國立臺灣大學電機資訊學院資訊網路與多媒體研究所
碩士論文
Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

多重實例迴歸於定界框內物件輪廓之估測

Augmented Multiple Instance Regression For Inferring Object Contours Within Bounding Boxes

許洸睿
Kuang-Jui Hsu

指導教授：莊永裕 博士 ; 林彥宇 博士
Advisor: Chuang Yung-Yu, Ph.D. ; Yen-Yu Lin, Ph.D.

中華民國 102 年 7 月
July, 2013

Abstract

In this thesis, we address the problem of the high annotation cost of acquiring training data for semantic segmentation. Most modern approaches to semantic segmentation are based upon graphical models, such as the conditional random fields, and rely on sufficient training data in form of object contours. To reduce the manual effort on pixel-wise annotating contours, we consider the setting in which the training dataset for semantic segmentation is a mixture of a few object contours and an abundant set of bounding boxes of objects. Our idea is to borrow the knowledge derived from the object contours to infer the unknown object contours enclosed by the bounding boxes. The inferred contours can then serve as training data for semantic segmentation. To this end, we generate multiple contour hypotheses for each bounding box with the assumption that at least one hypothesis is close to the ground truth. This thesis proposes an approach, called *augmented multiple instance regression* (AMIR), that formulates the task of hypothesis selection as the problem of MIR, and augments information derived from the object contours to guide and regularize the training process of MIR. In this way, a bounding box is treated as a bag with its contour hypotheses as instances, and the positive instances refer to the hypotheses close to the ground truth. The proposed approach has been evaluated on the Pascal VOC segmentation task. The promising results demonstrate that AMIR can precisely infer the object contours in the bounding boxes, and hence provide effective alternatives to

manually labeled contours for semantic segmentation.

Keywords: Semantic segmentation, weakly supervised learning, multiple instance regression (MIR), segment selection.

Contents

Abstract **i**

1 Introduction **1**

2 Related Work **6**

2.1 Semantic Segmentation 6

2.2 Figure-Ground Segmentation 7

2.3 Multiple Image Segmentations 7

2.4 Object Segmentation with Low Cost 8

2.5 Multiple Instance Learning (MIL) 9

3 Inferring Multiple Tight Segments in a Bounding Box **10**

3.1 Tight Segment via Bounding Box Prior 10

3.2 Multiple Tight Segments 12

3.3 Problem Definition 14

3.3.1 Object Contours 15

3.3.2 Positive Bounding Boxes 15

3.3.3 Negative Bounding Boxes 15

4 The Proposed AMIR approach **17**

4.1 On Least Square Fitting over L 17

4.2 On Multiple Instance Regression over $U^+ \cup U^-$ 19

5 Tight Segment Feature Representation **24**

5.1 Segment-Level Features 24

5.2 Pixel-Level Features 25

6 Experiment Results **27**

6.1 Dataset: Pascal VOC 2007 27

6.2 Experiment I: Multiple Tight Segment Generation 28

6.3 Experiment II: AMIR for Tight Segment Selection 29

6.4 Experiment III: Semantic Segmentation 35

7 Conclusion **38**

Bibliography **40**

List of Figures

1.1 The example of the semantic segmentation. The images in the first row are the input images. Those in the second row are the output images, and different colors represent different predefined classes. Each color in the third row represents the corresponding predefined class. 2

1.2 Framework overview. We unify three different information sources, i.e., object contours, positive and negative bounding boxes, in the basis of tight segments. The proposed AMIR minimizes the square errors of the labeled tight segments, while carrying out multiple instance learning over the unlabeled ones. After optimization, the most plausible tight segments in the positive bounding boxes are picked, and can serve as the training data in semantic segmentation. 3

1.3 (a) The bounding box of a kitty can be defined by the four clicks (*purple stars*). Background seeds are sampled outside the bounding box, i.e., the blue highlighted region in the case. (b) The superpixels of the bounding box. (c) The object segment (*ground truth*). (*Rest*) A few tight segments, marked by red contours, together with their accuracies in IoU shown below. These tight segments are generated from different foreground seeds (shown as cyan circles). The seeds are sampled at different locations (*columns*) and with five different radii (*rows*), details seen in Chapter 3. . 4

6.1 The IoU distributions of the best tight segments w.r.t. each of the 20 object classes. The edges of each box are the 25th and 75th percentiles, while the red line indicates the median. Outliers are marked as red-cross signs. The green cycle denotes the median IoU of the baseline that treats full bounding boxes as objects. The average number of the generated tight segments is given in parentheses. 28

6.2 Several examples of the generated tight segments by our approach. (a) The bounding box of an object. (b) The ground truth. (c) The best tight segment. (d) $\sim$ (i) Some of the other generated tight segments. The IoU of each tight segment is also reported. 30

6.3 Inferred object contours and their IoU scores by various approaches. (a) Bounding box. (b) Ground truth. (c) GrabCut [1]. (d) TS [2]. (e) OP [3]. (f) FG [4]. (g) DCCoSeg [5]. (h) LR. (i) AMIR. 33

6.4 The IoU distributions of AMIR (displayed by `boxplot`) as well as the median IoU scores of other baselines. 34

6.5 The performances, average median and mean IoU scores over the 20 object classes, of LR and AMIR with different fractions of the labeled object contours. The performance upper bounds, i.e., the IoU scores of the best tight segments, are also given. 34

6.6 The performances, average median and mean IoU scores over the 20 object classes, of AMIR with different values of γ. 35

6.7 Some results by applying the method in [6] to semantic segmentation. The first two columns give the testing images and the ground truth, respectively. The third and forth columns show the results by learning with manually labeled segments and with the object segments inferred by AMIR, respectively. 37

List of Tables

6.1 The Mean IoU (%) of The Inferred Segments by Various Approaches on Pascal VOC segmentation task. 32

6.2 The numbers of data in L, U^+, and U^- in each object class, together with the running (training) time of AMIR. 35

6.3 IoU scores (%) of [6] on Pascal VOC segmentation task with training data generated by various approaches. 36

6.4 IoU scores (%) of [7] on Pascal VOC segmentation task with training data generated by various approaches. 36

Chapter 1

Introduction

Semantic segmentation [6–23] attempts to assign one of predefined object classes or background to each pixel in an image, shown in Figure 1.1. Distinct from the conventional image segmentation task, e.g., [24–31], semantic segmentation not only determines the shapes or boundaries of objects, but also identifies the object classes of interest. Namely, it involves solving two of the most fundamental problems in image analysis: recognition and segmentation. Accordingly, semantic segmentation plays an essential role in many high-level computer vision applications, such as scene understanding [7, 9, 11], object recognition [11, 13], and image categorization [20, 22].

Recently, significant progress on semantic segmentation has been made with advances in many aspects, such as more powerful features [11, 20], combination of information from different levels of image quantization [6, 17, 19], exploration of contextual relations among object classes [8, 12], and integrating deformable templates and appearance models [14]. These approaches are often built on graphical models such as *Markov random fields* (MRFs) [14] or *conditional random fields* (CRFs) [**?**, 32] for their merits in fusing diverse visual evidences and ensuring spatial consistency. However, learning graphical models for the complex semantic segmentation tasks, e.g., *Pascal VOC* [33], often requires sufficient training data in form of *object contours*, i.e., pixel-wise annotation. In general, the object contours are manually drawn or delineated by tools with intensive user

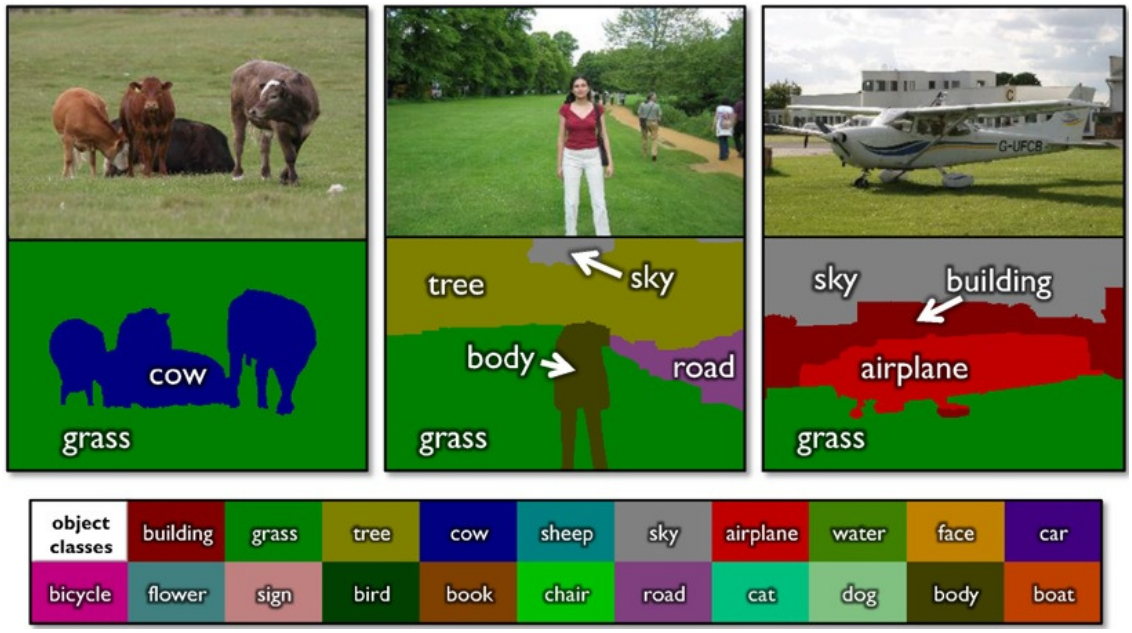

Figure 1.1: The example of the semantic segmentation. The images in the first row are the input images. Those in the second row are the output images, and different colors represent different predefined classes. Each color in the third row represents the corresponding predefined class.

interaction. Therefore, the heavy annotation cost of collecting training data undoubtedly hinders the advances in semantic segmentation.

In this thesis, we aim at reducing the annotation cost in semantic segmentation, and consider training data as a mixture of a few *contours of objects* and an abundant set of *bounding boxes of objects*. This is motivated by the fact that annotating the bounding box of an object is pretty simple by just clicking the four outermost points in the object boundary. If the unknown object contours enclosed by the bounding boxes can be accurately inferred, they can then be used as the training data for semantic segmentation. It reveals the potential of remarkably reducing the annotation cost. The bounding box of an object here is regarded as a *positive* bounding box. We also consider the use of *negative* bounding boxes, each of which has no sufficient overlaps with positive ones. We consider negative bounding boxes because they carry rich information regarding the background, and can activate discriminant learning in inferring the object contours of the positive bounding boxes.

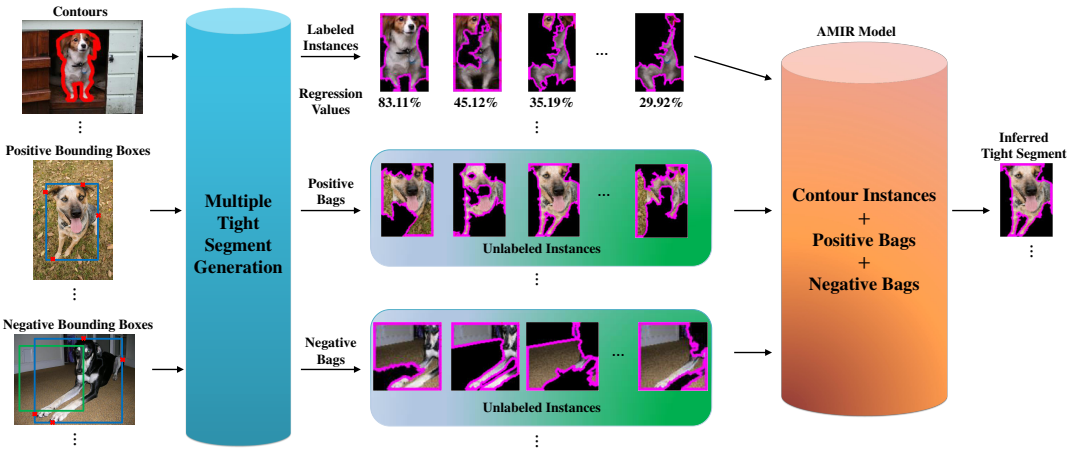

Figure 1.2: Framework overview. We unify three different information sources, i.e., object contours, positive and negative bounding boxes, in the basis of tight segments. The proposed AMIR minimizes the square errors of the labeled tight segments, while carrying out multiple instance learning over the unlabeled ones. After optimization, the most plausible tight segments in the positive bounding boxes are picked, and can serve as the training data in semantic segmentation.

