UNIVERSITY OF CALIFORNIA, MERCED

Data-Driven Object Segmentation in Single Images with Random Field Models

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering and Computer Science

by

Jimei Yang

Committee in charge:

Professor Ming-Hsuan Yang, Chair
Professor Honglak Lee
Professor Roummel Marcia

2015

The dissertation of Jimei Yang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Professor Roummel Marcia

Professor Honglak Lee

Professor Ming-Hsuan Yang Chair

University of California, Merced

2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

VITA

2006	B.S. in Electrical Engineering and Information Science, China Agricultural University, Beijing
2009	M.S. in Automation, University of Science and Technology of China, Hefei
2015	Ph.D. in Electrical Engineering and Computer Science, University of California, Merced

PUBLICATIONS

Jimei Yang, Brian Price, Scott Cohen, Zhe Lin and Ming-Hsuan Yang, *PatchCut: Data-Driven Object Segmentation via Local Shape Transfer*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

Jimei Yang, Simon Safar and Ming-Hsuan Yang, *Max Margin Boltzmann Machines for Object Segmentation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

Jimei Yang, Brian Price, Scott Cohen and Ming-Hsuan Yang, *Context Driven Scene Parsing with Attention to Rare Classes*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

Jimei Yang, Yi-Hsuan Tsai and Ming-Hsuan Yang, *Exemplar Cut*, In International Conference on Computer Vision (ICCV), 2013

ABSTRACT OF THE DISSERTATION

Data-Driven Object Segmentation in Single Images with Random Field Models

by

Jimei Yang
Doctor of Philosophy in Electrical Engineering and Computer Science
University of California Merced, 2015
Professor Ming-Hsuan Yang, Chair

As humans, we have a remarkable ability of telling objects apart from cluttered background and tracing their contours even with occlusions. This ability has long fascinated computer vision researchers to study the principles and algorithms for object segmentation. Object segmentation has both theoretical and practical interests as it is an essential step towards 3D image understanding and intelligent image editing.

To segment an object, we have to recognize it in order to obtain knowledge of what parts should be grouped together. In this thesis, we formulate object segmentation as an image labeling problem in random field models to facilitate integrating top-down recognition knowledge with bottom-up image cues. The integration can be driven by either bottom-up segmentation or top-down recognition. The segmentation-driven process requires object-level segmentation hypotheses drawn from bottom-up cues while the recognition-driven process needs shape and context to be effectively represented. This thesis addresses these issues in a data-driven approach. First, we propose to generate object segmentation proposals from segmentation trees using exemplars. Compared to previous parametric methods, our data-driven method takes advantage of both diversity and informativeness of exemplars and thus produce a compact set of highly plausible proposals. Second, we propose novel random field models that enjoy joint learning of shape representation and

object segmentation. Different from previous works that use shape representation as prior, our model emphasizes the structured prediction from the recognition model to the shape model. This difference ensures the the shape is well preserved in the resulting segmentation masks with robustness to partial occlusions. Third, we develop a novel nonparametric method based on multiscale shape transfer, which in turns forms a higher-order random field. Compared to previous works that transfer rigid or deformable masks in image subwindows, our method explores shape masks in multiple granularities and is able to produce high quality segmentations in an efficient way. The last but not least, we develop a novel scene parsing system where small objects are segmented in context. With extensive use of context in multiscale and particular care to the long-tailed label distribution, our system demonstrates state-of-the-art results in large-scale problems.

Chapter 1

Introduction

1.1 Problem Definition

Object segmentation is a computational process of separating objects of interest from background with clear boundaries/contours in images. It has long been an active research area in computer vision due to its theoretical and practical interests. With the extracted contours, object segmentation provide visual cues for pose estimation, 3D reconstruction and occlusion reasoning, and thus considered as a critical intermediate step that lifts 2D scene understanding to 3D physical world recovery. In the areas of photography and artistic design, there is a demand for modifying the color and shape of one particular object, deleting distractive objects in a photograph and composing new images from existing photographs. Selecting objects with accurate contours is the key function to enable effective, fast and intelligent editing [2]. This thesis is motivated by these potential impacts object segmentation could make to science, art and society.

Bottom-up image segmentation vs. object segmentation. Bottom-up image segmentation aims at grouping pixels into local regions that have consistent colors and textures. There are roughly three kinds of bottom-up image segmentation based on the trade-off between region locality and consistency. First, regions are small and spatially regular, usually

referred as superpixels, such as SLIC [1]. Second, regions are more adaptive to image colors and textures, such as graph-based segmentation [35] and normalized cuts [93]. Third, regions are organized into a tree structure with growing sizes from leaves to root, such the gPb-owt-ucm algorithm [7]. Bottom-up algorithms are closely related to image edge detection [26]. In contrast, object segmentation also requires regions to be consistent with object appearances and shapes of pre-defined classes. For example, human segmentation needs to group pixels of different colors due to dressing. Thus, object segmentation is a process of integrating bottom-up image cues (boundaries) and top-down object knowledge (appearance and shape). Since boundary cues can be preserved in bottom-up regions (superpxiels) or detected edges, object segmentation can operate in pixel-level or region-level.

Single-image vs. multi-image vs. video. Single-image object segmentation is a classic computer vision problem and deeply related to object recognition. Recently, object segmentation is extended to multi-image scenarios, referred as object co-segmentation [107, 88]. Those multiple images can be different views of one object instance [9] or random samples of particular object category [88]. In most of the co-segmentation algorithms the object category is unknown and thus it is of primary interest to discover object knowledge based on appearance or shape consistencies across different images. Compared to image based methods, video object segmentation involves motion as another important cue. Thus, video object segmentation is often coupled with optical flow estimation or tracking [19, 69]. This thesis focuses on object segmentation in a single image.

1.2 Solution

Early efforts are devoted to active contour models [48, 13]. The basic idea is to evolve a curve around one target object to fit its boundary. The evolution is driven by minimizing an energy function that controls the smoothness of the contour and at the same time attracts the contour to image edges. Although the success in medical imaging [115], ac-

tive contours face challenges when applied to complex natural images due it their limited capacity of representing irregular shapes and their vulnerability to cluttered background.

1.2.1 Random Field Models

In this thesis, we pose object segmentation as an image labeling problem in random field models [52] that assigns every pixel or region a binary label of being object or non-object. Compared to active contours, labeling masks are more flexible shape representations and compatible with classification models for object recognition. Random fields are a general class of undirected probabilistic models that represent joint distributions of input image, labels and possible hidden variables. Image labeling is thus solved by maximum a posteriori (MAP) inference. Markov Random Fields (MRFs) [5] and Conditional Random Fields (CRFs) [62] are commonly used for object segmentation [58, 59, 36, 11, 63]. Both MRFs and CRFs can be represented as energy functions with potentials of different orders. Their difference lies in that all the factors in CRFs are data-dependent while some of factors in MRFs are data-independent priors. The advantages of MRFs and CRFs are twofold.

1. Different object recognition models, bottom-up image cues and labeling priors can be easily integrated into a single energy function.

2. There exist efficient combinatorial optimization algorithms [16, 54] for minimizing labeling energy functions given submodular assumption.

Random field models are not limited to MRFs and CRFs. Restricted Boltzmann Machines (RBMs) [90] and Conditional Boltzmann Machines (CBMs) [79] are another group of random field models with extensive hidden variables that facilitate distributed representation learning. We will discuss how to apply RBMs and CBMs to object segmentation in Chapter 4.

1.2.2 Data-Driven Approach

Many works in object segmentation focus on the design of generic higher-order potentials to improve the robustness of labeling or to endows the labeling with certain properties [81, 51, 86, 91]. At the same time, some works apply hand-crafted models [58, 18, 108] to segment objects of particular category, e.g. humans. Both of these two approaches are confronted with performance limitations when generalized to arbitrary object categories. On one hand, it is very hard for one designed higher-order potential to deal with all the object variations and background clutters. On the other hand, it is almost not realistic to hand-craft a model for every object class.

An alternative is leveraging big data to overcome above-mentioned limitations. Nowadays image capturing becomes much easier with various cameras. As a result, there are a tremendous amount of images available in the internet and the size is still growing. At the same time, one can easily collect segmentation annotations at a reasonable price with Mechanic Turk [4]. Large-scale data collections usually cover a wide range of image variations, and thus offer a great opportunity to directly learn object segmentation models [97, 66] or to explore nonparametric methods for label transfer [73, 57]. This thesis considers both learning-based and nonparametric methods for data-driven object segmentation.

1.2.3 Challenges

Since recognition plays an important role in object segmentation models, the performance of object segmentation methods is indeed bounded by the recognition models. In early stage of object segmentation, most of the algorithms are mainly concerned with simple images and their performance also depends on effective interaction with human users. When SIFT [74] and HoG [25] features are successfully applied to object recognition (categorization and detection), automatic object segmentation methods demonstrate the promise of applying to complex natural images, e.g. PASCAL VOC datasets [31]. Recent

fast progress of deep learning significantly improve the performance of object recognition due to the rich features learned from massive data [56]. Object segmentation thus faces a new round of revolution. A thorough discussion about object recognition methods is beyond the scope of this thesis.

Regardless of what recognition models are used, object segmentation faces several fundamental challenges. An immediate one is how to integrate global recognition models with image boundary cues. Object recognition is usually based on holistic representations [64], although sometimes supported by parts [34], which leads to incompatibilities with pixel or region representations in random field models. One solution is to design higher-order or global potentials in random fields that host the output from object categorization or detection [61, 95]. Although solvable by efficient combinatorial optimization, higher-order random fields are difficult to learn and thus often use pre-trained recognition models. Another solution is to decompose object segmentation into two stages: 1) generating object-level segmentation proposals from random fields [20, 28, 49] and 2) evaluating proposals with recognition models [68, 103]. As one can easily access to various recognition models, the focus of this solution is to generate object proposals. The key to its success is to maintain a high recall rate of segmentation proposals, because the true positive can not be recovered in the recognition stage once missed at the first place.

In addition to appearance and boundary, shape serves as another important cue for object segmentation by providing regularizations for pixel/region grouping, especially when handling object instance segmentation with occlusions. However, how to build effective shape representations for segmentation is still an open question. Most of works use templates as shape representations that are either learned from training data [15, 67, 118] or simply a set of exemplars [37, 57, 114]. Boltzmann machines are recently used to learn distributed shape representations and have demonstrated their generalization performance and robustness to occlusions [29]. However, new challenges arise for learning large-size, fine-grained shape models and for applying them to object segmentation.

The discussions so far are based on the object-centric assumption that objects are

brought into focus of the camera. If we take a zoom-out view of scene, the background become dominant and the small size of objects make recognition very challenging. On the other hand, objects are usually occluded with each other in both indoor and outdoor scenes. Therefore, challenges for object segmentation in scenes lie in modeling contexts, i.e. object-object [60, 33] and object-scene [102, 94, 96] relationships.

1.3 Thesis Outline and Contributions

In Chapter 3, we present a hybrid parametric and nonparametric algorithm, exemplar cut, for generating class-specific object segmentation hypotheses. For the parametric part, we train a pylon model on a hierarchical region tree as the energy function for segmentation. For the nonparametric part, we match the input image with each exemplar by using regions to obtain a score which augments the energy function from the pylon model. Our method thus generates a set of highly plausible segmentation hypotheses by solving a series of exemplar augmented graph cuts. Experimental results on the Graz and PASCAL datasets show that the proposed algorithm achieves favorable segmentation performance against the state-of-the-art methods in terms of visual quality and accuracy.

In Chapter 4, we present Max-Margin Boltzmann Machines (MMBMs) for object segmentation. MMBMs are essentially a class of Conditional Boltzmann Machines that model the joint distribution of hidden variables and output labels conditioned on input observations. In addition to image-to-label connections, we build direct image-to-hidden connections to facilitate global shape prediction, and thus derive a simple Iterated Conditional Modes algorithm for efficient maximum a posteriori inference. We formulate a max-margin objective function for discriminative training, and analyze the effects of different margin functions on learning. We evaluate MMBMs using three datasets against state-of-the-art methods to demonstrate the strength of the proposed algorithms.

In Chapter 5, we propose a nonparametric method for generic object segmentation. As similar objects tend to share similar local shapes, we match query image patches with

example images in multiscale to enable local shape transfer. The transferred local shape masks constitute a patch-level segmentation solution space and we thus develop a novel cascade algorithm, PatchCut, for coarse-to-fine object segmentation. In each stage of the cascade, local shape mask candidates are selected to refine the estimated segmentation of the previous stage iteratively with color models. Experimental results on various datasets (Weizmann Horse, Fashionista, Object Discovery and PASCAL) demonstrate the effectiveness and robustness of our algorithm.

In Chapter 6, we presents a scalable scene parsing algorithm based on image retrieval and superpixel matching. We focus on rare object classes, which play an important role in achieving richer semantic understanding of visual scenes, compared to common background classes. Towards this end, we make two novel contributions: rare class expansion and semantic context description. First, considering the long-tailed nature of the label distribution, we expand the retrieval set by rare class exemplars and thus achieve more balanced superpixel classification results. Second, we incorporate both global and local semantic context information through a feedback based mechanism to refine image retrieval and superpixel matching. Results on the SIFTflow and LMSun datasets show the superior performance of our algorithm, especially on the rare classes, without sacrificing overall labeling accuracy.

We conclude this thesis in Chapter 7 and discuss future work.

Chapter 2

Literature Review

In this chapter, we review the literature related to the research work presented in the following chapters.

2.1 Generating Multiple Proposals

Parametric Min Cut. The parametric min cut algorithm [53] introduces a constant value to the node potentials of the graph cut energy function, which changes the decision threshold of classifying the nodes into foreground and background. By varying the constant value, a series of graph cuts are solved to produce a set of segmentation hypotheses. This technique has been used in [20, 49] for category independent object segmentation hypotheses. As the classification thresholds are changed uniformly for all the nodes, the parametric min cut usually produces noisy segmentation results where good segments are companied by false negatives.

M-Best Solutions. When the single MAP solution becomes less satisfactory, it is beneficial to find M best solutions. A potential issue is that the top M most probable solutions may be similar to each other if many noisy local minimal solutions exist close to the MAP one [119]. To address that, Batra et al. [10] propose to explore different local modes of

the energy function by enforcing the solution diversity. In their work, the energy function is augmented with dissimilarity constraints that isolate the current solution from previous ones by a pre-defined threshold. This strategy entails a greedy algorithm to find solutions sequentially.

Multiple Choice Learning. This approach aims to generate multiple structured outputs [39] by learning a set of sub-models simultaneously, rather than inferring multiple solutions from a single model. Recall that we need a diverse set of segmentation hypotheses. It is thus essential to enforce sub-models as different as possible. In fact, the multiple choice learning approach realizes this objective in the training phase by discriminative clustering. It assigns the training exemplars to sub-models by evaluating their segmentation errors so that a sub-model is eventually optimized towards a subset of exemplars. This approach is constrained by the clustering structure of training exemplars and sensitive to the initialization. When the number of sub-models is not properly chosen, the segmentation capabilities of learned sub-models may be imbalanced (some too strong and others too weak) so that the weak predictor degenerates in the training phase.

2.2 Learning Shape Representations

Combining RBMs with CRFs. Recent work [47, 71] on object segmentation realizes the power of Boltzmann Machines to represent high-order interactions in combining RBMs with CRFs. Li et al. [71] combine pairwise, data-dependent potentials with a one-layer RBM prior in CRFs (referred as Compositional High Order Pattern Potentials (CHOPPs) in Figure 4.1(b)), and show the relationship between the marginalized RBM free energy and high-order potentials [86]. Kae et al. [47] augment CRFs with an RBM shape prior in a two-layer model for image labeling. Their lower layer has nodes for every superpixel of the image, with pairwise weights connecting them. The labels for this layer are then pooled into a raster structure, enabling them to use a RBM to provide shape priors.

Combining DBMs with variational models. Another attempt is to combine Deep Boltz-

mann Machines shape prior with a variational segmentation model [23], showing the effectiveness of strong shape priors for simple object segmentation. In all of the above approaches, the only inference pathway between the image features $\mathbf{x}$ and the hidden variables $\mathbf{h}$ representing shapes leads through the labels assigned to image pixels $\mathbf{y}$ while the shape only works as a prior. To perform inference and learning, the hidden variables are usually marginalized through an EM-like procedure. The shape information is thus not fully explored.

2.3 Transferring Shape Masks

Rigid window masks. In [57], the test image is matched with example images by window proposals. By adding up the matched window masks, the estimated segmentation prior contains more information about object location but less information about object shape. As a result, its segmentation performance largely depends on the final iterative GraphCut refinement step.

Deformable window masks. The algorithm in [3] involves two-step image matching. The window proposals of the test image are first localized on example images and then each localized image window is aligned using SIFT flow [73] with its corresponding test window proposal to achieve deformable mask transfer. Although a better shape prior could be obtained this way, running SIFT flow for thousands of window proposals with tens of examples inevitably results in considerable computational cost.

2.4 Putting Objects in Context

Context-driven maching. Context has been investigated in various semantic segmentation algorithms from co-occurrence statistics to scene categories [60, 27, 96, 42]. The key idea is to first label the input image only using appearance information, and then extract context information from initial semantic labels. Eigen and Fergus [27] investigate

context information in superpixel neighborhoods. By observing that the initial labeling usually produces reliable results on background classes, they build a context index for each superpixel using background labels in its four-directional neighborhood. Therefore, in additional to image retrieval, their method is able to find more relevant superpixels for matching. However, in large scale, even similar background (indoor) can encapsulate many object categories, which will result in the uncontrollable size of retrieved superpixels. Singh and Kosecka [96] instead focus on global semantic context. They construct a semantic label descriptor for each image in a three-layer spatial pyramid to refine image retrieval. Compared to appearance image descriptors, semantic label descriptors are much lower dimensional and allow fast image matching, but also more vulnerable when the initial labeling fails.

Context-driven classification. Tu and Bai [105] exploits local context in an iterative manner. They pose object segmentation as a pixel-wise classification problem. A classifier is first trained only using appearance features to predict a label for each pixel. The classification likelihood maps are then used as context features to train a second classifier together with appearance features. This procedure iterates until convergence. Labeling is improved gradually as each iteration will bring more local semantic information (context) into local prediction.

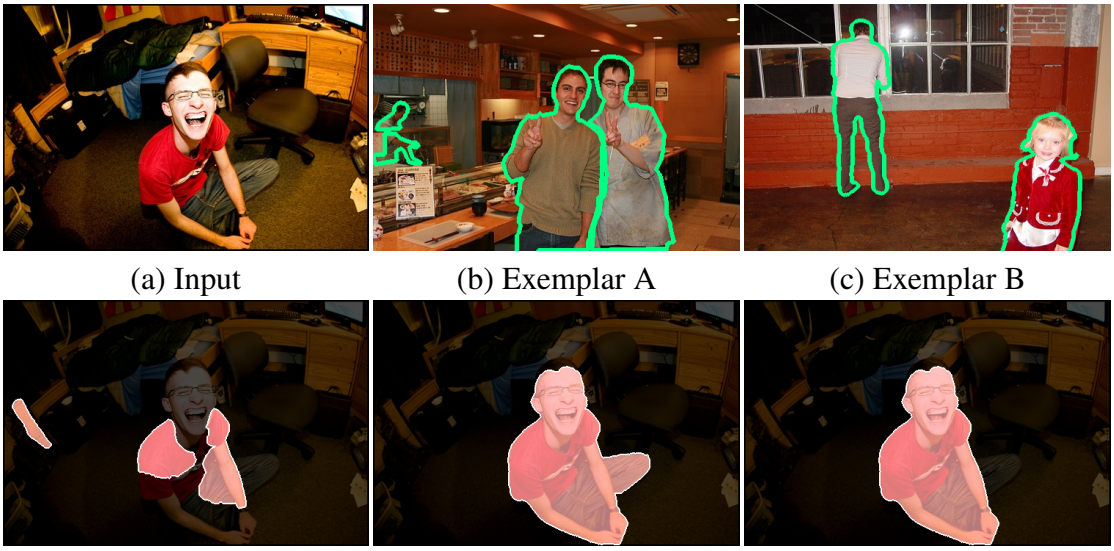
Chapter 3

Generating Object Segmentation Proposals with Exemplar Cut

3.1 Introduction

Category level object segmentation is one of the core problems in computer vision. Its main challenges lie in that small visual elements (pixels or superpixels) contain insufficient information that admits category level object recognition. One line of research aims at effectively propagating high level recognition results back to low level segmentation through superpixel neighborhood [36], high-order Conditional Random Fields (CRFs) [59] or object detector outputs [6, 114]. Another line makes efforts to generate object segmentation hypotheses so that recognition can be achieved more efficiently by classification or ranking [68].

Object segmentation hypotheses could be category independent or category specific. Recent work for category independent object segmentation [28, 20, 49] exploit hierarchical image segmentations, grouping strategies and cross-category shape priors in order to increase the chance of recovering true object regions. As a result, such methods are likely to generate thousands of instance-level object region hypotheses which entail la-



(a) Input (b) Exemplar A (c) Exemplar B

(d) MAP solution (e) Hypothesis A (f) Hypothesis B

Figure 3.1: Generating class-specific segmentation hypotheses from exemplars (person in this example).

borious post-processing to filter out low-quality solutions. Category specific approaches instead, [15, 58, 11] generate single object segmentation by using efficient maximum a posteriori (MAP) inference tools (e.g., graph cut [54]), which perform well when target objects appear dominantly in the images with simple backgrounds. In real-world applications, however, target objects more often appear in cluttered backgrounds with large appearance variations and interact with the objects of other categories (e.g., PASCAL VOC datasets [31]). In these cases, the single MAP solution becomes less satisfactory (Figure 3.1(d)) due to the limited model capacity and training errors. A natural choice to resolve this issue is to generate multiple object segmentation hypotheses from class-specific models [10, 39] (Figure 3.1(e)(f)). This choice not only benefits from learning but also increases the probability of finding all the target objects.

In this chapter, we propose a hybrid parametric and nonparametric model for generating a small set of highly plausible class-specific object segmentations, thereby reducing

ambiguities and computational loads for sequential classification or ranking. Towards that, we first learn a pylon model [66] to obtain the parametric object segmentation energy function. Building on a bottom-up hierarchical segmentation [7], the pylon model combines a flat CRF with a region tree. The resulting energy function remains submodular and admits efficient inference by graph cut, which brings conveniences to max-margin learning. Second, we match the test image with each exemplar by regions. For each region in the test image, we retrieve k nearest neighbors (K-NN) from the matching exemplar, so that the node potentials of the pylon model are augmented by K-NN matching scores. Therefore, an object segmentation hypothesis can be generated by solving a graph cut with the exemplar augmented energy function, which we refer as *exemplar cut*.

Our method leverages both the generalizability of parametric models and the flexibility of nonparametric models. Parametric models usually make assumptions on image segmentations. For example, CRFs and pylons assume that regions are classifiable in the node potentials, and labels between adjacent regions are consistent up to the Potts pairwise potentials. Under these assumptions, the MAP inference usually produces reasonably smooth labeling around the target (Figure 3.1(d)). The reason of missing some parts and predicting a false negative lies in that the node classifiers are less effective in handling heterogeneous appearance in complex background.

On the other hand, the nonparametric segmentations [99, 73, 57, 85] are more flexible to model assumptions. These methods are able to segment an image by transferring prior knowledge (e.g., labels and shape masks) from retrieved exemplars or regions in a database of segmentation exemplars. However, considering the statistical instability of using exemplars, challenges arise from integrating the retrieved or matched segmentation results into a single solution. Our method avoids such issue and instead queries each exemplar to generate one segmentation hypothesis. By adjusting the pylon energy function by the exemplar matching score, we fuse the parametric and nonparametric classifiers [22] on the node potentials and still take advantage of the label consistency assumption and learned parameters on the pairwise potentials. Consequently, we increase the possibili-

ties of correcting the mistakes of parametric models and prevent segmentation from noisy labeling.

We carry out experiments on the Graz-02 [82] and PASCAL VOC 2010 datasets [31]. We use the intersection/union overlap scores [31] to evaluate the upper bound performance of segmentation hypotheses. The results show that the proposed exemplar cut algorithm generates better segmentation hypotheses than the MAP solution and performs favorably against the state-of-the-art methods based on parametric min cut [53], diverse M-Best solutions [10] and multiple choice learning [39]. We also analyze the performance of hypotheses at different MAP quality levels. The results on the Graz-02 dataset suggest that exemplar cut maintains high recall rates when MAP solutions miss the target objects.

3.2 Exemplar Cut

In this section, we present the proposed exemplar cut algorithm for class-specific object segmentation in details. We first introduce the underlying segmentation model and then present our approach of generating multiple segmentation hypotheses with exemplars.

