8	72.1	33.0	40.5	38.6	12.2	64.6	23.6	40.4
Exemplar [13]	FS	55.9	37.9	45.6	43.8	47.3	**83.6**	57.8	69.4	22.7	68.5	37.1	72.8	63.7	69.0	57.5	**43.9**	66.6	38.3	**75.1**	**56.7**	56.2
Oquab [22]	WS	48.9	42.9	37.9	47.1	31.4	68.4	39.9	66.2	**27.2**	54.0	38.3	48.5	56.5	70.1	43.2	42.6	52.2	34.8	68.1	43.4	48.1
Proposed	WS	**71.2**	22.3	**74.9**	39.9	**52.5**	82.7	**58.9**	**83.4**	27.1	**81.1**	**49.3**	**82.4**	**77.9**	**74.2**	**69.8**	31.9	**81.4**	**49.8**	63.2	53.3	**61.4**

method gives a mean pixel-level precision rate at EER of 73.1% which is better than the 70.16% and 70.49% reported by [6] and [7] respectively in this test setting. In all the three test settings, the proposed modifications enabled our current model to outperform [124] by more than 18% in accuracy, achieving state-of-the art performance. The use of CNN features contributes mainly to this performance boost.

6.9.3.2 PASCAL VOC-2012 segmentation dataset

In Table 6.2, we compare a recent CNN-based fully supervised top-down saliency [13] with our method by evaluating on PASCAL VOC-2012 segmentation-validation set consisting of 1449 images. Similar to [13], each object category is evaluated only on positive images of that category. We did not fine-tune the convolution layers for this dataset, which took nearly 8 days on a GPU in [13]. Moreover, we only need a single CNN forward pass to extract features, while [13] requires 500 forward passes for an image. The presence of multiple, visually similar object classes in a single image is challenging for a weakly supervised approach. Inspite of this, we outperform the state-of-the art fully supervised approach [13] and the CNN-based weakly supervised object localization approach [22] in mean accuracy by 5% and 13%, respectively. We outperform [22] in 15 out of the 20 categories. The top-down selection of bottom-up approach along with contextual saliency plays an important role in this improved performance, especially in classes like aeroplane and sheep.

6.9.3.3 PASCAL VOC-2007 segmentation dataset

Similar to our previous chapters, saliency models are evaluated on 210 segmentation test images. We used the models trained on PASCAL VOC-2012 training set in this

Table 6.3: Precision rates at EER(%) on PASCAL VOC-2007.

Method	Yang and Yang [1]	LCCSC [7]	CG-TD [123]	WS-SC[124]	Proposed
Supervision	FS	FS	FS	WS	WS
Mean of 20 classes	16.7	23.4	23.81	18.6	42.1

Table 6.4: Intersection over union (IOU) for semantic segmentation on validation set and test set of PASCAL VOC-2012.

Method	BG	plane	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	Mean
											Val											
MIL-FCN [127]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.9
MIL-Base with ILP [113]	73.2	25.4	18.2	22.7	**21.5**	28.6	39.5	44.7	46.6	11.9	40.4	11.8	45.6	40.1	35.5	35.2	20.8	41.7	17.0	34.7	30.4	32.6
EM adapt [128]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
Pathak *et al.* [129]	68.5	25.5	18.0	25.4	20.2	36.3	46.8	**47.1**	48.0	**15.8**	37.9	**21.0**	44.5	34.5	46.2	40.7	**30.4**	36.3	**22.2**	38.8	**36.9**	35.3
Proposed	**77.7**	**57.4**	**18.6**	**58.2**	19.3	**40.5**	**62.0**	40.9	**69.7**	11.0	**50.7**	14.3	**65.9**	**49.3**	**50.9**	**54.8**	13.5	**54.2**	21.5	**47.0**	36.2	**43.5**
											Test											
Pathak *et al.* [129]	-	24.2	19.9	26.3	18.6	38.1	51.7	42.9	48.2	15.6	37.2	18.3	43.0	38.2	52.2	40.0	33.8	36.0	21.6	33.4	38.3	35.6
EM adapt [128]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.6
Proposed	**79.6**	**58.2**	**25.6**	**65.4**	19.2	**44.3**	**60.2**	39.5	**62.1**	10.6	**45.5**	**22.7**	**65.2**	**50.7**	**56.5**	**53.7**	14.3	**51.8**	**24.9**	**40.8**	34.9	**44.1**

experiment. Separate sparse codes of size 512 are computed for each category in [1, 6] and [123]. [124] uses sparse coding on a common dictionary of 1536 atoms for all object classes. Similarly, a common feature code of $20 \times 512 = 10240$ elements is used in [7]. In our method, we compute 512 dimensional relu5 features which are common for all object categories.

Table 6.3 compares the pixel-level performance of the proposed WS method and patch-level results of FS top-down saliency approaches [1, 7] (these approaches did not report their pixel-level results on this dataset). We outperform [1], [123] and [7] in almost all categories and in mean precision rate at EER across 20 classes. A performance drop of 5 to 10% is reported by [101] while converting patch-level results of [1] to pixel-level, which further increases the performance gain of the proposed approach. Khan and Tappen [63] report pixel-level precision rates at EER only for cow category (8.5%) which is much lower than the proposed weakly supervised approach (52.3%).