We propose to use *tight segments* [2] as the common basis for fusing the information from three different sources, i.e., object contours, positive and negative bounding boxes, and formulate the inference of the object contours enclosed by the positive bounding boxes as a variant of the *multiple instance regression* problem. Figure 1.2 gives an overview of the framework.

Specifically, we integrate the *bounding box prior* [2] into the concept of *multiple image segmentations* [19,34,35] to develop a new algorithm that can automatically generate a set of tight segments for each positive bounding box. By assuming that at least one of these tight segments is close to the ground truth, the inference of the object contour for a bounding box can be achieved by *picking* the best tight segment from the generated set. Figure 1.3 gives an example of a positive bounding box and the generated tight segments. The same procedure is also applied to the bounding boxes of the labeled object contours and the negative bounding boxes. Although the tight segments generated from negative bounding boxes are not object contours, they are helpful in assessing the tight segments yielded from positive bounding boxes owning to the shared background information.

It can be observed that the bounding boxes and tight segments match the two-layer

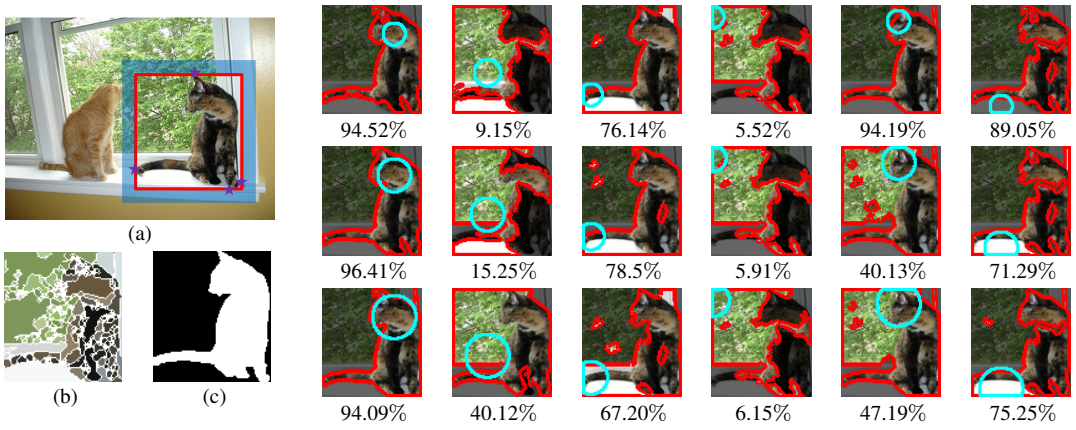

Figure 1.3: (a) The bounding box of a kitty can be defined by the four clicks (*purple stars*). Background seeds are sampled outside the bounding box, i.e., the blue highlighted region in the case. (b) The superpixels of the bounding box. (c) The object segment (*ground truth*). (*Rest*) A few tight segments, marked by red contours, together with their accuracies in IoU shown below. These tight segments are generated from different foreground seeds (shown as cyan circles). The seeds are sampled at different locations (*columns*) and with five different radii (*rows*), details seen in Chapter 3.

structure in *multiple instance learning* (MIL) [36]. Namely, each positive (negative) bounding box can be treated as a *positive (negative) bag* with its tight segments as *instances*. The tight segment closest to the contour in a positive bounding box corresponds to the positive instance. We propose an approach, called *augmented multiple instance regression* (AMIR), that casts the task of tight segment selection as the problem of MIR, and augments information derived from the labeled object contours to guide and regularize the training process of MIR. The *target value* of a tight segment indicates its goodness in regression. Thus, the target values of the tight segments yielded from labeled contours are known, while those from bounding boxes are unknown. AMIR performs least square fitting for labeled tight segments, and carries out multiple instance learning for unlabeled ones. After completing the optimization, the most plausible tight segment for each positive bounding box is determined. Any off-the-shelf approach to semantic segmentation can then be adopted and trained using the inferred object contours for the semantic segmentation task.

With the aim of inferring object contours enclosed by bounding boxes, this work distinguishes itself with the following three main contributions. First, it greatly reduces the

manual annotation in semantic segmentation by using training data as a mixture of a few object contours and a set of bounding boxes, and casting object segment estimation as an MIR problem. The proposed AMIR approach jointly exploits the data labels in the levels of bags and instances, and better solves the MIR problem. Second, AMIR adopts the *smooth maximum function* [37] to handle unlabeled instances. The resulting objective function is differentiable. It follows that AMIR can be very efficiently optimized by gradient descent methods. Third, we demonstrate that the estimated object contours by AMIR are of high quality, and can replace the manually labeled contours even in challenging semantic segmentation tasks such as PASCAL VOC.

The rest of the thesis is organized as follows. Related work is reviewed in chapter 2. In chapter 3, we describe the developed algorithm that generates a set of tight segments for a given bounding box, and satisfies the constraint that at least one tight segment is close to the object contour. Our problem statement and the proposed approach AMIR are introduced in chapter 4. The adopted segment descriptors and the experimental results are given in chapter 5 and chapter 6, respectively. Finally, the conclusion is made in chapter 7.

Chapter 2

Related Work

We review several research topics that are relevant to the development of our approach in the chapter.

2.1 Semantic Segmentation

Approaches to semantic segmentation, such as [6–22], aim to perform multi-class object recognition and segmentation jointly. Most of these approaches are established upon MRFs or CRFs, which can effectively model the dependencies among random variables and observations, and enforce the consistency of labeling. For instance, Shotton et al. [11] proposed a rich set of features to capture the texture, layout and contextual information of object classes in the pixel level, and combined these features via solving an energy minimization problem over CRFs. Zhou et al. [14] developed *dynamic hybrid Markov random field* (DHMRF) that can combine middle-level object shapes and low-level visual appearance to solve the semantic segmentation problem. Various high order potential functions for CRFs have been introduced for semantic segmentation. For example, Kohli et al. [12] and Boix et al. [8] expressed the contextual information among object classes and integrated features extracted from different levels of image quantization by developing hierarchical generalizations of CRFs. Some variants of CRFs were developed

by integrating with other prediction models. For instance, Payet and Todorovic [13] combined *Hough forest* and CRFs for joint object recognition and segmentation in images. Despite the effectiveness of the aforementioned approaches, training CRFs for semantic segmentation requires a vast amount of manual drawing on labeling object contours. The heavy annotation cost of compiling sufficient training data hinders the advances in semantic segmentation.

2.2 Figure-Ground Segmentation

Some notable methods of this category, such as *graph-cut* [38], *GrabCut* [1], *constrained parametric min-cuts* [35], cast this task as an energy minimization problem over graph structures. A latter improvement of GrabCut was made by Lempitsky et al. [2] with the so-called *bounding box prior*. They showed that the resulting foreground regions are sufficiently tight with respect to the given bounding boxes. Instead of working on individual images, the authors of [39, 40] extended figure-ground segmentation for a set of images of an object class. In this way, additional class-specific cues can be included to benefit figure-ground segmentation. Due to the inherent difficulties of unsupervised segmentation, the steps of segmenting objects and learning class models in [39, 40] are carried out either alternately or sequentially. However, segmentation methods aware of object classes may suffer from the problems caused by large intra-class variations, especially when there are only a few training data available.

2.3 Multiple Image Segmentations

Classic image based segmentation methods, such as [24–31], were developed with theoretic support. However, the general conclusion [41] is still that the resulting segmentations typically are not good enough for discovering object contours. Since there is barely universal single-shot solution to segment out various objects with satisfactory results, the

strategy of multiple image segmentations, e.g., [3, 19, 34, 35, 42], arises, in which many segmentation hypotheses are computed with different segmentation algorithms, parameter settings, and/or seeds. In [34, 35], the authors assumed that each object can be discovered by at least one segmentation hypothesis. In [19], Pantofaru et al. sought the most probable objects based on the intersections of multiple segments. Distinct from these approaches, we are motivated by the fact that the bounding box of an object can be acquired with low labeling cost, i.e., four clicks, and it contains rich information for inferring the object contour within it. We couple the concepts of the bounding box prior and multiple image segmentations to develop an algorithm that generates a set of tight segments for each bounding box of an object with the assumption that at least one tight segment is similar to the ground truth. Furthermore, the generated tight segments will serve as the common basis for fusing data in forms of object contours, positive and negative bounding boxes. Thus, we can work with different types of data without worrying about their variations in representations.

2.4 Object Segmentation with Low Cost

Recent research efforts have been made on reducing the labeling cost for object segmentation. *Weakly supervised methods*, e.g., [21, 23, 40, 42–44], support training data labeled in the levels of images or bounding boxes, instead of object maps. As information regarding object classes has been annotated, the class-specific clues can be extracted to facilitate object segmentation. However, restricted by the nature of weakly supervised labeling, the large intra-class variations often obstruct the discovery of the latent object contours. Another type of methods, e.g., [1, 2, 4, 45], for saving manual labeling is interactive segmentation, in which the segmentation process is guided by user inputs. Distinct from these approaches, our approach adopts MIR to model the uncertainty in training data, and automatically infers object contours enclosed by bounding boxes via leveraging knowledge transferred from a few labeled contours.

2.5 Multiple Instance Learning (MIL)

MIL [36] is a variant of binary-class, supervised learning. The training data in MIL are labeled on *bags*, each of which is composed of *instances*. A bag is positive if there is at least one positive instance in it, while a bag is negative if all its instances are negative. The task of MIL is to predict whether a test bag (and its instances) is positive or not. MIL has been applied to image analysis applications for handling the ambiguity in training data, such as face detection [46] and content-based image retrieval [47]. Motivated by the fact that the objective functions of MIL may not be convex, Li and Sminchisescu [48] reformulated non-convex MIL problems as convex ones by optimizing on the *likelihood ratio* between the positive and the negative class for each instance.

In this thesis, we derive a regressor to rank the hypotheses of object contours, and hence consider *multiple instance regression* (MIR) [49,50], instead of the widely adopted multiple instance classification. Each bag in MIR is associated with a *target value* in the training phase. Ray and Page [49] considered that a bag is well predicted if there is at least one instance whose regression value is close to the target value. On the other hand, Cheung et al. [50] designed their MIR algorithm in the manner that a bag is well predicted if the maximal regression value of all its instances is close to the target value. The proposed AMIR adopts the criterion by Cheung et al. because it fits well to our task. AMIR can derive a robust regressor by taking not only labeled bags but also labeled instances into account. Furthermore, with the introduction of the *smooth maximum function* [37] to handle instances with uncertain labels, AMIR can be more efficiently optimized by simply applying the gradient descent methods.

Chapter 3

Inferring Multiple Tight Segments in a Bounding Box

In this chapter, we represent an algorithm that automatically generates a set of *tight segments* for the bounding box of an object, and at least one of these tight segments would approach the object segment. Our goal in this step is to account for the information asymmetry between an object segment and its bounding box, since the latter can be determined once the former is given, but not vice versa. Specifically, we model the ambiguity in inferring the object segment from a bounding box by generating multiple segment hypotheses. If at least one of them is close to the object segment, the underlying task of inferring the object segment from a bounding box is reduced to a segment selection problem.

In the following, the approach by Lempitsky et al. [2] that yields one tight segment for a given bounding box is first reviewed. We then specify how to generalize their approach to obtain a few tight segments and make sure that at least one of them approaches the object contour.