3.2.1 Category Specific Object Segmentation

Pylon Model. We use the two-class pylon model [66] as the underlying mechanism for category specific object segmentation. We first segment an image I into a hierarchical region tree $\mathbf{S} = \{S_1, S_2, ..., S_{2L-1}\}$ by the *gPb* contour detector [7]. We index the leaf segments from 1 to L, the intermediate segments from $L+1$ to $2L-2$ and the root segment (the entire image) as the last one, $2L-1$. We also denote $a(i)$ as the ancestor of segment i and $p(i,j)$ as the shortest path from segment i to segment j. Note that segment i and its ancestor $a(i)$ are overlapped, so we only need to keep one of them to explain the image. Each segment $S_i \in \mathbf{S}$ thus could be assigned a label $f_i \in \{0,1,2\}$, where $f_i = 1$ indicates the foreground, $f_i = 2$ the background, and $f_i = 0$ not being used for explaining

the image. To produce a consistent labeling $\mathbf{f} = \{f_1, ..., f_{2L-1}\}$, the pylon model requires that for any leaf segment, there is only one non-zero label along its path to the root node in the tree,

$$\forall i = 1, ..., L, \forall j \in p(i, 2L-1), f_j > 0 \wedge f_j \cdot f_{a(j)} = 0. \tag{3.1}$$

This constraint guarantees the complete and non-overlapping labeling. We present a simplified two-class pylon model in Figure 3.2. Each node represents a segment at a different

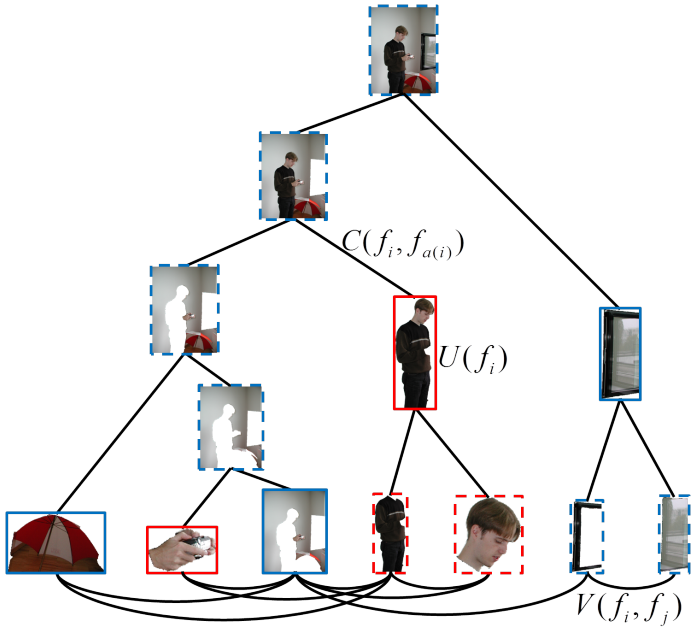


Figure 3.2: A two-class pylon model illustration.

level and its figure/ground assignment energy is given by $U(f_i)$ in (3.3). The edges between the leaf nodes $V(f_i, f_j)$ in (3.4) denotes the pairwise smoothness term. The edge between a segment and its ancestor represents the consistency constraint $C(f_i, f_{a(i)})$ in (3.1). A labeling of the image is visualized by colored bounding boxes around the segments. The red box indicates the figure label ($f_i = 1$) while the blue box indicates the ground label ($f_i = 2$). The dashed box indicates the segment is not being used to explain the image ($f_i = 0$) and all the solid boxes constitute a complete image.

We formulate an energy function for the pylon model similar to a conventional CRF,

$$E(\mathbf{f}) = \sum_{i=1}^{2L-1} U(f_i) + \sum_{(i,j)\in\mathcal{N}} V(f_i, f_j), \tag{3.2}$$

where the unary term $U(f_i)$ specifies the cost of assigning a label f_i for the segment i, and the pairwise term $V(f_i, f_j)$ instantiates the non-negative boundary cost between any two adjacent segments $(i, j) \in \mathcal{N}$. The set of adjacent segments is denoted by $\mathcal{N}$. In particular, we define the unary term as linear models,

$$U(f_i) = \begin{cases} |S_i| \cdot <\mathbf{w}_1, \mathbf{h}(\mathrm{S_i})>, & \text{for } f_i = 1, \\ |S_i| \cdot <\mathbf{w}_2, \mathbf{h}(\mathrm{S_i})>, & \text{for } f_i = 2, \\ 0, & \text{for } f_i = 0, \end{cases} \tag{3.3}$$

where $\mathbf{h}(S_i)$ denotes the feature vector extracted from the segment S_i and $\mathbf{w}_1, \mathbf{w}_2$ denotes the unary parameter vectors. We use the size of segment $|S_i|$ as a weighting factor to encourage the pylon inference to select larger segments. The pairwise term is defined by a weighted Potts model,

$$V(f_i, f_j) = <\mathbf{w}_3, \mathbf{b}(\mathrm{S_i}, \mathrm{S_j})> \cdot \delta[\mathrm{f_i} \neq \mathrm{f_j}], \tag{3.4}$$

where $\mathbf{b}(S_i, S_j)$ is a vector of exponentiated boundary strength with different bandwidths and $\mathbf{w}_3$ is the smoothness parameter vector.

Inference. It is not trivial to infer the pylon model with both semantic labels $f_i = 1, 2$ and exclusive label $f_i = 0$. In order to leverage the strength of graph cut for optimal inference on binary Markov random fields, Lempitsky et al. [66] decompose the target label f_i of each segment into a pair of equivalent binary variables x_i^1 and x_i^2, which indicate whether any of the segments along the path $p(i, 2L-1)$ falls entirely into the regions of class 1 or 2. Based on this re-parametrization technique, both the unary terms (except the root) and consistency constraint in (3.1) are absorbed into a new energy function $E(\mathbf{x}^1, \mathbf{x}^2)$ as pairwise terms. With some manipulations, this new energy function becomes sub-modular

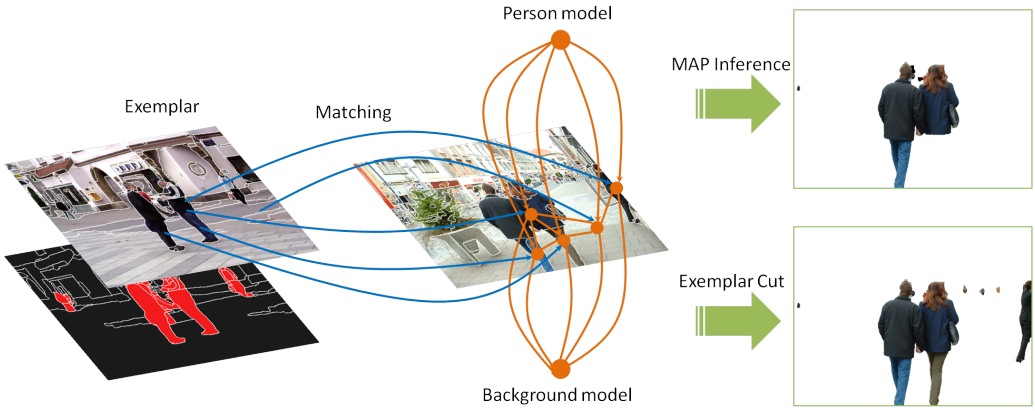


Figure 3.3: Exemplar cut algorithm diagram.

and can be minimized by graph cut. To recover the target labeling $\mathbf{f}$ from equivalent labelings $(\mathbf{x}^1, \mathbf{x}^2)$, we can simply examine if its ancestors up the tree are assigned non-zero labels. More details about this re-parametrization method can be found in [66].

Max-Margin Learning. The pylon model can be learned in a max-margin fashion. The optimization is formulated as,

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_n \xi_n, \quad \text{s.t.} \quad \forall \xi_n \geq 0 \\ &\max_{\mathbf{x}} [\Delta(\mathbf{x}^{(n)}, \mathbf{x}) - E(\mathbf{x}, \mathbf{y}^{(n)}; \mathbf{w}) + E(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}; \mathbf{w})] \leq \xi_n \\ &\mathbf{w_3} \geq 0 \end{aligned} \tag{3.5}$$

where $\mathbf{x}$ denotes all the binary variables $\{\mathbf{x}^1, \mathbf{x}^2\}$, $\mathbf{y}$ represents all the feature vectors $\{\mathbf{h}_i; \mathbf{b}_{ij}\}$ and $\mathbf{w}$ is the full parameter vector $[\mathbf{w}_1; \mathbf{w}_2; \mathbf{w}_3]$. We develop a stochastic gradient (sub-gradient) descent algorithm for efficiently optimizing the max-margin learning objective in (4.18).

3.2.2 Hybrid Parametric/Nonparametric Model

Pylon models have been demonstrated to be effective for figure/ground segmentation and semantic scene parsing [66], due to their ability of selecting larger regions in the

segmentation tree to reduce the classification ambiguity. When the pylon models are applied to object class segmentation, their MAP solution becomes less effective in dealing with complex object appearance and their interactions. On one hand, the Markov random field assumption about image structures may be invalid for the objects with heterogeneous appearance. For example, the strong edges between different body parts and partial occlusions break the smoothness assumption while the cluttered background violates the discontinuity assumption around object boundaries. On the other hand, the max-margin objective and the use of slack variables in (4.18) usually introduce bias to the process of model learning, although they improve the generalizability to unseen images and the resistance to noises.

On the contrary, exemplar based nonparametric approaches, which involve no model assumptions and training process, can generate segmentations by image matching and prior transfer. Although they are flexible of exploiting prior knowledge contained in exemplars, the nonparametric approaches still face the challenge of filtering the matching results due to the large variance of exemplars.

We herein present a hybrid method by integrating the pylon model with exemplars. Figure 3.3 illustrates the proposed algorithm. The goal is to segment an input image in the **center** into the foreground object and the background. We construct a pylon graph [66] on the regions generated by [7]. The node energies (orange links) are constructed by the pre-learned person and background models (orange nodes). The pairwise energies are constructed on the adjacent regions by measuring their shared boundaries (white lines in the image) and compatibilities. Note that we omit the hierarchical structure for ease of illustration. On the **left** are an exemplar image (**top**) and its segmentation mask (**bottom**). The exemplar is also segmented into regions by [7] and each region is assigned a label based on the mask. We find the best match (blue links) in the exemplar image for each region of the test image. Based on their labels, we incorporate the node energy of each region with the matching similarities. We solve a graph cut to this augmented energy function for generating an exemplar cut solution, which is shown on the **bottom right**. As the test

image shares similar appearance with the exemplar image, the resulting segmentation has a very high accuracy. The original MAP solution on the **top right** instead misses many small targets and one occluded person. To segment an input image I, we first compute its parametric energy from the learned pylon model through (3.2), which provides a basis of generating smooth segmentations close to the ground truth. The exemplars are then used to generate segmentation hypotheses different from the MAP solution, through region matching. We represent an exemplar by an image $I^{(n)}$ and its segmentation $\mathbf{f}^{(n)}$, and denote the matching energy by $\Delta(\mathbf{f}, \mathbf{f}^{(n)}; I, I^{(n)})$. Therefore, a segmentation hypothesis can be generated by solving graph cut to the augmented energy function,

$$\tilde{E}(\mathbf{f}, \mathbf{f}^{(n)}) = (1 - \lambda)U(\mathbf{f}) + \lambda\Delta(\mathbf{f}, \mathbf{f}^{(n)}) + V(\mathbf{f}), \tag{3.6}$$

where $\lambda \in [0, 1]$ controls the tradeoff between the parametric and nonparametric energies. In this exemplar cut energy function (3.6), we actually resolves the learning bias of pylon model by the matching variance with exemplars.

3.2.3 K-Nearest Neighbor Region Matching

In this work, we compute the nonparametric matching energy function $\Delta(\mathbf{f}, \mathbf{f}^{(n)})$ by using the K-NN matching algorithm. We parse an exemplar image $I^{(n)}$ into a set of hierarchical regions $\mathbf{S}^{(n)}$. For each region, we extract a feature vector $\mathbf{h}_i^{(n)}$ and assign a ground truth label $f_i^{(n)}$ from its annotated object masks. The exemplar is thus represented by a set of feature-label pairs $\{(\mathbf{h}_i^{(n)}, f_i^{(n)})\}_{i=1,2,...,2L-1}$. Note that we assign non-zero labels to regions that fall entirely into ground truth foreground, which increases the chance of recovering object parts. We assume that segments are independent of each other so that we approximate the matching energy by

$$\Delta(\mathbf{f}, \mathbf{f}^{(n)}) = \sum_{i=1}^{2L-1} \Delta(f_i, \mathbf{f}^{(n)}). \tag{3.7}$$

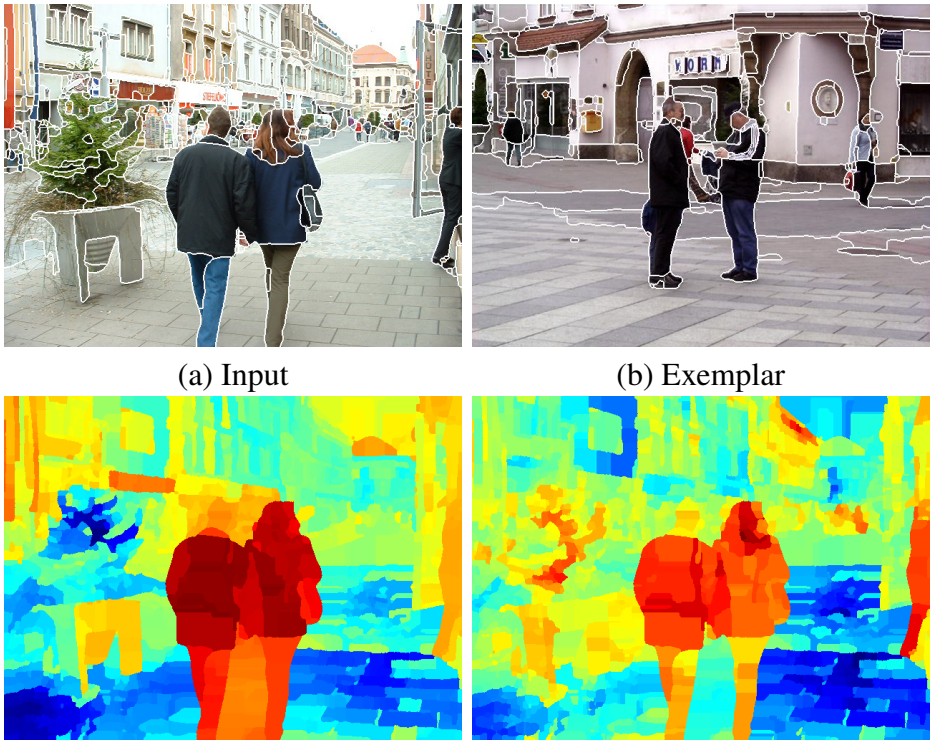
For each region i in the test image, we retrieve K best segments in the exemplar based on their feature similarities,

$$(\mathbf{h}_k^{(n)}, f_k^{(n)})_{k=1,2,\ldots,K} \leftarrow \text{K-NN}(\mathbf{h}_i, \{(\mathbf{h}_j^{(n)}, f_j^{(n)})\}). \tag{3.8}$$

We define the matching energy of region i by the K-NN output,

$$\Delta(f_i) = -\frac{1}{K}\sum_{k=1}^{K} <\mathbf{h}_\text{i}, \mathbf{h}_\text{k}^{(\text{n})}> \cdot \delta[\text{f}_\text{i} = \text{f}_\text{k}^{(\text{n})}]. \tag{3.9}$$

This matching energy can be easily merged into the parametric unary term (3.3) so that the augmented energy function (3.6) remains the same form as the original energy function (3.2) and can be solved by the inference algorithm developed in [66]. Figure 3.4 shows



(a) Input (b) Exemplar

(c) Pylon confidence map (d) K-NN confidence map

Figure 3.4: Foreground confidence maps of the pylon model and K-NN matching method.

the foreground confidence maps of pylon model and K-NN matching, and their complementary effects. The confidence maps are computed from the node energy. The K-NN matching recovers small targets at a distance and a partial figure near the right boundary, which is missed by the pylon model. The pylon model removes the false positives of the K-NN matching method around the tree region.

3.3 Experiments

We present the experimental results using the Graz-02 [82] and PASCAL VOC 2010 datasets [31] with evaluations of the proposed algorithm against several state-of-the-art class-specific methods for generating segmentation hypotheses, i.e., the parametric min cut (PMCut) method, the diverse M Best solutions (MBest) [10] and the multiple choice learning (MCL) approach [39]. We implement these three algorithms based on the learned pylon model. We use the intersection/union overlap scores to evaluate the upper bound performance of these segmentation algorithms.

3.3.1 Graz-02

Setup. The Graz-02 dataset includes 3 object classes (bike, car and person) and background images, which is challenging for object segmentation due to large pose variations, scale changes and partial occlusions. Each class consists of 300 images $I^{(n)}$ of 640×480 pixels and all the images are annotated by foreground masks. The odd-numbered images per class are used for training pylon models and for exemplars, and the rest for evaluation. We set the parameter $\lambda = 0.9$ as the tradeoff of parametric and nonparametric energy, and the parameter $K = 7$ as the number of nearest neighbors in the K-NN matching method[1].
Representations. We represent an image by a bottom-up segmentation tree using the *gPb* algorithm [7]. The number of segments is around 7000 on average. Each segment S_i is

[1]We empirically determine these values by evaluating $\lambda \in [0.7, 1.0]$ and $K = [5, 9]$.

represented by three kinds of features $\mathbf{h}_i = [\mathbf{h}_i^{SIFT}; \mathbf{h}_i^{color}; \mathbf{h}_i^{shape}]$. We extract dense SIFT descriptors using the VLFeat toolbox [106] and train a codebook of size 512 using all the training images of three classes. The SIFT histogram $\mathbf{h}_i^{SIFT}$ is computed using the locality constrained linear coding method [109] and max pooling. The color histogram $\mathbf{h}_i^{color}$ is computed from a color codebook of size 128 by assigning each RGB pixel to its closest codeword. The contour shape descriptor $\mathbf{h}_i^{shape}$ is extracted from a spatial pyramid of oriented *gPb* edge responses [38]. We map the concatenated feature vectors to a high-dimensional space with the explicit χ^2 kernel [106].

Ground truth labeling. To train the pylon models and use exemplars, we need to determine the ground truth labels for each region in a training image, generated by the *gPb* algorithm. Due to the errors from the *gPb* contour detection and blurred boundaries of annotated masks, the average overlap score of the pylon ground truth segmentations with annotated masks is 87.8%.

Results. We use the learned pylon models of each class to generate MAP segmentations of test images. For the diverse MBest approach, we observe that its performance improves slightly when the number of solutions is close to 30, thereby its overlap scores are reported by using 30 solutions per image. For the MCL method, we train up to 5 sub-models for each class and observe some sub-models are underutilized (as no training exemplars belong to some sub-models). For the PMCut, we follow the method in [49] and draw 41 samples by varying the constant parameter from -2.0 to 2.0 with the increment 0.1. By using exemplar cuts, we generate 150 raw segmentation hypotheses per test image. Note that all these methods generate redundant segmentations. We evaluate their performance from raw results without considering post-processing to merge the duplicates and remove the low quality ones. Table 3.1 shows the overlap scores of segmentation approaches by using the "oracle" evaluation protocol [10]. The exemplar cut algorithm outperforms the MAP solution by 15.3% on average and the second best MCL approach by 5.1%.

We present some qualitative results in Figure 3.5. Bikes, cars and people are highlighted by green, blue and red masks, respectively. From left to right, the segmentations

Table 3.1: "Oracle" overlap scores in the Graz-02 dataset.

overlap	Bicycle	Car	Person	mean
MAP	66.4	61.4	60.6	62.8
MBest	69.8	67.3	66.0	67.7
PMCut	73.9	70.6	70.7	71.7
MCL	73.1	75.1	71.0	73.0
ExemplarCut	**77.4**	**78.9**	**78.0**	**78.1**

are generated by (a) MBest, (b) MCL, (c) PMCut and (d) ExemplarCut. It shows that the segmentation hypotheses generated by exemplar cut are usually able to deal with occlusions, suppress the background clutters and recover the missing targets.

We generate up to 150 segmentations per test image. Moreover, we are interested in the performance of exemplar cut segmentations when the MAP solutions fail to detect the targets. We consider the hypotheses as true positives if their overlap with ground truth are greater than 50%; otherwise as false positives. In Figure 3.6, we compare the recall rates of segmentation hypotheses at different MAP quality levels. From the left to the right, we compare the evaluated four algorithms for bike, car and person, respectively. When the MAP overlap score is smaller than 10%, almost losing the targets, the hypotheses generated by exemplar cut (red curves) achieve recall rates 75.0%, 52.6% and 46.7% for bike, car and person datasets. The overall recall rates are 97.3%, 88.7% and 90.0% for bike, car and person datasets, respectively.

3.3.2 PASCAL VOC 2010

Setup. In the PASCAL VOC 2010 dataset, each of the training, validation and test sets includes 964 annotated images from 20 object categories and one background class. The images in this dataset may include multiple objects from several categories. We use the training set to learn pylon models, and evaluate the proposed algorithm on the validation set. For class-specific object segmentations, we first parse the multi-class segmentation masks into class-specific object masks, and then train a pylon model on the positive exem-

(a) MBest (b) MCL (c) PMCut (d) ExemplarCut

Figure 3.5: Segmentation results on the Graz-02 test set.

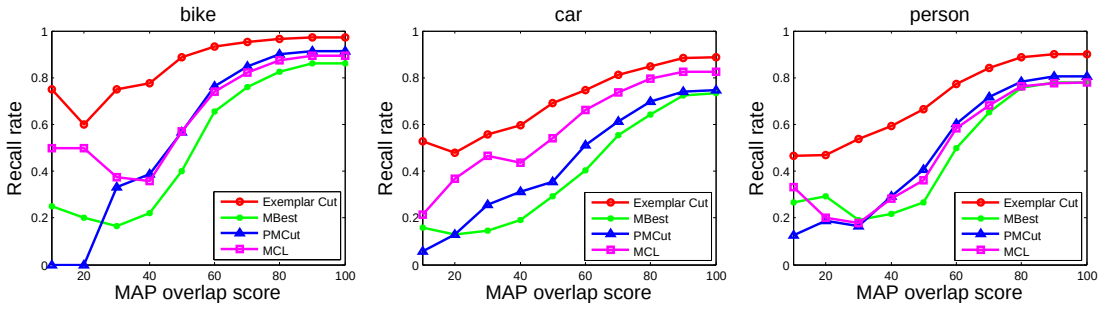


Figure 3.6: Recall rates at different MAP quality levels on the Graz-02 datasets.

plars for each class.

Implementations. Similar to the Graz-02 experiments, we represent an image with a bottom-up segmentation tree using the *gPb* method and represent each segment by SIFT, color and shape features. Considering the large appearance variations, we train a SIFT codebook of size 8096. We use the same method in the Graz-02 experiments to obtain ground truth pylon labelings and use the same values for the parameter λ and K. For exemplar cut of each class, we use the positive training images as exemplars and thus generate about 50 segmentation hypotheses per class on average. We implement the PMCut method by varying the parameter from -2 to 2 by increasing 0.1 each step. For the MCL algorithm, we choose to initialize one predictor per 10 images since each class has different training images. We segment one test image using the models from 20 classes, because we have no information about the object categories before running segmentation algorithms.

Results. We evaluate the quality of segmentation hypothesis sets by "oracle" overlap scores in Table 3.2. In the top panel, we present the overlap scores for class-specific object segmentation hypotheses. The exemplar cut performs the best on 18 of 20 classes, and achieves 60.8% average overlap score, with an improvement over the second best, PMCut, by 9.7%. To elucidate the quality of our segmentation hypotheses, we present one image per class and compare its exemplar cut and the second best segmentation in Figure 3.7. In each panel of Figure 3.7, the left image is produced by exemplar cut and the right image

Table 3.2: "Oracle" class-wise overlap scores on the VOC 2010 validation set.