6.9.3.4 Category-independent salient object detection

Category-independent saliency maps are obtained using the top-down models trained on PASCAL VOC-2012 training set through simple pixel-level maximum operation as

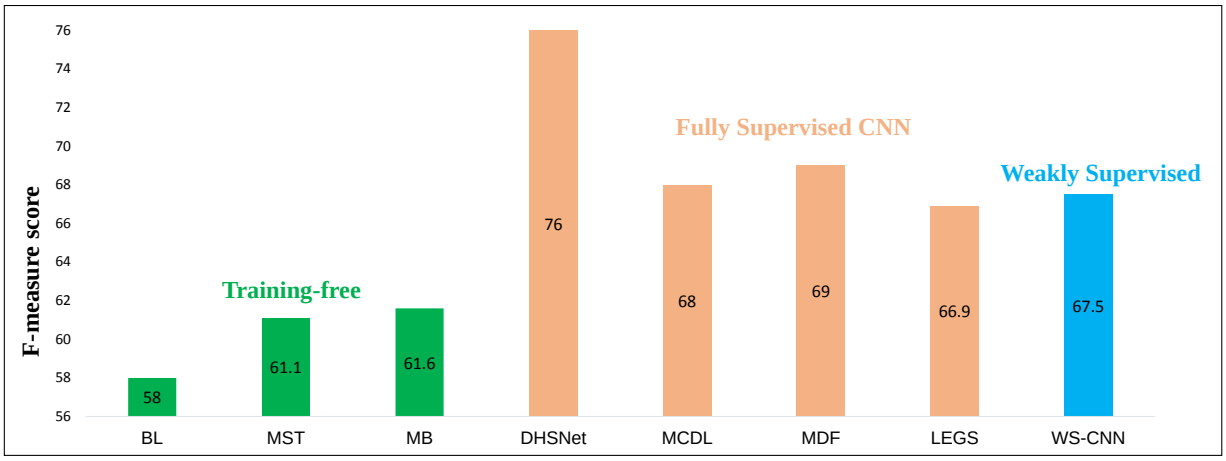


Figure 6.8: Comparison of the proposed weakly supervised approach with state-of-the-art category-independent saliency approaches on PASCAL-S dataset. We achieve a performance comparable with deep learning-based fully supervised approaches.

explained in section 6.7.3. Saliency values less than 0.5 are considered as background, and those between 0.5 and 1 are normalized to [0, 1]. The performance is evaluated on PASCAL-S dataset. Fig. 6.8 compares the proposed method against state-of-the-art category-independent approaches that include deep learning based fully supervised approaches such as MCDL [37], LEGS [55], MDF [56] and DHSNet [36]. The performance metric, F-measure is

$$f_\eta = \frac{(1+\eta^2) \cdot Precision \cdot Recall}{\eta^2 \cdot Precision + Recall}, \tag{6.15}$$

where $\eta^2 = 0.3$ [57]. Following [36], precision and recall are computed by binarizing each saliency map at an image adaptive threshold, which is twice the average value of the saliency map.

The proposed weakly supervised method achieves an f-measure of 67.5, which is comparable with fully supervised LEGS, MCDL and MDF. We use only 5717 images from PASCAL VOC-2012 training set, which is much smaller compared to the training data used by fully supervised approaches shown in Fig. 6.8. For example, DHSNet uses nearly 10,000 fully annotated images from multiple datasets such as MSRA 10K [130] and DUT-OMRON [51]. Data augmentation is used to further increase the number of training images. With less supervision and lesser training data, we achieve a performance comparable with these fully supervised approaches. Qualitative results are shown in Fig. 6.9.

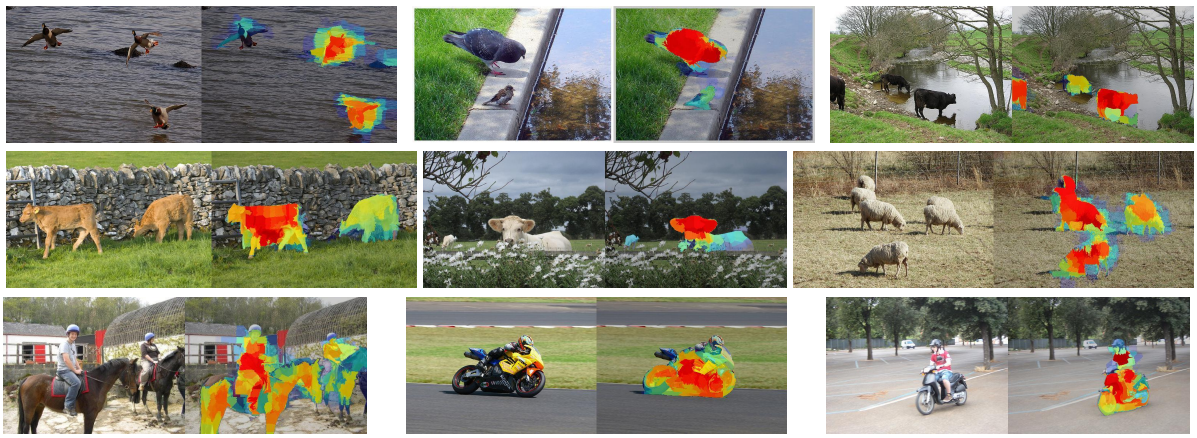


Figure 6.9: Category-independent saliency maps produced by the proposed method on PASCAL-S dataset.

6.9.4 Computation time

Training of the proposed framework is significantly faster than [1, 6] and [13] since we do not use iterative dictionary learning or fine-tuning of convolutional filter weights. MATLAB implementations of all approaches were evaluated on a PC running on Intel Xeon 2.4 GHz processor. Despite a non-parallelized implementation, our approach requires 1 hour to train all 3 object categories of Graz-02. This is significantly faster compared to [1] which takes 4 hours and 49 minutes, and also to Chapter 3, which takes 30 hours and 10 minutes. The fully supervised algorithm proposed in Chapter 4 also needs 3 hours and 34 minutes to train Graz-02 models. Similarly, the proposed saliency models for all 20 categories of PASCAL VOC-2012 are trained within 6 hours and 20 minutes. Inspite of a parallel execution using GPU, [13] needs 8 days to train their model on PASCAL VOC-2012.