3.1 Tight Segment via Bounding Box Prior

Let us consider a bounding box $\mathcal{B}$ of an object segment. We start by partitioning $\mathcal{B}$ into *superpixels* by mean-shift [24], which attains a fast and stable over-segmentation. In

practice, the bandwidth parameters in mean-shift algorithm are adjusted by binary search, so that about 50 superpixels are obtained. Let $\mathcal{P}$ denote the set of the superpixels. A figure-ground segmentation or a segment can then be represented by a labeling vector $\mathbf{m} = [m_p] \in \{0,1\}^{|\mathcal{P}|}$, where m_p takes the value 1 if superpixel p belongs to foreground, otherwise 0.

We are particularly interested in *tight segments* within bounding box $\mathcal{B}$. Here a segment is tight with respect to $\mathcal{B}$ if the smallest rectangle covering this segment is $\mathcal{B}$ itself. It is obvious that any non-tight segments won't be the object segment. In [2], Lempitsky et al. introduce the *crossing paths* of a bounding box, and prove that a segment is tight if and only if it intersects all the crossing paths. It turns out that a tight segment $\mathbf{m}$ can be obtained by solving

$$\min_{\mathbf{m}} \quad \sum_{p\in\mathcal{P}} U_p \cdot m_p + \lambda \sum_{(p,q)\in\mathcal{E}} V_{p,q} \cdot |m_p - m_q| \tag{3.1}$$

$$\text{subject to} \quad \forall p \quad m_p \in \{0,1\}, \tag{3.2}$$

$$\forall C \in \Gamma \quad \sum_{p\in C} m_p \geq 1, \tag{3.3}$$

where $\mathcal{E}$ is the set of pairs of adjacent superpixels. The *unary potential* U_p specifies the preference of assigning superpixel p to either foreground or background. The *pairwise potential* $V_{p,q}$ ensures the smoothness between superpixel p and q. The nonnegative coefficient λ controls the importance tradeoff between the unary and pairwise terms. Γ is the set of all the crossing paths of $\mathcal{B}$.

Note that the constraints (3.3) cause that the energy minimization problem (3.1) can no longer be solved by an efficient algorithm, like graph-cut [38]. Thus Lempitsky et al. [2] instead solve a series of its linear relaxation, in which active constraints in (3.3) are added incrementally.

3.2 Multiple Tight Segments

The resulting segment by solving Eq. (3.1) is tightly enclosed by the given bounding box, and hence the aspect ratio of the object is maintained. Due to the unsupervised nature, a satisfactory figure-ground segmentation is not always guaranteed in our empirical test. When addressing bounding boxes of objects with multi-modal color distributions and/or with clutter background, this shortcoming becomes even more evident. Alas, it is usually the case in nowadays benchmark databases of object segmentation, like MSRC-21 [11] or Pascal VOC [33].

We resolve this difficulty by implementing multi-segmentation relaxation. Namely, we generate a few tight segments with different *seeds* [3, 35], and relax the requirement to that at least one of them closely approaches the unknown object segment. It can be observed that apart from the property of tightness, the bounding box of an object also gives two additional hints for discovering the object segment: (1) Its outside borders provide strong cues for identifying the background in the bounding box; (2) It exists a few ROIs that are fully filled by the foreground. If we can retrieve one of them, it helps much in revealing the object segment. Specifically, we maintain the aspect ratio and expand the bounding box by 10%. The *background seeds* are the pixels outside the bounding box and inside the expanded one, i.e., those in the blue highlighted region in Figure 1.3 (a). We sample multiple sets of *foreground seeds* to account for the uncertainty on the locations and scales of those ROIs fully filled by the object. One circular seed region for foreground is constructed for the centroid of each superpixel and with each of predefined radii. The cyan circles in Figure 1.3 show some of the seed regions for foreground.

We leverage the flexibility in developing potential functions $\{U_p\}$ and $\{V_{p,q}\}$ in Eq. (3.1), and derive one tight segment for each set of foreground seeds. A Gaussian mixture model GMM_f with five components is learned with the foreground seeds in RGB color space. Similarly GMM_b is acquired with the background seeds. For each superpixel p,

the unary potential U_p is defined as

$$U_p = \sum_{u \in p} \log P(c_u|GMM_b) - \log P(c_u|GMM_f), \tag{3.4}$$

where u is an image pixel and c_u is its RGB color vector. On the other hand, the pairwise potential $V_{p,q}$ between superpixels p and q is given by

$$V_{p,q} = \sum_{u \in p, v \in q, (u,v) \in \mathcal{N}} \frac{1}{dist(u,v)} \cdot \exp\left(-\beta ||c_u - c_v||^2\right), \tag{3.5}$$

where $\mathcal{N}$ is set of neighboring pixels. We use 8-connected neighbors, and $dist(u, v)$ is the Euclidean distance between pixels u and v. β is a positive constant. One tight segment is inferred by optimizing Eq. (3.1) with these redefined potentials in Eq. (3.4) and Eq. (3.5). The procedure is repeated for each combination of foreground seed regions and parameter settings (λ in Eq. (3.1) and β in Eq. (3.5)). Multiple tight segments of the bounding box are then produced.

An example is shown in Figure 1.3. The left three figures give the bounding box, its representation in superpixels, and the ground truth (GT) respectively. The others are 18 of the yielded tight segments for the bounding box. We evaluate the goodness or the score of a segment, say $\mathbf{m}$, by the following

$$IoU(\mathbf{m}, GT) = \frac{|\mathbf{m} \cap GT|}{|\mathbf{m} \cup GT|} \times 100\,(\%), \tag{3.6}$$

where IoU (intersection over union) is known as the Jaccard index; the binary mask $\mathbf{m}$ indicates each pixel of the bounding box as either foreground or background; and GT is the ground truth. Hereafter, we use IoU as the evaluation metric, and estimate the accuracy of a segment by comparing it with the ground truth. In Figure 1.3, the accuracies of the 18 tight segments in IoU are shown. It can be observed that seed regions of foreground located within the object and with proper radii often lead to satisfactory tight segments.

Since the object must appear in some location of the bounding box with a particular scale, the seeding strategy discovers at least one tight segment close to the ground truth with high chance.

Redundance Removal. Each generated tight segment is parameterized by the location and scale of the seed region for foreground, and the values of λ and β. In our implementation, the number of the tight segments generated for a bounding box is in the order of 10^3. Since many of them are redundant, we develop a $(1-\epsilon)$-approximation procedure to compile the tight segmentations into a smaller set of representative ones. In initialization, all the tight segments are sorted in a queue according to their scores measured by *ratio cut* [30]. We *pop* the first tight segment, add it into the representative set, and remove all the tight segments of more than $1-\epsilon$ overlapping with it from the queue. The process is done repeatedly until the queue is empty. It is obvious that the best tight segment remained in the representative set shares at least $1-\epsilon$ overlapping with the original best one. We empirically set ϵ as 0.05 in all the experiments.

3.3 Problem Definition

After generating multiple segmentation, the problem definition and goal are given before introducing our proposed approach. Suppose a few contours as well as bounding boxes have been annotated in the images of an object class, say `kitty` in Figure 1.3 (a). We crop the ROIs of the contours as well as the bounding boxes in the images. For ROIs of the object contours, we denote them by $L = \{(\mathcal{B}_i, GT_i)\}_{i=1}^{\ell}$, where ℓ is the number of the contours. $\mathcal{B}_i$ is the ROI of the ith labeled object, while GT_i is the ground truth in form of the figure-ground map, as the one shown in Figure 1.3 (c). As for the ROIs of the bounding boxes, we have $U^+ = \{\mathcal{B}_i\}_{i=\ell+1}^{\ell+u^+}$, where u^+ is the number of the positive bounding boxes of the objects, and $\mathcal{B}_i$ is a positive bounding box. Note that the ground truth of the object segment in U^+ is not available. Typically, we have much more positive

bounding boxes than labeled object contours, i.e., $u^+ \gg \ell$. In most of our experiments, we set $u^+ = 9\ell$. For each ROI in $L \cup U^+$, we partition it into superpixels by mean-shift, and generate a set of tight segments by the aforementioned approach.

The proposed AMIR approach works with three kinds of data, including labeled object contours, positive and negative bounding boxes. We describe the three kinds of data, give the notations, and specify the goal of this thesis in the following.

3.3.1 Object Contours

After generating the tight segments for each $(\mathcal{B}_i, GT_i) \in L$, we have $(\mathcal{B}_i, GT_i) = \{(\mathbf{x}_{ij}, y_{ij})\}_{j=1}^{N_i}$, where N_i is number of the generated tight segments for the ith labeled object, and $\mathbf{x}_{ij}$ is the feature representation of the jth tight segment, which will be described in Chapter 5. Since ground truth GT_i is available, the goodness of segment $\mathbf{x}_{ij}$, $y_{ij} \in [0, 1]$, can be computed via Eq. (3.6), and y_{ij} will be the target value for $\mathbf{x}_{ij}$ in our regression problem.

3.3.2 Positive Bounding Boxes

By applying the tight segment generation algorithm to each $\mathcal{B}_i \in U^+$, we have $\mathcal{B}_i = \{\mathbf{x}_{ij}\}_{j=1}^{N_i}$, where N_i is the number of the yielded tight segments, and $\mathbf{x}_{ij}$ is the feature vector of the jth tight segment. Note that the target value of $\mathbf{x}_{ij}$ is not available due to the lack of ground truth. Nevertheless, as we mentioned previously that at least one tight segment is close to the ground truth, the maximal regression value of $\{\mathbf{x}_{ij}\}_{j=1}^{N_i}$ is supposed to approach 1. We will exploit this property to regularize the learning of the regressor, and alleviate the problems caused by insufficient labeled training data.

3.3.3 Negative Bounding Boxes

While the object contours in L and the positive bounding boxes in U^+ are manually labeled, the set of negative bounding boxes can be automatically generated. Consider an

image $\mathcal{I}$ from which one or multiple positive bounding boxes are cropped. We randomly sample a few bounding boxes in $\mathcal{I}$, and consider a sampled bounding box, say $\mathcal{B}_N$, as negative if there is no sufficient overlap between $\mathcal{B}_N$ and all the positive bounding boxes in $\mathcal{I}$. That is, $\mathcal{B}_N$ satisfies the following criterion in our implementation:

$$IoU(\mathcal{B}_P, \mathcal{B}_N) \leq 50\%, \forall \mathcal{B}_P \text{ in } \mathcal{I}, \tag{3.7}$$

where function IoU is defined in Eq. (3.6).

There are often numerous bounding boxes that satisfy Eq. (3.7) in $\mathcal{I}$. In the empirical test, negative bounding boxes with similar areas and aspect ratios to the positive bounding boxes give the most information for contour inference. Thus for each positive bounding box $\mathcal{B}_P$ in $\mathcal{I}$, we randomly sample at most 30 negative bounding boxes with the same size as $\mathcal{B}_P$. The procedure is repeated for each $\mathcal{B} \in U^+$. The tight segments of each negative bounding box are also compiled. It follows that a set of negative bounding boxes is collected and can be represented as $U^- = \{\mathcal{B}_i\}_{i=\ell+u^++1}^{\ell+u^++u^-}$, where $\mathcal{B}_i = \{\mathbf{x}_{ij}\}_{j=1}^{N_i}$ and $u^- \simeq 30u^+$. Although the tight segments of the negative bounding boxes don't contain complete objects, they hold effective information to identify the bad tight segments generated from positive bounding boxes owing to the common background (and the incomplete object foreground).

Given L, U^+, and U^-, our goal is to learn a regressor f, which can effectively pick the most plausible tight segment for each positive bounding box. Then the picked tight segments are used in place of the manually drawn contours, and serve as the training data in semantic segmentation for saving manual annotation efforts. Thus, the goal of our algorithm is to predict the goodness values of tight segments and pick up the one closest to the unknown ground truth.

Chapter 4

The Proposed AMIR approach

The proposed AMIR approach jointly works with three different sources of data, L, U^+, and U^-, and adopts tight segments as the unified basis, upon which knowledge derived from the three sources can be mutually transferred and beneficial. AMIR performs least square fitting over the tight segments in L, while carrying out multiple instance regression for bounding boxes in $U^+ \cup U^-$. In this way, positive (negative) bounding boxes are treated as positive (negative) bags with their tight segments as unlabeled instances. Besides, the tight segments in L are considered as the labeled instances, which are used as augmented information in learning the regressor.