	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
MAP	52.9	19.1	35.6	39.3	23.3	49.9	43.1	47.4	11.8	56.9
MCL	57.9	**29.3**	45.7	36.1	32.8	57.4	50.9	61.0	23.9	68.8
PMCut	60.3	24.3	47.9	51.0	39.2	67.1	51.6	66.4	22.7	69.3
ExemplarCut	**66.8**	28.0	**66.6**	**60.7**	**44.1**	**71.1**	**59.8**	**72.7**	**34.5**	**76.4**

	dtable	dog	horse	mbike	person	plant	sheep	sofa	train	tv	average
MAP	36.8	45.1	50.6	44.3	37.9	20.9	58.1	34.5	48.9	33.1	39.5
MCL	49.7	60.1	58.5	55.1	50.0	20.9	65.4	45.9	61.1	35.7	48.3
PMCut	50.7	55.9	56.7	55.7	50.3	29.6	**71.6**	45.2	62.7	43.9	51.1
ExemplarCut	**71.4**	**68.9**	**68.2**	**64.9**	**63.9**	**39.5**	71.0	**62.1**	**70.4**	**55.2**	**60.8**

is produced by the most competing algorithm (MCL or PMCut). For the bicycle image, we present a case that exemplar cut is slightly worse than MCL. For the sheep image, we present a failure case of exemplar cut where PMCut performs the best. The masks for different classes follow the VOC color codes.

In order to see the potentials of class-specific hypotheses for category level object segmentation, we compute the upper bound overall overlap score in Table 3.3. For a test image, we select the best segmentation from each class as its category confidence map. The class segmentation is thus given by the maximum confidence score from all the classes at pixels. Intuitively, if there is an oracle segmentation selector, we can achieve 68.7% over-

Table 3.3: "Oracle" multi-class overlap scores on the VOC 2010 validation set.

	MAP	MBest [10]	MCL	PMCut	ExemplarCut
overlap	44.86	48.0	54.3	58.8	68.7

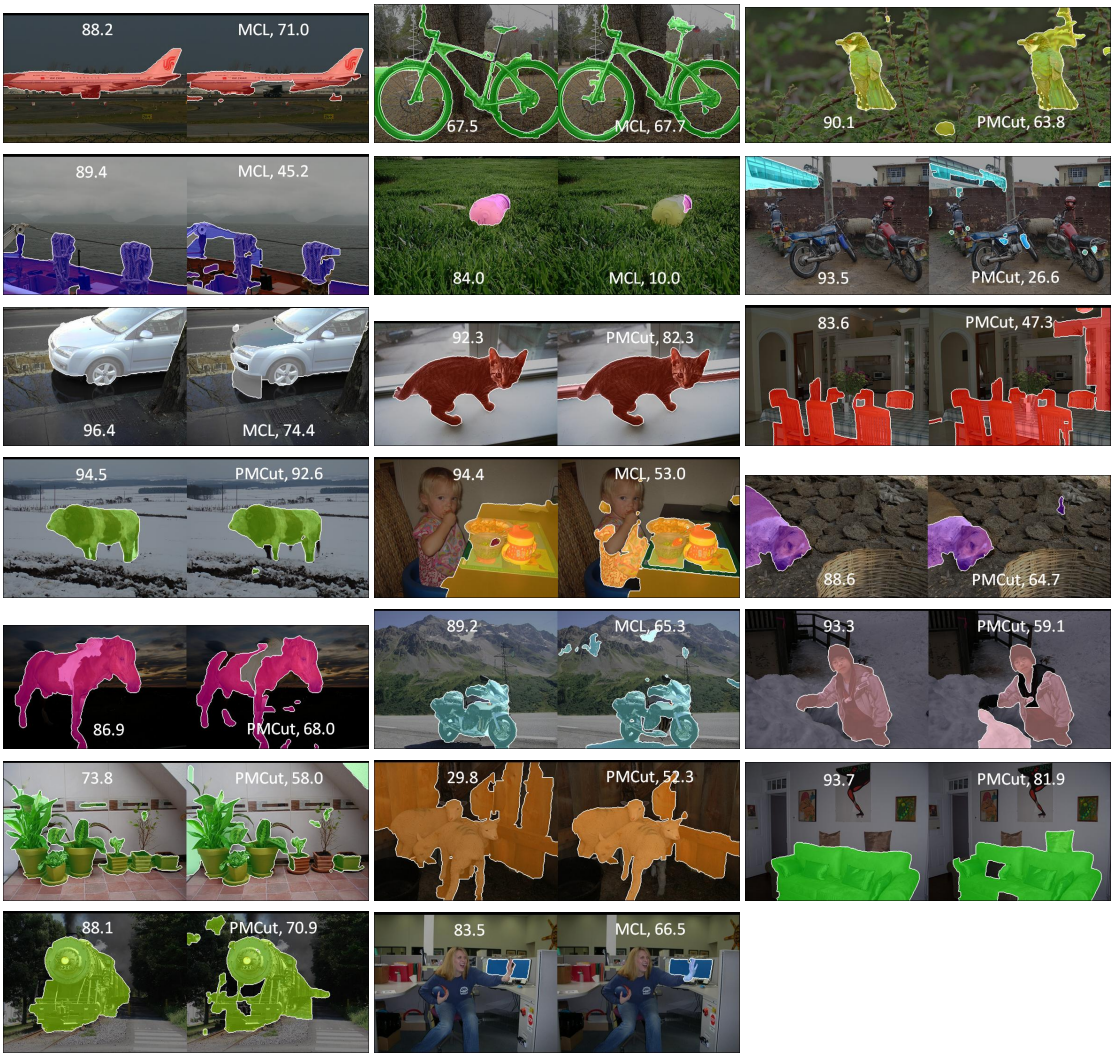


Figure 3.7: Comparing segmentations for 20 classes on the VOC 2010 evaluation datasets.

lap accuracy on the PASCAL VOC 2010 validation set. This result suggests the potential of exemplar cut for category level object segmentation. Note that the MBest result [10], is computed from a multi-class CRF model [59]. We do not present the MBest result by the pylon model because we find it difficult to tune a single dissimilarity parameter for all the classes. The number of segmentation hypotheses for exemplar cut equals the number of exemplars (i.e., 50). In both Graz-02 and VOC 2010 datasets, exemplars are limited for each class. Thus it is convenient to use all of them to generate hypotheses. When the number of exemplars is large, it will be infeasible to use all the exemplars. Similar to the algorithms for scene parsing [73, 99], we can also use image retrieval techniques to find the most relevant exemplars for hypothesis generation. The segmentation results in this work may include some isolated false negatives in the background area and duplicate segments from different hypotheses. It will be worthwhile developing segment filters to remove noise and [20] before using them for classification purposes.

3.4 Summary

We present a novel exemplar cut algorithm for generating class-specific object segmentation hypotheses. It combines a learning based parametric segmentation model and a matching based nonparametric segmentation algorithm in a principled way. Experimental results on the Graz-02 and PASCAL VOC 2010 datasets demonstrate that the proposed exemplar cut algorithm achieves favorable results in terms of visual quality and accuracy. In addition, the results show that the proposed algorithm is especially effective when the MAP approaches fail to generate good segmentation results on images with complicated scenes.

Chapter 4

Learning Shape Representations for Object Segmentation with Max-Margin Boltzmann Machines

4.1 Introduction

Object segmentation can be formulated as a structured output problem that involves making predictions collectively over correlated output labels $\mathbf{y} \in \mathcal{Y}$ from input observations $\mathbf{x} \in \mathcal{X}$. One of the core issues in structured output prediction problems is how to represent complex output variable interrelations effectively while carrying out inference and learning efficiently.

In Markov Random Fields (MRFs), output structures are represented by pairwise and high-order potential functions $p(\mathbf{y}) = \prod_{\mathbf{y}_i \subset \mathbf{y}} \phi(\mathbf{y}_i)/\mathbf{Z}$ where $\mathbf{Z}$ is the partition function. The prediction from the observations $\mathbf{x}$ to the labels $\mathbf{y}$ is usually realized in the conditional models $p(\mathbf{y}|\mathbf{x})$, i.e., Conditional Random Fields (CRFs) [62], which allow flexible use of various long-range features from observations $\mathbf{x}$. Pairwise potentials [97], although admitting efficient inference, can only capture limited local structure, such as smoothness

and edges. High-order potentials are able to capture long-range interactions between pixel labels through bottom-up segmentation [51], pattern-based priors [86, 91]. Beyond the generic high-order priors, the ObjCut algorithm [58] introduces category-specific object models into MRFs and has shown good segmentation performance on articulated objects. In ObjCut, the hidden variables of pictorial structures encode the positions of object parts, but their interactions with pixel labels are manually designed.

Alternatively, Restricted Boltzmann Machines (RBMs) render more flexible models for structured output representation that learn high-order interrelations through a joint distribution of labels and a set of hidden (latent) variables $\mathbf{h}$ in $p(\mathbf{y}, \mathbf{h})$. By omitting lateral connections in a single layer, RBMs admit efficient inference and sampling from conditional probabilities. When operating with a small number of training samples, layered architectures [89] have been shown more effective in terms of model expressiveness and learning efficiency. Eslami et al. [29] propose a two-layer Boltzmann Machine $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2)$ (where $\mathbf{h}^1$ and $\mathbf{h}^2$ denote hidden variables in two layers) for modeling object shapes (ShapeBMs), and apply it onto object segmentation in a generative model $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{x})$ [30].

In this chapter, we present a general class of Conditional Boltzmann Machines (CBMs) for object segmentation in the form of $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ and $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$. In addition to the connections from image to labels, our models also include the connections from the image to hidden variables, which allows direct shape inference from image features. Based on layer-wise conditional independence of BMs, we derive a simple but efficient Iterated Conditional Modes [12] algorithm for maximum a posteriori (MAP) inference.

Learning with CRFs and CBMs is challenging as it requires handling exponentially large numbers of output combinations in data-dependent partition functions. Approximate learning algorithms are easily trapped in local optima, thereby limiting their generalization performance. Another line of research for structured output prediction is developed on max-margin formulations [98, 104, 46], that facilitates model generalizability to unseen test data. This technique has been applied to CRFs for object segmentation [97, 11].

In a similar spirit, we propose a max-margin formulation of CBMs, referred as MMBMs, and develop an online Concave-Convex Procedure (CCCP) [120] algorithm for learning efficiently with hidden variables. Note that large margin BMs have been proposed in [78] with a focus on theoretical analysis while our max-margin method is proposed for training a particular class of CBMs with applications to object segmentation. We investigate the effects of four kinds of margin functions on discriminative training, and demonstrate the importance of combined hidden and visible margin functions. We study two variants of MMBMs with a single hidden layer $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ as well as two hidden layers $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$, and compare them with two state-of-the-art models: superpixel based CRFs [7] and Compositional High Order Pattern Potentials [71]. We carry out experiments on the Weizmann horse [15], Penn-Fudan pedestrian [14] and Caltech-UCSD birds 200 [112] datasets. Experimental results show that the proposed MMBMs perform better than existing methods both quantitatively and qualitatively.

4.2 Models

In this section, we first introduce two variants of Boltzmann Machines, RBMs and ShapeBMs, for modeling object shapes, and then describe the proposed conditional models and the maximum a posteriori inference algorithm.

4.2.1 Boltzmann Machines

Given a labeled image of an object, we represent the mask as a set of visible variables $\mathbf{y} \in \{0, 1\}^n$. RBMs use one layer of hidden variables $\mathbf{h} \in \{0, 1\}^m$ to capture global dependencies between visible variables (See Figure 4.1(a))

$$p(\mathbf{y}, \mathbf{h}) = \exp(-E(\mathbf{y}, \mathbf{h}))/\mathbf{Z}, \tag{4.1}$$

where $\mathbf{Z}$ is the partition function. RBMs do not have lateral connections within visible and hidden layers so that the energy function takes the form,

$$E(\mathbf{y}, \mathbf{h}) = -\mathbf{y}^\top \mathbf{W}\mathbf{h} - \mathbf{b}^\top \mathbf{y} - \mathbf{c}^\top \mathbf{h}, \tag{4.2}$$

parametrized by $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{W}$. One attractive property of RBMs is that visible variables are conditionally independent given hidden variables and vice versa. The conditional probability of each variable is essentially the sigmoid function $\sigma(y) = 1/(1 + \exp(-y))$,

$$p(y_i = 1|\mathbf{h}) = \sigma(\sum_j w_{ij} h_j + b_i), \tag{4.3}$$

$$p(h_j = 1|\mathbf{y}) = \sigma(\sum_i w_{ij} y_i + c_j), \tag{4.4}$$

which facilitates efficient inference.

Although RBMs have the capacity of modeling complex distributions, they require a large set of hidden variables and numerous training examples. For object segmentation, it is labor intensive to collect a large number of training examples with ground truth masks, and challenging to train RBMs with a large set of variables. It is, however, possible to ameliorate this problem by considering the spatial structure of images. Eslami et al. [29] propose a particular form of Boltzmann Machine with two hidden layers $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2)$ (referred as ShapeBM) for object shape modeling. The first layer of hidden variables $\mathbf{h}^1$ is partitioned into several disjoint subsets $\{\mathbf{h}^1_k = \mathbf{h}^1(\mathbf{J}_k)\}_{k\in\mathbf{G}}$ of same size m^1_k, where $\mathbf{J}_k \in \{0, 1\}^m$ denotes the subset indexing. Each of them has a restricted receptive field and only connects to a local patch of the object mask. The local patches $\{\mathbf{y}_k = \mathbf{y}(\mathbf{I}_k)\}_{k\in\mathbf{G}}$ have the same size n_k and they overlap each other along the boundaries, where $\mathbf{I}_k \in \{0, 1\}^n$ denotes the patch index. Therefore, the pairwise potentials between visible variables $\mathbf{y}$ and the first layer hidden variables $\mathbf{h}^1$ can be represented by $\sum_{k\in\mathbf{G}} \mathbf{y}^\top_k \mathbf{W}^1_k \mathbf{h}^1_k$. Furthermore, different patches can share the same weights $\mathbf{W}^1 = \mathbf{W}^1_k, k \in \mathbf{G}$. The second layer of hidden variables $\mathbf{h}^2$ connects to all the variables $\mathbf{h}^1$ of the first layer. Similar to RBMs, there are no lateral connections between variables within any single layer. The energy function can be thus described by

$$\begin{aligned}\mathbf{E}(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2) = &- \sum_{k\in\mathbf{G}} \mathbf{y}^\top_k \mathbf{W}^1 \mathbf{h}^1_k - \mathbf{b}^\top \mathbf{y} - \\ &\mathbf{c}^{1\top} \mathbf{h}^1 - \mathbf{h}^{1\top} \mathbf{W}^2 \mathbf{h}^2 - \mathbf{c}^{2\top} \mathbf{h}^2.\end{aligned} \tag{4.5}$$

The pairwise term of the first layer can be rewritten in the same form as RBMs by some matrix manipulation:

$$\sum_{k \in \mathbf{G}} \mathbf{y}_k^\top \mathbf{W}^1 \mathbf{h}_k^1 = \mathbf{y}^\top \tilde{\mathbf{W}}^1 \mathbf{h}^1, \quad \text{where } \tilde{\mathbf{W}}^1(\mathbf{I}_k, \mathbf{J}_k) = \mathbf{W}^1. \tag{4.6}$$

Due to its structure, ShapeBM uses much fewer parameters than conventional two-layer RBMs [89], thereby facilitating efficient learning for smaller datasets. The pairwise term $\mathbf{y}^\top \tilde{\mathbf{W}}^1 \mathbf{h}^1$ models the compatibility between pixels and parts while the term $\mathbf{h}^{1\top} \mathbf{W}^2 \mathbf{h}^2$ defines the possible configuration of parts. Thus, when an unit of $\mathbf{h}^1$ is activated, a template stored in $\mathbf{W}^1$ is selected to enforce the group of pixels to obey a binary image pattern. Also, when an unit of $\mathbf{h}^2$ is activated, it triggers a particular configuration of parts (due to varying pose or viewpoint). The ShapeBM architecture also enjoys the property of conditional independence $p(\mathbf{y}|\mathbf{h}^1), p(\mathbf{h}^1|\mathbf{y}, \mathbf{h}^2)$ and $p(\mathbf{h}^2|\mathbf{h}^1)$, although exact inference is not tractable for this model.

4.2.2 Conditional Boltzmann Machines

While generative RBMs and ShapeBMs are capable of modeling object shape priors, it is still challenging to efficiently infer a binary object mask $\mathbf{y}$ from an image $\mathbf{x}$. Intuitively, we can construct a fully generative model for object images and their binary masks $p(\mathbf{y}, \mathbf{x})$ such that object shape can be inferred from an image by the conditional distribution $p(\mathbf{y}|\mathbf{x})$, and an image generated from a shape mask by $p(\mathbf{x}|\mathbf{y})$. As an example, Eslami et al. [30] present a generative multinomial joint model of appearance (object images) and shape (parts-based segmentation).

Nevertheless, constructing a joint model of object images and shape masks poses significant difficulties as the conditional distribution of images given the shape masks are intrinsically multimodal and full of ambiguities. In order to estimate the object mask $\mathbf{y}$ from an image $\mathbf{x}$, we instead propose to directly train the conditional models $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ for RBMs (MMBM1, Figure 4.1(c)) and $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$ for ShapeBMs (MMBM2, Fig-

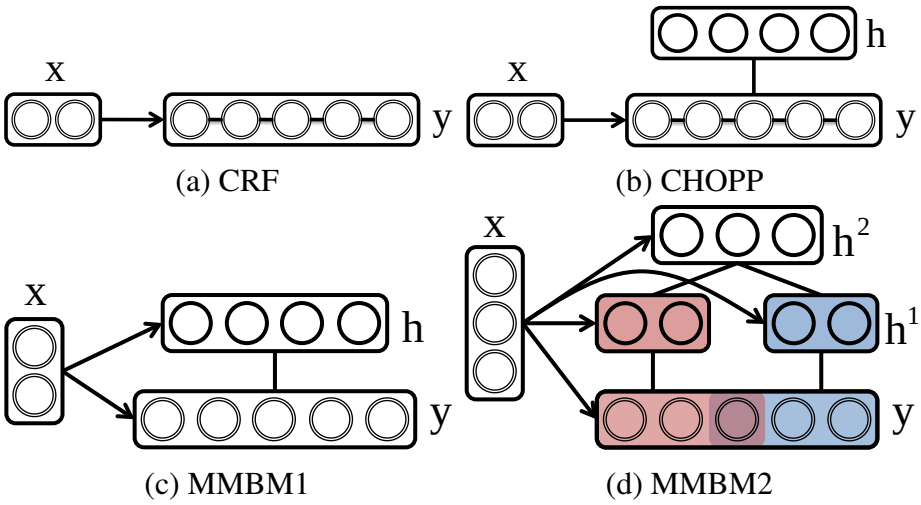


Figure 4.1: Comparing graphical models of MMBMs ((c) and (d)) with pairwise CRF (a) and CHOPP [71].

ure 4.1(d)). In these conditional models, the activations of variables depend on the observations or image features, so the energy function of $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ can be represented by

$$\mathbf{E}(\mathbf{y}, \mathbf{h}, \mathbf{x}) = -\mathbf{y}^\top \mathbf{W}\mathbf{h} - \mathbf{h}^\top(\mathbf{V}^1\mathbf{x}^1 + \mathbf{c}) - \mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b}), \tag{4.7}$$

while the energy function of $p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x})$ takes the form,

$$\begin{aligned}\mathbf{E}(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{x}) &= -\mathbf{y}^\top \tilde{\mathbf{W}}^1\mathbf{h}^1 - \mathbf{h}^{1\top}\mathbf{W}^2\mathbf{h}^2 \\ &- \mathbf{h}^{1\top}(\mathbf{V}^1\mathbf{x}^1 + \mathbf{c}^1) - \mathbf{h}^{2\top}(\mathbf{V}^2\mathbf{x}^2 + \mathbf{c}^2) - \mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b}).\end{aligned} \tag{4.8}$$

In the above equations, $\mathbf{x}^0$ represents low-level image features that indicate foreground and background assignments. The variable $\mathbf{x}^1$ represents features of object parts and $\mathbf{V}^1$ contains templates of object parts. The variable $\mathbf{x}^2$ describes the holistic object features, and $\mathbf{V}^2$ is composed of object templates of different poses and viewpoints. In these two models, we connect the observations $\mathbf{x}$ to both visible and hidden layers, which enables the direct inference pathway from image features to shapes.

4.2.3 MAP Inference

Given a set of image features $\mathbf{x}$, the most likely configuration of $\mathbf{y}$ is computed from

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}). \tag{4.9}$$

In the proposed MMBM with single hidden layer, the marginal distribution $p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ can be represented by its free energy form $\exp(-F(\mathbf{y}, \mathbf{x}))/\mathbf{Z}$), and

$$\begin{aligned} -F(\mathbf{y}, \mathbf{x}) =& \mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b}) + \\ & \sum_j \log(1 + \exp(c_j + \mathbf{y}^\top \mathbf{W}_{.j} + \mathbf{V}^1_{j.}\mathbf{x}^1)). \end{aligned} \tag{4.10}$$

where $\mathbf{W}.j$ and $\mathbf{W}_{j.}$ denote j-th column and row of $\mathbf{W}$, respectively. As the partition function $\mathbf{Z}$ is constant given $\mathbf{x}$, the MAP inference in (4.9) is exactly equivalent to optimizing the free energy function

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} -F(\mathbf{y}, \mathbf{x}). \tag{4.11}$$

Note that the free energy $F(\mathbf{y}, \mathbf{x})$ is not a linear function of $\mathbf{y}$, and we need to take gradients to find the optimal $\hat{\mathbf{y}}$. However, the analytic free energy is not available in the MMBM with two hidden layers. We instead optimize the variational upper bound of log-likelihood $\log p(\mathbf{y}|\mathbf{x})$ using the EM algorithm in spirit similar to techniques that have been effectively applied to training generative BMs. However, in MMBMs, both visible $\mathbf{y}$ and hidden variables are conditioned on input variables $\mathbf{x}$. The conditional distributions $p(\mathbf{y}|\mathbf{h}^1, \mathbf{x})$, $p(\mathbf{h}^1|\mathbf{y}, \mathbf{h}^2, \mathbf{x})$ and $p(\mathbf{h}^2|\mathbf{h}^1, \mathbf{x})$ are likely highly peaked, if not unimodal, and thus they can be approximated by optimizing

$$\{\hat{\mathbf{y}}, \hat{\mathbf{h}}^1, \hat{\mathbf{h}}^2\} = \arg\max p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x}). \tag{4.12}$$

Similar to the block Gibbs sampling method, the independent property of conditional distributions induces an efficient Iterated Conditional Modes (ICM) algorithm (See Algorithm 1). The ICM algorithm also provides a good approximate solution to the free energy optimization problem in (4.11) and (4.10) for single layer MMBMs. Essentially, the second term in (4.10) can be approximated by

Algorithm 1 MAP inference by the ICM algorithm.

1: Initialize $\mathbf{h}^1$
2: **while** do not converge **do**
3: $\quad \mathbf{h}^2 \leftarrow \max p(\mathbf{h}^2|\mathbf{h}^1, \mathbf{x})$
4: $\quad \mathbf{y} \leftarrow \max p(\mathbf{y}|\mathbf{h}^1, \mathbf{x})$
5: $\quad \mathbf{h}^1 \leftarrow \max p(\mathbf{h}^1|\mathbf{y}, \mathbf{h}^2, \mathbf{x})$
6: **end while**

$$\begin{aligned} &\sum_j \log(1 + \exp(c_j + \mathbf{y}^\top \mathbf{W}_{.j} + \mathbf{V}^1_{j.}\mathbf{x}^1)) \approx \\ &\max_{\mathbf{h}}(\mathbf{c}^\top \mathbf{h} + \mathbf{y}^\top \mathbf{W}\mathbf{h} + \mathbf{h}^\top \mathbf{V}^1 \mathbf{x}^1), \end{aligned} \tag{4.13}$$

which can be solved by the ICM algorithm.

4.3 Learning

Given a training set of object image-mask pairs $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, we learn MMBMs for object segmentation. As the proposed learning algorithm can be applied to both MMBMs with single ($p(\mathbf{y}, \mathbf{h}|\mathbf{x}; \omega)$) or two hidden layers ($p(\mathbf{y}, \mathbf{h}^1, \mathbf{h}^2|\mathbf{x}; \omega)$), we denote the MMBM by a general form $p(\mathbf{y}, \mathbf{H}|\mathbf{x}; \omega)$ where $\mathbf{H} = \mathbf{h}$ for one single hidden layer or $\mathbf{H} = \{\mathbf{h}^1, \mathbf{h}^2\}$ for two hidden layers, and $\omega = \{\mathbf{W}^{1,2}, \mathbf{V}^{0,1,2}, \mathbf{c}^{1,2}, \mathbf{b}\}$ are the model parameters. The MMBMs consist of image-independent and image-dependent parts. We first initialize the image-independent part by generative pre-training, and then reformulate the joint learning problem into a max-margin optimization task which is solved effectively by a CCCP algorithm.

4.3.1 Pre-training

Generative pre-training $p(\mathbf{y}, \mathbf{H})$ is of crucial importance for the MMBM models. It provides the MMBM models with proper regularization between output and hidden vari-

Algorithm 2 Stochastic Gradient Descent algorithm for max-margin learning MMBMs.