On a Graz-02 test image of size 640×480 pixels, our unoptimized, non-parallel MATLAB execution took 8 seconds for inference. The multi-scale superpixel averaging step takes up a major part of this running time, which can be removed at the expense of a 2% reduction in accuracy. With similar settings, [1] needs 5.5 seconds and [6] needs 28 seconds for inference. Since all the saliency models share common relu5 features, it reduces the computational time of the proposed method while inferring multiple saliency models on the same test image. However, [1] and [6] calculate sparse codes for each model

Figure 6.10: Semantic segmentation using our top-down saliency map. Input image, semantic segmentation result produced by our framework and the ground truth for semantic segmentation are shown in adjacent columns.

separately and hence, the inference pipeline needs to be repeated for each category on a test image. This requires multi-fold inference time on images with multiple categories. On PASCAL VOC-2012 test images of size 500×350 pixels, our parallel MATLAB execution without GPU took an average inference time of 1.8 seconds as compared to 4 seconds in [13].

6.9.5 Applications

6.9.5.1 Weakly supervised semantic segmentation

In PASCAL VOC-2012 semantic segmentation task, each pixel in the image needs to be classified to one of 21 categories comprising background and 20 object categories. The proposed approach achieves state-of-the-art performance on both validation set and test set of PASCAL VOC-2012 segmentation challenge as shown in Table 6.4. The segmentation results on the validation set and test set are denoted Val and Test, respectively. Our results are reported from PASCAL VOC-2012 evaluation server which uses intersection over union (IoU)) as the evaluation metric. We outperform [129] in 14 out of 21 classes in the validation set and by nearly 8% in the mean IOU in both validation set and test

Figure 6.11: Object segmentation using our top-down saliency map. Input image and the object segmentation results produced by our framework are shown in adjacent columns.

set. A larger training set of 10582 and 12000 images are used by [129] and [128], respectively. It comprises additional images collected by Hariharan *et al.* [131] along with PASAL VOC 2012 training set. We trained our models using 5717 images from PASCAL VOC-2012 training set.

Fig. 6.10 compares the qualitative results obtained by the proposed method against the ground truth. Majority of the horse pixels are classified accurately in row 2, despite the presence of person. Similarly most of the person pixels are classified correctly, despite the size of person being small compared to motorbike in the top row. We do not use Grabcut [109] or similar energy minimization techniques for semantic segmentation. The class label for each pixel is produced by simple maximum operation on the top-down saliency maps.

6.9.5.2 Weakly supervised object segmentation

Similar to Chapter 5, object segmentation accuracy is evaluated on 100 test images from each category of Object Discovery dataset [10]. 300 images from each category are used to train our saliency model, along with 300 negative images from Graz-02 dataset. Qualitative results are shown in Fig. 6.11. Multiple instances of car are accurately segmented out as shown in the row 1. In row 2, the proposed approach could accurately segment out the horse. Quantitative comparisons with state-of-the-art co-segmentation approaches

Table 6.5: Comparison of proposed weakly supervised approach with object segmentation approaches on Object Discovery dataset, evaluated using Jaccard similarity.

Method	Airplane	Car	Horse	Mean
Joulin *et al.* [11]	15.4	37.2	30.2	27.6
Joulin *et al.* [96]	11.7	35.2	29.5	25.5
Kim *et al.* [97]	7.9	0.04	6.43	4.79
Object Discovey [10]	55.8	64.4	51.6	57.3
Koteshwar *et al.* [134]	56	69	55	60
Zhang *et al.* [133]	53.5	58.8	52.2	54.8
Quan *et al.* [132]	56.3	66.8	58.1	60.4
WS-SC[124]	57.3	67.4	50.51	58.4
Object selection [28]	64.3	71.8	55.1	63.7
Proposed	**65.0**	**77.3**	**61.6**	**68.0**

Table 6.6: Average precision of object localization on PASCAL VOC-2012 detection validation set.

Method	SV	plane	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	Mean
		Exact																				
RCNN [30]	FS	86.5	72.1	74.2	66.7	43.1	78.3	68.8	80.8	44.9	62.3	51.1	74.4	73.6	83.0	83.0	49.2	78.4	40.6	74.1	69.2	67.7
Exemplar [13]	FS	86.8	87.2	72.7	46.8	31.7	91.0	58.6	95.2	44.5	94.8	41.5	87.0	91.4	**94.3**	**89.2**	57.7	**93.5**	59.2	84.7	60.5	73.4
ProNet [23]	WS	89.4	78.1	79.2	73.7	39.9	84.2	61.2	86.4	42.1	67.7	53.2	84	81.8	82.3	84.1	39.2	81.9	48.2	80.8	58.1	69.8
ProNet + classifier [23]	WS	90.3	82	81.2	**75.4**	44.5	84.9	62.5	89.8	47.2	78.9	55.6	88.1	84.3	85.7	85.6	44.4	84.3	50	81.8	65.1	73.1
Proposed	WS	**97.6**	**93.1**	**95.6**	73.4	**60.6**	**91.8**	**77.9**	**97.4**	**52.0**	**80.8**	**84. 1**	**96.6**	**91.4**	91.6	71.9	**62.4**	65.1	**84.4**	**92.5**	**84.1**	**82.2**
		18 Pix.																				
RCNN [30]	FS	92.0	80.8	80.8	73.0	49.9	86.8	77.7	87.6	50.4	72.1	57.6	82.9	79.1	89.8	88.1	56.1	83.5	50.1	81.5	76.6	74.8
Fast RCNN [135]	FS	95.2	88.2	88.4	77.9	49.0	**93.4**	**83.6**	95.1	**59.4**	86.6	71.0	92.6	93.1	93.0	**92.2**	58.2	88.0	63.6	91.9	77.3	81.9
Oquab *et al.* [22]	WS	90.3	77.4	81.4	79.2	41.4	87.8	66.4	91.0	47.3	83.7	55.1	88.8	**93.6**	85.2	87.4	43.5	86.2	50.8	86.8	66.5	74.5
ProNet [23]	WS	91.6	82	85.1	78.6	45.9	87.9	67.1	92.2	51	72.9	60.8	89.3	85.1	85.3	86.4	45.6	83.5	55.1	85.6	65.9	74.8
ProNet + box classifier [23]	WS	92.6	85.6	87.4	**79.6**	48.3	88.7	68.9	94.2	54.6	83.2	62.8	92.0	89.9	88.2	87.1	49.2	86.9	57.2	86.8	70.0	77.7
Bency *et al.* [136]	WS	90.1	86.4	86.4	77.6	56.8	90.3	68.3	89.9	54.7	**86.8**	66.4	88.5	89.0	88.1	78.5	64.1	**90.0**	67.0	89.9	82.6	79.7
Proposed	WS	**98.2**	**94.1**	**96.0**	79.1	**66.2**	91.8	79.5	**98.1**	58.5	85.4	**87.4**	**96.8**	92.0	**93.8**	76.0	**64.5**	69.3	**86.7**	**93.4**	**88.0**	**84.7**