In the following, we first depict the least square fitting over L. Then, multiple instance regression over $U^+ \cup U^-$ is specified, followed by the derivation on how the regressor in AMIR can be efficiently optimized by simply using gradient descent methods. Finally, the tight segments with the highest regression values in positive bounding boxes are selected and used as training data for semantic segmentation.

4.1 On Least Square Fitting over L

We first consider a linear regressor, $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, to perform least square fitting over all the tight segments in $L = \{(\mathbf{x}_{ij}, y_{ij})\}_{i=1,j=1}^{\ell,N_i}$. For compact representation, we let $\mathbf{x}_{ij} \leftarrow [\mathbf{x}_{ij} \;\; 1]^\top$ and $\mathbf{w} \leftarrow [\mathbf{w} \;\; b]^\top$ such that regression can be represented as $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$.

The linear regressor f can be derived by minimizing the l_2-regularized squared error, i.e.,

$$\mathbf{w} = \arg\min_{\mathbf{w}} \lambda_1 J_1(\mathbf{w}) + \|\mathbf{w}\|^2, \tag{4.1}$$

$$\text{where } J_1(\mathbf{w}) = \frac{1}{N_L} \sum_{i=1}^{\ell} \sum_{j=1}^{N_i} \|\mathbf{w}^\top \mathbf{x}_{ij} - y_{ij}\|^2 \text{ and } N_L = \sum_{i=1}^{\ell} N_i. \tag{4.2}$$

In Eq. (4.1), $\|\mathbf{w}\|^2$ is the regularization term that penalizes the parameter set $\mathbf{w}$ with a large norm, and λ_1 is a positive constant controlling the trade-off between the fitness and regularization terms.

Steepest gradient descent is one efficient and effective way to optimize Eq. (4.1) with derivatives

$$\frac{\partial \lambda_1 J_1(\mathbf{w}) + \|\mathbf{w}\|^2}{\partial \mathbf{w}} = \frac{\lambda_1}{N_L} 2(XX^\top \mathbf{w} - XY^\top) - 2\mathbf{w}, \tag{4.3}$$

where $X = [\mathbf{x}_{11} \ \cdots \ \mathbf{x}_{ij} \ \cdots \ \mathbf{x}_{\ell N_\ell}] \in \mathbb{R}^{D \times N_L}$ is the instance matrix where D is the dimensionality of the features, and $Y = [y_{11} \ \cdots \ y_{ij} \ \cdots \ y_{\ell N_\ell}] \in \mathbb{R}^{1 \times N_L}$ is the target value vector.

We could use the learned regressor f in Eq. (4.1) to regress tight segments in the positive bounding boxes, and select tight segments with high regression values to serve as training data for semantic segmentation. However, two unfavorable effects may occur in this case, and lead to suboptimal performance. First, there are only a few data used for deriving the regressor f. The learned parameters $\mathbf{w}$ are hence at a high risk of *overfitting*. Second, if *large intra-class variation* exists among the object segments in $L \cup U^+$, the regressor f learned with L will not predict the tight segments in U^+ well. Unfortunately, it is usually the case in nowadays benchmarks of semantic segmentation, such as Pascal VOC [33]. We have to address these two unfavorable effects to ensure the quality of the selected tight segments.

4.2 On Multiple Instance Regression over $U^+ \cup U^-$

We resolve the two aforementioned problems, overfitting and large intra-class variation, by taking the bounding boxes of $U^+ \cup U^-$ into account. First, the abundant bounding box data can be exploited to guide and regularize the training process of the regressor. It compensates for the lack of the labeled training data, and alleviates the unfavorable effect of overfitting. Second, our goal is to predict the goodness of tight segments in positive bounding boxes. By introducing U^+ into the training process, the regressor is derived with the access to the tight segments in U^+. It will relieve the problems caused by large intra-class variation, since the tight segments to be predicted are included in the training process.

As mentioned previously, the bounding boxes and tight segments in this case respectively meet the definitions of bags and instances in multiple instance learning. Namely, at least one tight segment in a positive bounding box is positive, while all the tight segments in a negative bounding boxes are negative. We formulate it as a problem of multiple instance regression, and couple it with least square fitting. To begin with, we define the regression value of a bag (bounding box) $\mathcal{B}_i = \{\mathbf{x}_{ij}\}_{j=1}^{N_i} \in U^+ \cup U^-$ as

$$R(\mathcal{B}_i) = \max_j f(\mathbf{x}_{ij}) \tag{4.4}$$

$$= \max_j \mathbf{w}^\top \mathbf{x}_{ij}. \tag{4.5}$$

It is shown in Eq. (4.5) that the regression value of a bag is the maximum of the regression values of its instances. For each positive bag $\mathcal{B}_i \in U^+$, its regression value is supposed to be close to 1. We adopt the log-loss function to enforce this property in learning the regressor, i.e.,

$$J_2(\mathbf{w}) = \frac{1}{u^+} \sum_{\mathcal{B}_i \in U^+} \log\left(1 + \exp\left(1 - R(\mathcal{B}_i)\right)\right), \tag{4.6}$$

where u^+ is number of bags in U^+. The log-loss function in Eq. (4.6) will penalize the positive bags whose regression values do not approach 1. We here use the log-loss function, instead of l_1- or l_2-error, because it is more robust to noisy data and outliers. According to our empirical test, the log-loss function gives much better performance than l_1- and l_2-error in our approach to tight segment regression and semantic segmentation.

For negative bags in U^-, the loss function can be oppositely designed as

$$J_3(\mathbf{w}) = \frac{1}{u^-} \sum_{\mathcal{B}_i \in U^-} \log\left(1 + \exp\left(R(\mathcal{B}_i) - \tau_i\right)\right), \tag{4.7}$$

where u^- is number of bags in U^-. As shown in Eq. (4.7), the negative bags are penalized if their regression values are not small enough. Note that we give an upper bound τ_i for each bag $\mathcal{B}_i$ rather than force its regression value to approach zero. We set τ_i as the highest overlap ratio between $\mathcal{B}_i$ and a positive bounding box. Here the overlap ratio is defined as the first term in Eq. (3.7).

By jointly considering positive and negative bounding boxes, the induced optimization problem of multiple instance regression in this thesis becomes

$$\mathbf{w} = \arg\min_{\mathbf{w}} \lambda_2 J_2(\mathbf{w}) + \lambda_3 J_3(\mathbf{w}) + \|\mathbf{w}\|^2. \tag{4.8}$$

Notice that the $\max$ operator in Eq. (4.5) makes the optimization problem in Eq. (4.8) non-differentiable, and gradient descent methods are no longer applicable. It makes the coupling of least square fitting over L and multiple instance regression over $U^+ \cup U^-$ difficult. To address this problem, we introduce the *softmax activation function* (or the log-sum-exp trick) [37] to give a differentiable surrogate of $R(\mathcal{B}_i)$ in Eq. (4.5) by

$$R(\mathcal{B}_i) = \max_j \mathbf{w}^\top \mathbf{x}_{ij} \approx \frac{1}{\gamma} \log\left(\sum_j \exp\left(\gamma \mathbf{w}^\top \mathbf{x}_{ij}\right)\right), \tag{4.9}$$

where the *smoothness parameter* γ is a positive constant, and is used to control the degree of precision in approximation. We use γ and the exponential function to scale up

the values so that the largest one dominates the rest. It can be verified that, if γ is large enough, $\max_j \exp(\gamma \mathbf{w}^\top \mathbf{x}_{ij}) \approx \sum_j \exp(\gamma \mathbf{w}^\top \mathbf{x}_{ij})$ and there is a unique maximum. By taking the log and dividing both sides by γ, we have Eq. (4.9). Larger γ yields more precise approximation, but may result in the overflow problem in implementation. We empirically set $\gamma = 2$, which provides sufficiently good approximation in our dataset. Hereafter, we re-define $R(\mathcal{B}_i)$ as the differentiable approximation in Eq. (4.9).

The optimization problem in Eq. (4.8) is now differentiable, and can be efficiently solved by gradient descent methods. In the following, we give the derivatives with respect to the set of the optimization variables $\mathbf{w}$. We start by showing the gradients of $R(\mathcal{B}_i)$, the common component in $J_2(\mathbf{w})$ and $J_3(\mathbf{w})$, as below:

$$\frac{\partial R(\mathcal{B}_i)}{\partial \mathbf{w}} = \sum_j \kappa_{ij} \mathbf{x}_{ij}, \tag{4.10}$$

$$\text{where } \kappa_{ij} = \frac{\exp(\gamma \cdot \mathbf{w}^\top \mathbf{x}_{ij})}{\sum_{j'} \exp(\gamma \cdot \mathbf{w}^\top \mathbf{x}_{ij'})}. \tag{4.11}$$

It can be observed in Eq. (4.10) that the derivative of $R(\mathcal{B}_i)$ w.r.t. $\mathbf{w}$ is a convex combination of its associated instances. The derivative of $J_2(\mathbf{w})$ is then derived by using the chain rule, and we show it as follows:

$$\frac{\partial J_2(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{u^+} \sum_{\mathcal{B}_i \in U^+} \frac{-\exp(1 - R(\mathcal{B}_i))}{1 + \exp(1 - R(\mathcal{B}_i))} \times \frac{\partial R(\mathcal{B}_i)}{\partial \mathbf{w}} \tag{4.12}$$

$$= \frac{1}{u^+} \sum_{\mathcal{B}_i \in U^+} \underbrace{\frac{-\exp(1 - R(\mathcal{B}_i))}{1 + \exp(1 - R(\mathcal{B}_i))}}_{\text{bag term}} \times \underbrace{\sum_j \kappa_{ij} \mathbf{x}_{ij}}_{\text{inst. term}}. \tag{4.13}$$

The derivative of $J_3(\mathbf{w})$ with respect to $\mathbf{w}$ is similarly obtained

$$\frac{\partial J_3(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{u^-} \sum_{\mathcal{B}_i \in U^-} \underbrace{\frac{\exp(R(\mathcal{B}_i) - \tau_i)}{1 + \exp(R(\mathcal{B}_i) - \tau_i)}}_{\text{bag term}} \times \underbrace{\sum_j \kappa_{ij} \mathbf{x}_{ij}}_{\text{inst. term}}. \tag{4.14}$$

With the derivatives of $J_2(\mathbf{w})$ and $J_3(\mathbf{w})$, the problem of multiple instance regression

in Eq. (4.8) can be optimized by simply using gradient descent methods.

The derivatives in Eq. (4.13) and Eq. (4.14) have intuitive meanings. For a bag $\mathcal{B}_i$ in either Eq. (4.13) or Eq. (4.14), the corresponding derivative is the product of two terms: the instance term and the bag term. For the instance term, it is a weighted combination of all the instances in $\mathcal{B}_i$. The weights $\{\kappa_{ij}\}_{j=1}^{N_i}$ in Eq. (4.11) are non-negative, and sum to one, $\sum_j \kappa_{ij} = 1$. The κ_{ij} grows exponentially as the regression value of the jth instance increases. It indicates that the instance terms in both Eq. (4.13) and Eq. (4.14) put emphasis on the instance with the largest regression value in $\mathcal{B}_i$. It is consistent with the rationale of MIR, since instances with the largest regression values in the bags are more relevant to the determination of the regressor. On the other hand, the bag term of positive bag $\mathcal{B}_i$ in Eq. (4.13) is negative, and that of a negative bag in Eq. (4.14) is positive. While their signs control the directions in the procedure of gradient descent, the magnitudes of the bag terms in both Eq. (4.13) and Eq. (4.14) show that the algorithm will focus on the training bags that are not well predicted by the current regressor $\mathbf{w}$. Namely, the positive bags whose regression values do not approach 1 or the negative bags whose regression values are larger than the corresponding upper bounds. In brief, our MIR part weights the bags as well as its instances in gradients, and uses weighted data to conduct the subsequent learning.