1: Set $t = 0$, initialize ω_0, α_0 and define γ
2: **while** $t < T$ **do**
3: Randomly select a training instance $(\mathbf{x}_i, \mathbf{y}_i)$
4: Solve (4.16): $\mathbf{H}_i^* \leftarrow \max_{\mathbf{H}}[-E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega_t)]$
5: Solve (4.17): $\hat{y}_i, \hat{\mathbf{H}}_i \leftarrow \max_{\mathbf{y},\mathbf{H}}[-E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega_t) + \Delta(\mathbf{y}, \mathbf{y}_i, \mathbf{H}, \mathbf{H}_i^*)]$
6: Update $\omega_{t+1} \leftarrow (1 - \alpha_t\gamma)\omega_t + \alpha_t(\frac{\partial E(\hat{y}_i, \hat{\mathbf{H}}_i, \mathbf{x}_i; \omega)}{\partial \omega} - \frac{\partial E(\mathbf{y}_i, \mathbf{H}_i^*, \mathbf{x}_i; \omega)}{\partial \omega})$
7: Decrease α_t
8: **end while**

ables, and feed sensible hidden variables to discriminative learning in the following stage. By omitting image-dependent components, the MMBM with one single hidden layer reduces to the RBM while the one with two hidden layers reduces to the ShapeBM. We thus can utilize the generative training algorithms of these methods. Indeed, the general training procedure of BMs requires minimizing the differences between the data-dependent and model-dependent expectations. We train the RBM by minimizing contrastive divergence [41]. For the ShapeBM, each layer is greedily trained.

4.3.2 Max-Margin Learning

To generate accurate prediction on test images, we seek for the parameters ω that assign training labels $\mathbf{y}_i$ a greater than or equal log-likelihood of any other labeling $\mathbf{y}$ for instance i,

$$\log p(\mathbf{y}_i, \mathbf{H}|\mathbf{x}_i; \omega) \geq \log p(\mathbf{y}, \mathbf{H}|\mathbf{x}_i; \omega), \forall \mathbf{H}, \forall \mathbf{y}, \forall i. \tag{4.14}$$

We can cancel the partition function $\mathbf{Z}$ for both sides of (4.14), and express the constraints by energies,

$$-E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega) \geq -E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega), \forall \mathbf{H}, \forall \mathbf{y}, \forall i. \tag{4.15}$$

We refer the left term of (4.15) as data-dependent energy and the right term as model-

dependent energy. Since the number of constraints in (4.15) is exponentially large, we look for the hidden variables $\mathbf{H}_i^*$ that best explain the training instance $(\mathbf{x}_i, \mathbf{y}_i)$ in the data-dependent energy

$$\mathbf{H}_i^* = \arg\max_{\mathbf{H}} - E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega). \tag{4.16}$$

For the model dependent energy, we compute the best prediction from $\mathbf{x}_i$ by augmenting an energy margin $\Delta(\mathbf{y}, \mathbf{y}_i, \mathbf{H}, \mathbf{H}_i^*)$,

$$\{\hat{y}_i, \hat{\mathbf{H}}_i\} = \arg\max_{\mathbf{y},\mathbf{H}} - E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega) + \Delta(\mathbf{y}, \mathbf{y}_i, \mathbf{H}, \mathbf{H}_i^*). \tag{4.17}$$

These two decoding problems (4.16) and (4.17) can be solved efficiently by the ICM algorithm in Algorithm 1 where the only difference is to initialize with random $\mathbf{H}$.

To deal with noisy training image data, we relax the margin constraints by introducing slack variables ξ_i. Thus, we formulate the MMBM learning with the following max-margin objective function,

$$\begin{aligned} &\min_{\omega} \frac{\gamma}{2}\|\omega\|^2 + \sum_i \xi_i, \quad \text{s.t.} \\ -E(\mathbf{y}_i, \mathbf{H}_i^*, \mathbf{x}_i; \omega) \geq &\max_{\mathbf{y},\mathbf{H}} [\Delta(\mathbf{y}_i, \mathbf{y}, \mathbf{H}_i^*, \mathbf{H}) - E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega)] \\ &- \xi_i, \xi_i \geq 0, \forall i, \\ \text{where} \quad \mathbf{H}_i^* = &\arg\max_{\mathbf{H}} - E(\mathbf{y}_i, \mathbf{H}, \mathbf{x}_i; \omega). \end{aligned} \tag{4.18}$$

This formulation is equivalent to minimizing the loss function,

$$\begin{aligned} \min_{\omega} \frac{\gamma}{2}\|\omega\|^2 + \sum_i \max_{\mathbf{y},\mathbf{H}} [&\Delta(\mathbf{y}_i, \mathbf{y}, \mathbf{H}_i^*, \mathbf{H}) - E(\mathbf{y}, \mathbf{H}, \mathbf{x}_i; \omega) + \\ &E(\mathbf{y}_i, \mathbf{H}_i^*, \mathbf{x}_i; \omega)]. \end{aligned} \tag{4.19}$$

To optimize the loss function (4.19), we initialize the parameters ω_0 with pre-trained $\mathbf{W}^{1,2}, \mathbf{c}^{1,2}, \mathbf{b}$ and random matrices $\mathbf{V}^{1,2}$. We develop a stochastic gradient descent algorithm (See Algorithm 2) for optimizing (4.19) by applying the Concave-Convex Procedure [120]. Note that it is easy to compute the gradients of energy functions with respect to ω as both data energy $E(\hat{y}_i, \hat{\mathbf{H}}_i, \mathbf{x}_i; \omega)$ and model energy $E(\mathbf{y}_i, \mathbf{H}_i^*, \mathbf{x}_i; \omega)$ are linear functions of parameters ω given fixed hidden and output variables.

Comparing Margin Functions. Choosing a proper margin penalty function $\Delta(\cdot)$ is crucial to effective learning. Taking the single layer MMBM as an example, we find its energy function consists of three components: hidden-visible interaction (H-V), hidden-image interaction (H-I) and visible-image interaction (V-I), which correspond to the three kinds of edges in the graphical model of MMBMs,

$$E(\mathbf{y}, \mathbf{h}, \mathbf{x}) = -\underbrace{\mathbf{y}^\top \mathbf{W}\mathbf{h}}_{\text{H-V}} - \underbrace{\mathbf{h}^\top(\mathbf{V}^1\mathbf{x}^1 + \mathbf{c})}_{\text{H-I}} - \underbrace{\mathbf{y}^\top(\mathbf{V}^0\mathbf{x}^0 + \mathbf{b})}_{\text{V-I}}, \tag{4.20}$$

We analyze four cases of $\Delta(\cdot)$ and evaluate their performance in the experiments.

Case 1: $\Delta(\cdot) = 0$**.** If we set $\Delta(\cdot) = 0$, then the loss function in (4.19) reduces to the perceptron loss used in [79]. As the data-dependent and model-dependent energies remain the same form, there exist several possibilities that can explain the perceptron loss, considering the potential combinations of three components. For example, learning with $\Delta = 0$ may end up with a strong H-V component but weak H-I and V-I components, as the H-V component is pre-trained. This result is clearly deficient for prediction.

Case 2: $\Delta(\cdot) = \Delta(\mathbf{y}, \mathbf{y}_i)$**.** If we set $\Delta(\cdot) = \Delta(\mathbf{y}, \mathbf{y}_i)$, then the loss function in (4.19) is closely related to the one used in latent Structured SVM [120]. The energy margin $\Delta(\mathbf{y}, \mathbf{y}_i)$ only depends on $\mathbf{y}$ so that the V-I component will be better constrained to dominate the energy loss between the data energy and the augmented model energy. Considering the pre-trained H-V component, we may obtain strong H-V and V-I components but a weak H-I component. However, the H-I and V-I components take different input features and should be complementary to each other. The unoptimized H-I component very likely constrains the model generalizability to unseen data.

Case 3: $\Delta(\cdot) = \Delta(\mathbf{H}, \mathbf{H}_i^*)$**.** If we set $\Delta(\cdot) = \Delta(\mathbf{H}, \mathbf{H}_i^*)$, then the loss function in (4.19) indirectly corresponds to the output through hidden variables. That is, the H-V component functions as clustering. The margin on hidden variables essentially encourages the H-I component to correctly predict the cluster labels $\mathbf{H}_i^*$, i.e., the hidden variables that best explain the training instance $(\mathbf{x}_i, \mathbf{y}_i)$. Thus, the energy difference is likely dominated by the H-I and H-V components, which leaves the V-I component unoptimized. This

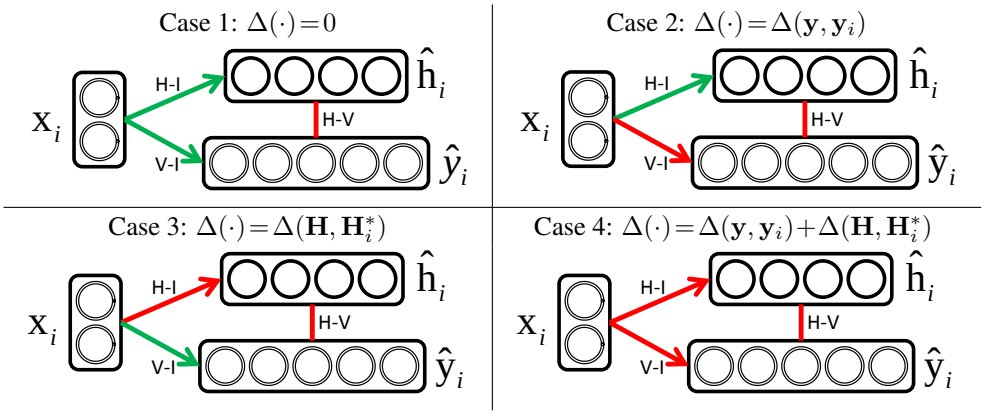


Figure 4.2: Comparing margin functions.

approach has the same generalizability problem as Case 2.

Case 4: $\Delta(\cdot) = \Delta(\mathbf{y},\mathbf{y}_i) + \Delta(\mathbf{H},\mathbf{H}_i^*)$**.** Based on the above analysis, we use $\Delta(\cdot) = \Delta(\mathbf{y},\mathbf{y}_i) + \Delta(\mathbf{H},\mathbf{H}_i^*)$ as the margin penalty function. Since $\Delta(\mathbf{y},\mathbf{y}_i)$ and $\Delta(\mathbf{H},\mathbf{H}_i^*)$ are absorbed into the V-I component and H-I component, respectively, all three components are optimized during learning.

Figure 4.2 illustrates learning single-layer MMBMs with four kinds of margin functions. The particular margin functions induce the red connections between any two layers dominate the energy loss during learning while leaving the green connections unoptimized.

4.4 Experiments

4.4.1 Datasets

Penn-Fudan Pedestrians This dataset [111] consists of 170 images with bounding box annotations and ground truth foreground-background segmentation masks. The images all include one or more pedestrians. For our experiments we extracted 423 patches, each adjusted to include one person only. We resize the patches to an uniform size of 32×64

pixels, cropping the original image so that we can keep the original aspect ratio while resizing them.

In order to increase the number of training and test samples, we subsequently mirror all patches, resulting in 846 samples, some of which with severe occlusions. We then select 400 samples for training and use the rest for tests. The training-test split is done randomly except for keeping original images in the same set as their mirrored pairs.

Weizmann Horses. This dataset [15] contains 328 horse images, with a high variability of poses and scales. Before processing, we resize every image to 128x128, padding images with different aspect ratios with mirrored versions of the image itself. To get comparable results to [71], we calculate 32x32 foreground-background segmentation masks with all of our models. Also, we use their training-test split (into 200 training and 128 test images).

Caltech-UCSD Birds 200. The dataset [112] includes 6033 images of 200 bird species, each image usually including one dominant bird in the scene. The images are annotated with a bounding box and a coarse-grained segmentation mask. As the accuracy of this isn't sufficient to evaluate our segmentation methods, we manually annotate these images with accurate masks (available on the website `https://eng.ucmerced.edu/people/jyang44`). We crop 6033 bird patches and the corresponding segmentation masks from bounding boxes, and resize the image patches to 128×128 pixels. We use the same training/test partition as in [112], i.e., 3000 samples for training and the rest for tests.

4.4.2 Implementations

Architectures. For the MMBM with a single hidden layer (MMBM1) and RBM, we use 500 hidden units $\mathbf{h} \in \{0,1\}^{500}$. For the MMBM with two hidden layers (MMBM2), we use 500 hidden units in the first layer $\mathbf{h}^1 \in \{0,1\}^{500}$, and 200 hidden units in the second layer $\mathbf{h}^2 \in \{0,1\}^{200}$. For the birds and the horses, each mask is partitioned into 2×2 four patches $\{\mathbf{y} = \vee \mathbf{y}_k, k = 1, \ldots, 4, \mathbf{y}_k \in \{0,1\}^{36 \times 36}\}$ with 8 pixels overlapping between adjacent patches, such that each part is connected to 125 hidden units in the first layer $\mathbf{h}^1$.

For the pedestrians, we also use four patches $\{\mathbf{y} = \vee \mathbf{y}_k, k = 1, \ldots, 4, \mathbf{y}_k \in \{0,1\}^{22\times 32}\}$ but in a 4x1, vertical organization with 14 pixel overlaps between neighbors.

Features. One of the advantages of the proposed method is that it can handle a diverse set of features: local descriptors can be connected to the visible layer while features covering larger image areas are better suited as conditionals for one of the hidden layers.

For MMBM1, we use two sets of features: $\mathbf{x}^0$ for the visible and $\mathbf{x}^1$ for the hidden layer. For $\mathbf{x}^0$, we first segment the image into superpixels using the gPb algorithm [7]. For each superpixel, we compute dense SIFT, color and contour histograms. The histograms of densely sampled SIFT words are computed by using a codebook of size 512 and the locality-constrained linear coding method [109]. The color histograms of RGB values are computed from a codebook of size 128, and finally, the contour histograms are computed from the oriented gPb edge detector responses [7]. For per-pixel visible features we simply use those of the superpixel containing the pixel in question.

For the hidden layers of MMBM1, we use the HOG descriptors for the entire input image as $\mathbf{x}^1$. For MMBM2, the features $\mathbf{x}^2$ for the top layer is calculated the same way, while for the middle layer feature vector $\mathbf{x}^1$ we use the HOG descriptors for the four patches.

Training. For the MMBM1, we run 2000 epochs with 100-sample mini-batches in the generative training phase (RBM training). For the MMBM2, we run 2000 epochs for the first layer pre-training in the generative training phase (ShapeBM training) and 1000 epochs for the second layer pre-training. In addition, we run 5 cycles in the max-margin training phase in both cases. We set the learning rate $\alpha_0 = 0.001$ and the constant $\gamma = 0.01$ for all the experiments. The MATLAB source code and the labeled datasets will be made available for research purposes.

Baseline. We study two discriminative models for comparison: a superpixel based CRF model using bottom-level features $\mathbf{x}^0$ and Compositional High Order Potentials (CHOPPs) [71]. For the CRF model $p(\mathbf{y}|\mathbf{x}^0)$, we use the implementation in [71].

For CHOPPs, we used the code provided by the authors for the inference but we didn't

get the same results, likely due to differences in our unary / pairwise potential generation code. To make the comparison fair, in the experiments we used the same unary features as in our MMBM implementation instead. As Table 5.2 shows, this improved their results compared to the original published in [71].

Since, unlike the combined RBM-CRF models of [71] and [47], our model doesn't have pairwise weights in the visible layer. For a better comparison with these models, we also ran Graph Cut on the output mask, using the probabilities given by the model as unary potentials and a pairwise term taken from [17], based on the magnitude of the gradients of color channels. We report results for both the original and refined masks.

4.4.3 Results

We use two metrics for performance evaluation: the average pixel accuracy (AP) of foreground and background classification and the foreground intersection-over-union score (IoU) of entire test set [1]. We first present segmentation results on the Penn-Fudan Pedestrians in Table 4.1. Overall, the MMBM1 (76.92% IoU, Case 4) and MMBM2 (77.30% IoU, Case 4) outperform the CRF (68.35% IoU) and CHOPPs (71.33% IoU) algorithms. The results show that the MMBMs are effective models for object segmentation by integrating image features and a strong shape prior. Also, as the last two rows of the table indicate, introducing pairwise constraints further improves results.

Our results for Weizmann horses are shown in Table 5.2. Again, both the advantage of augmenting the loss function with multiple margins and the benefits of using a two-layered architecture are demonstrated. Also, comparisons using different margin functions (Cases 1-4) for the MMBM1 model demonstrate the importance of a max-margin formulation with multiple margins for output prediction. By using margin functions (MMBM1 Cases 2-4), we obtain 19% AP improvement and more than 30% IoU improvement over the

[1] The IoU score is defined as $\frac{|\mathbf{Y} \bigcap \hat{\mathbf{Y}}|}{|\mathbf{Y} \bigcup \hat{\mathbf{Y}}|}$, where $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ are the sets of ground truth and predicted foreground pixels.

Table 4.1: Quantitative results on the Penn-Fudan Pedestrians dataset.

		AP	IoU
CRF		84.87	68.35
CHOPPs [71]		86.55	71.33
MMBM1	Case 1	82.66	64.80
	Case 2	85.27	69.20
	Case 3	83.35	65.78
	Case 4	89.91	76.92
MMBM2		89.74	77.30
MMBM1 Case 4 w/ GC		90.42	77.97
MMBM2 Case 4 w/ GC		**90.77**	**79.42**

non-margin (perceptron loss) algorithm in Case 1 of the MMBM1. The best results for the MMBM1 (89.43% AP, 69.59% IoU) from Case 4 indicate that the combining multiple margin functions $\Delta(\cdot) = \Delta(\mathbf{y}, \mathbf{y}_i) + \Delta(\mathbf{H}, \mathbf{H}_i^*)$ alleviates degenerating effects by providing stronger constraints. Our results on other datasets also strengthen this observation. The two-layer hierarchical hidden architecture also helps generating better results than a single hidden layer, as shown in the Case 4 of MMBM2 (89.80%AP, 72.09% IoU) over MMBM1.

In addition to the comparison to CRF and CHOPPs, for this dataset we also added the results from [14]. Their aim was to identify body parts and got the foreground-background segmentation as a byproduct.

Finally, the segmentation results on the Caltech-UCSD Birds 200 dataset are presented in Table 4.3. Different from pedestrians and horses, this dataset has large shape variations but more distinct appearances (e.g., color, textures). Thus, the appearance-based CRF model performs less well. Similar is the case of CHOPP: as its hidden nodes are not directly connected to image features, so they can only refine and correct the shape of results that are mostly right just based on local, visible-layer features, which is hard to accomplish on this dataset. In contrast, the features-to-hidden connections in the MMBM models make it possible to exploit global shape information even without reliable local

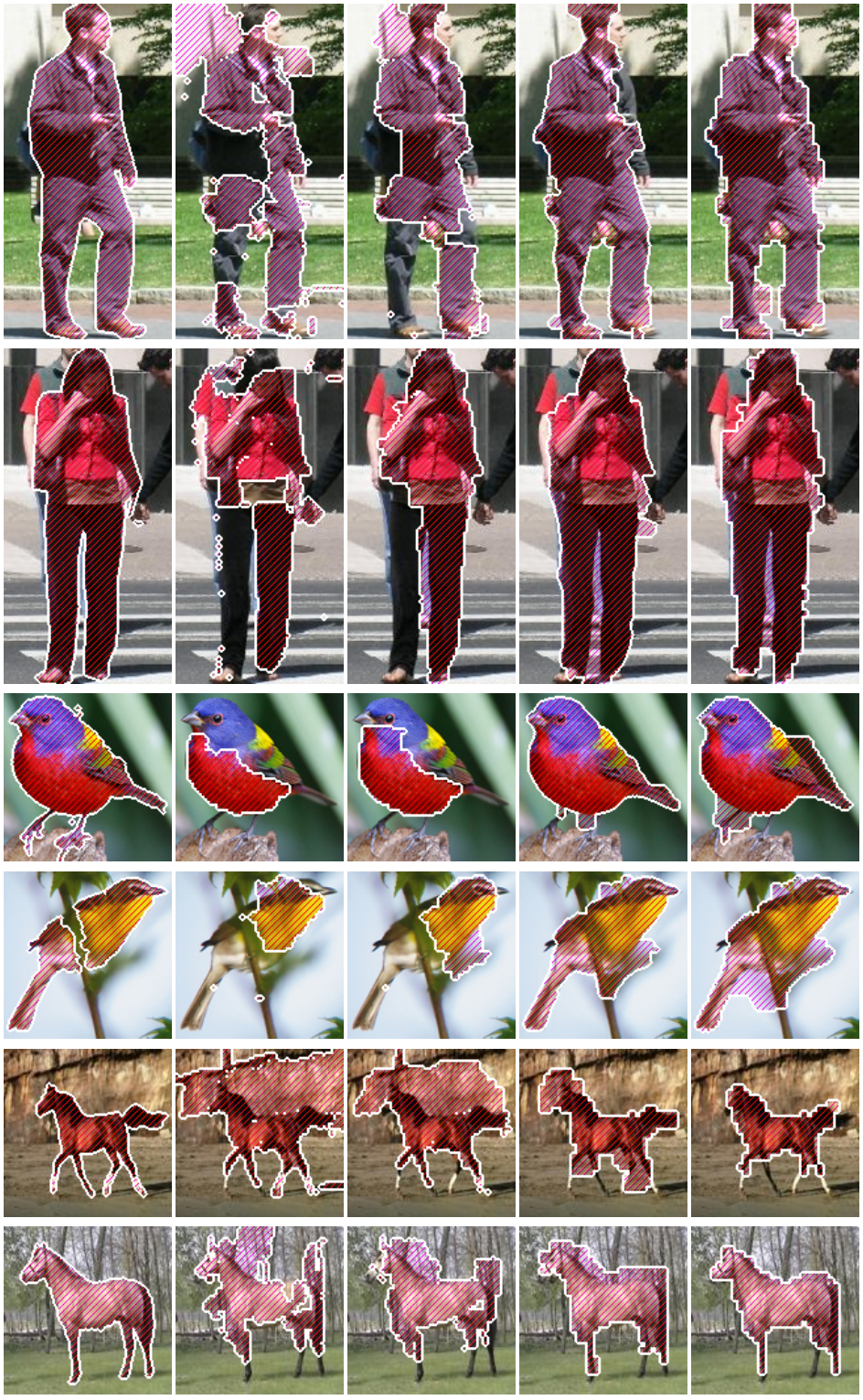
Table 4.2: Quantitative results on the Weizmann Horses dataset.

		AP	IoU
CRF		87.46	67.44
Bo and Fowlkes [14]		77.2	N/A
CHOPPs [71]		88.67	71.60 (69.90 in [71])
MMBM1	Case 1	70.59	38.01
	Case 2	85.87	62.97
	Case 3	85.37	59.35
	Case 4	89.43	69.59
MMBM2		89.80	72.09
MMBM1 Case 4 w/ GC		90.62	74.12
MMBM2 Case 4 w/ GC		**90.71**	**75.78**

Table 4.3: Quantitative results on the Caltech-UCSD Birds 200 dataset.

		AP	IoU
CRF		83.50	38.45
CHOPPs [71]		74.52	48.84
MMBM1	Case 1	80.96	60.37
	Case 2	87.73	72.45
	Case 3	75.73	63.22
	Case 4	88.07	72.96
MMBM2		86.38	69.87
MMBM1 Case 4 w/ GC		**90.42**	**75.92**
MMBM2 Case 4 w/ GC		90.77	72.40

features. The results, similar to the observations on the other two datasets for evaluating different margin functions demonstrate the significance of max-margin formulation and combining margin functions (Case 4). In the bird data, we observe better performance by using just one hidden layer compared to using the two-layered MMBM2 model. A possible reason is that while the weight replication for the four windows in MMBM2 is beneficial when given a small number of training samples (such as for horses and pedestrians), but for larger datasets we can learn a better prior using simple architectures (RBMs) with more parameters from the data.



(a) GT (b) CRF (c) CHOPP (d) MMBM1 (e) MMBM2

Figure 4.3: Qualitative results on the Penn-Fudan Pedestrians, Caltech-UCSD Birds 200 and Weizmann horse datasets.

We present some qualitative results in Figure 4.3, from which we can see more directly the importance of features-to-hidden connections for shape prediction. For example, CRF finds the most colorful parts of birds, which is corrected by CHOPP to be shaped more birdlike, but it's only MMBMs that discover the entire bird well.