are shown in Table 6.5. The Jaccard similarity, i.e, intersection over union (IOU) with the ground-truth is evaluated as in [10]. In all the three categories, we achieve state-of-the-art performance compared to related co-segmentation [10, 132] and co-saliency [133] approaches. The semantic object selection [28] uses additional supervision by collecting positive training images with white background using an internet search. Inspite of this modification, they could only achieve an average accuracy of 63.73%, which is lower than our mean accuracy of 68.0% across 3 categories.

Figure 6.12: Object localization using our top-down saliency map.

6.9.5.3 Weakly supervised object localization

Presence of multiple objects in an image makes object localization on PASCAL VOC-2012 detection set a challenging task, especially in a weakly supervised setting. The location of the maximum value in the top-down saliency map of an object category is used for its localization as explained in section 6.8. Since an accurate estimate of object boundaries are not required, we replaced the multi-scale superpixel averaging with an averaging filter on a rectangular window of size 64×64 pixels for faster inference. The location that falls exactly within any ground truth bounding box associated for a given category is assumed correct and the average precision is calculated as in [13]. In [22], average precision is evaluated by giving an error tolerance of 18 pixels to the predicted location. We evaluated our model in both these settings denoted Exact and 18 Pix and corresponding results are compared with state-of-the art approaches as shown in Table 6.6. In both the evaluation settings, we achieve a performance which is comparable to fully supervised top-down saliency approaches and dedicated object detectors such as fast RCNN [135]. Fig. 6.12 shows some qualitative results obtained using the proposed method in localizing multiple objects. Partially occluded objects such as motorbike and car are localized accurately despite the presence of other distracting objects.

Table 6.7: Comparison with weakly supervised object detection approaches on PASCAL VOC-2012 validation dataset, measured by average precision.

Method	Oquab *et al.* [22]	ProNet [23]	ProNet+Classifier [23]	Li *et al.* [137]	Proposed
mAP (Mean of 20 Classes)	11.74	13	15.5	29.1	20.4

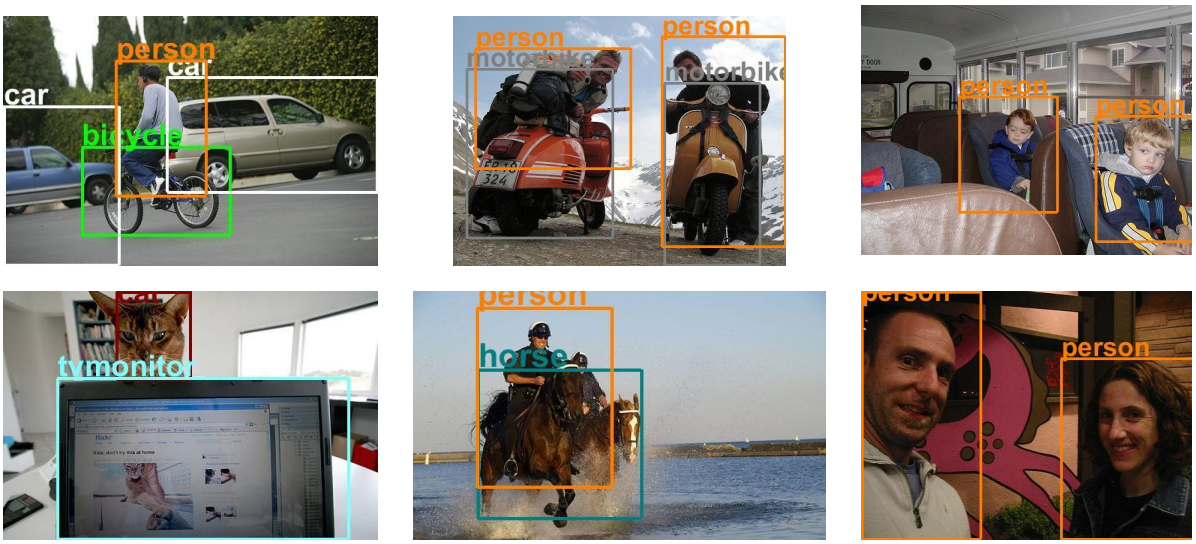
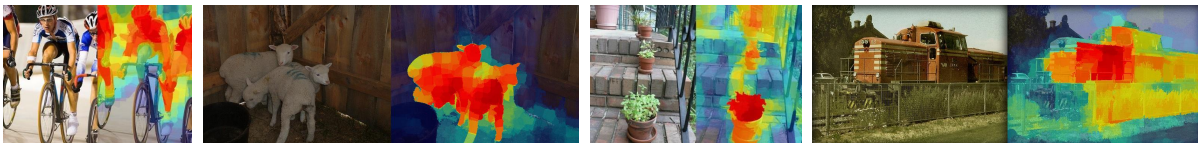


Figure 6.13: Object detection using our top-down saliency map.