By jointly considering the least square fitting over L in Eq. (4.1) and multiple instance regression over $U^+ \cup U^-$ in Eq. (4.8), the optimization problem of the proposed AMIR is defined as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \lambda_1 J_1(\mathbf{w}) + \lambda_2 J_2(\mathbf{w}) + \lambda_3 J_3(\mathbf{w}) + \|\mathbf{w}\|^2, \tag{4.15}$$

where $J_1(\mathbf{w})$, $J_2(\mathbf{w})$, and $J_3(\mathbf{w})$ are given in Eq. (4.2), Eq. (4.6), and (4.7), respectively.

Since all terms in Eq. (4.15) are differentiable, we use steepest gradient descent to optimize AMIR for its simplicity. The optimization variable $\mathbf{w}$ is set as the zero vector in initialization, and the optimization is done until convergency of the objective values. The

parameters, λ_1, λ_2, and λ_3, are determined by five-fold cross validation in the experiments.

After completing the optimization of Eq. (4.15), we can pick the most plausible tight segments in positive bounding boxes. Specifically, for each $\mathcal{B}_i = \{\mathbf{x}_{ij}\}_{j=1}^{N_i}$ in U^+, we pick its j^*th tight segment, where

$$j^* = \arg\max_j \mathbf{w}^\top \mathbf{x}_{ij}. \tag{4.16}$$

We can complete the collection of a set of training data $D = L \cup \tilde{U}^+$, where $L = \{GT_i\}_{i=1}^{\ell}$, $\tilde{U}^+ = \{TS_{ij^*}\}_{i=\ell+1}^{\ell+u^+}$, and TS_{ij^*} is the picked tight segment in the ith positive bounding box. D can then be used as the input to any of the off-the-shelf semantic segmentation methods. The training and test procedures of AMIR are summarized in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 AMIR Training Procedure

Input: Labeled object contours L; Positive bounding boxes U^+; Negative bounding boxes U^-;

1: **Initialization**: $\mathbf{w} \leftarrow \mathbf{0}$;
2: **repeat**
3: Compute gradients $\{\frac{\partial J_i(\mathbf{w})}{\partial \mathbf{w}}\}_{i=1}^{3}$ via Eq. (4.3), (4.13) and (4.14), respectively;
4: Compute the step size α by using line search;
5: Update $\mathbf{w} \leftarrow \mathbf{w} - \alpha(\sum_{i=1}^{3} \frac{\partial J_i(\mathbf{w})}{\partial \mathbf{w}} + 2\mathbf{w})$;
6: **until** Convergence

Output: Regressor $f(\mathbf{x}) = \mathbf{w}^{*\top}\mathbf{x}$;

Algorithm 2 AMIR Test Procedure

Input: Labeled object contours L; Positive bounding boxes U^+; Learned regressor $f(\mathbf{x}) = \mathbf{w}^{*\top}(\mathbf{x})$;

1: $D \leftarrow L$;
2: **for** each positive bounding box $\mathcal{B}_i \in U^+$ **do**
3: Pick the best tight segment, TS_{ij^*}, using Eq. (4.16);
4: $D \leftarrow D \cup TS_{ij^*}$;
5: **end for**

Output: Training dataset D for semantic segmentation;

Chapter 5

Tight Segment Feature Representation

We give a brief introduction to the adopted features for representing the yielded tight segments. The features are roughly divided into two categories: the segment-level and the pixel-level features. The segment-level features are designed to characterize the mid-level structural and context information regarding the segments. The pixel-level features instead capture the low-level visual cues of pixels within the segments, and explore the distributions of local evidences. The two types of features are complementary, and jointly adopted to account for the appearance variation of object segments.

5.1 Segment-Level Features

We implemented a set of segment-level features, most of which are suggested in [35], for characterizing each tight segment, including

- *Percentage of boundary pixels (one dim)*: the ratio of the number of boundary pixels to the number of foreground pixels. It measures how complicated the shape of a segment is.

- *Boundary edge strength (one dim)*: the edge strengths along the object contour. We used Canny edge detector to evaluate the edge strengths of the pixels. A good tight segment is supposed to have higher boundary edge strength.

- *Centroid (two dims)*: the normalized coordinates of the mass center of the segment. Each object class may have its own particular distribution of the segment centroids.

- *Major and minor axis length (two dims)*: we used an ellipse to approximate a segment, and selected normalized lengths of its major and minor axes as the features. They measure the aspect ratio of a segment.

- *Convexity and area (two dims)*: the ratios of the number of foreground pixels to the area of the convex hull and to the whole bounding box. They describe geometric properties of segments.

- *Foreground and background dissimilarity (three dims)*: the dissimilarity is respectively measured by three kinds of features, including RGB, *SIFT* [51, 52], and *Texton* [11, 20, 53]. By using the BoW (*Bag-of-Words*) model with 500 visual words, a pair of histograms, one for foreground and one for background, over the visual words are generated for each feature. The χ^2 distance is employed as the dissimilarity measure.

In our implementation, `MATLAB` function `regionprops` was used to extract the first four types of the segment-level features, while the packages provided in [52] and [53] were used to compute SIFT and Texton features, respectively. We concatenate all the segment-level features into an 11-dimensional feature vector for each tight segment.

5.2 Pixel-Level Features

The segment-level features capture the global and structural properties of segments. They don't make the most of information provided by the associated bounding boxes. The pixels outside the bounding boxes, i.e., the background pixels, carry the rich information to identify background pixels inside the bounding boxes, and are hence helpful for evaluating the quality of a tight segment. To this end, we develop pixel-level features to exploit

this evidence.

For each bounding box, we expand its region by 50% without changing the aspect ratio, and collect pixels that are inside the extended bounding box but outside the original one. A GMM model $\mathbf{g}$ is fit to the RGB features of the collected pixels. The pixels within the bounding box are then sorted according to their probabilities measured by $\mathbf{g}$. Like [2], two GMMs $\mathbf{f}$ and $\mathbf{b}$ are learned with the last 33% and the first 33% of the sorted pixels respectively. The probability of each pixel belonging to foreground and background can be estimated by $\mathbf{f}$ and $\mathbf{b}$, respectively. We compute the *relative probability* [35] of pixel i as $p(x_i) = \ln(p_{\mathbf{f}}(x_i)) - \ln(p_{\mathbf{b}}(x_i))$. For each tight segment of the bounding box, we can obtain its mean relative probability by averaging the relative probabilities of all pixels within the tight segment. The mean relative probability is then taken as one feature of the tight segment.

To exploit other kinds of pixel features, the aforementioned procedure is repeated by changing the pixel features from RGB color vectors to SIFT [51, 52] and Texton [11, 20, 53] respectively. Thus, for each tight segment, there are totally three pixel-level features which are the mean relative probabilities for RGB, SIFT and Texton.

To sum up, a tight segment is described by a 15-dimensional feature vector, in which 11 dimensions are for segment-level features, 3 for pixel-level features, and one additional dimension for a bias term.

Chapter 6

Experiment Results

We designed three sets of experiments to evaluate the proposed approach on Pascal VOC segmentation task [33]. In the first experiment, we checked whether our approach can generate at least a tight segment close to the ground truth in each positive bounding box, since the MIR formulation is established upon this assumption. Second, AMIR was evaluated by measuring the IoU scores of the tight segments picked by the derived regressor. Third, we assessed if the picked tight segments by AMIR can replace the manually labeled contours in semantic segmentation, the underlying goal of this thesis.

6.1 Dataset: Pascal VOC 2007

The Pascal VOC 2007 Segmentation Challenge is composed of 20 object classes. Each object category contains about $30 \sim 100$ annotated objects, except the class of `person`, which has 345 ones. The dataset consists of highly deformable objects with large intra-class variation, and results in substantial annotation costs for manual contour labeling. Thus, it serves as a good test bed for verifying the effectiveness of our approach. In view of that most segmentation algorithms are often sensitive to image resolutions, we computed the corresponding bounding box for each annotated object contour. Then, we excluded the bounding boxes whose resolutions are less than $2,000$, and resized each of

the rest to around $80,000$ pixels, without altering their aspect ratios.

6.2 Experiment I: Multiple Tight Segment Generation

The effectiveness of multiple segmentation strategy lives with the underlying assumption that there is at least one tight segment close to the object segment. To inspect if this assumption is valid, we cropped the bounding box of each annotated object segment in the training and validation sets in Pascal VOC 2007, and generated a set of tight segments for the bounding box using the approach in Chapter 3. The resulting tight segments are compared to the annotated ground truth via Eq. (3.6). The IoU of the best tight segments indicates the theoretical upper bound on the performance of our AMIR method.

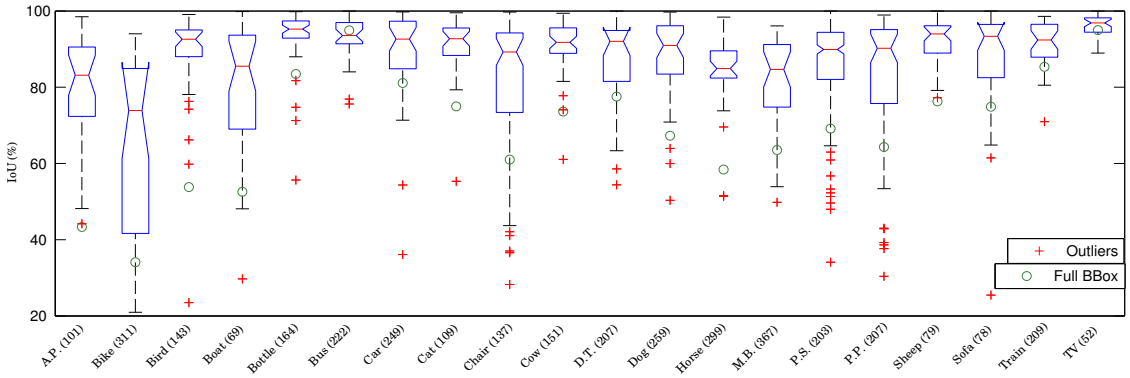

Figure 6.1: The IoU distributions of the best tight segments w.r.t. each of the 20 object classes. The edges of each box are the 25th and 75th percentiles, while the red line indicates the median. Outliers are marked as red-cross signs. The green cycle denotes the median IoU of the baseline that treats full bounding boxes as objects. The average number of the generated tight segments is given in parentheses.

Figure 6.1 depicts the distributions of the IoU scores of the best tight segments for each of the 20 object classes using `MATLAB's boxplot`. The average number of the generated tight segments in the positive boxes of each class is also reported in parenthesis in Figure 6.1. The number is with the range between 52 and 367. The IoU of the best tight segment for each bounding box certainly depends on the complexity of object appearance and the foreground/background discernibility. Nevertheless, it can be found that most bounding boxes hold *good* tight segments with their median IoU higher than 80%, except

for those of the class bike. Even for bike, the median IoU is still higher than 70%. The experiment shows there is a good chance to find a satisfying tight segment from the generated candidate set for each bounding box. As a reference, we consider the baseline that treats the whole bounding boxes as objects, and show the median IoU scores of the baseline in Figure 6.1. It is obvious that the baseline only works well on the classes where objects can be approximated by rectangles, e.g., bus and TV. The baseline performs poorly on the rest.

Figure 6.2 shows several examples of the object bounding boxes as well as the generated tight segments by our approach for visual assessment. The best yielded tight segments are given in the third column. It can be observed that our approach to multiple tight segment generation still works well even on highly deformable objects with clutter background.

6.3 Experiment II: AMIR for Tight Segment Selection

The experiment II was designed to assess the quality of the selected tight segments by AMIR. The proposed AMIR learned a regressor for each of the 20 object categories in Pascal VOC 2007 segmentation task. For each category, randomly selected 10% of bounding boxes come with the ground truth (object contours), i.e., L, while the rest were treated as positive bounding boxes, i.e., U^+, and their object contours were assumed to be unknown. We randomly generated at most 30 negative bounding boxes around each positive bounding box, and collected all the negative bounding boxes to yield U^-. AMIR was used to infer the object contour enclosed by each positive bounding box. Similar to the experiment I, the quality of the inferred tight segment is evaluated with the IoU score. For performance comparison, we implemented eight baselines of the following three categories.