4.5 Summary and Future Work

In this chapter, we propose MMBMs for structured output prediction problems and investigate two variants of MMBMs with single and two hidden layers for object segmentation. Instead of using BMs as shape priors, we build connections between input observations with hidden variables that opens an inference pathway from image features to object shapes. We derive a simple yet efficient ICM algorithm for MAP inference. We formulate MMBMs with a max-margin objective function for discriminative training, and discuss four margin functions as well as their effects on learning performance. The results on horses, pedestrians, birds datasets show that our algorithms perform favorably against the state-of-the-art methods.

In experiments, we have found that the pairwise edge potentials can after all improve the segmentation quality, given the predicted shapes from our models. In the future, we plan to extend MMBM models by adding pairwise potentials to the visible layer. Considering the alternating procedure of the MAP inference algorithm, this extension will not significantly increase the complexity of inference and learning because we only need to replace Line 4 in Algorithm 1 with Graph Cut. We are also interested in integrating object detection with segmentation in MMBM models.

Chapter 5

Local Shape Transfer for Generic Object Segmentation

5.1 Introduction

Object segmentation, separating a foreground object from its background with a clear boundary, has long been an important challenge for computer vision. It not only provides mid-level representations for high-level recognition tasks [20] but also has immediate applications to image editing [2]. Object segmentation is typically formulated as a binary labeling problem on Markov Random Fields (MRFs) with foreground/background appearance models [17].

Recent methods [87, 65] show that object segmentation can be solved efficiently with a carefully prepared bounding box around the target and further refined by user inputs. In these interactive algorithms [17, 87, 65], color is commonly used to separate foreground from background. Although more complex image cues such as textures are shown to be useful to improve segmentation performance [113], a critical source of information, object shape, is clearly missing in these algorithms. Similar situations exist in salient object segmentation [83, 76, 24] in that most of algorithms work well when the images have high

foreground-background color contrast, but work poorly in cluttered images. A notable exception is Li et al.'s latest work [70] that achieves impressive object segmentation results on the PASCAL images [31] by integrating shape sensitive object proposals [20] with a classic saliency map [40]. On the other hand, in model-based algorithms [15, 63, 58, 11, 117], shape is always the major driving force for segmentation. Category-specific shape models are usually designed [58] based on prior knowledge or learned offline from training data [67, 117]. This category-specific nature limits the generalizability of model based algorithms to handle unseen objects.

In this chapter, we propose a data-driven object segmentation algorithm that addresses the problems mentioned above by using a database of existing segmentation examples. Our algorithm requires neither offline training of category-specific shape models nor prior knowledge of object shapes. Instead we transfer shape masks from similar segmentation examples to the test images by image retrieval and matching. Compared to user- and saliency-driven algorithms, the transferred shape cues help resolve segmentation ambiguities from appearance models.

Existing data-driven object segmentation algorithms [57, 3, 49, 100] mostly focus on transferring entire shape masks by either window based or local features based image matching. In this chapter, however, we investigate a patch-level local shape transfer scheme that finds candidate local shape masks for each patch of a test image in multiple scales through dense correspondences between query and example images built by the PatchMatch algorithm [8]. Those candidate local shape masks indeed constitute an online structured label space where object segmentation solutions can be found. We thereby develop a novel cascade algorithm for coarse-to-fine object segmentation. At each stage, we define a color based MRF energy function with the coarse shape mask estimated in the previous stage, and select the local shape mask for each patch independently with the minimum MRF energy to estimate a new shape mask with finer details. This patch-wise segmentation provides an approximate solution to global energy minimization, but a solution which is easier to solve in parallel. We carry out local shape mask selection iteratively

while updating the foreground/background color models. This iterative procedure shares a similar idea with GrabCut [87], but it operates patch-wise in a structured label space. Thus we name our method *PatchCut*.

Our proposed idea of using local masks as solution space is inspired by recent structured forest based image labeling algorithms [55, 26]. In the training stage, the clustering structure of label patches is exploited in branching functions so that each leaf node stores one example label patch. The label patches in all the leaf nodes constitute a structured label space for edge detection [26] and semantic labeling [55]. In our algorithm, the local shape masks transferred from examples constitute another kind of structured label space for object segmentation. In spirit, both structured forests and our algorithm aim at preserving output structures (local context and shape) when making predictions. However, an important difference is that the structured label space in our algorithm is constructed online by matching with examples, which is more flexible and easier to generalize than offline training in structured forests [55, 26]. We carry out experiments on various object segmentation benchmark datasets with comparisons to leading example-, learning- and saliency-based algorithms.

The contributions of this chapter are summarized below:

- a novel nonparametric high-order MRF model via patch-level label transfer for object segmentation;
- an efficient iterative algorithm (PatchCut) that solves the proposed MRF energy function in patch-level without using graph cuts;
- state-of-the-art performance on various object segmentation benchmark datasets.

5.2 Our Algorithm

Given a test image $\mathbf{I}$, our goal is to estimate its segmentation $\hat{\mathbf{Y}}$ by using example images $\{\mathbf{I}_m, m = 1, 2, ..., M\}$ and their segmentation ground truth $\{\mathbf{Y}_m, m = 1, 2, ..., M\}$. Figure 5.1 presents an overview of the proposed algorithm.

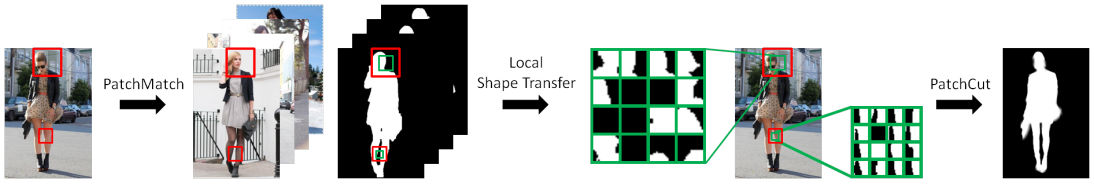


Figure 5.1: Overview of proposed object segmentation algorithm using examples.

5.2.1 Local Shape Transfer

Our algorithm performs image matching to achieve shape transfer from examples like most data-driven algorithms. However, transferring entire masks in large image windows may result in poor boundary quality [57], while alignment, although improving the boundary quality, significantly increases the computational cost [3]. In this work, we propose transferring local shape masks from multiple scales. We build three-layer image pyramids

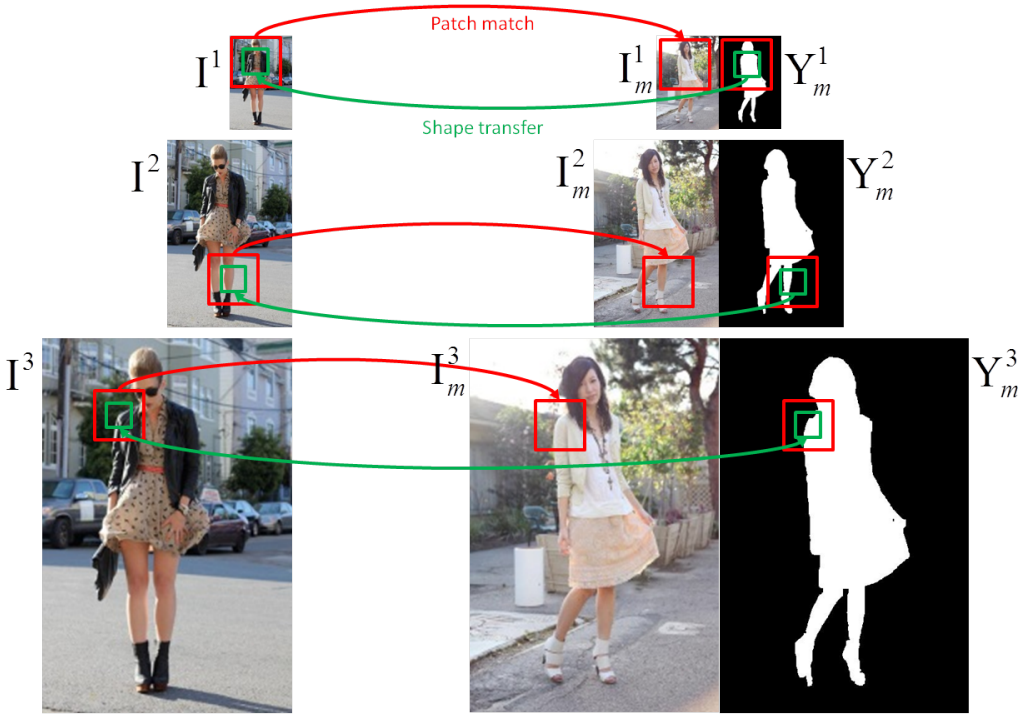


Figure 5.2: Local shape transfer with multiscale PatchMatch.

by downsampling both the test $\{\mathbf{I}^s, s=1,2,3\}$ and example images $\{\mathbf{I}^s_m, \mathbf{Y}^s_m, s=1,2,3\}$. If the size of image $\mathbf{I}$ is $[h, w]$, the size of downsampled image in the s^{th} layer is $[\frac{h}{2^{3-s}}, \frac{w}{2^{3-s}}]$. For all three scales, we use image patches of the same size to perform matching and mask transfer as demonstrated in Figure 5.2. In each scale $(s = 1, 2, 3)$, we densely sample image patches of 16×16 at every 2 pixels $\{\Delta^s_k, k = 1, 2, ..., K\}$, where $K = \frac{h \times w}{4 \times 2^{6-2s}}$ [1]. For each patch of the test image Δ^s_k (green boxes in Figure 5.2), we extract a SIFT descriptor $\mathbf{x}^s_k$ from its extended 32×32 patch (red boxes in Figure 5.2). Therefore, the matching problem between the test $\mathbf{I}$ and the m^{th} example $\mathbf{I}_m$ can be described by $\arg\min_{k'} \|\mathbf{x}^s_k - \mathbf{x}^s_{k'm}\|_1, \forall k = 1, 2, ..., K$, where $\mathbf{x}^s_{k'm}$ is the SIFT descriptor extracted from the k'^{th} patch $\Delta^s_{k'}$ of the m^{th} example. This nearest neighbor field problem is solved efficiently by the PatchMatch algorithm [8]. As a result, the test patch Δ^s_k finds its match $\Delta^s_{k^*}$ in the m^{th} example with the cost $d^s_{km} = \|\mathbf{x}^s_k - \mathbf{x}^s_{k^*m}\|_1$.

We denote the local segmentation masks from the matched patches in m^{th} example as $\mathbf{z}^s_{km} = \mathbf{Y}^s_m(\Delta^s_{k^*})$, which provide location and shape information for segmenting the test image. We argue that those local masks $\mathbf{z}^s_{km}$ constitute a patch-wise segmentation solution space for the test image; in other words, the segmentation mask of test image $\mathbf{Y}$ can be well approximated by $\mathbf{z}^s_{km}$. Note that different methods for image dense correspondences have been explored in [73, 50] to enable pixel-wise label transfer, but their results are either constrained by the local flow [73] or contaminated by relaxation noise [50]. Compared to [73, 50], our method achieves structured label transfer (local masks) through a more flexible matching algorithm.

To examine the quality of local shape masks $\mathbf{z}^s_{km}$, for each patch Δ^s_k we calculate the mean of its local masks $\bar{\mathbf{z}}^s_k = \frac{1}{M}\sum_m \mathbf{z}^s_{km}$, and also find the best possible $\tilde{\mathbf{z}}^s_k$ using the ground truth as reference. Note that $\tilde{\mathbf{z}}^s_k$ actually defines the upper bound for local shape transfer. Obviously, the mean shape prior mask $\bar{\mathbf{Q}}^s$ can be immediately estimated by adding up $\bar{\mathbf{z}}^s_k$ similar to [57]. Similarly, we can estimate the oracle shape prior mask $\tilde{\mathbf{Q}}^s$ from $\tilde{\mathbf{z}}^s_k$.

[1] Padding is needed to ensure $[\frac{h}{2^{3-s}}, \frac{w}{2^{3-s}}]$ divisible by 2.

Figure 5.3 demonstrates the mean and oracle shape prior masks of different scales. The masks are upsampled to the size of original image for better visualization. At the

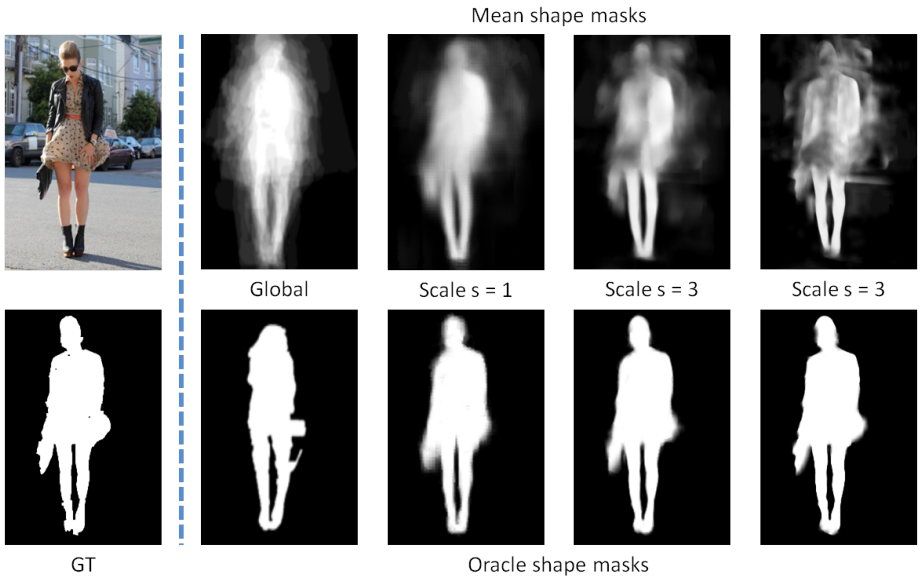


Figure 5.3: Shape prior masks estimated from mean masks (top row) and best masks (bottom row) at different scales.

coarse scale, the object is well located but its boundary is blurry in the mean shape prior masks. Moving towards finer scales, although some parts of mean shape prior (legs) become clearer, other parts (head and shoulder) turn out to be very noisy. This is because the very local structures of image patches at the finer scales preserve well the edge patterns during matching, but local masks may have inconsistent foreground/background relationship. Meanwhile, both location and boundary qualities of oracle shape prior masks keep getting better from coarse to fine scales. This divergent result indicates that good segmentation solutions can be obtained if we find the right label patches at a fine scale, but without that knowledge, the average results are far from satisfactory. The above observations motivate the coarse-to-fine strategy where we start with a good approximation at the coarse scale which then leads us to choose the right label patches at the fine scale.

5.2.2 PatchCut

In this section, we introduce a novel algorithm to gradually estimate the shape prior $\hat{\mathbf{Q}}^s$ in a coarse-to-fine manner. In particular, at the s^{th} scale, given the shape prior from the previous scale $\hat{\mathbf{Q}}^{s-1}$, the finer shape prior $\hat{\mathbf{Q}}^s$ is estimated using candidate local shape masks $\mathbf{z}_{km}^s$. At the end, the binary segmentation $\hat{\mathbf{Y}}$ can be computed by thresholding the shape prior at the finest scale.

MRF with shape prior. We start with reviewing a typical object segmentation method based on shape prior, which provides the fundamentals for our algorithm. Note that we temporarily omit the scale index s to keep the description clean. Object segmentation with shape prior is commonly formulated as a MRF energy function [57, 3],

$$E(\mathbf{Y}) = \sum_{i \in \mathcal{V}} U(y_i) + \gamma \sum_{i,j \in \mathcal{E}} V(y_i, y_j) + \lambda \sum_{i \in \mathcal{V}} S(y_i, q_i). \tag{5.1}$$

where y_i is the binary label at pixel i, q_i is the probability at pixel i of shape prior $\mathbf{Q}$. The unary term for each pixel $U(y_i)$ is the negative log probability of the label y_i given the pixel color $\mathbf{c}_i$ and Gaussian Mixture Models (GMMs) $\mathbf{A}_1$ and $\mathbf{A}_0$ for foreground and background color,

$$U(y_i) = -\log P(y_i | \mathbf{c}_i, \mathbf{A}_1, \mathbf{A}_0). \tag{5.2}$$

The pairwise term $V(y_i, y_j)$ measures the cost of assigning different labels to two adjacent pixels, which is usually based on their color difference,

$$V(y_i, y_j) = \exp(-\alpha \|\mathbf{c}_i - \mathbf{c}_j\|^2) \mathbb{I}(y_i \neq y_j), \tag{5.3}$$

where the parameter α is estimated by the mean color difference over the image and $\mathbb{I}(\cdot)$ is an indicator function. The shape term $S(y_i, q_i)$ measures the inconsistency with shape prior $\mathbf{Q}$,

$$S(y_i, q_i) = -\log q_i^{y_i} (1 - q_i)^{1 - y_i}. \tag{5.4}$$

This energy function can be solved by alternating two steps in a way similar to the GrabCut algorithm [87]: 1) updating GMM color models in (5.2) from the current segmentation

$\{\mathbf{A}_1, \mathbf{A}_0\} \leftarrow \mathbf{Y}$; 2) solving the MRF energy function in (6.3) with updated color models by GraphCut: $\mathbf{Y} \leftarrow \{\mathbf{A}_1, \mathbf{A}_0\}$. However, this method is too sensitive to the parameter λ. On one hand, if the λ is large, the color models cannot correct the mistakes in the shape prior; on the other hand, if the λ is small, the segmentation may deviate from the good shape prior.

High order MRF with local shape transfer To use candidate local shape masks to resolve segmentation ambiguities, we can naturally extend (6.3) to include a patch likelihood $P_{\text{cand}}(\mathbf{Y}(\Delta_k))$ that encourages the label patch $\mathbf{Y}(\Delta_k)$ for image patch $\mathbf{I}(\Delta_k)$ to be similar to some candidate local shape mask $\mathbf{z}_{km} = \mathbf{Y}_m(\Delta_{km})$ for database image patch $\mathbf{I}_m(\Delta_{km})$:

$$E'(\mathbf{Y}) = E(\mathbf{Y}) - \sum_k \log(P_{\text{cand}}(\mathbf{Y}(\Delta_k))). \tag{5.5}$$

The last term is the negative Expected Patch Log Likelihood (EPLL), a formulation that Zoran and Weiss [121] use for image patches to produce state-of-the-art results on inverse problems such as deblurring. Here we define the patch likelihood on local shape masks by marginalizing out over a hidden variable m_k^* that indicates which database patch Δ_{km} is selected for transfer to the output patch $\mathbf{Y}(\Delta_k)$:

$$\begin{aligned} P_{\text{cand}}(\mathbf{Y}(\Delta_k)) &= \sum_{m=1}^{M} P(\mathbf{Y}(\Delta_k), m_k^* = m) \\ &= \sum_{m=1}^{M} P(\mathbf{Y}(\Delta_k) | m_k^* = m) P(m_k^* = m) \\ &= \sum_{m=1}^{M} \frac{\exp(-\eta ||\mathbf{Y}(\Delta_k) - \mathbf{z}_{km}||_2^2)}{Z_1} \frac{\exp(-\tau d_{km})}{Z_2}, \end{aligned}$$

where the second term expresses the probability by image appearance that we want to transfer the m^{th} candidate label patch and the first term expresses that the output label patch should be similar to the transferred patch. Note that Z_1, Z_2 are normalization terms, and d_{km} is the match cost introduced in the previous section. We assume that η is large

to encourage the output label patches $\mathbf{Y}(\Delta_k)$ to be as similar to the selected candidate patches $\mathbf{z}_{km_k^*}$ as possible. For large η and distinct $\mathbf{z}_{km}$, we have

$$P_{\text{cand}}(\mathbf{Y}(\Delta_k)) \approx \begin{cases} \exp(-\tau d_{km})/Z_2 & \text{if } \mathbf{Y}(\Delta_k) = \mathbf{z}_{km} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

and

$$E'(\mathbf{Y}) \approx E(\mathbf{Y}) + \tau \sum_k H(\mathbf{Y}(\Delta_k)), \tag{5.7}$$

where

$$H(\mathbf{Y}(\Delta_k)) = \begin{cases} d_{km} & \text{if } \mathbf{Y}(\Delta_k) = \mathbf{z}_{km} \\ \infty & \text{otherwise} \end{cases} . \tag{5.8}$$

Note that this approximate energy (5.7) is related to the Non-parametric Higher-order Random Field (NHRF) introduced in [77] that considers top-down local appearance matching (5.8) but not the bottom-up global image cues (6.3).

Approximate optimization on patches. The high order term $H(\mathbf{Y}(\Delta_k))$ actually enforces label patch selection among all the local shape masks $\mathbf{z}_{km}$, and thus the solutions of energy function $E'(\mathbf{Y})$ do not exist when selected label patches disagree in any overlapping areas. To address this issue, we introduce an auxiliary variable $\mathbf{z}_k$ to indicate the selected label patch $\mathbf{z}_k \in \{\mathbf{z}_{k1}, \mathbf{z}_{k2}, ..., \mathbf{z}_{kM}\}$ on the k^{th} patch Δ_k and thus rewrite the energy (5.7) by

$$E'(\mathbf{Y}, \{\mathbf{z}_k\}) \approx E(\mathbf{Y}) + \tau \sum_k H(\mathbf{z}_k), \text{s.t. } \mathbf{Y}(\Delta_k) = \mathbf{z}_k.$$

We notice that the energy $E(\mathbf{Y})$ can be further decomposed into a summation of local energies on $\mathbf{Y}(\Delta_k) = \mathbf{z}_k$

$$E'(\mathbf{Y}, \{\mathbf{z}_k\}) \approx \kappa \sum_k E(\mathbf{z}_k) + \tau \sum_k H(\mathbf{z}_k), \text{s.t. } \mathbf{Y}(\Delta_k) = \mathbf{z}_k,$$

where the constant κ is inversely proportional to the number of patches superimposing on a single pixel. To tolerate the inconsistency between $\mathbf{Y}(\Delta_k)$ and $\mathbf{z}_k$, we convert it into an

unconstrained problem by introducing a quadratic penalty on each patch

$$E'(\mathbf{Y}, \{\mathbf{z}_k\}) \approx \sum_k (\kappa E(\mathbf{z}_k) + \tau H(\mathbf{z}_k) + \frac{\beta}{2} \|\mathbf{Y}(\Delta_k) - \mathbf{z}_k\|^2). \tag{5.9}$$

In a similar spirit with the dual decomposition method [108], this quadratic penalty energy function (5.9) can be minimized by alternatively solving a series of independent slave problems on patch $\mathbf{z}_k$ and a master problem on $\mathbf{Y}$. However, for sufficiently large β, we can approximately solve (5.9) by a simple two-step minimization:

$$\hat{\mathbf{z}}_k = \arg\min_{\mathbf{z}_k} \kappa E(\mathbf{z}_k) + \tau H(\mathbf{z}_k), \forall k \tag{5.10}$$

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \sum_k \frac{1}{2} \|\mathbf{Y}(\Delta_k) - \hat{\mathbf{z}}_k\|^2. \tag{5.11}$$

Note that (5.10) can be immediately solved by evaluating the energies of all the local mask candidates in parallel. To solve (5.11), we need to consider the inconsistency of overlapping $\hat{\mathbf{z}}_k$. By introducing a soft segmentation mask (shape prior) $0 \leq \mathbf{Q} \leq 1$ for $\mathbf{Y}$, we first solve $\hat{\mathbf{Q}} = \arg\min_{\mathbf{Q}} \sum_k \|\mathbf{Q}(\Delta_k) - \hat{\mathbf{z}}_k\|^2$ by averaging all the selected candidates $\hat{\mathbf{z}}_k$ and then compute $\hat{\mathbf{Y}}$ by thresholding $\hat{\mathbf{Q}}$ at 0.5.

Algorithm 3 The single scale PatchCut algorithm.

1: **while** not converged **do**
2: for each patch Δ_k, select the candidate local shape mask $\hat{\mathbf{z}}_k$ by (5.10)
3: estimate the shape prior $\hat{\mathbf{Q}}$ by averaging $\hat{\mathbf{z}}_k$ and the segmentation $\hat{\mathbf{Y}}$ by (5.11)
4: update the foreground and background GMM color models $\{\mathbf{A}_1, \mathbf{A}_0\}$ by (5.2).
5: **end while**

Given the current segmentation $\hat{\mathbf{Y}}$, we further update the color models $\{\mathbf{A}_1, \mathbf{A}_0\}$ in (5.2). Iteratively, the high-order MRF energy (5.7) is minimized using local shape mask candidates. We summarize this procedure, dubbed as *PatchCut*, in Algorithm 3.