6.9.5.4 Weakly supervised object detection

The object detection boxes produced by a simple binarization of our saliency maps is shown to be comparable with dedicated weakly supervised object detectors in Table 6.7. We outperform [23] which uses an additional box classifier to classify their object proposal boxes. We consider all category-specific object boxes as positive detections. PASCAL VOC 2012 evaluation server is used to estimate object detection accuracy, where a detection having an $IOU > 0.5$ with the ground truth rectangular bounding box is considered as true positive. [137] use EdgeBoxes for object proposals and for identifying the class-specific object proposals in an image, they mask the image regions that belongs to each object proposal separately. This approach requires thousands of forward passes through the network to identify the class-specific object proposals, which is time consuming, where as the proposed approach need only one forward pass through the network. Moreover, fine-tuning of convolution layers is required for detection task, where as we use the CNN trained for image classification, without fine-tuning. Inspite of all these computational



Person sheep Potted plant Train

Figure 6.14: Failure cases in the saliency map.

Figure 6.15: Failure cases in object detection, due to the inability of the proposed saliency map to discriminate among multiple instances of an object which are spatially connected.

requirements, their AlexNet based framework achieved a mAP of 22.4% and their VGGNet based framework achieved a mAP of 29.1 on PASCAL VOC 2012 object detection dataset. The proposed strategy achieve a mAP of 20.4.

In Fig. 6.13, multiple overlapping objects are accurately detected by the proposed strategy. Multiple instances of person, motorbike and car are also detected. The horse and bicycle are accurately detected despite the presence of other categories in the image. Similarly, an accurate bounding box around the cat is marked in an image that also contains a TV monitor.

Limitations. Similar to other weakly supervised approaches [23, 84], the proposed approach has limited ability to discriminate among multiple instances of an object which are spatially adjacent as shown in person and sheep images of Fig. 6.14. This causes low performance for object detection on such images as shown in Fig. 6.15.

6.10 Conclusion

In this chapter, a CNN feature-based weakly supervised salient object detection approach is proposed. A novel strategy to select a bottom-up saliency map that suits a top-down task is proposed. Contribution of relu5 features at different spatial locations to a ReluSPM image classifier are estimated to compute a novel R-ReluSPM saliency. The

top-down R-ReluSPM saliency is integrated with the bottom-up saliency map and produces a combined saliency which is further integrated with contextual saliency. The proposed weakly supervised top-down saliency model achieves state-of-the-art performance in top-down salient object detection by outperforming even fully supervised CNN-based approaches. Moreover, the top-down saliency maps of different object categories are combined to produce a category-independent saliency map that can estimate salient objects under free-viewing condition. Finally, through quantitative comparisons, we demonstrated the usefulness of proposed saliency map for four different applications.

Chapter 7

Conclusions and Future Research

7.1 Conclusions

This thesis focused on development of classifier-based algorithms for top-down salient object detection and demonstrating their utility in various computer vision applications. We proposed four different classifier based methods for top-down salient object detection in which an image classifier is an integral part. The first two methods are fully supervised approaches while the last two are weakly supervised. The major contributions of this thesis can be summarized as follows:

(a) A feature coding strategy for salient object detection.

(b) A joint framework for image classification and top-down salient object detection.

(c) A novel strategy for weakly supervised top-down salient object detection, by analyzing the probabilistic contribution of image regions to image classifier confidence.

(d) A novel strategy to automatically select the best bottom-up saliency map for an image in a given task.

(e) CNN-feature based hybrid approach that can be trained in a weakly supervised setting.

Besides illustrating the accuracy of saliency maps produced by the proposed methods, we have demonstrated the effectiveness of top-down salient object detection in various applications like image classification, weakly supervised semantic segmentation, semantic

object selection, object localization, object detection and action-specific patch discovery.

The major conclusions from each contribution are discussed next:

(1) **Locality constrained contextual sparse coding:** LCCSC has been developed to ensure that the feature codes for top-down salient object detection satisfy the three locality constraints, i.e. spatial locality, feature-domain locality and category-domain locality constraints. The first constraint ensures that features from spatially nearby regions have similar code and the second constraint assigns similar features with similar code as in [26]. Finally, the category-domain locality constraint ensures that similar atoms from each partition of the dictionary is used for feature coding, where each partition corresponds to an object category. For each object category, the dictionary contains a partition of 512 atoms. This results in large feature code with more than 10,000 elements in PASCAL VOC 2007 dataset with 20 object categories. To avoid this limitation, in Chapter 5, we fixed the feature code size to only 1536 elements across datasets by slightly compromising on accuracy.

(2) **Joint framework for image classification and top-down salient object detection:**
In Chapter 4 we showed that image classification and top-down salient object detection can be integrated so that they are mutually beneficial to each other. The image classifier improves top-down salient object detection by incorporating a high-level idea about the presence of object of interest in an image. Similarly, top-down salient object detection improves image classification by suppressing features from image regions with low saliency. The classifier module is built on feature codes computed using all object dictionaries. Hence, this approach is suitable for applications such as semantic segmentation where the saliency maps of every category needs to be evaluated before classifying each image pixel in to any one of the predefined categories. On the other hand, it is less desirable to use this approach for applications such as object detection, where the objective is to detect only one user defined object category because computing feature codes with dictionaries of every object category is less efficient.

(3) **Weakly supervised top-down saliency by backtracking ScSPM image classifier:**

Image classifiers can be trained with image-level labels only. On the other hand, conventional top-down saliency models require pixel-accurate annotation for their training. Based on our findings in Chapter 4 that the image classifiers are helpful in top-down saliency frameworks, we used image classifiers to train top-down saliency models in a weakly supervised settings, which is less addressed in the literature. The probabilistic contribution of image regions to an ScSPM image classifier confidence is used to identify salient regions in an image through a novel R-ScSPM strategy. The fully supervised training of contextual saliency used in Chapter 3 is replaced with patch-labels obtained through R-ScSPM saliency, enabling the complete model to be trained with a binary image-level label. This approach does not use any bottom-up saliency information, which limits this model to be used only for category-specific applications. Moreover, the handcrafted SIFT features used in this approach are less efficient as compared with CNN features.