Single image figure-ground segmentation. Methods in this category perform figure-ground segmentation for contour estimation by considering a positive bounding box at a

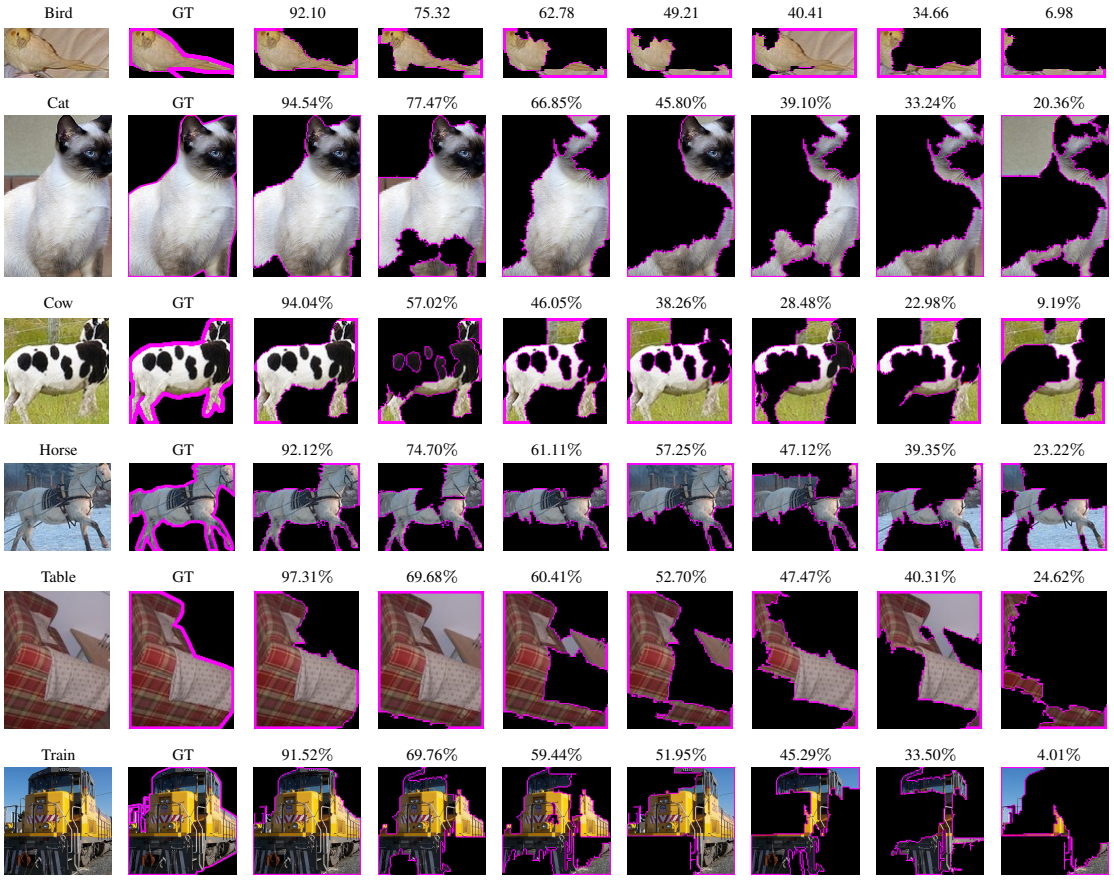

Figure 6.2: Several examples of the generated tight segments by our approach. (a) The bounding box of an object. (b) The ground truth. (c) The best tight segment. (d) $\sim$ (i) Some of the other generated tight segments. The IoU of each tight segment is also reported.

time. Specifically, we adopted the following four approaches, each of which is denoted below in bold and in abbreviation:

- **GrabCut**: GrabCut [1] works with initial foreground/background models. Here the foreground model is initialized with the whole bounding box, while the background model is fitted from the region outside the bounding box.

- **TS** (Tight Segments): Following [2], bounding box prior is integrated into figure-ground segmentation. It ensures that the resulting foreground segment is tight with respect to the given object bounding box.

- **OP** (Object Proposals): Endres & Hoiem [3] also proposed a two-stage process, in which a set of object proposals were produced. A pretrained regressor is used to

rank these proposals and pick the best one.

- **FG** (F-G Classification): The approach by Chen et al. [4] estimates the background and various foreground priors. Different contour hypotheses are generated by changing the foreground priors, and the one that maximizes the score of segmentation quality is selected.

Class-based object contour estimation. Methods of this category work by considering all the bounding boxes of an object class simultaneously. Thus, the class-specific knowledge can be derived to benefit object contour estimation. Specifically, we implemented the following two approaches:

- **DCCoSeg** (Discriminative Clustering for Co-Segmentation): We implemented the discriminative clustering algorithm by Joulin et al. [5] to jointly segment out the objects enclosed by the positive bounding boxes.

- **SVR** (Support Vector Regression): Our prior work [42] learns a regressor by considering both the object contours and the positive bounding boxes at the same time i.e., $L \cup U^+$. It formulates the task of object contour estimation as a support vector regression problem.

The variants of AMIR. We consider two degenerate variants of AMIR, and describe them as follows:

- **LR** (Linear Regression): The linear regressor solves the optimization problem in Eq. (4.1), in which only data in form of object contours, i.e., L, are considered.

- **MIR** (Multiple Instance Regression): The MIR regressor solves the optimization problem in Eq. (4.8), in which only data in form of bounding boxes, i.e., $U^+ \cup U^-$, are considered.

We applied AMIR and the eight baselines to infer the object contours in the positive bounding boxes for the 20 object classes. For each method, TABLE 6.1 reports the mean

IoU score for each class and the average mean IoU score of all classes (the second column). Because GrabCut, TS, OP, and FG, work on a single bounding box where only restricted information is accessible, they often result in suboptimal performance. On the other hand, the two class-based methods, DCCoSeg and SVR, give diverse IoU scores. SVR jointly utilizes labeled object contours and object bounding boxes, and leads to good results. DCCoSeg seeks object contours that share common appearance. However, this assumption may not hold, since there exist large intra-class variations in Pascal VOC. It is interesting that LR and MIR achieve similar average mean IoU scores, but their class-wise IoU scores are differently distributed. It means object contours, used in LR, and bounding boxes, used in MIR, often carry complementary information. The proposed AMIR significantly and consistently outperforms LR and MIR. It indicates that AMIR can effectively make the most of both types of information, leading to promising results. In addition, AMIR outperforms all baselines in terms of the average IoU.

Table 6.1: The Mean IoU (%) of The Inferred Segments by Various Approaches on Pascal VOC segmentation task.

	Avg.	A.P.	Bike.	Bird	Boat	Bottle.	Bus	Car	Cat	Chair	Cow	D.T.	Dog	Horse	M.B.	P.S.	P.P.	Sheep	Sofa	Train	Tv
GrabCut [1]	64.90	52.89	39.61	63.98	51.17	72.86	79.25	66.21	72.07	51.68	66.26	70.47	69.45	62.52	64.12	60.05	**65.59**	73.68	69.15	73.24	83.65
TS [2]	62.80	63.57	**52.44**	71.69	**64.66**	64.28	64.99	58.23	63.26	55.17	67.91	61.14	66.65	**66.34**	59.70	60.82	63.60	69.64	53.55	64.60	64.09
OP [3]	55.16	55.42	38.06	51.60	41.00	65.50	64.59	41.90	54.65	55.46	67.88	48.97	55.47	60.79	63.10	49.89	40.47	63.90	62.69	61.61	60.13
FG [4]	53.77	**63.79**	44.57	**73.31**	50.29	61.21	60.50	43.58	65.50	43.43	49.00	51.76	67.32	63.95	**67.94**	34.66	54.11	52.66	44.16	57.35	26.38
SVR [42]	67.26	56.81	32.33	64.22	52.58	**85.49**	89.30	**78.69**	70.66	47.47	63.03	72.71	73.81	55.81	65.49	64.62	53.30	72.17	74.62	80.05	92.10
DCCoSeg [5]	53.35	47.05	38.08	42.38	55.06	68.38	63.90	42.69	50.95	56.14	66.35	48.99	48.02	58.08	56.33	42.19	46.01	64.90	54.25	54.25	62.98
LR	60.87	54.43	33.26	64.60	52.21	70.66	64.65	56.75	74.04	60.72	64.75	65.83	58.56	52.35	48.43	67.30	60.28	56.05	72.14	56.12	84.31
MIR	62.91	44.92	37.64	51.55	45.43	62.74	87.02	71.99	73.56	57.93	73.17	73.30	64.43	43.68	62.51	65.92	60.40	71.65	71.56	62.25	75.47
AMIR	**72.00**	58.57	41.04	72.71	60.44	85.02	**89.64**	78.61	**76.62**	**62.06**	**79.14**	**74.58**	**73.93**	60.76	66.98	**67.85**	61.27	**79.93**	**76.44**	**82.02**	**92.47**

Figure 6.4 shows the IoU distributions of the inferred object segments by AMIR for the twenty object classes (using `MATLAB's boxplot`). Along with the distributions, the median IoU scores of GrabCut [1], TS [2], OP [3], FG [42], LR, and MIR are also plotted in Figure 6.4. It can be seen that AMIR performs best in most classes.

For visual assessment, Figure 6.3 gives some of the inferred object segments by various approaches. The object segments by AMIR are very close to the ground truth no matter the objects are rigid or highly deformable. This indicates that manual annotation can be replaced with the inferred segments by AMIR with low quality degradation but

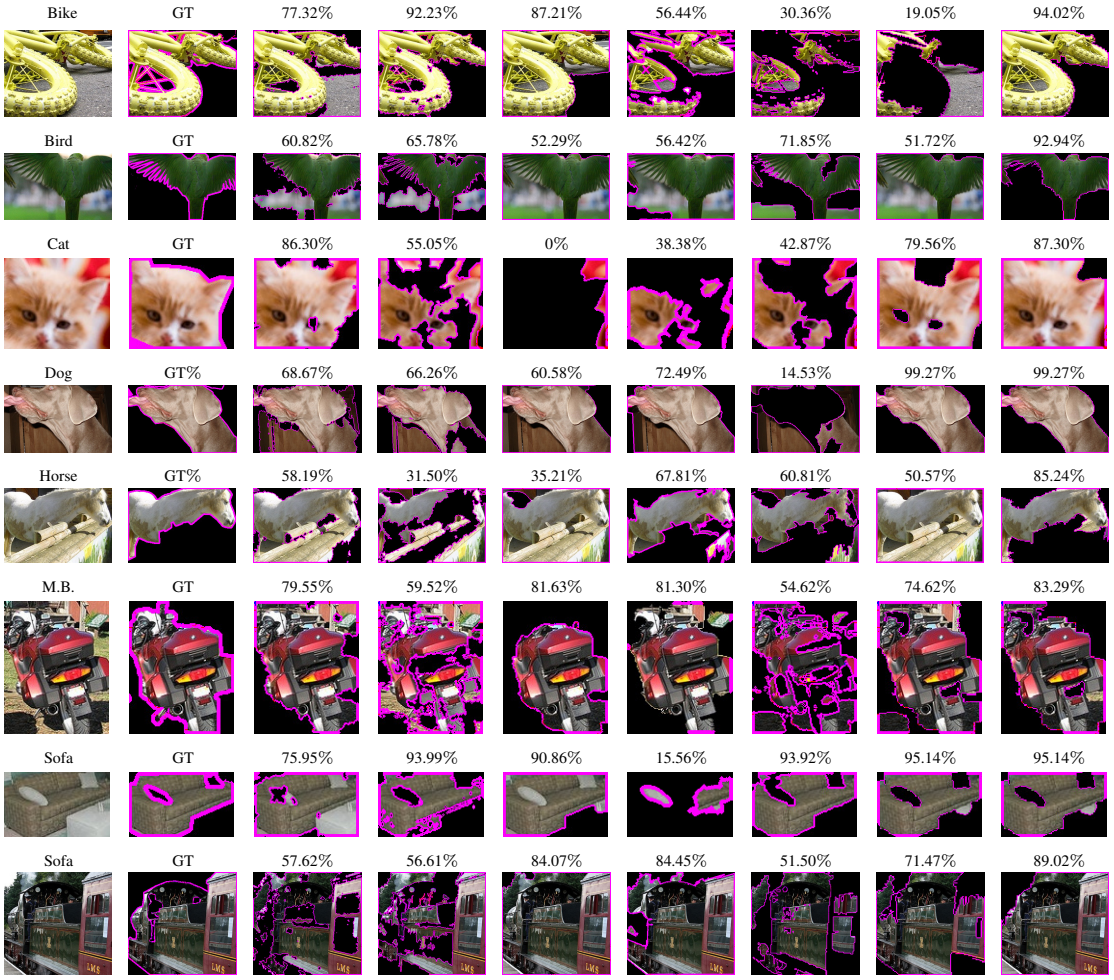

Figure 6.3: Inferred object contours and their IoU scores by various approaches. (a) Bounding box. (b) Ground truth. (c) GrabCut [1]. (d) TS [2]. (e) OP [3]. (f) FG [4]. (g) DCCoSeg [5]. (h) LR. (i) AMIR.

with high annotation cost reduction.