Cascade. Using the PatchCut algorithm at a single scale, we assemble the cascade object segmentation algorithm in Figure 5.4. The cascade is initialized by averaging the global

shape masks transferred from examples at the coarsest scale $\hat{\mathbf{Q}}^0 = \frac{1}{M}\sum_m \mathbf{Y}^1_m$. Note that other soft segmentation methods can also be used for initialization [40, 70]. At each scale, we run Algorithm 3 with the previously estimated shape prior $\hat{\mathbf{Q}}^{s-1}$, color models $\mathbf{A}_1, \mathbf{A}_0$ and candidate local shape masks $\mathbf{z}^s_{km}$. The algorithm proceeds untill the scale $s = 3$ is reached. The final object segmentation is inferred by thresholding the shape prior $\hat{\mathbf{Q}}^3$, denoted as $\hat{\mathbf{Y}}_t$, or further refined by iterative graph cuts in (6.3), denoted as $\hat{\mathbf{Y}}_r$.

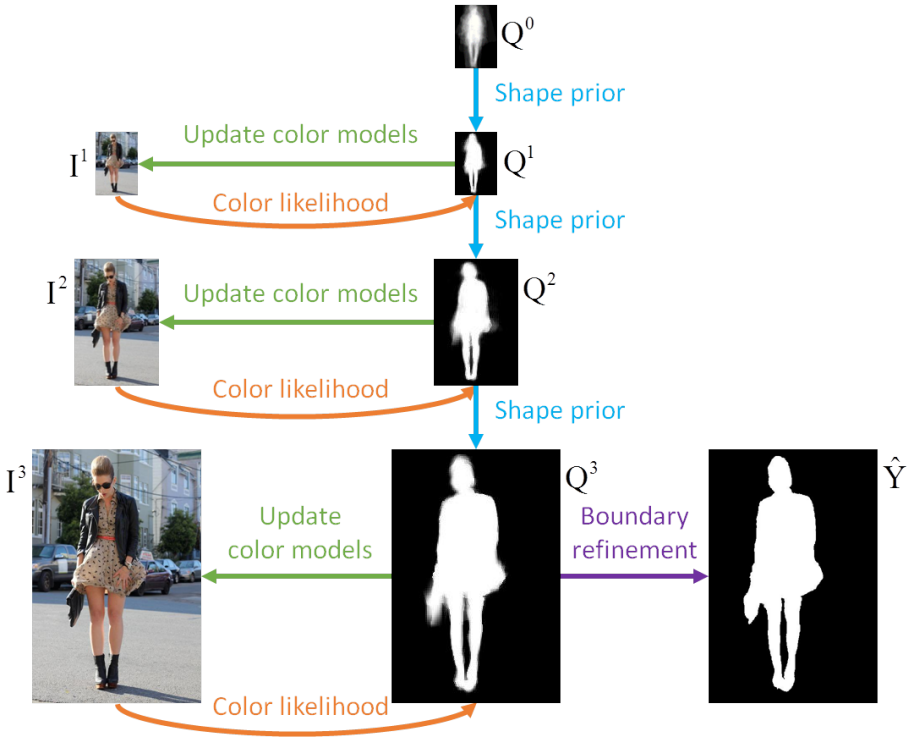


Figure 5.4: PatchCut cascade for coarse-to-fine object segmentation.

5.3 Experimental Results

We present experimental results on various object segmentation datasets (Fashionista [116], Weizmann Horse [15], Object Discovery [88], and PASCAL [31]). More results can be found in our website `https://eng.ucmerced.edu/people/jyang44`.

The term *PatchCut_soft* denotes the shape mask $\hat{\mathbf{Q}}^3$, *PatchCut_thres* denotes the binary segmentation after thresholding $\hat{\mathbf{Y}}_t$, and *PatchCut* denotes the binary segmentation after refinement $\hat{\mathbf{Y}}_r$.

Implementation Details. To perform our algorithm, we retrieve relevant examples from a database of existing segmentations. We use Bag-of-Words (BoW) features [92] on category-specific datasets such as Fashionista, Weizmann Horse and Object Discovery, and use image features extracted from the 7^{th} layer of convolutional networks (ConvNet [2]) [44] on the PASCAL dataset for nearest neighbor image search. The number of retrievals is set to $M = 16$. We set $\gamma = 0.5$ for the pairwise term, $\lambda = 0.5$ for the shape term in (6.3), $\tau = 1.0$ for the match cost term in (5.7), the number of Gaussian components to 5 for both foreground and background GMM color models in (5.2). We use the same set of parameters in all the experiments.

5.3.1 Fashionista

This dataset [116] consists of 700 street shots of fashion models with various poses, cluttered background and complex appearance. All the images have the same size of 600x400 pixels. We run *leave-one-out* tests on this dataset, which means that for each image, we run object segmentation by using the remaining 699 images as the database. We present some segmentation results in Figure 5.5. In this experiment, we compare our algorithm with the widely used GrabCut baseline [87]. We use the OpenCV implementation for GrabCut by providing a bounding box with 8 pixels from the image borders at each side. We evaluate the object segmentation performance by mean Jaccard (Intersection-Over-Union) score ($|\hat{\mathbf{Y}} \cap \mathbf{Y}|/|\hat{\mathbf{Y}} \cup \mathbf{Y}|$) in Table 5.1. By simply thresholding the estimated shape masks, the PatchCut algorithm significantly outperforms (by more than 20%) the GrabCut baseline, and the results can be further improved by GrabCut refinement from 86.25% to 88.33%. Figure 5.5 shows that the refinement mainly occurs around the object

[2]The ConvNet is pre-trained on the ImageNet dataset [56].

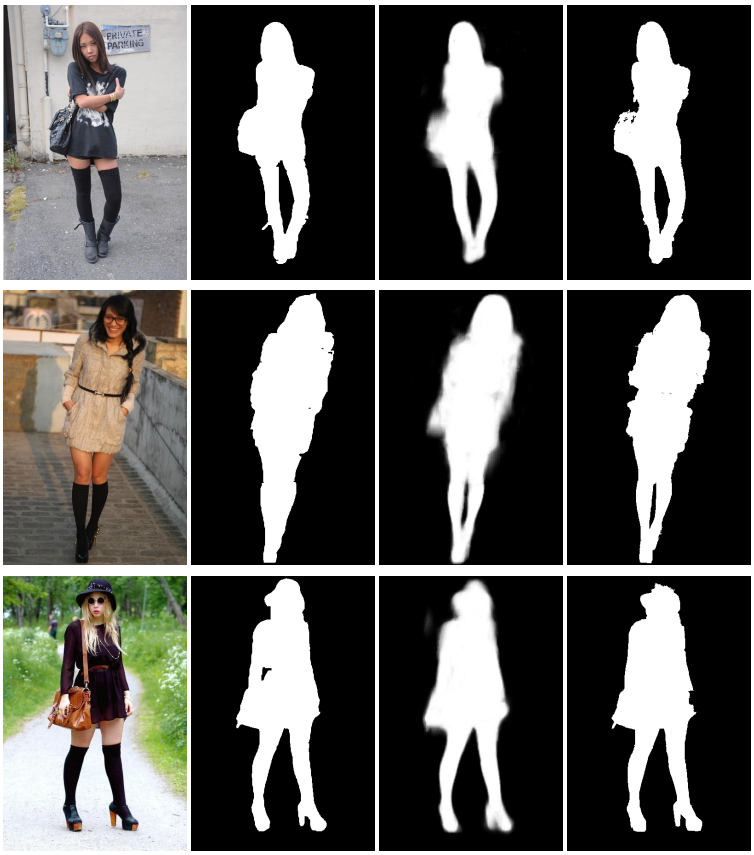


Figure 5.5: Qualitative results on Fashionista.

Table 5.1: Segmentation performance on Fashionista.

	Jaccard (%)
GrabCut	64.23
PatchCut_thres	**86.25**
PatchCut	**88.33**
PatchCut_thres upper bound	95.72
PatchCut upper bound	95.20

contours. To take a closer look at the PatchCut performance, we calculate the segmentation success rate as the percentage of tests that achieve Jaccard scores above certain levels. Figure 5.6 shows that about 58% of tests achieve more than 90% Jaccard score while about 22% of tests achieve more than 95% Jaccard score.

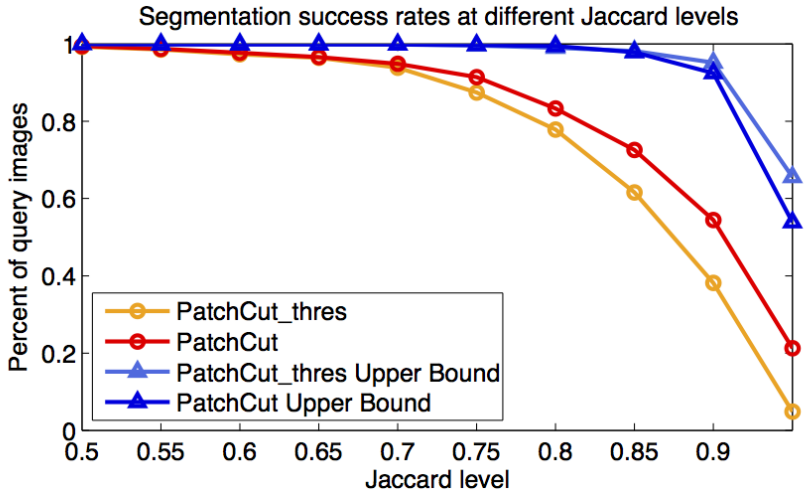


Figure 5.6: Segmentation success rates on Fashionista.

Upper bound performance. We also evaluate the upper bound performance of the PatchCut algorithm. For each test image, we estimate the oracle shape prior masks $\tilde{\mathbf{Q}}$ using the ground truth segmentation $\mathbf{Y}$, and thus produce segmentation results. The mean Jaccard scores from the upper bound segmentation results are as high as 95.72% without refinement (Table 5.1) and the segmentation success rate at Jaccard score level 95% is near 70% (Figure 5.6). These upper bound results prove that the transferred local shape masks from examples constitute a valid structured label space for object segmentation.

5.3.2 Weizmann Horse

Th Weizmann Horse dataset [15] is widely used for benchmarking object segmentation algorithms. This dataset consists of 328 horse images with side views. We follow a typical

evaluation protocol that uses 200 images for the database and the remaining 128 for the test set. We present some qualitative results in Figure 5.7. We evaluate object segmentation

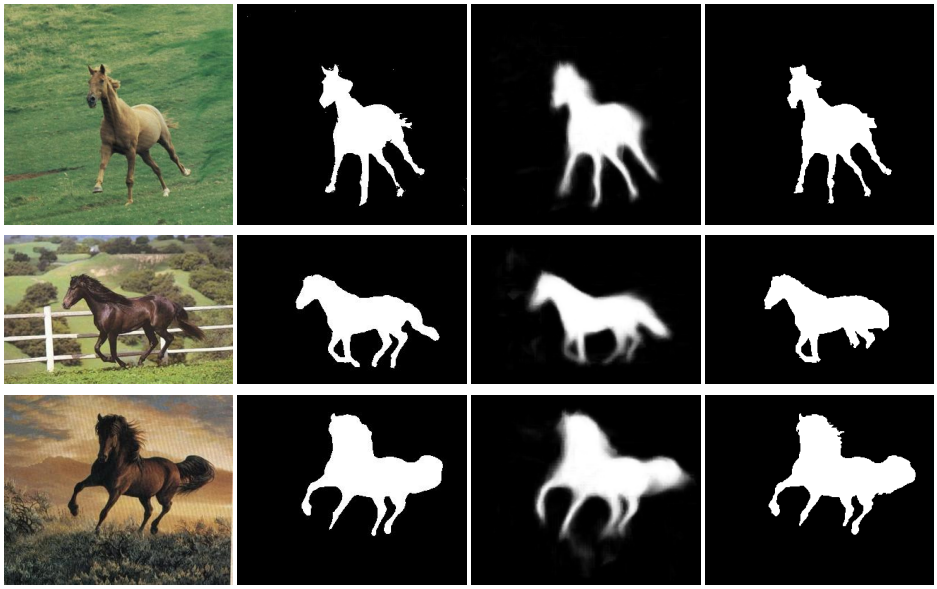


Figure 5.7: Qualitative results on Weizmanm Horse.

performance in terms of mean Jaccard score and overall pixel-wise classification accuracy ($Acc = |\hat{\mathbf{Y}} == \mathbf{Y}|/|\mathbf{Y}|$). In Table 5.2, we compare PatchCut algorithm with state-of-the-art example based algorithm using Window Mask Transfer [57], and various leading learning based algorithms based on Kernelized Structured SVM [11], CRFs [71, 67] and Margin-Margin Boltzmann Machines (BMs) [117]. Our algorithm performs better in terms of both mean Jaccard score and Accuracy. Especially, our algorithm improves about 4% on mean Jaccard score. For horse images, our algorithm usually generates high quality shape prior around legs (Figure 5.7), but the iterated graph cuts refinement tends to cut off the legs because of the shrinking bias towards shorter boundaries.

Table 5.2: Performance evaluation on Weizmann Horse.

	Jaccard (%)	Acc (%)
PatchCut_thres	**80.33**	**94.78**
PatchCut	**84.03**	**95.81**
Kernelized Structured SVM [11]	80.10	94.60
Fragment-based CRFs [67]	N/A	95.0
High-Order CRFs [71]	69.90	N/A
Max-Margin BMs [117]	75.78	90.71
Window Mask Transfer [57]	N/A	94.70

5.3.3 Object Discovery

This dataset consists of three object categories: airplane, car and horse and their images are collected from Internet. It is originally designed for evaluating object co-segmentation [88] and recently used for object segmentation by Ahmed et al. [3]. This dataset is more challenging because the images generally have more complex appearance. Some images include more than one small target and some images are outliers. For each category, we use the same 100 test images as in [88, 3] and the rest as the database. Figure 5.8 shows some qualitative results.

We compare our algorithm with the GrabCut baseline (same implementation as Fashionista), a state-of-the-art co-segmentation algorithm [88] and the latest example based method [3] in Table 5.3. In the Airplane and Horse experiments the refinement step improve the results only slightly while in the Car experiment our algorithm achieves better results without using refinement. The possible reason is that the pixel-wise color models may confuse the shadows with the bottom of the car while the shape prior estimated from

Table 5.3: Jaccard scores on Object Discovery.

Jaccard (%)	Airplane	Car	Horse
GrabCut	63.29	67.63	50.32
Co-segmentation [88]	55.81	64.42	51.65
Ahmed et al. [3]	64.27	71.84	55.08
PatchCut_thres	**70.44**	**86.40**	**63.19**
PatchCut	**70.49**	**84.52**	**64.80**

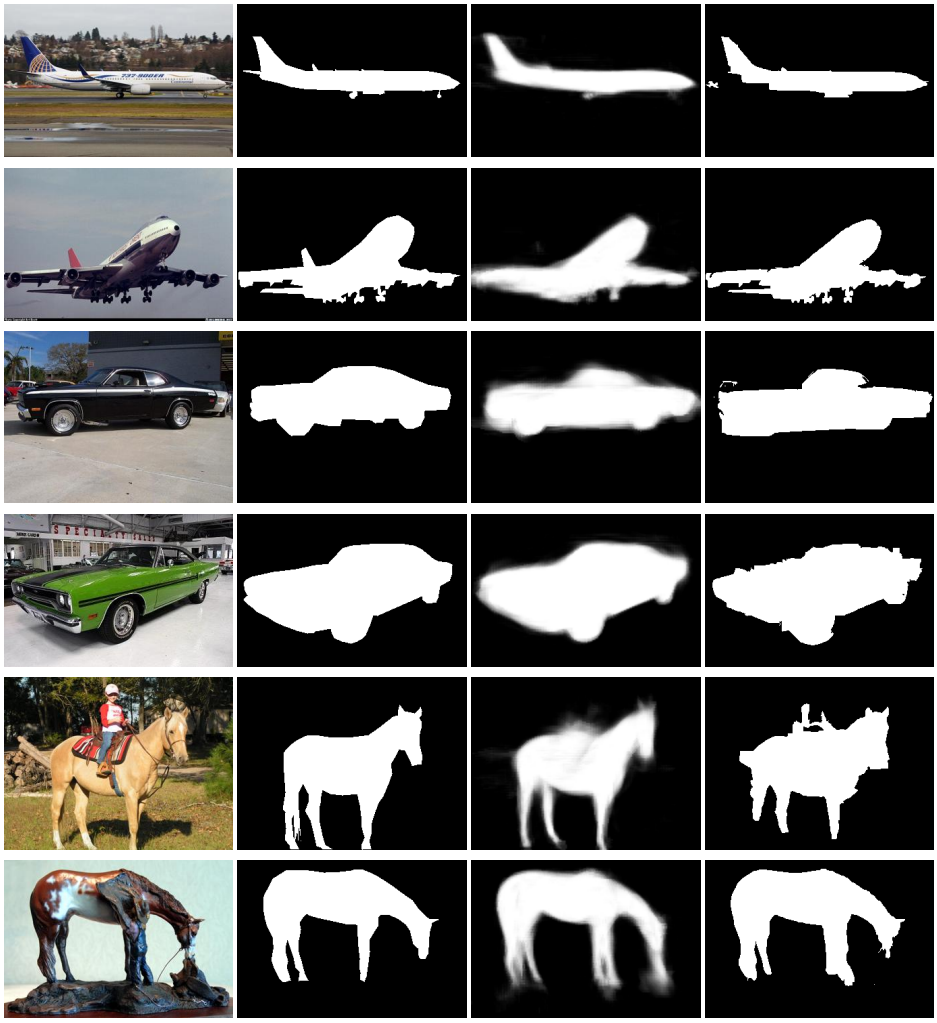


Figure 5.8: Qualitative results on Object Discovery.

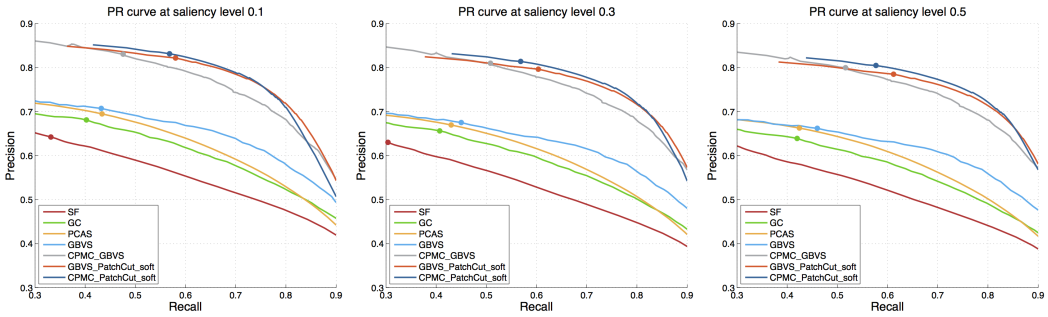


Figure 5.9: Comparing soft segmentation results at different saliency levels in terms of precision-recall curves. The dot on each curve indicates the operating point that gives the best F-score.

local masks better preserves high-level structures.

5.3.4 PASCAL

In this experiment, we present results for salient object segmentation using the PASCAL VOC 2010 dataset [31]. This dataset is more challenging because the images are from 20 object classes with large pose, shape and appearance variations and occlusions. Li et al. [70] collect salient object segmentation masks from human subjects for 850 images in the validation set. We use these images as the test set. Note that salient object segmentation masks may not be binary as subjects may disagree on the choice of salient objects as shown in Figure 5.10.

On the other hand, we use all the images in the training set to build our example database, and collect salient object segmentation ground truth in a similar way as in [70]. Basically, for each image, we use the semantic labeling provided by [80] as full segmentation, and ask 6 subjects to select the salient object regions by clicking on them, so the saliency value for each segment is defined as the number of clicks it receives divided by the number of subjects. Differently from previous experiments, we initialize the PatchCut algorithm with the saliency maps generated by the GBVS algorithm [40], and its results are denoted as GBVS_PatchCut_soft, GBVS_PatchCut_thres and GBVS_PatchCut. We mainly

compare with the state-of-the-art algorithm, CPMC_GBVS, presented in [70] which also uses the GBVS saliency maps. Figure 5.10 shows some qualitative results from PatchCut and CPMC_GBVS for comparisons.

Quantitative evaluation. We convert the ground truth segmentation saliency maps into binary masks with three thresholds: 0.1, 0.3, 0.5. Larger threshold means that less objects with higher saliency values are selected in the ground truth. We first evaluate the soft segmentation masks (saliency maps) in terms of precision-recall curves. We compare the GBVS_PatchCut_soft results with CPMC_GBVS and three recent saliency algorithms, SF [83], GC [24] and PCAS [76]) in Figure. 5.9. Our algorithm (GBVS_PatchCut_soft) performs favorably against CPMC_GBVS and clearly above other saliency algorithms. Second, we evaluate binary segmentation results in terms of mean Jaccard scores. We convert the CPMC_GBVS results into binary segmentation by tuning the threshold, and find that its best mean Jaccard scores are obtained at the threshold 0.3. Table 5.4 shows that our algorithm (GBVS_PatchCut) performs slightly better than CPMC_GBVS, especially at low saliency levels. This result also means that our algorithm tends to select more objects than CPMC_GBVS (see examples in Figure 5.10).

Table 5.4: Jaccard scores on PASCAL.

Saliency level	0.1	0.3	0.5
GBVS_GrabCut	45.84	45.25	44.90
CPMC_GBVS [70]	59.43	60.58	60.75
GBVS_PatchCut_thres	60.08	60.22	59.27
GBVS_PatchCut	**62.02**	**62.15**	**61.14**
CPMC_PatchCut_thres	61.37	62.64	62.76
CPMC_PatchCut	**63.74**	**64.92**	**64.97**

We also initialize our PatchCut algorithm with the soft segmentation masks generated by CPMC_GBVS, and its results are denoted as CPMC_PatchCut_soft, CPMC_PatchCut_thres and CPMC_PatchCut. With this high quality initialization, PatchCut clearly outperforms the state-of-the-art in terms of both precision-recall curve (CPMC_PatchCut_soft in Figure 5.9) and the mean Jaccard scores (CPMC_PatchCut in Table 5.4).

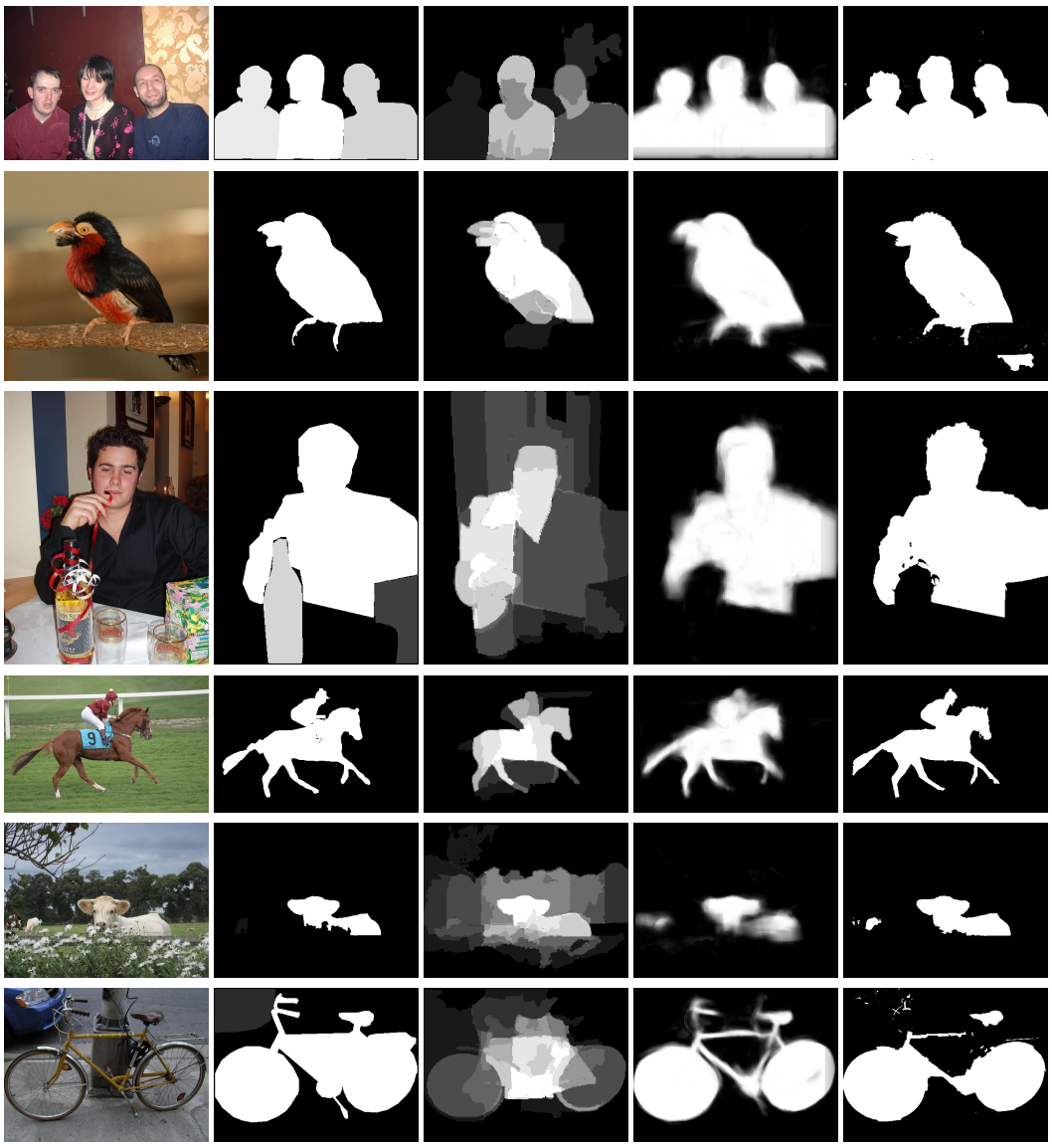


Figure 5.10: Comparing salient object segmentation results on PASCAL.