(4) **CNN-based hybrid approach for salient object detection:**

Replacing ScSPM image classifier in Chapter 5 with an improved image classifier based on CNN-features helped to largely improve the accuracy of weakly supervised salient object detector in Chapter 6. This shows that the quality of saliency map can be further improved by fine-tuning the CNN-based image classifier. The R-ReluSPM is combined with the bottom-up saliency map selected by our selection strategy to produce a hybrid saliency framework that can be configured for both category-independent applications as well as for category-specific top-down salient object detection.

(5) **Bottom-up saliency map selection:**

In Chapter 6, we proposed a weakly supervised strategy to select a bottom-up saliency map which is best suited for a given application. Even though state-of-the-art bottom-up saliency approaches produces good quality saliency maps in most of the images, they still lag behind top-down approaches in many applications such

as segmentation, due to lack of prior knowledge about the task. The proposed strategy incorporates this prior knowledge through a selection criterion that uses an image classifier. The performance of this approach largely depends on the choice of bottom-up saliency algorithms. If appropriate bottom-up algorithms are selected to be complementary with top-down saliency framework, we can expect significant improvement in saliency map accuracy. First we evaluated the performance improvement of this selection strategy alone in our experiments in section 6.9.1. Here, the top-down information is used only to select a bottom-up saliency map, but not to compute the saliency of image regions. Hence, only a small performance improvement was achieved by incorporating top-down information. Large improvement is observed in section 6.9.2.1 when we integrated this selection strategy with our R-ReluSPM saliency.

The LCCSC based approach in Chapter 3 is effective for applications having limited number of training images but with pixel-accurate annotation. On the other-hand, if the goal is to develop an integrated system that can perform both image classification and top-down salient object detection in a fully supervised settings, the approach in Chapter 4 can be used. The algorithms in Chapter 5 suits those applications where limited number of training images with only image-level annotation are available.

Among our four salient object detection frameworks, CNN-based, weakly supervised approach described in Chapter 6 achieves stat-of-the-art performance. It outperforms even CNN-based fully supervised top-down salient object detection frameworks [13] across multiple datasets. Our experiment shows that this approach is effective for various applications such as semantic segmentation, object detection, semantic object selection, object localization etc. It is suitable for applications where the objective is to obtain a pixel-accurate saliency maps using image-level label only.

7.2 Future research

In this section, we discuss some of the possible research directions for top-down salient object detection and its applications.

(1) In this thesis, we have considered the effectiveness of classifiers for top-down salient object detection. There are several other machine learning techniques such as reinforcement learning, regression, and various other classifier techniques such as decision trees, random forest whose usefulness in top-down salient object detection is yet to be explored.

(2) In the CNN framework in Chapter 6, we used the filter weights pre-trained for ImageNet image classification task to reduce training time. The quality of CNN features and hence the saliency maps can be largely improved by fine-tuning the convolution weights for a given application. An approach similar to our method in Chapter 4 can be developed to improve convolution weights and hence the CNN image classifier by using top-down saliency maps in the fine-tuning step.

(3) Our weakly supervised approaches can be further improved by semi-supervised training where pixel-accurate object annotations are available for few training images. These annotated images can be used to evaluate and improve our saliency models in a cross validation setting, by modifying the model parameters responsible for false positive detection or false negative detection.

(4) Top-down salient object detection in videos is less explored in the literature. Extending our models by incorporating motion cues for spatio-temporal top-down saliency detection in videos will be useful for various applications such as video tracking, semantic labeling of video contents and object detection for surveillance. Searching for a particular object or person in multiple terabytes of surveillance videos is a highly challenging task. Faster top-down saliency approaches can largely speed-up this content based video search tasks.

(5) Visual Question Answering (VQA) : To answer a set of questions related to the attributes of image contents, it is essential to identify image regions which are relevant for a particular question [20]. Our weakly supervised top-down saliency approaches can be modified to replace dedicated object detectors in such applications.

(6) For real-time streaming of surveillance videos it is essential to compress the video. Intruders appearing in a surveillance video may be of very small resolution and quite

often with low contrast with background. So applying general video compression algorithms may remove these important information from the video. Identifying task relevant regions in a video using top-down saliency can avoid this problem by applying less compression in such image regions.

List of Publications

1. H. Cholakkal, J. Johnson and D. Rajan, "Backtracking ScSPM image classifier for weakly supervised top-down saliency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

2. H. Cholakkal, J. Johnson and D. Rajan, "A Classifier-guided Approach for Top-down Salient Object Detection," in *Signal Processing: Image Communication*, Elsevier, Volume 45, July 2016.

3. H. Cholakkal, D. Rajan, and J. Johnson, "Top-down saliency with locality-constrained contextual sparse coding," in *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, 2015.

4. H. Cholakkal, J. Johnson and D. Rajan, Weakly Supervised Top-down Salient Object Detection," (under review)

Bibliography

[1] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 2296–2303.

[2] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2009, pp. 1794–1801.

[3] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch, "Minimum barrier salient object detection at 80 fps," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, June 2015, pp. 1404–1412.

[4] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 2334–2342.

[5] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 37, no. 3, pp. 569–582, March 2015.

[6] A. Kocak, K. Cizmeciler, A. Erdem, and E. Erdem, "Top down saliency estimation via superpixel-based discriminative dictionaries," in *Proc. British Mach. Vis. Conf. (BMVC)*, September 2014.

[7] H. Cholakkal, D. Rajan, and J. Johnson, "Top-down saliency with locality-constrained contextual sparse coding," in *Proc. British Mach. Vis. Conf. (BMVC)*, September 2015, pp. 159.1–159.12.

[8] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 31, no. 6, pp. 989–1005, 2009.

[9] F. Moosmann, E. Nowak, and F. Jurie, "Randomized clustering forests for image classification," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 30, no. 9, pp. 1632–1646, September 2008.

[10] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 1939–1946.

[11] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2010, pp. 1943–1950.

[12] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. PP, no. 99, pp. 1–1, 2016.

