Learning to Segment Under Various Forms of Weak Supervision

Jia Xu[1] Alexander G. Schwing[2] Raquel Urtasun[2]
[1]University of Wisconsin-Madison [2]University of Toronto
jiaxu@cs.wisc.edu {aschwing, urtasun}@cs.toronto.edu

Abstract

Despite the promising performance of conventional fully supervised algorithms, semantic segmentation has remained an important, yet challenging task. Due to the limited availability of complete annotations, it is of great interest to design solutions for semantic segmentation that take into account weakly labeled data, which is readily available at a much larger scale. Contrasting the common theme to develop a different algorithm for each type of weak annotation, in this work, we propose a unified approach that incorporates various forms of weak supervision – image level tags, bounding boxes, and partial labels – to produce a pixel-wise labeling. We conduct a rigorous evaluation on the challenging Siftflow dataset for various weakly labeled settings, and show that our approach outperforms the state-of-the-art by 12% *on per-class accuracy, while maintaining comparable per-pixel accuracy.*

1. Introduction

Semantic segmentation is one of the most fundamental challenges in computer vision, and conventional fully supervised algorithms have demonstrated promising results [21, 10, 8, 33, 35, 36, 44, 19, 20, 45, 31]. However, in order to train fully supervised systems, a set of training examples with semantic labels for each pixel in an image is required. Considering the recent performance improvements obtained when employing millions of data points, it is obvious that the size of the training data is one of the bottlenecks for semantic segmentation. This is not astonishing since labeling each pixels with a semantic category is a very expensive and time-consuming process, even when utilizing crowd-sourcing platforms such as MTurk.

Compared to the massive size of modern visual data – everyday, more than 300 million images are uploaded to Facebook – only a tiny fraction is assigned accurate pixel-wise annotations. For instance, in the ImageNet dataset [7], 14 million images are assigned with scene categories; 500,000 images are annotated with object bounding boxes; but only $4,460$ images are segmented at the pixel level [14]. The reason for this difference is rather obvious: for almost any given image we are quickly able to decide whether an object is illustrated in a scene, but careful delineation is tedious. Therefore, certainly image tags, but sometimes even bounding boxes, and occasionally partial labels in the form of user scribbles are either easily collected or are even readily available in large photo collections of websites such as Flickr and Facebook.

Figure 1. Our semantic segmentation algorithm learns from various forms of weak supervision (image level tags, bounding boxes, partial labels), and produces pixel-wise labels.

In the absence of large pixel-wise annotated datasets, development of visual parsing algorithms that benefit from weakly labeled data is key to further improve the performance of semantic segmentation. Supervision in the form of partial labels has been effectively utilized in interactive object segmentation via graph-cuts [3], random walks [12], geodesic shortest path [2], and geodesic star convexity [15]. Recursive propagation of segmentations from labeled masks to unlabeled images has also been investigated [14]. An alternative form of weak supervision are bounding boxes. Grabcut [27] has been a great success for binary object segmentation, taking advantage of a bounding box around the foreground object. Recent research has extended this idea to semantic segmentation by building object detectors from bounding boxes [41]. A more challenging setting is inference of a pixel-wise labeling, given only image level tags. Encouraging results have been presented for this weakly supervised semantic segmentation task by connecting super-pixels across images, and jointly inferring pixel labels for all images [38, 39, 37]. An alternative are Markov random fields with latent variables denoting the super-pixels, and observed variables representing image tags [43, 30].

In this work, we propose a unified approach that takes any form of weak supervision, *e.g.*, tags, bounding boxes, and/or partial labels, and learns to semantically segment images. We refer the reader to Fig. 1 for an illustration of the problem. When compared to existing weakly labeled methods [39, 43], our approach is very efficient, taking only 20 minutes for learning, and a fraction of a second for inference. We conduct a rigorous evaluation on the challenging Siftflow dataset for various weakly labeled settings, and show that our method outperforms the state-of-the-art by 12% in per-class accuracy [29], while maintaining a result comparable in the per-pixel metric.

2. Related Work

Semantic segmentation: Semantic segmentation, sometimes called scene parsing, is widely studied in computer vision. A large variety of algorithms have been developed for the fully supervised setting, requiring access to a fully labeled training set. Three types of approaches are very popular. Non-parametric methods [21, 8, 33, 35, 36, 44] build pixel-wise potentials using nearest neighbors. These methods are motivated by the observation that similar semantic pixels lie close in some feature space. The second set of approaches frames semantic segmentation as an inference task using Markov random fields (MRF) [19, 20, 45]. These methods handle supervision at different levels (tags, bounding boxes, scene types) by adding variables to the energy function. The final set of effective methods are based on object proposal [4, 9, 1, 11, 16], where class-independent segments are generated, and subsequently classified into different categories using features computed on those segments. These conventional methods tend to work well, given a sufficient amount of fully labeled data. Unfortunately, such pixel-wise labelings are very expensive and challenging to obtain.