5.4 Summary

In this chapter, we present a data-driven object segmentation algorithm using examples, which requires no offline training of category-specific models and generalizes well to novel objects. Our algorithm constructs an online structured label space for object segmentation by transferring local shape mask candidates from examples. The MRF labeling problem is decomposed into a set of independent label patch selection sub-problems that are easier to solve in parallel. Our algorithm operates in a coarse-to-fine manner and achieves leading results in many object segmentation benchmarks with low computational cost (about 10 seconds for segmenting a 200x200 image with unoptimized MATLAB code on a typical desktop).

Chapter 6

Scene Paring: Object Segmentation in Context

6.1 Introduction

The goal of scene parsing is to associate a semantic label such as sky, trees, cars, etc. with every pixel in a still image. Such an image description has broad applications, e.g. image editing, image search and autonomous vehicles. Considering potentially hundreds or thousands of semantic labels in common outdoor environments and indoor scenes, it is of great interest to endow the scene parsing system with the ability to operate in a large scale. Large scale scene parsing faces two main challenges. First, the distribution of objects in natural images tends to be heavy-tailed, with many pixels in the images coming from common background classes (the sky, water, and sand in Figure 6.1) and far fewer pixels coming from any given one of the thousands of possible object types. The large number of rare object classes and their relatively small sizes in many images make it difficult for algorithms to accurately segment important objects (the persons and boat in Figure 6.1). In fact, when evaluating error on a per-pixel basis, the performance of algorithms can often be improved by eliminating the rare classes altogether if their

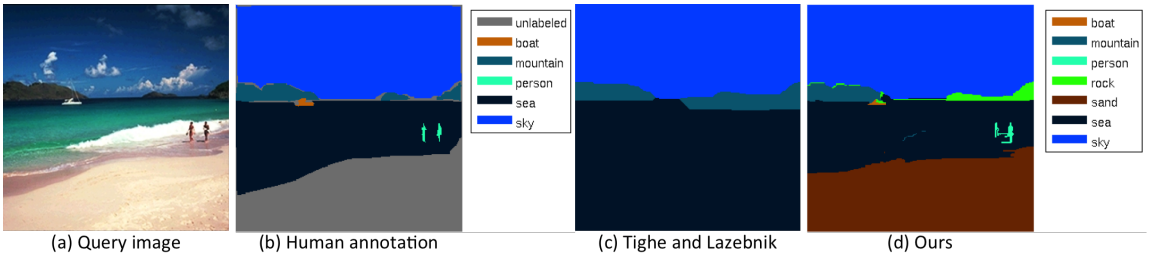


Figure 6.1: Given a query image (a), our method (d) recognizes small objects of rare classes (people, boat) while state-of-the-art systems (c) tend to miss them. Note that our method also recognizes the sand while the human annotator leaves it unlabeled (b).

sizes in the images are usually small, despite the rare classes being very important to human observers. Secondly, it is expensive to optimize image labeling problems with hundreds or thousands of labels. For example, given efficient optimization algorithms such as graph-cut [16], it still takes minutes to solve a pairwise Markov Random Field (MRF) on a 600×800 image with 100 labels. These two challenges make learning based algorithms [94, 59, 32] less applicable.

An alternative is to use nonparametric approaches [73, 27, 96, 101, 100]. To parse an input image, these algorithms first retrieve a small set of similar images and their associated semantic labels from the database, and compute classification confidence maps by matching the query with retrieved images in pixels or superpxiels. The final semantic labeling is obtained by solving a pairwise MRF model. The core of these nonparametric algorithms is motivated by two important observations. On one hand, a single image usually contains very few labels that are constrained by the scene category of the image, compared to the hundreds in total. For example, a photo taken at the beach usually contains sand, sea, boat and person (Figure 6.1(a)). Image retrieval constrains the potentially large number of candidate labels to the ones present in the similar scenes, which greatly reduce the labeling efforts. On the other hand, segments usually capture different partial appearances of objects and are difficult to fit into unified category models. Matching-based algorithms break the category barrier and recognition can be realized by transferring labels from matched

segments to query segments. These nonparametric approaches [73, 27, 96, 101, 100] achieve good performance on overall per-pixel labeling accuracy. However, by taking a closer look at their semantic labeling results, we find that the overall performance is in fact dominated by several common background classes, e.g. mountain, building, water, road and wall, while the performance on interesting object classes is still lagging, e.g. people, animals and man-made objects. To address this problem, Tighe and Lazebnik [100] propose to augment their previous superpixel based parsing system [101] with pre-trained exemplar SVMs [75]. Although per-exemplar detectors makes it possible to transfer object shape masks, their training indeed requires considerable amounts of computational resources and their output also needs to be calibrated with superpixel parsing output in a complicated post-classification step. This hybrid system shows state-of-the-art labeling performance in general, but is still constrained by the quality of per-exemplar detectors and the over-smoothing property of post-processing.

In this chapter, we propose a novel context-driven scene parsing system. Different from previous approaches, we focus more on rare object classes aiming at generating richer and more structured semantic labelings. Beyond the three basic components of nonparametric algorithms, i.e. image retrieval, superpixel matching and MRF labeling, we make two novel contributions:

1. We propose to regularize the retrieval set by a dictionary of rare class superpixels, since the semantic labels of retrieval images usually follow a long-tailed distribution. Therefore, we obtain more balanced classification results.

2. Visual context plays an important role in scene parsing [84]. Beyond the traditional co-occurrence statistics, we bring local and global spatial context into superpixel scoring process to refine image retrieval and superpixel classification, which gives us more contextually sensible parsing results.

We demonstrate our system on the SIFTflow dataset (2688 images, 33 labels) and the LMSun dataset [101] (45576 images, 232 labels). The results show that the proposed

algorithm achieves superior labeling performance than the previous state-of-the-art algorithms in terms of per-class accuracy and per-pixel accuracy on the rare classes, while still achieving similar or superior results on all classes.

6.2 The Baseline System

Our base system consists of three components: image retrieval, superpixel matching and MRF labeling.

6.2.1 Image Retrieval

Image retrieval is a critical step for our system. It determines the labels we use to parse the input image. If the algorithm fails to retrieve relevant images and true labels, we are not able to recover them in later steps. In this chapter, we use the method in [92] to compute the spatially constrained image similarity $m(I_q, I_d)$ between the query image I_q and database images I_d, and retrieve top-K most similar images $\{I_d^1, I_d^2, ..., I_d^K\}$,where $m(I_q, I_d^k) > m(I_q, I_d^{k+1})$. Based on the Bag-of-Words image matching algorithm, this method incorporates spatial voting of local features (SIFT and RGB color) into image scoring. Therefore, the retrieved images usually have similar scene layout to the query, which is desirable for our parsing system. We use a SIFT vocabulary of 10,000 words and a RGB color vocabulary of 1,000 words for local feature quantization. The retrieval top-K images also determine a subset of candidate labels $\mathcal{L}' \subset \mathcal{L}$ for the query image, where $\mathcal{L}$ is the overall label set. This method shares a similar spirit with commonly used spatial pyramid matching [64], but favors scene retrieval more than scene classification in terms of implementation.

6.2.2 Superpixel Matching

We intend to assign semantic labels to every pixel of the query image, based on retrieval images and their corresponding ground truth semantic labels (annotated by human). As a single pixel alone does not contain sufficient information for recognition, we thus choose to recognize pixels in their proper neighboring regions, i.e. superpixels. We use the fast graph-based segmentation algorithm in [35] for producing superpixels for both query and retrieved images. Different from traditional methods, we harvest superpixels of retrieved images from multiple scales. This increases the chance to find good matches for the query superpixel at a controllable computational cost. In experiments, we segment the retrieved images in four scales by varying the k value $k = 50, 100, 200, 400$ in [35]. The smaller k means fine-scale segmentation while the larger k means coarse-scale segmentation. Note that many superpixels from multi-scale segmentation may include labels from different classes, and cannot be assigned a single category label. We thus screen the superpixels by checking their label purity, which is defined as the percent of label majority. We assign a semantic label y_i to a superpixel s_i if its label purity is greater than 95%; otherwise, we remove it from retrieval set. For the query image, we segment it in the finest scale by setting $k = 50$ to control their purity.

We represent each superpixel by four kinds of features, SIFT histogram, RGB histogram, location histogram and PHOG histogram. We extract SIFT descriptors of four scales per 4 pixels by using VLFeat package [106] and encode them by 5 words from a vocabulary of size 1024 using the LLC algorithm [110]. For each superpixel, we compute a 128-dimensional color histogram by quantizing the RGB features from a vocabulary of 128 color words, and a 36-dimensional location histogram by quantizing the (x,y)-locations into a 6×6 grid. In addition, the 168-dimensional PHOG histogram is extracted from the bounding box of each superpixel in a $1\times1, 2\times2, 4\times4$ pyramid. To incorporate the contextual features into the superpixel representation, we also dilate the superpixel masks by 10 pixels and extract the same four kinds of features in the dilated superpixel regions. We thus obtain a 2712-dimensional ($(1024+128+36+168)\times2$) feature vector x_i for each

superpixel s_i.

We compute the classification cost of each input superpixel $s_i \in \mathcal{Q}$ by its k-nearest neighbors $\mathcal{N}_k(i)$ in retrieval set $\mathcal{R} = \{s_j, x_j, y_j\}$,

$$U(y_i = c|s_i) = 1 - \frac{\sum_{j \in \mathcal{N}_k(i), y_j = c} \mathcal{K}(x_i, x_j)}{\sum_{j \in \mathcal{N}^k(i)} \mathcal{K}(x_i, x_j)}, \tag{6.1}$$

where $\mathcal{K}(x_i, x_j)$ denotes the intersection kernel between two histogram feature vectors x_i and x_j.

To reduce the computational complexity, we further map feature vectors into a high-dimensional space $\phi(x_i)$ where the inner product approximates the intersection kernel [106],

$$\mathcal{K}(x_i, x_j) \approx < \phi(x_i), \phi(x_j) > \tag{6.2}$$

6.2.3 MRF Labeling

We build a four-connected pairwise MRF for semantic labeling. The energy function is given by

$$E(Y) = \sum_p U(y_p = c) + \lambda \sum_{pq} V(y_p = c, y_q = c'), \tag{6.3}$$

where p, q are pixel indices, c, c' are candidate labels that belong to retrieval label subset $\mathcal{C}'$ and λ is the weight of pairwise energy. The unary energy of one pixel is given by its superpixel,

$$U(y_p = c) = U(y_i = c|s_i), p \in s_i. \tag{6.4}$$

The pairwise energy on edges is given by spatially variant label cost,

$$V(c, c') = d(p, q) \cdot \mu(c, c'), \tag{6.5}$$

where $d(p, q) = \exp(-\|I(p) - I(q)\|^2 / 2\sigma^2)$ is the color dissimilarity between two adjacent pixels and $\mu(c, c')$ is the penalty of assigning label c and c' to two adjacent pixels. We define $\mu(c, c')$ by the log-likelihood of label co-occurrence statistics,

$$\mu(c, c') = -\log[(P(c|c') + P(c'|c))/2] \times \delta[c' \neq c] \tag{6.6}$$

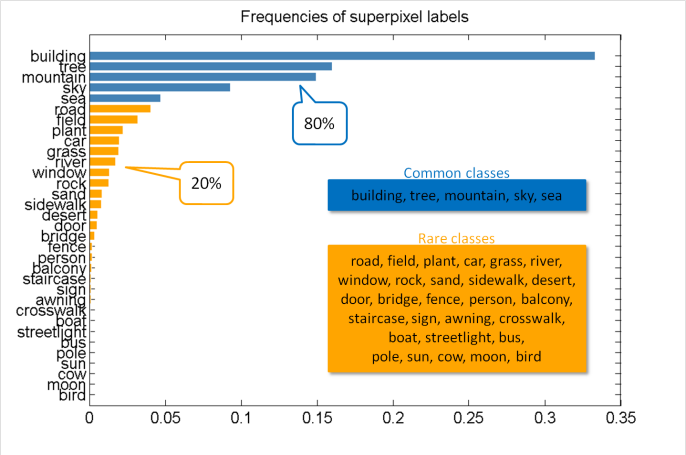


Figure 6.2: The long tailed superpixel label distribution on the SIFTflow training set with the orange bar denotes the rare classes while the red bars denote the common classes.

which we estimate from the training images by calculating conditional probabilities $P(c|c')$ of adjacent superpixels. We obtain the semantic labeling by performing MAP inference on $E(Y)$ by alpha-beta swap algorithm [16].

6.3 Rare Class Expansion

In scene images, the salient regions usually capture the attention of human observers [43], as they provide more information than generic background for scene understanding. It is thus crucial to recognize these objects of interest for generating rich semantic description of images. The saliency property of interesting objects also result in their insufficient representations in the retrieval set. For example, in the "Before expansion" portion of the plot in Figure 6.3, the superpixels of retrieved images are dominated by sky, sea and sand while boat and people classes are in the very tail of the distribution. This fact brings challenges to recognizing those objects of rare classes.

In this chapter, we propose to enrich the retrieved set of superpixels with exemplars of rare classes from the entire database. The label distribution of the retrieval set could

be noisy due to the possibly irrelevant images. It is thus difficult to tell if a class in the tail part are interesting objects (boat and people) or simply outliers (building). We instead define "rare" classes by examining the superpixel label distribution of the entire training set. We partition this distribution into head and tail parts based on the 80%-20% Pareto rule, and define the classes in the tail as "rare" $\mathcal{L}_r$ while the other classes in the head as "common" $\mathcal{L}_c$. In Figure 6.2, we present the superpixel distribution of the SIFT flow training set [73], and our definition of rare and common classes. Given this definition, we can partition the label subset of retrieval superpixels into two parts $\mathcal{L}' = \mathcal{L}'_r \bigcup \mathcal{L}'_c$ and populate the superpixels in the classes of $\mathcal{L}'_r$ with exemplars. Note that the reduced label set remains the same after expansion. In Figure 6.3, we expand 9 classes (road, field, plant, grass, river, rock, sand, person, boat) and obtain a more balanced, noise resistant superpixel distribution as shown in the "After expansion" portion of the plot.

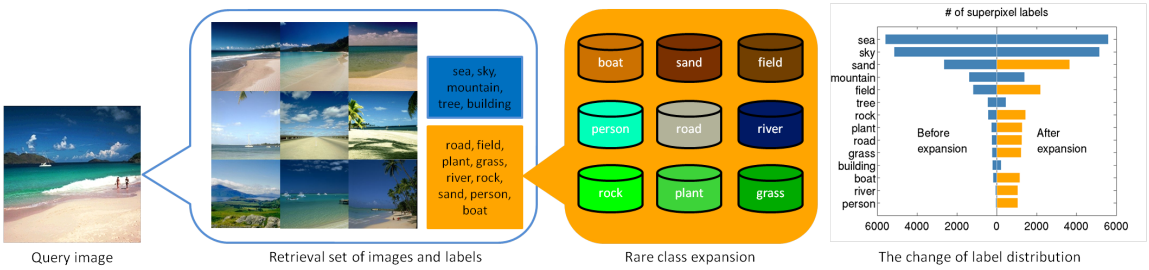


Figure 6.3: Rare class expansion (orange bars).

6.3.1 Building a Dictionary of Exemplar Superpixels

We build a dictionary $\mathcal{D}_c$ of exemplar superpixels for each class $c \in \mathcal{L}$. We project superpixel feature vectors x_i into a low-dimensional space by PCA and cluster them into 1000 centers by using k-means. We select those superpixels which are closest to the centers as exemplars of particular class. Note that we use this method to build dictionaries for its simplicity, although a more sophisticated algorithm such as in [72] could help to mine more discriminative and diverse exemplars.

6.3.2 Superpixel Classification

We classify the superpixels of query image $s_i \in \mathcal{Q}$ by using both the retrieval set $\mathcal{R}$ and auxiliary set of rare class exemplars $\mathcal{D}_c, c \in \mathcal{L}'_r$. Similar to our base system in Section 6.2.2, we compute the classification cost of one query superpixel s_i by its κ-nearest neighbors $\mathcal{N}'_k(i) \in \{\mathcal{R} \bigcup \mathcal{D}_c, c \in \mathcal{L}'_r\}$,

$$U_1(y_i = c|s_i) = 1 - \frac{\sum_{j \in \mathcal{N}'_k(i), y_j = c} \mathcal{K}(x_i, x_j)}{\sum_{j \in \mathcal{N}'^k(i)} \mathcal{K}(x_i, x_j)}. \tag{6.7}$$

We set $\kappa = 37$ through all the experiments. To increase the classification accuracy of κ-NN, systems in [27, 96] learn weights for superpixels feature vectors. In this work, we choose to hybridize the κ-NN classifier with an SVM classifier [22]. We train the SVM classifier only using the exemplars in our dictionary. Given this small and balanced set of training samples, we can train linear SVM classifiers $\{w_c, b_c\}, c \in \mathcal{L}$ in a very efficient way [21]. The SVM classification cost for the query superpixel $s_i \in \mathcal{Q}$ is given by $U_2(y_i = c|s_i) = - < w_c, \phi(x_i) > -b_c$. The total superpixel classification cost is thus computed by combining the κ-NN cost and the SVM cost,

$$U(y_i{=}c|s_i) = \alpha U_1(y_i{=}c|s_i) + (1 - \alpha)U_2(y_i{=}c|s_i), \tag{6.8}$$

where α is the combination coefficient.

6.4 Semantic Context

Context is an important source of information for scene labeling. Although there are many ways to explore this information, we choose to a simply, yet effective feedback mechanism based approach, inspired by [27, 96, 105]. In the base system with rare class expansion, we transfer semantic labeling information from the database to the input image through image retrieval and superpixel classification. In this feedforward process, we obtain the initial semantic knowledge of the input image that are represented by the pixel-

wise classification likelihood maps.

$$\ell(p,c) = \frac{1}{1+\exp(U(y_p = c))}, c \in \mathcal{L}', \tag{6.9}$$

where $U(y_p = c)$ is the cost of assigning label c to pixel p in (6.4) and $\mathcal{L}' \subset \mathcal{L}$ is the candidate label subset. These classification maps in the reduced label set grant us naturally sparse representation of semantic information without an extra sparse coding step [45]. The question is how we can use this initial result as a feedback to reinforce the labeling process, in particular the two key components, image retrieval and superpixel classification. First, the likelihood maps in (6.9) can serve as global context, which has the potential to improve the appearance based image retrieval with semantic scene description. Second, we can generate local semantic descriptors from the likelihood maps for superpixels. Together with local appearance descriptors, we can achieve more contextually consistent classification results. We introduce the algorithms to construct both global and local context descriptors from the likelihood maps in (6.9) below. Note that to compare semantic context between the query and the database images, we compute the classification likelihood maps for all the training images in a leave-one-out fashion.

6.4.1 Global Context Descriptor

We define the global context of an image as the spatial layout of semantic content in multi-scale. To this end, we partition an image into a three-layer spatial pyramid $\{I = \bigcup_i \Omega_i^l, l = 1,2,3, i = 1,...,2^{l-1}\}$, and for each cell Ω_i^l, we compute its $|\mathcal{L}| \times 1$ sparse context vector $\mathbf{z}_i^l = [z_{ic}^l]_{c=1,2,...,|\mathcal{L}|}$ by max pooling of classification likelihood maps,

$$z_{ic}^l = \begin{cases} \max_{p \in \Omega_i^l} \ell(p,c) & \text{if } c \in \mathcal{L}'; \\ 0, & \text{otherwise.} \end{cases} \tag{6.10}$$

The global context descriptor is thus formed by concatenating the sparse vectors from all the cells $\mathbf{z} = [\mathbf{z}_i^l]_{l=0,1,2,i=1,...,2^{l-1}}$. Figure 6.4 demonstrates the process of computing the global context descriptor of one query image. We use global context descriptor $\mathbf{z}$ to

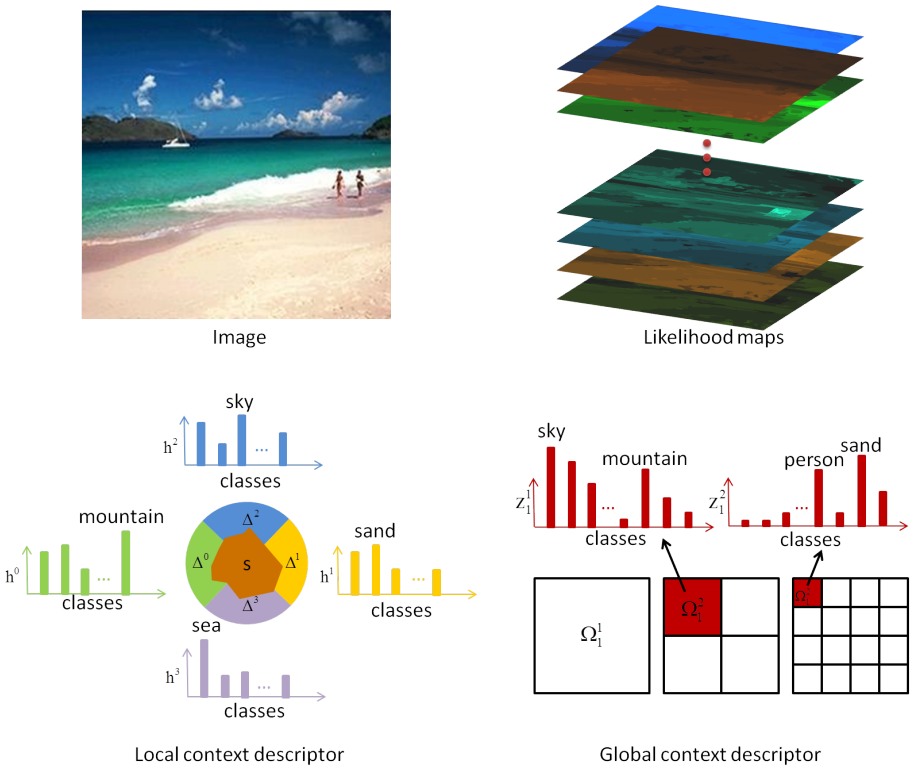


Figure 6.4: Computing global and local context descriptors from the likelihood maps.

update the image similarity between the query image and database images, $m'(I_q, I_d) = m(I_q, I_d) + < \mathbf{z}_q, \mathbf{z}_d >$ and obtain a new set of retrieved images.

6.4.2 Local Context Descriptor

We describe superpixels by their local context for robust matching. For each superpixel s_i, we divide its neighborhood into left, right, top, bottom four cells $\{\Delta_i^0, \Delta_i^1, \Delta_i^2, \Delta_i^3\}$ as illustrated in Figure 6.4, and for each cell Δ_i^j, we compute its $|\mathcal{L}| \times 1$ sparse context vector $\mathbf{h}_i^j = [h_{ic}^j], c = 1, 2, ..., |\mathcal{L}|$ by the same operation as for global context descriptors,

$$h_{ic}^j = \begin{cases} \max_{p \in \Delta_i^j} \ell(p, c) & \text{if } c \in \mathcal{L}'; \\ 0, & \text{otherwise.} \end{cases} \tag{6.11}$$

We represent the superpixel s_i by concatenating the visual feature vector $\mathbf{x}_i$ and spatial context descriptor $\mathbf{h}_i = [\mathbf{h}_i^0; \mathbf{h}_i^1; \mathbf{h}_i^2; \mathbf{h}_i^3]$. Therefore, we can classify superpixels of the query image using the same procedure described in Section 6.3, but with new feature vectors $\phi([\mathbf{x}_i; \mathbf{h}_i])$.