[13] S. He, R. W. Lau, and Q. Yang, "Exemplar-driven top-down saliency detection via deep association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 5723–5732.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 779–788.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, December 2015, pp. 91–99.

[17] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 33, no. 2, pp. 353–367, Feb 2011.

[18] J. Li and W. Gao, "Visual saliency computation-a machine learning perspective," ser. Lect. Notes Comput. Sci. Springer, 2014, vol. 8408.

[19] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," *arXiv preprint arXiv:1411.5878*, 2014.

[20] Z. Bolei, K. Aditya, L. Agata, O. Aude, and T. Antonio, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 2921–2929.

[21] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 3506–3513.

[22] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 685–694.

[23] C. Sun, M. Paluri, R. Collobert, R. Nevatia, and L. Bourdev, "Pronet: Learning to propose object-specific boxes for cascaded neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 3485–3493.

[24] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[25] J. Zhu, Y. Qiu, R. Zhang, J. Huang, and W. Zhang, "Top-down saliency detection via contextual pooling," *J. Signal Process. Syst.*, vol. 74, no. 1, pp. 33–46, 2014.

[26] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2010, pp. 3360–3367.

[27] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell, "Deep compositional captioning: Describing novel object categories without paired training data," *arXiv preprint arXiv:1511.05284*, 2015.

[28] E. Ahmed, S. Cohen, and B. Price, "Semantic object selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 3150–3157.

[29] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2011, pp. 1577–1584.

[30] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[31] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vision (IJCV)*, vol. 104, no. 2, pp. 154–171, 2013.

[32] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2010, pp. 73–80.

[33] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, June.

[34] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 438–445.

[35] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 35, no. 1, pp. 185–207, Jan 2013.

[36] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 678–686.

[37] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 1265–1274.

[38] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of intelligence*. Springer, 1987, vol. 188, pp. 115–141.

[39] J. J. Clark and N. J. Ferrier, "Modal control of an attentive vision system." in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, December 1988, pp. 514–523.

[40] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," in *MIT Technical Report*, 2012.

[41] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[42] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2007, pp. 545–552.

[43] A. Garcia-Diaz, X. Fdez-Vidal, X. Pardo, and R. Dosil, "Decorrelation and distinctiveness provide with human-like saliency," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lect. Notes Comput. Sci., 2009, vol. 5807, pp. 343–354.

[44] A. Torralba, "Modeling global scene factors in attention," *J. Opt. Soc. Am. A*, vol. 20, no. 7, pp. 1407–1418, 2003.

[45] P. Jiang, N. Vasconcelos, and J. Peng, "Generic promotion of diffusion-based salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec 2015, pp. 217–225.

[46] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, June 2007, pp. 1–8.

[47] H.-D. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.

[48] L. Itti, C. Koch, E. Niebur *et al.*, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 20, no. 11, pp. 1254–1259, November 1998.

[49] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 1884–1892.

[50] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing markov chain," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec 2013, pp. 1665–1672.

[51] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 3166–3173.

[52] C. Aytekin, S. Kiranyaz, and M. Gabbouj, "Automatic object segmentation by quantum cuts." in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Aug 2014, pp. 112–117.

[53] Ç. Aytekin, E. C. Ozan, S. Kiranyaz, and M. Gabbouj, "Visual saliency by extended quantum cuts," in *Proc. Int. Conf. Image Proc. (ICIP)*, Sept 2015, pp. 1692–1696.

[54] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 2083–2090.

[55] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 3183–3192.

[56] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 5455–5463.

[57] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 280–287.

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision (IJCV)*, vol. 120, pp. 211–252, 2015.

[59] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search." *Psychol. Rev.*, vol. 113, no. 4, pp. 766–786, October 2006.

[60] A. Oliva, A. Torralba, M. S. Castelhano, and J. M. Henderson, "Top-down control of visual attention in object detection," in *International Conference on Image Processing*, vol. 1. IEEE, September 2003, pp. I–253.

[61] C. Kanan, M. H. Tong, L. Zhang, and G. W. Cottrell, "Sun: Top-down saliency using natural statistics," *Vis. cogn.*, vol. 17, no. 6-7, pp. 979–1003, 2009.

[62] M. Marcin and S. Cordelia, "Accurate object recognition with shape masks," *Int. J. Comput. Vision (IJCV)*, vol. 97, no. 2, pp. 191–209, 2012.

[63] N. Khan and M. F. Tappen, "Discriminative dictionary learning with spatial priors." in *Proc. Int. Conf. Image Proc. (ICIP)*, 2013.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision (IJCV)*, vol. 60, pp. 91–110, 2004.

[65] F. Moosmann, D. Larlus, and F. Jurie, "Learning saliency maps for object categorization," in *ECCV Workshop on the Representation and Use of Prior Knowledge in Vision*, 2006.

[66] F. Shahbaz Khan, J. van de Weijer, and M. Vanrell, "Top-down color attention for object recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sept 2009, pp. 979–986.

[67] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision (IJCV)*, vol. 42, no. 3, pp. 145–175, 2001.

[68] F. Khan, J. Weijer, and M. Vanrell, "Modulating shape features by color attention for object recognition," *Int. J. Comput. Vision (IJCV)*, vol. 98, no. 1, pp. 49–64, 2012.

[69] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, June 2006, pp. 2049–2056.

[70] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision research*, vol. 40, no. 10, pp. 1489–1506, 2000.

[71] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," in *Joint Pattern Recognition Symposium*. Springer, 2005, pp. 117–124.

[72] A. Garcia-Diaz, V. Leboran, X. R. Fdez-Vidal, and X. M. Pardo, "On the relationship between optical variability, visual saliency, and eye fixations: A computational approach," *Journal of vision*, vol. 12, no. 6, pp. 17–17, 2012.

[73] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2006, pp. 2169–2178.

[74] T. Ge, K. He, and J. Sun, "Product sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 939–946.