The quantitative and qualitative results in TABLE 6.1, Figure 6.4, and Figure 6.3 are measured when 10% of the object contours in Pascal VOC are used as L, and the bounding boxes of the rest 90% form U^+. We evaluated the performance of AMIR with different fractions of labeled object contours. To this end, we randomly divided the object contours in Pascal VOC into two disjoint subsets. One consists of $k\%$ of the object contours, and serves as L. The rest yields U^+. We respectively set $k = 1, 3, 5, 10, 20, 35$ and 50, and report the performances of LR and AMIR in Figure 7. As references, the performance upper bounds, i.e., the IoU scores of the best tight segments, are included in the figure. It can be observed that AMIR consistently outperforms LR especially when k is small. In

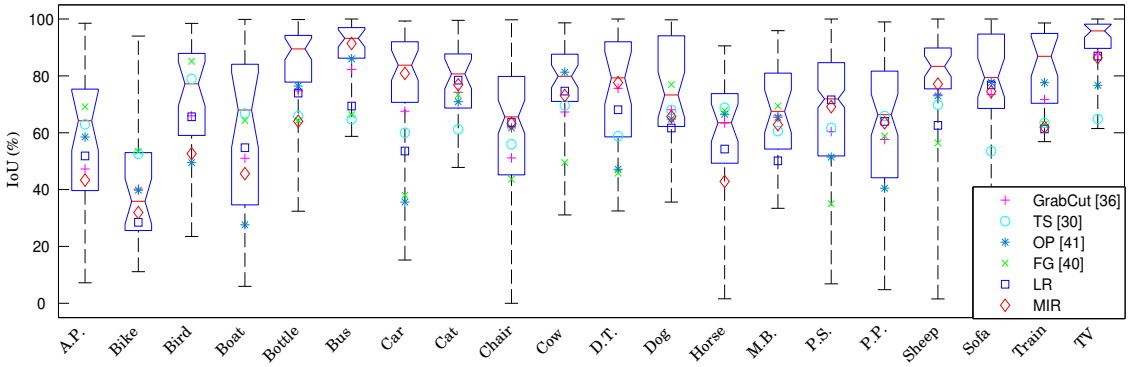

Figure 6.4: The IoU distributions of AMIR (displayed by `boxplot`) as well as the median IoU scores of other baselines.

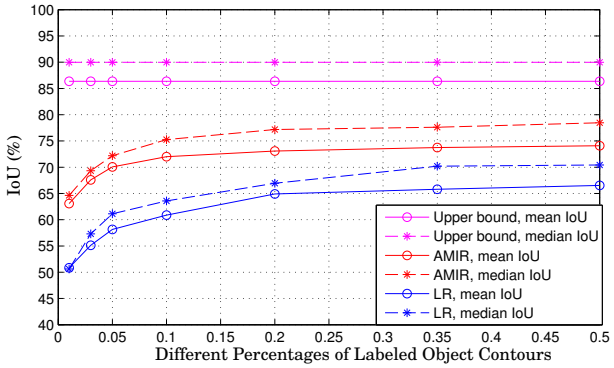

Figure 6.5: The performances, average median and mean IoU scores over the 20 object classes, of LR and AMIR with different fractions of the labeled object contours. The performance upper bounds, i.e., the IoU scores of the best tight segments, are also given.

addition, the performance of AMIR rapidly converges with a small fraction, say 10%, of labeled object contours.

The training time of AMIR, the running time of solving Eq. (4.15), is reported in TABLE 6.2. Since AMIR is trained class by class, the numbers of labeled contours, positive and negative bounding boxes of each class are also given in TABLE 6.2. The training of AMIR took less than 20 seconds in most classes, because AMIR can be efficiently optimized by steepest gradient descent. It is also worth mentioning that although AMIR only slightly outperforms its prior work [42] in object contour inference in the aspect of accuracy, AMIR gives a two order speed-up in the training process.

We also tested the stability of AMIR by evaluating its performance with different values of γ in Eq. (4.9). According to the results shown in Figure 6.6, AMIR works stably,

Table 6.2: The numbers of data in L, U^+, and U^- in each object class, together with the running (training) time of AMIR.

	A.P.	Bike.	Bird	Boat	Bottle.	Bus	Car	Cat	Chair	Cow	D.T.	Dog	Horse	M.B.	P.S.	P.P.	Sheep	Sofa	Train	Tv
# of Contours in L	3	3	4	4	5	3	4	4	4	9	5	4	4	4	28	7	4	4	4	4
# of Positive ROIs	26	26	30	31	41	27	36	29	76	38	28	35	30	27	247	59	34	32	27	33
# of Negative ROIs	628	747	836	902	1101	600	1025	688	2231	1074	807	986	900	689	1350	1755	940	895	696	985
Running Time (sec)	5.3	11.4	11.5	7.6	12.4	11.3	15.0	8.2	20.4	12.0	8.5	11.7	10.2	8.4	27.8	51.0	10.5	12.8	7.4	10.2

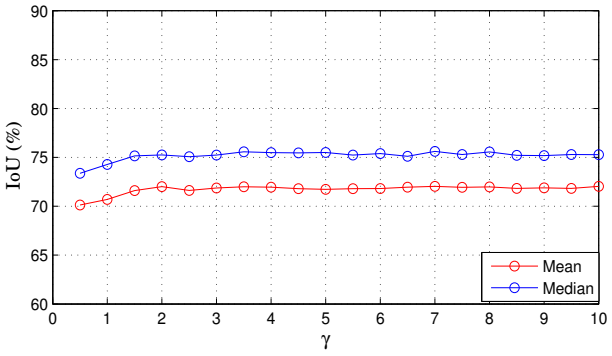

Figure 6.6: The performances, average median and mean IoU scores over the 20 object classes, of AMIR with different values of γ.

and is not very sensitive to the value of γ. We used $\gamma = 2$ throughout all experiments in the thesis.

6.4 Experiment III: Semantic Segmentation

The experiment III aims to verify the effectiveness of AMIR in acquiring training data for semantic segmentation. To this end, the inferred object contours by AMIR and the eight baselines were used as training data for two state-of-the-art semantic segmentation algorithms [6, 7]. That is, we used the automatically inferred object contours in place of the manually labeled ground truth in Pascal VOC segmentation task. For performance evaluation, the semantic segmentation method [6] was trained 10 times respectively with ground truth, the results of AMIR and the eight baselines, and evaluated by the testing data, `test`, on the Pascal VOC 2007 segmentation dataset. TABLE 6.3 reports the quantitative results. TABLE 6.4 reports results for another semantic segmentation method [7].

It can be observed in both TABLE 6.3 and TABLE 6.4 that, compared with the eight baselines, training with AMIR's results gives the best performance in semantic segmen-

Table 6.3: IoU scores (%) of [6] on Pascal VOC segmentation task with training data generated by various approaches.

	avg.	A.P.	Bike.	Bird	Boat	Bottle.	Bus	Car	Cat	Chair	Cow	D.T.	Dog	Horse	M.B.	P.S.	P.P.	Sheep	Sofa	Train	Tv
GT	**13.27**	5.53	10.45	9.32	0.60	4.48	12.25	17.74	21.42	3.83	2.43	3.56	11.61	9.23	26.95	26.39	6.87	8.36	4.27	16.98	10.67
GrabCut [1]	10.71	2.08	2.21	5.20	1.37	0.44	13.18	14.67	15.96	2.90	4.14	2.60	7.96	9.29	23.66	22.12	2.88	8.39	6.74	14.92	9.03
TS [2]	12.05	6.14	6.41	9.73	1.63	0.99	14.37	16.63	16.42	2.58	2.29	3.38	11.10	6.07	26.73	23.89	3.68	7.84	4.47	15.52	12.54
OP [3]	10.89	5.55	1.05	6.90	1.54	2.41	12.57	14.72	13.02	3.40	2.84	5.76	11.95	4.50	17.13	29.91	3.23	7.93	4.70	13.76	9.66
FG [4]	11.83	1.60	13.66	0.89	1.01	0.17	10.27	16.23	16.75	3.45	0.63	3.54	11.38	9.11	27.69	24.22	3.51	10.22	6.43	14.76	10.07
SVR [42]	11.67	3.32	0.66	2.79	0.84	1.26	13.67	21.67	19.18	2.02	5.36	11.78	8.41	4.39	23.99	22.66	3.64	11.30	3.98	13.31	6.85
DCCoSeg [5]	7.78	3.35	2.55	2.98	2.02	2.12	7.58	13.64	12.05	3.09	3.34	6.33	8.89	6.19	20.30	12.07	2.68	5.51	1.82	12.18	4.05
LR	12.11	3.31	1.78	8.84	1.91	0.03	16.01	16.89	12.87	3.66	2.02	2.19	11.57	8.14	27.40	22.57	4.18	12.46	3.72	22.27	7.86
MIR	12.33	0.64	3.03	11.15	0.04	0.65	13.87	18.91	16.94	2.44	1.52	1.76	6.36	7.66	32.12	27.04	9.47	7.13	8.27	17.39	4.28
AMIR	12.81	0.00	5.12	8.13	0.67	4.39	14.07	23.67	19.66	4.36	0.51	4.88	1.31	9.46	30.41	25.58	4.40	5.50	5.86	19.98	13.85

Table 6.4: IoU scores (%) of [7] on Pascal VOC segmentation task with training data generated by various approaches.

	avg.	A.P.	Bike.	Bird	Boat	Bottle.	Bus	Car	Cat	Chair	Cow	D.T.	Dog	Horse	M.B.	P.S.	P.P.	Sheep	Sofa	Train	Tv
GT	**12.99**	12.29	5.03	19.50	0.38	0.31	4.77	5.74	17.97	3.79	4.29	3.68	12.91	7.41	16.28	28.58	3.33	23.83	4.71	20.73	4.07
GrabCut [1]	11.94	9.21	0.38	8.49	0.31	2.02	4.87	1.86	15.41	4.45	1.14	2.50	16.27	13.80	20.06	28.77	4.84	25.00	1.94	13.49	3.18
TS [2]	11.18	16.04	5.44	6.25	2.97	3.60	3.06	0.60	21.84	1.21	1.95	2.35	27.15	3.90	7.83	24.23	5.52	17.13	0.75	9.40	0.99
OP [3]	10.55	10.99	1.71	8.04	0	1.76	4.76	1.04	14.69	5.94	0.70	1.64	14.04	4.05	7.25	27.44	0.39	18.48	3.00	21.52	1.67
FG [4]	10.16	5.55	0.66	7.49	0.11	0.56	4.21	2.38	21.46	3.99	1.31	0.87	15.94	7.06	12.14	25.28	5.48	17.79	0.10	10.62	2.99
SVR [42]	12.19	0	0.36	6.12	0	8.84	4.16	1.20	27.73	2.19	1.73	4.21	19.28	7.91	18.21	31.96	4.73	23.47	5.68	13.61	1.22
DCCoSeg [5]	8.96	6.02	0.96	5.89	0	10.86	2.88	1.11	15.17	0.82	0.11	1.91	7.17	11.05	10.68	9.80	0.68	10.24	2.26	19.42	0
LR	10.92	2.06	2.10	14.04	0.01	1.92	1.13	3.97	19.81	4.12	1.90	3.92	3.46	9.17	18.58	30.90	4.77	19.72	2.17	10.41	2.38
MIR	11.02	8.74	0.65	4.31	0	14.26	0.65	1.62	11.83	3.62	2.44	1.19	20.22	2.43	13.04	31.88	5.09	28.28	2.07	4.04	2.54
AMIR	12.56	5.92	3.86	9.71	1.72	5.49	2.86	3.21	21.06	4.52	4.06	2.60	9.50	5.07	17.34	25.40	4.03	33.66	9.11	18.19	3.30

tation. AMIR also achieves similar performance to training with ground truth, i.e., 13.27 vs. 12.81 using [6] or 12.99 vs. 12.56 using [7]. It shows that AMIR can automatically infer object segments enclosed by bounding boxes with sufficient quality. The experimental results indicate that AMIR can be an effective alternate for the manually drawn contours in the task of semantic segmentation, and can save the expensive annotation cost. We complete this section by showing some examples of semantic segmentation in Figure 6.7, for visually assessing the similar segmentation quality using AMIR's inferred object segments and manually labeled ones for training.