Co-segmentation: A large number of researchers have therefore explored ways to make use of unlabeled or weakly labeled data. One possibility is co-segmentation [28, 40, 23], where the task is to segment the shared foreground from multiple images. This is a data driven approach based on the assumption that common foreground objects look alike, while differing significantly from the background. Co-segmentation is further extended to the multi-class case via discriminative clustering [18, 17], and the multi-object case using subspace analysis [24].

Segmentation with tags: Weakly supervised semantic segmentation and co-segmentation share the same motivation. Consider a case where one image is tagged with labels for cow and grass, and another one is assigned the categories for cow and road. It is reasonable to assume that pixels which are similar in both images take on the class label for cow, while the remaining image content may take the other assigned categories. Researchers have attempted to tackle this challenge by connecting super-pixels across images, and jointly inferring pixel labels for all images [38, 39, 37, 22]. Alternatively, propagation via dense image correspondences [29], or learning a latent graphical model between tags and super-pixel labels [43] has been considered. Promising results have been demonstrated, even though training is expensive. Recent research [34] also built a one-shot object detection algorithm with object tags.

Segmentation with semi-supervision: Another form of weak annotation is semi-supervision. Hereby a user provides partial labels for some pixels within an image. This is a convenient setting, as it is reasonably easy for annotators to perform strokes which partially label an image. Such a form of supervision has been effectively utilized in interactive object segmentation with graph-cuts [3], random walks [12], geodesic shortest path [2], geodesic star convexity [15], and topological constraints [42].

Segmentation with bounding boxes: Also of interest for weak supervision are bounding boxes. Among the biggest successes is GrabCut [27], where a user provided bounding box is employed to learn a Gaussian Mixture Model (GMM) differentiating between foreground and background. Recent work has extended this idea to semantic segmentation by building object detectors from multiple bounding boxes [41]. [25] utilizes bounding boxes to locate objects of interest, within a latent structured SVM framework. 3D bounding boxes as a form of weak supervision have been shown to produce human-level segmentation results [5].

In this work, we make a first attempt to employ all of the aforementioned forms of weak supervision within a single unified model. We then build an efficient learning algorithm, which is parallelizable and scalable to large image sets.

3. Unified Model for Various Forms of Weak Supervision

In this paper, we address the problem of semantic segmentation using various forms of weak supervision, like image level tags, strokes (*i.e.*, partial labels) as well as bounding boxes. More specifically, we are interested in inferring pixel-level semantic labels for all the images, as well as learning an appearance model for each semantic class. The latter permits prediction in previously unseen test examples. Note that we never observe a single labeled pixel in most of our settings. We formulate the task using a max-margin clustering framework, where knowledge from supervision is included via constraints, restricting the assignment of pix-

els to class labels. We obtain a unifying formulation that is able to exploit arbitrary combinations of supervision.

3.1. Unified Model

Following recent research [39, 43], we first over-segment all images into a total of n super-pixels. For each super-pixel $p \in \{1, \dots, n\}$, we then extract a d dimensional feature vector $\mathbf{x}_p \in \mathbb{R}^d$. Let the matrix $H = [\mathbf{h}_1, \dots, \mathbf{h}_n]^T \in \{0,1\}^{n \times C}$ contain the hidden semantic labels for all super-pixels. We use a 1-of-C encoding, and thus a C-dimensional column vector $\mathbf{h}_p \in \{0,1\}^C$, with C denoting the number of semantic classes and h_p^c referring to the c-th entry of the vector $\mathbf{h}_p$.

Our objective is motivated by the fully supervised setting and the success of max-margin classifiers. As the assignments of super-pixels to semantic labels is not known, not even for the training set, supervised learning is not possible. Instead, we take advantage of max-margin clustering (MMC) [47, 46] which searches for those assignments that maximize the margin. We therefore aim at minimizing the regularized margin violation

$$\frac{1}{2}\text{tr}(W^T W) + \lambda \sum_{p=1}^{n} \sum_{c=1}^{C} \xi(\mathbf{w}_c; \mathbf{x}_p, h_p^c), \qquad (1)$$

where $W = [\mathbf{w}_1, \dots, \mathbf{w}_C] \in \mathbb{R}^{d \times C}$ is a weight matrix encoding the learned appearance model, $\mathbf{w}_c \in \mathbb{R}^d$ is the c-th column of matrix W, and λ is a hyper-parameter of our framework.

Note that in most semantic segmentation tasks the class categories are distributed according to a power law. Hence, instead of using a standard hinge loss for the margin violation $\xi(\mathbf{w}_c; \mathbf{x}_p, h_p^c)$, we want to take into account the fact that class labels typically occur in a very unbalanced way. Therefore, we let

$$\xi(\mathbf{w}_c; \mathbf{x}_p, h_p^c) = \begin{cases} \max(0, 1 + (\mathbf{w}_c^T \mathbf{x}_p)), & h_p^c = 0 \\ \mu^c \max(0, 1 - (\mathbf{w}_c^T \mathbf{x}_p)), & h_p^c = 1 \end{cases} \qquad (2)$$

where $\mu^c = \frac{\sum_{p=1}^{n} 1(h_p^c == 0)}{\sum_{p=1}^{n} 1(h_p^c == 1)}$. If the number of negative examples ($h_p^c = 0$) is bigger than the positive ones, this asymmetric loss penalizes more if we incorrectly label a positive instance. Note that if the matrix of super-pixel class assignments H was known, the cost function given in Eq. (1) is identical to a one-vs-all support vector machine (SVM).