[75] H. Bilen, V. Namboodiri, and L. Gool, "Object and action classification with latent window parameters," *Int. J. Comput. Vision (IJCV)*, vol. 106, no. 3, pp. 237–251, 2014.

[76] A. Bergamo, L. Bazzani, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," *arXiv preprint arXiv:1409.3964*, 2014.

[77] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 39, no. 1, pp. 189–203, Januvary 2016.

[78] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 1717–1724.

[79] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 818–833.

[80] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 5188–5196.

[81] A. Dosovitskiy and T. Brox, "Inverting convolutional networks with convolutional networks," *CoRR abs/1506.02753*, 2015.

[82] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[83] A. Z. K. Simonyan, A. Vedaldi, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *ICLR Workshop*, 2014.

[84] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 2846–2854.

[85] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, september 2014, pp. 391–405.

[86] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec 2015, pp. 1431–1439.

[87] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer Berlin Heidelberg, October 2012, pp. 73–86.

[88] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 494–502.

[89] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, October 2008, pp. 179–192.

[90] F. Brian, V. Andrea, and S. Stefano, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, September 2009, pp. 670–677.

[91] D. Aldavert, A. Ramisa, R. L. de Mantaras, and R. Toledo, "Fast and robust object segmentation with the integral linear classifier," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2010, pp. 1046–1053.

[92] D. Singaraju and R. Vidal, "Using global bag of features models in random fields for joint categorization and segmentation of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 2313–2319.

[93] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[94] A. Jain, L. Zappella, P. McClure, and R. Vidal, "Visual dictionary learning for joint object categorization and segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, October 2012, pp. 718–731.

[95] K. R. Jerripothula, J. Cai, F. Meng, and J. Yuan, "Automatic image co-segmentation using geometric mean saliency," in *Proc. Int. Conf. Image Proc. (ICIP)*, October 2014, pp. 3277–3281.

[96] J. Armand, B. Francis, and P. Jean, "Multi-class cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2012, pp. 542–549.

[97] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade, "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, November 2011, pp. 169–176.

[98] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 801–808.

[99] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2011, pp. 1697–1704.

[100] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 28, no. 3, pp. 416–431, March 2006.

[101] Khan, Nazar, Tappen, and M. F, "Discriminative dictionary learning with spatial priors." in *Proc. Int. Conf. Image Proc. (ICIP)*, September 2013, pp. 166–170.

[102] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, December 2013, pp. 281–288.

[103] S. Gao, I.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Trans. Image Process. (TIP)*, vol. 23, no. 2, pp. 623–634, Feb 2014.

[104] M. Marszałek and C. Schmid, "Accurate object recognition with shape masks," *Int. J. Comput. Vision (IJCV)*, vol. 97, no. 2, pp. 191–209, 2012.

[105] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. British Mach. Vis. Conf. (BMVC)*, september 2011, pp. 76.1–76.12.

[106] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[107] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2005, pp. 994–1000 vol. 2.

[108] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2002.

[109] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[110] Y.-H. Tsai, J. Yang, and M.-H. Yang, "Decomposed learning for joint object segmentation and categorization," in *Proc. British Mach. Vis. Conf. (BMVC)*, september 2013.

[111] "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31 – 71, 1997.

[112] R. G. Cinbis, J. Verbeek, C. Schmid *et al.*, "Multi-fold mil training for weakly supervised object localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2014, pp. 2409–2416.

[113] P. O. Pinheiro and R. Collobert, "Weakly supervised semantic segmentation with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, no. 5, June 2015, p. 6.

[114] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[115] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei, "Object-centric spatial pooling for image classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, October 2012, pp. 1–15.

[116] M. H. Nguyen, L. Torresani, F. de la Torre, and C. Rother, "Weakly supervised discriminative localization and classification: a joint learning process," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, September 2009, pp. 1925–1932.

[117] P. Siva and T. Xiang, "Weakly supervised object detector learning with model drift detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, November 2011, pp. 343–350.

[118] P. Siva, C. Russell, T. Xiang, and L. Agapito, "Looking beyond the image: Unsupervised learning for object saliency and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2013, pp. 3238–3245.

[119] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. British Mach. Vis. Conf. (BMVC)*, september 2014.

[120] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 37, no. 9, pp. 1904–1916, 2015.

[121] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[122] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *J. Mach. Learn. Res.*, vol. 14, no. Feb, pp. 567–599, 2013.

[123] H. Cholakkal, J. Johnsan, and D. Rajan, "A classifier-guided approach for top-down salient object detection," *Signal Process. Image Commun.*, vol. 45, pp. 24–40, 2016.

[124] H. Cholakkal, J. Johnson, and D. Rajan, "Backtracking scspm image classifier for weakly supervised top down saliency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 5278–5287.

[125] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 34, no. 11, pp. 2274–2281, 2012.

[126] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[127] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," in *Proc. ICLR*, 2015.

[128] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a dcnn for semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, December 2015, pp. 1742–1750.

[129] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, December 2015, pp. 1796–1804.

[130] B. Ali, C. Ming-Ming, J. Huaizu, and L. Jia, "Salient object detection: A benchmark," *IEEE Trans. Image Process. (TIP)*, vol. 24, no. 12, pp. 5706–5722, 2015.

[131] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, November 2011, pp. 991–998.

[132] R. Quan, J. Han, D. Zhang, and F. Nie, "Object co-segmentation via graph optimized-flexible manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 687–695.

[133] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int. J. Comput. Vision (IJCV)*, vol. 120, no. 2, pp. 1–18, November 2016.

[134] K. R. Jerripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Trans. Multimedia (TMM)*, vol. 18, no. 9, pp. 1896–1909, 2016.

[135] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[136] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. S. Manjunath, *Weakly Supervised Localization Using Deep Feature Maps*, October 2016, pp. 714–731.

[137] D. Li, J. B. Huang, Y. Li, S. Wang, and M. H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 3512–3520.