6.5 Experimental Results

6.5.1 SIFTflow

The SIFTflow dataset consists of 2488 training images and 200 test images. All the images are 256×256 pixels from 33 semantic labels. We retrieve $K = 40$ images for each query. By applying the 80%-20% rule to all the superpixels of training dataset, we identify 5 classes as common while 28 classes as rare (Fig. 6.2). We set $\alpha = 0.7$ to combine the κ-NN and SVM classifiers in (6.8), and set $\lambda = 6$ for the pairwise term of MRF energy function in (6.3). We compare our results with recent work in Table 6.1. Compared

Table 6.1: Comparing accuracy (%) on the SIFTflow dataset. Note that in our results, Full=baseline+RCE+LCD+GCD.

SIFTflow	Per-pixel	Per-class
Liu et al. [73]	76.7	N/A
Farabet et al. [32]	78.5	29.5
Farabet et al. [32] balanced	74.2	46.0
Eigen et al. [27]	77.1	32.5
Singh and Kosecka [96]	79.2	33.8
Tighe and Lazebnik [101]	77.0	30.1
Tighe and Lazebnik [100]	78.6	39.2
Full	79.8	48.7
baseline + RCE + LCD	79.4	46.9
baseline + RCE	78.4	45.4
baseline	78.0	27.5

to nonparametric methods, our method (79.8%) outperforms the state-of-the-art per-pixel

rate (79.2%) in [96], and per-class rate (39.2%) in [100] by a large margin (8.5%). The state-of-the-art learning based method in [32] can achieve close per-class rate (46.0%) to our system at a cost of more than 4% performance drop on per-pixel rate (74.2%). In contrast, we achieve overall performance improvement using the proposed rare class expansion and semantic context descriptors. We present some qualitative results in Fig. 6.5. In the bottom of Table 6.1, we evaluate the contributions of important components to our system: rare class expansion (RCE), local context descriptors (LCD) and global context descriptors (GCD). The results show that rare class expansion plays a central role on per-class rates while semantic context boosts the system performance in general. To further investigate the influence of the rare classes, we compare the performance of our full system with [100] only on the 28 rare classes. Table 6.2 shows that our system outperforms the state-of-the-art by more than 10% for both per-pixel and per-class rates on those rare classes.

Table 6.2: Accuracy (%) on the 28 rare classes of SIFTflow dataset.

Rare classes	Per-pixel	Per-class
Tighe and Lazebnik [100]	48.8	29.9
Our full system	59.4	41.9

Runtime. For this dataset, it takes < 1 sec. to retrieve relevant images, ~ 5 sec. for feature loading, ~ 5 sec for superpixel classification, and < 1 sec. to solve the MRF. The use of context descriptors doubles the classification time.

6.5.2 LMSun

The LMSun dataset consists of 45176 training images and 500 test images. The size of images ranges from 256×256 pixels to 800×600 pixels. There are 232 semantic labels in total. By using the same 80%-20% rule on all the superpixels in the training set, we identify 47 common classes and 185 rare classes. Since this is more complex dataset, we retrieve $K = 120$ images to cover large appearance variations. We set $\alpha = 0.9$ to trade-off

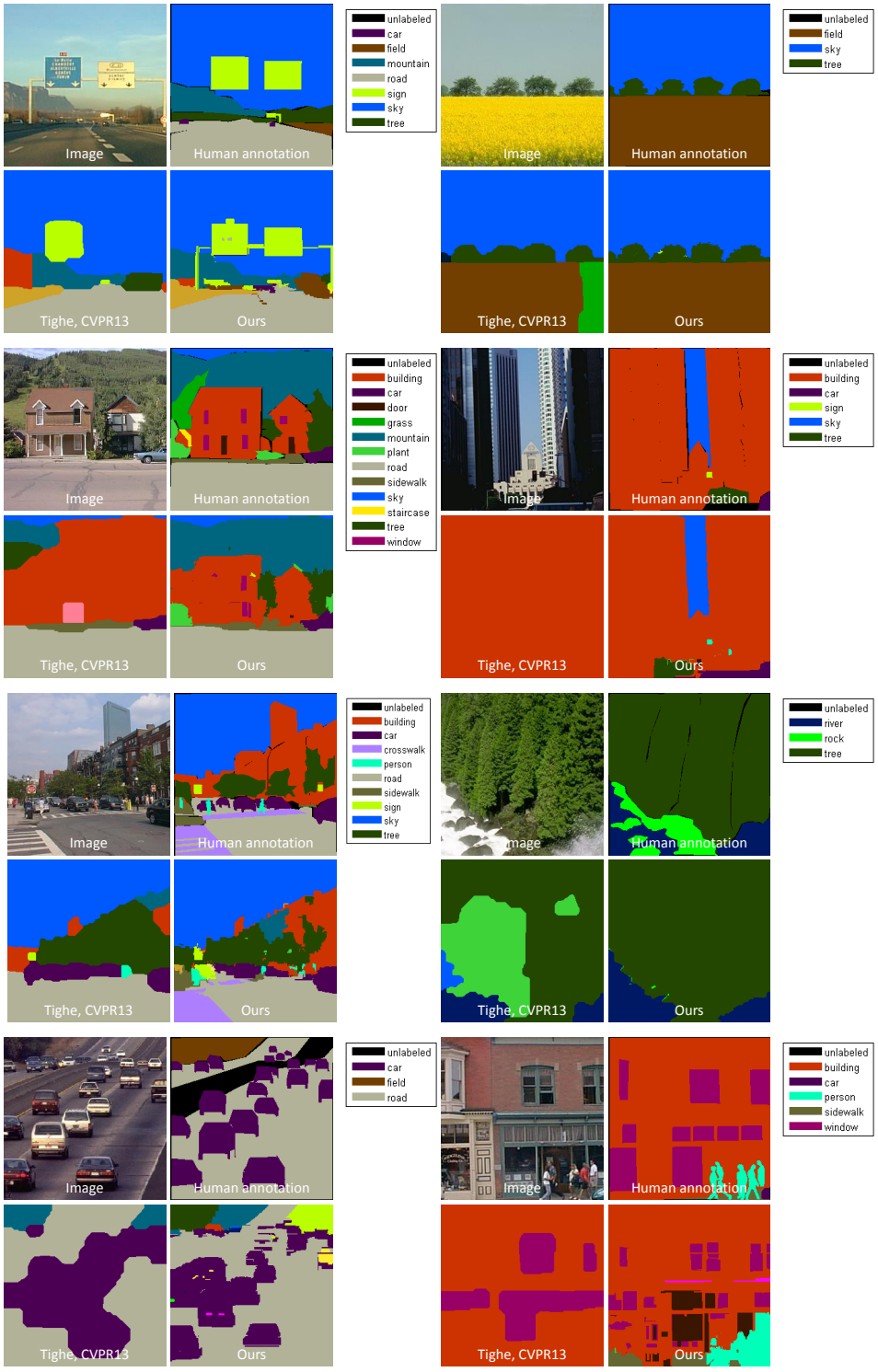


Figure 6.5: Some representative scene parsing results on the SIFTflow dataset

the KNN and SVM classifiers, and set $\lambda = 6$ for the pairwise term in MRF energy function. We compare our results with recent work in Table 6.3. As we focus on the rare classes, our method indeed produces the superior per-class result 18.0% to the previous one 15.2%, while remaining competitive on the per-pixel rate 60.6% vs. 61.4% in [100]. By looking at the accuracy on outdoor (65.4 per-pixel, 17.7 per-class) and indoor (41.8 per-pixel, 16.1 per-class) separately, we observe our system loses the per-pixel performance mainly on indoor images (4.5% lower than [100] vs. 0.1% lower than [100] for outdoor). In Table 6.4,

Table 6.3: Comparing accuracy (%) on the LMSun dataset.Note that in our results, Full=baseline+RCE+LCD+GCD.

	Per-pixel	Per-class
Tighe and Lazebnik [101]	54.9	7.1
Tighe and Lazebnik [100]	61.4	15.2
Full	60.6	18.0
baseline + RCE + LCD	59.4	17.8
baseline + RCE	57.1	14.5
baseline	58.5	9.0

we present the results on the 185 rare classes. It turns out that our system outperforms the state-of-the-art for both per-pixel and per-class rates, which further demonstrates our contributions to rare class boosting. We present some qualitative results in Figure 6.6.

Runtime. Scene parsing is more expensive on the LMSun dataset than the SIFTflow dataset. It takes < 20 sec to retrieve relevant images, ~ 60 seconds for feature loading, ~ 60 sec for superpixel classification, and ~ 60 sec to solve MRF for an 600×800 image with 50-100 labels. Using context descriptors doubles the superpixel classification time.

Image retrieval has significant influence on our system. Superpixel matching and MRF inference becomes much easier to solve within a compact set of relevant images to the query; on the contrary, we notice most of the failure cases are caused by incorrect retrieval. We plan to investigate more effective image retrieval techniques, such as convolutional neural networks [56].

Our system faces challenges in indoor scenes. Indoor scenes are usually composed of

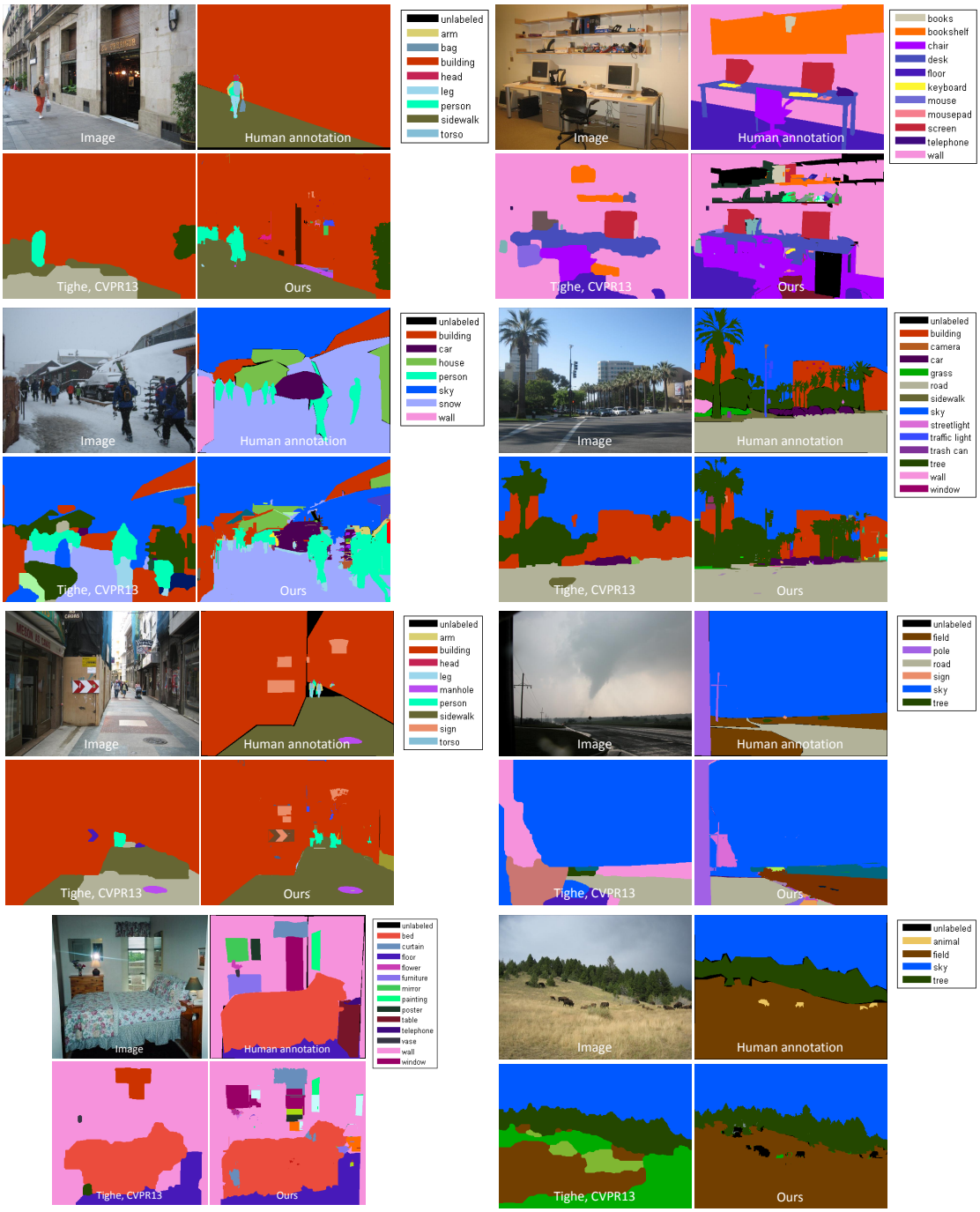


Figure 6.6: Some representative scene parsing results on the LMSun dataset

Table 6.4: Accuracy (%) on the 185 rare classes of LMSun dataset.

Rare classes	Per-pixel	Per-class
Tighe and Lazebnik [100]	19.0	12.9
Our full system	26.4	14.4

many man-made 3D objects (bed, cabinet, table, chairs) and thus have more line structures than textures. Our SIFT based superpixel representation becomes less applicable in this scenario, compared to the HOG feature used in object detectors [100]. We plan to develop better indoor object representations by exploring their 3D geometric structures.

The runtime efficiency is one of the most important factors in large scale problems. On one hand, we plan to incorporate hashing algorithms to accelerate superpixel feature loading and matching; on the other hand, we plan to investigate faster MRF inference algorithms.

6.6 Summary

We have presented a novel scene parsing algorithm, which can operate in large scale. By investigating the roles of rare classes in the database, we have proposed two novel techniques: rare class expansion and local/global semantic context descriptors, which are able to significantly boost the per-class performance of our system. Based on that, we have achieved the state-of-the-art results on the SIFTflow and the large scale LMSun datasets.

Chapter 7

Conclusion

7.1 Summary of Contributions

We have investigated object segmentation problems in single images. We have made several key contributions towards effective integration of top-down recognition and bottom-up segmentation.

Generating proposals with exemplar cuts. In the segmentation driven process, object segmentation is decomposed into two subproblems of grouping pixels or regions to form object proposals and evaluating proposals using recognition models. We have developed a novel proposal generation algorithm using exemplars. The algorithm is based on one regularized CRF energy function built on a region tree. To generate multiple solutions from the learned CRF energy function, our method introduces a nonparametric term that varies when matching with different exemplars. Compared to previous works that re-parameterize the energy function by a tuning parameter or an iterated penalty term our data-driven method takes advantage of both diversity and informativeness of existing segmentation exemplars and thus produce a compact set of highly plausible proposals. When evaluating the proposals as a whole, our method significantly improves the upper-bound performance on the PASCAL VOC dataset.

Learning shape representations with max-margin Boltzmann machines. Shape plays an important role on segmenting object instances with occlusions. Towards shape-guided object segmentation, we have proposed a novel model that unifies the distributed shape representation and bottom-up segmentation in a conditional Boltzmann machine. The proposed model is jointly learned in a max-margin fashion. We compared the performance of different margin functions and showed their relations to existing learning algorithms for conditional RBMs and latent structured support vector machines. Different from previous works that mainly use pre-trained Boltzmann machines as shape prior, our model emphasizes the structured prediction from recognition models to hidden variables of Boltzmann machines. This difference ensures the the shape is well preserved in the resulting segmentation masks. Evaluating on widely used pedestrian, horse and bird datasets, our algorithm achieved state-of-the-art results and demonstrated robustness to partial occlusions.

Transferring multiscale shape masks with Patch Cut. In many interactive applications, users would like to select an object of interest for segmentation. Since the selected object could be from any category, learning class-specific models become not realistic. We have proposed a novel nonparametric algorithm for generic object segmentation using a database of examples. We analyzed the relation between image matching and shape transfer in scale space and thus formulated a coarse-to-fine shape transfer scheme using multiscale dense matching. We combined the transferred local shape masks with image color models in a higher-order MRF and developed a novel approximate inference algorithm by alternating local mask selection and color model estimation, referred as Patch Cut. Compared to previous works that transfer rigid or deformable masks in image sub-windows, our method explores shape masks in multiple granularities and is able to produce high quality segmentations in an efficient way. We evaluated the Patch Cut algorithm against the state-of-the-art learning and nonparametric methods on various benchmarks and results show its superior performance.

Putting objects in context for scene parsing. In scenic (indoor or outdoor) images, objects usually only take a small portion of pixels and the rest belongs to background. This

phenomenon results in a fundamental ambiguity about object appearance. We have proposed a novel data-driven scene parsing system that resolves this ambiguity by introducing context from multiple levels into the recognition process. We also developed a novel technique to regularize the class distribution by expanding the statistically rare object classes in the recognition process. Our system demonstrates state-of-the-art results in two large-scale scene parsing dataset.

7.2 Future Work

Along with data-driven object segmentation, we are interested in the following questions for future investigation.

- **How can object segmentation benefit from deep learning?** Deep learning methods, e.g. convolutional networks, have made tremendous progress for object categorization. The learned rich feature hierarchy has shown its successful application in detection with object proposals. What challenges its application to object segmentation is an effective integration or unification of convolutional networks and random fields, which is also a long-standing question in machine learning. Early attempts include 1) learning potential functions of random fields with convolutional networks; 2) re-interpreting inference iterations of random fields as new layers of convolutional networks. Those pioneer methods, although improving the labeling accuracies, still lack knowledge of representing individual objects and their occlusion relationships.

- **How can one learn object segmentation with rich semantics?** For a human image, its segmentation can be holistic ”human” and ”background”, by parts ”head”, ”arms”, ”torso” and ”legs”, by attributes ”jumping”, ”happy” and ”girl”. These different labelings may confuse the learning algorithms. Thus, it is of great interest to integrate segmentation with language models so that one can collect training data

from arbitrary annotations and interpret results in multiple semantic levels.

- **How can object segmentation be done with humans in the loop?** Most of state-of-the-art object segmentation algorithms are data-hungry, but it is very challenging to collect large-scale segmentation datasets.Different from categorization, manually segmenting one image is a tedious task for humans and the quality decreases fast when human annotators lose their patience. First of all, we need to develop more advanced annotation tools. Second, instead of one-time data collection, it is more interesting to close the loop from data collection, annotation to model learning. In order to achieve that, we need an internet-scale platform where machines and humans can collaborate.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2281, 2012.

[2] Adobe Systems Inc. Photoshop. Creative Cloud, 2014.

[3] E. Ahmed, S. Cohen, and B. Price. Semantic object selection. In *CVPR*, 2014.

[4] Amazon. Mechanical turk.

[5] P. K. Andrew Blake and C. Rother, editors. *Markov Random Fields for Vision and Image Processing*. The MIT Press, 2011.

[6] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[8] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *ECCV*, 2010.

[9] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *IJCV*, 2011.

[10] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse m-best solutions in Markov random fields. In *ECCV*, 2012.

[11] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural svm learning for supervised object segmentation. In *CVPR*, 2011.

[12] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B*, 48(3):259–302, 1986.

[13] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1997.

[14] Y. Bo and C. Fowlkes. Shape-based pedestrian parsing. In *CVPR*, 2011.

[15] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002.

[16] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222 – 1239, 2001.

[17] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001.

[18] M. Bray, P. Kohli, and P. H. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *ECCV*, 2006.

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

[20] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010.

[21] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[22] P. Chaudhuri, A. K. Ghosh, and H. Oja. Classification based on hybridization of parametric and nonparametric classifiers. *PAMI*, 31(7):1153 – 1164, July 2009.

[23] F. Chen, H. Yu, R. Hu, and X. Zeng. Deep learning shape priors for object segmentation. In *CVPR*, 2013.

[24] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *PAMI*, 2014.

[25] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[26] P. Dollár and C. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[27] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *CVPR*, 2012.

[28] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[29] S. M. A. Eslami, N. Heess, and J. Winn. The shape Boltzmann machine: a strong model of object shape. In *CVPR*, 2012.

[30] S. M. A. Eslami and C. K. I. Williams. A generative model for parts-based object segmentation. In *NIPS*, 2012.

[31] 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".

[32] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*, 2012.

[33] A. Farhadi and M. A. Sadeghi. Recognition using visual phrases. In *CVPR*, 2011.

[34] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramana. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.

[35] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 59(2), 2004.

[36] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

[37] D. M. Gavrila. A Bayesian, exemplar-based approach to hierarchical shape matching. *PAMI*, 29(8):1408–1421, 2007.

[38] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009.

[39] A. Guzman-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *NIPS*, 2012.

[40] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006.

[41] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771 – 1800, 2002.

[42] Q. Huang, M. Han, B. Wu, and S. Ioffe. A hierarchical conditional random field model for labeling and images of street scenes. In *CVPR*, 2011.

[43] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene anaysis. *PAMI*, 20(11):1254–1259, 1998.

[44] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[45] L. jia Li, H. Su, E. P. Xing, and L. Fei-fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

[46] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[47] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller. Augmenting CRFs with Boltzmann Machine shape priors for image labeling. In *CVPR*, 2013.

[48] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988.

[49] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV*, 2012.

[50] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.

[51] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[52] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[53] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, 2007.

[54] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 26:65–81, 2004.

[55] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *ICCV*, 2011.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[57] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012.

[58] M. P. Kumar, P. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[59] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[60] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[61] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, 2010.

[62] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[63] D. Larlus and F. Jurie. Combining appearance models and markov random fields for category level object segmentation. In *CVPR*, 2008.

[64] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[65] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[66] V. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. In *NIPS*, 2011.

[67] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006.

[68] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010.

[69] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[70] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

[71] Y. Li, D. Tarlow, and R. Zemel. Exploring compositional high order pattern potentials for structured output learning. In *CVPR*, 2013.

[72] Z. Liao, A. Farhadi, Y. Wang, I. Endres, and D. Forsyth. Building a dictionary of image fragments. In *CVPR*, 2012.

[73] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 33:2368 – 2382, 2011.

[74] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[75] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[76] R. Margolin, A. Tal, and L. Zelnik-Manor. What makes a patch distinct? In *CVPR*, 2013.

[77] P. Marquez-Neila, P. Kohli, C. Rother, and L. Baumela. Non-parametric higher-order random fields for image segmentation. In *ECCV*, 2014.

[78] X. Miao and R. P. N. Rao. Large margin boltzmann machines. In *IJCAI*, 2009.

[79] V. Mnih, H. Larochelle, and G. E. Hinton. Conditional restricted Boltzmann machines for structured output prediction. In *UAI*, 2011.

[80] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[81] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *CVPR*, 2008.

[82] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *PAMI*, 28(3):416–431, 2006.

[83] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.

[84] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *CVPR*, 2007.

[85] A. Rosenfeld and D. Weinshall. Extracting foreground masks towards object recognition. In *ICCV*, 2011.

[86] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009.

[87] C. Rother, V. Kolmogorov, and A. Blake. Grabcut -interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 2004.

[88] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[89] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *AISTATS*, 2009.

[90] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *ICML*, 2007.

[91] A. Shekhovtsov, P. Kohli, and C. Rother. Curvature prior for MRF-based segmentation and shape inpainting. In *DAGM*, 2012.

[92] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn reranking. In *CVPR*, 2012.

[93] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22:888–905, 1997.

[94] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-Class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009.

[95] D. Singaraju and R. Vidal. Using global bag of features models in random fields for joint categorizationand segmentation of objects. In *CVPR*, 2011.

[96] G. Singh and J. Kosecka. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *CVPR*, 2013.

[97] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *ECCV*, 2008.

[98] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[99] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[100] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.

[101] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. *IJCV*, 101:329–349, 2013.

[102] A. Torralba, A. Oliva, M. Castelhano, and J. M. Henderso. Contextual guidance of attention in natural scenes: The role of global features on object search. *Psychological Review*, 113(10):766–786, 2006.

[103] Y.-H. Tsai, J. Yang, and M.-H. Yang. Decomposed learning for joint object segmentation and categorization. In *BMVC*, 2013.

[104] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453 – 1484, 2005.

[105] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *PAMI*, 32(10):1744 – 1757, 2010.

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

[107] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[108] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *CVPR*, 2011.

[109] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[110] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[111] L. Wang, J. Shi, G. Song, and I.-F. Shen. Object detection combining recognition and segmentation. In *ACCV*, 2007.

[112] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

[113] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, 2014.

[114] W. Xia, Z. Song, J. Feng, L. F. Cheong, and S. Yan. Segmentation over detection by coupled global and local sparse representations. In *ECCV*, 2012.

[115] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *TIP*, 7(3), 1998.

[116] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.

[117] J. Yang, S. Safar, and M.-H. Yang. Max-margin boltzmann machines for object segmentation. In *CVPR*, 2014.

[118] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object models for image segmentation. *PAMI*, 34(9):1731–1743, 2012.

[119] C. Yanover and Y. Weiss. Finding the m most probable configurations using loopy belief propagation. In *NIPS*, 2003.

[120] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009.

[121] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011.