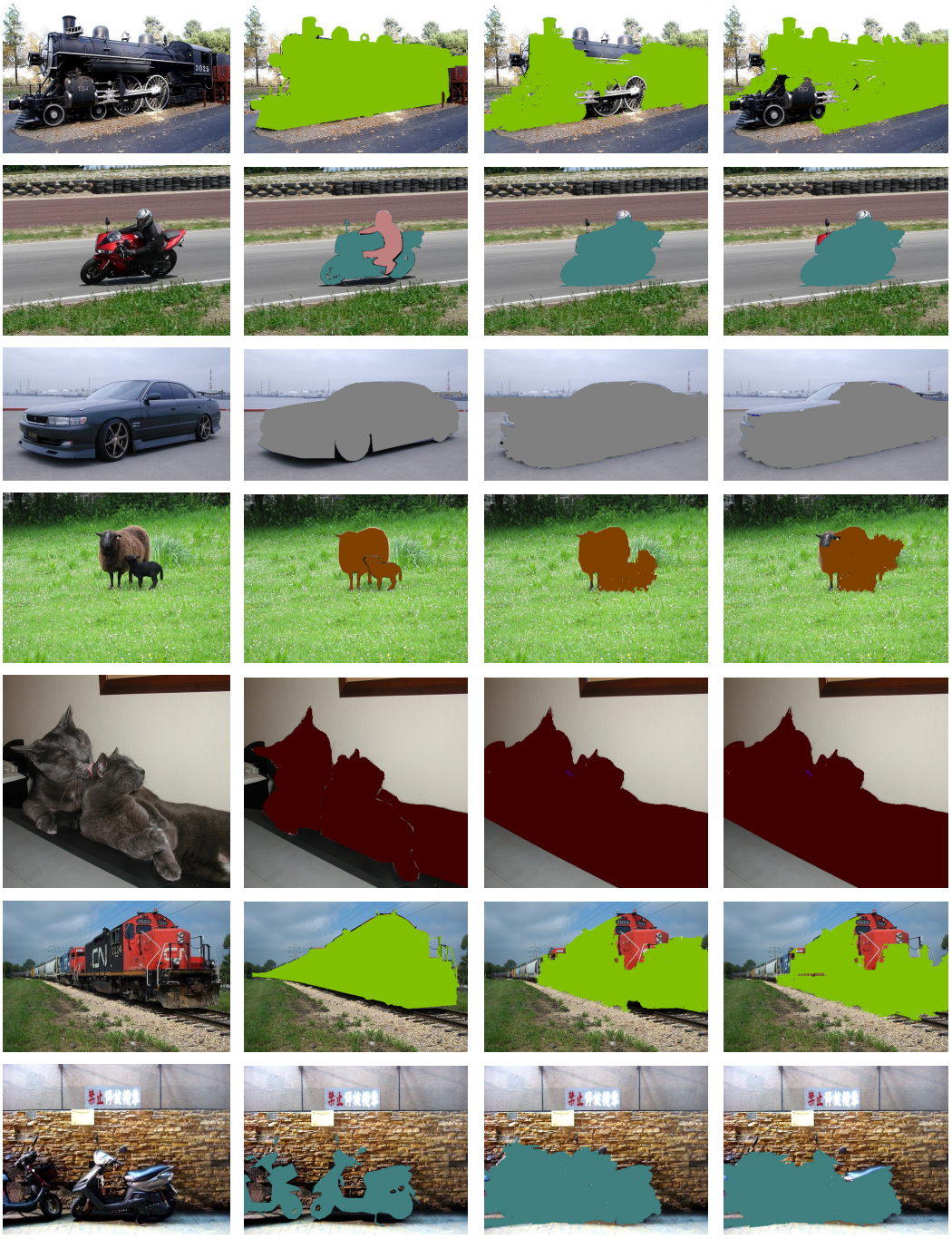

Figure 6.7: Some results by applying the method in [6] to semantic segmentation. The first two columns give the testing images and the ground truth, respectively. The third and forth columns show the results by learning with manually labeled segments and with the object segments inferred by AMIR, respectively.

Chapter 7

Conclusion

With the aim to reduce the heavy annotation cost of acquiring training data for semantic segmentation, we develop an effective and efficient approach, AMIR, to automatically infer the object segments enclosed by the bounding boxes. The main contribution of this thesis is three-fold. First, we propose to adopt training data for semantic segmentation in form of a few contours and a abundant set of bounding boxes, and formulate the estimation of the object contours as a variant of the multiple instance regression problem. Second, the proposed AMIR can jointly work with three different data sources including the labeled object contours, the positive and the negative bounding boxes, and fuses the information in the domain of tight segments. AMIR performs least square fitting for the labeled tight segments while carries out multiple instance learning for the unlabeled tight segments. It turns out that AMIR can resolve the problems caused by the large intra-class variations, and alleviate the unfavorable overfitting induced by the lack of labeled contours. Third, the proposed framework is evaluated in the benchmark of semantic segmentation, Pascal VOC 2007. The promising experimental results assert that the inferred object contours by AMIR are of high quality, and can replace the manually labeled contours.

For future work, we will extend our approach to take the correlation between objects into account, since the co-occurrence of object classes has been proved to be helpful in semantic segmentation. Furthermore, the structural information, i.e., the spatial configu-

ration and the label consistence of superpixels, of the tight segments should be exploited in the inference of object contours. We also plan to comprehensively analyze and assess our approach with more semantic segmentation benchmarks, such as the MSRC dataset.

Bibliography

[1] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 23(3):309–314, 2004.

[2] V. S. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *Proc. Int'l Conf. Computer Vision*, 2009.

[3] I. Endres and D. Hoiem. Category independent object proposals. In *Proc. Euro. Conf. Computer Vision*, 2010.

[4] Y. Chen, A. B. Chan, and G. Wang. Adaptive figure-ground classification. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[5] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2010.

[6] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Proc. Int'l Conf. Computer Vision*, 2009.

[7] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *Int. J. Computer Vision*, 101(2):329–349, 2013.

[8] X. Boix, J. M. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. S. Gual, and J. Gonzàlez. Harmony potentials - fusing global and local scale for semantic image segmentation. *Int. J. Computer Vision*, 96(1):83–102, 2012.

[9] C. Cheng, A Koschan, C.-H Chen, D L. Page, and M. A. Abidi. Outdoor scene image segmentation based on background recognition and perceptual organization. *IEEE Trans. on Image Processing*, 21(3):1007–1019, 2012.

[10] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *Int. J. Computer Vision*, 98(3):243–262, 2012.

[11] J. Shotton, J. Winn, A. Criminisi, and T. Darrell. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Computer Vision*, 81(1):2–23, 2009.

[12] P. Kohli, L. Ladický, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *Int. J. Computer Vision*, 82(3):302–324, 2009.

[13] N. Payet and S. Todorovic. Hough forest random field for object recognition and segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(5):1066–1079, 2013.

[14] Q. Zhou, J. Zhu, and W. Liu. Learning dynamic hybrid Markov random field for image labeling. *IEEE Trans. on Image Processing*, 22(6):2219–2232, 2013.

[15] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[16] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *Proc. Euro. Conf. Computer Vision*, 2012.

[17] L. Ladický, C. Russell, P. Kohli, and P. Torr. Associative hierarchical CRFs for object class image segmentation. In *Proc. Int'l Conf. Computer Vision*, 2009.

[18] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *Proc. Euro. Conf. Computer Vision*, 2010.

[19] C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. In *Proc. Euro. Conf. Computer Vision*, 2008.

[20] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.

[21] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured ouput learning for semantic segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[22] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[23] A. Müller and S. Behnke. Multi-instance methods for partially supervised image segmentation. In *Proc. IAPRW on Partially Supervised Learning*, 2011.

[24] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[25] S. Chen, L. Cao, Y. Wang, J. Liu, and X. Tang. Image segmentation by MAP-ML estimations. *IEEE Trans. on Image Processing*, 19(9):2254–2264, 2010.

[26] M. Mignotte. A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation. *IEEE Trans. on Image Processing*, 19(6):1610–1624, 2010.

[27] A. K. Qin and D. A. Clausi. Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Trans. on Image Processing*, 19(8):2157–2170, 2010.

[28] M. B. Salah, A. Mitiche, and I. B. Ayed. Multiregion image segmentation by parametric kernel graph cuts. *IEEE Trans. on Image Processing*, 20(2):545–557, 2011.

[29] C. Panagiotakis, I. Grinias, and G. Tziritas. Natural image segmentation based on tree equipartition, bayesian flooding and region merging. *IEEE Trans. on Image Processing*, 20(8):2276–2287, 2011.

[30] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2003.

[31] S. Xiang, C. Pan, F. Nie, and C. Zhang. Turbopixel segmentation using eigen-images. *IEEE Trans. on Image Processing*, 19(11):3024–3034, 2010.

[32] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. Int'l Conf. Machine Learning*, 2001.

[33] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[34] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *Proc. Euro. Conf. Computer Vision*, 2008.

[35] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012.

[36] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[37] G. Takács. Smooth maximum function based algorithms for classification, regression, and collaborative filtering. *Acta Technica Jaurinensis, Series Computatorica Intelligentica*, 3(1):27–63, 2010.

[38] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[39] T. Cour and J. Shi. Recognizing objects by piecing together the segmentation puzzle. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.

[40] B. Alexe, T. Deselaers, and V. Ferrari. ClassCut for unsupervised class segmentation. In *Proc. Euro. Conf. Computer Vision*, 2010.

[41] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(6):929–944, 2007.

[42] J.-Z. Cheng, F.-J. Chang, K.-J. Hsu, and Y.-Y. Lin. Knowledge leverage from contours to bounding boxes: A concise approach to annotation. In *Proc. Asian Conf. on Computer Vision*, 2012.

[43] M. P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *Proc. Int'l Conf. Computer Vision*, 2011.

[44] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[45] Z. Kuang, D. Schnieders, H. Zhou, K.-Y. K. Wong, Y. Yu, and B. Peng. Learning image-specific parameters for interactive segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2012.

[46] P. A. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems*, 2005.

[47] Q. Zhang, S. Goldman, W. Yu, and J. E. Fritts. Content-based image retrieval using multiple-instance learning. In *Proc. Int'l Conf. Machine Learning*, 2002.

[48] F. Li and C. Sminchisescu. Convex multiple-instance learning by estimating likelihood ratio. In *Advances in Neural Information Processing Systems*, 2010.

[49] S. Ray and D. Page. Multiple instance regression. In *Proc. Int'l Conf. Machine Learning*, 2001.

[50] P.-M. Cheung and J. T. Kwok. A regularization framework for multiple-instance learning. In *Proc. Int'l Conf. Machine Learning*, 2006.

[51] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, 2004.

[52] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011.

[53] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proc. Euro. Conf. Computer Vision*, 2002.