We now show how to incorporate different forms of weak supervision, *i.e.*, tags, partial labels and bounding boxes. To this end we add constraints to the program given in Eq. (1). Thus, in general our learning algorithm reads as follows:

$$\begin{aligned} \min_{W,H} \quad & \frac{1}{2}\text{tr}(W^T W) + \lambda \sum_{p=1}^{n} \xi(W; \mathbf{x}_p, \mathbf{h}_p) \\ \text{s.t.} \quad & H\mathbf{1}_C = \mathbf{1}_n, \quad H \in \{0,1\}^{n \times C}, \quad H \in \mathcal{S}, \end{aligned} \qquad (3)$$

where $\mathbf{1}_C$ is an all ones vector of length C, and $\mathbf{1}_n$ is an all ones vector of length n. The parameter λ balances the regularization term $\text{tr}(W^T W)$ and the loss contribution $\xi(W; \mathbf{x}_p, \mathbf{h}_p)$. Subsequently we describe the constrained space $\mathcal{S}$ for each form of weak annotation.

3.2. Incorporating Weak Supervision

Importantly, for all forms of weak supervision discussed subsequently, the set of constraints subsumed within $\mathcal{S}$ turns out to be linear.

Image level tags (ILT): Considering image level tags (ILT), each image $i \in \{1, \dots, m\}$ is assigned a set of categories, indicating which classes are present. However the specific location of the class, *i.e.*, the super-pixel is not specified in any way. Let the binary matrix $Z \in \{0,1\}^{m \times C}$ denote the image-level tag supervision: $Z_{ic} = 1$ if class $c \in \{1, \dots, C\}$ is present in image $i \in \{1, \dots, m\}$, and $Z_{ic} = 0$ otherwise. Let the binary matrix B be a super-pixel-image incidence matrix of size $n \times m$: $B_{pi} = 1$, if super-pixel $p \in \{1, \dots, n\}$ belongs to image $i \in \{1, \dots, m\}$, and $B_{pi} = 0$ otherwise. Given the binary matrices B and Z, we incorporate tag-level supervision by adding two sets of constraints. The first set expresses the fact that if a tag is not assigned to an image, super-pixels in that image can not be assigned to that class. This fact is encoded via

$$H \leq BZ. \qquad (4)$$

The second set of constraints encodes the fact that if an image tag is present, at least one super-pixel should take that class as its label. Such a statement is described with

$$B^T H \geq Z. \qquad (5)$$

To explain how these constraints work in practice, let us demonstrate the details using a toy example. Suppose we have $m = 2$ images, each partitioned into 2 super-pixels, *i.e.*, $n = 4$. Let the number of classes of interest $C = 3$. We further assume the first image is tagged with categories $\{1, 2\}$, while the second one is assigned labels $\{2, 3\}$. Then our matrices look as follows:

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad BZ = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Note that the product BZ 'copies' the image-level tags for super-pixels belonging to that particular image. Due to the less than or equal to constraint and the restriction to a binary matrix H, these super-pixels can not take classes which are not assigned to an image. Similarly, suppose we are given the following class assignment matrix H, then $B^T H$ counts how many instances are labeled for each class

within each image, making sure that at least one super-pixel takes on a required class-label:

$$H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B^T H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Semi-supervision: In the semi-supervised setting, we are given class labels for only a subset of the super-pixels. This is a useful setting, as users can simply scribble on an image, and label a subset of the super-pixels. This form of supervision is easily framed using an equality constraint for each super-pixel that is labeled, *i.e.*,

$$H_\Omega = \hat{H}_\Omega, \tag{6}$$

where $\Omega \subseteq \{1, \ldots, n\}$ corresponds to the set of annotated super-pixels and the matrix $\hat{H}_\Omega \in \{0,1\}^{|\Omega| \times C}$ refers to the user-specification.

Bounding boxes: Including the bounding box annotation follows the ILT case and adds additional restrictions. Note that the ILT setting, is equivalently phrased using a single bounding box of size corresponding to the image dimensions. Several tags are given for this bounding box. Here, we extend this setting to allow for smaller bounding boxes, each of which is assigned a single tag. Following the constraint given in Eq. (5) we can thus treat each box as a sub-image and enforce

$$(B^{pos})^T H^{pos} \geq Z^{box}, \tag{7}$$

where H^{pos} corresponds to label variables for super-pixels entirely inside the bounding boxes that were provided. In addition, $B^{pos}_{p,j} = 1$ if super-pixel p is entirely inside box j, and 0 otherwise. Further, Z^{box} is the binary label matrix for bounding boxes: $Z^{pos}_{j,c} = 1$, if c is the class of bounding box j and 0 otherwise. Note that $(B^{pos})^T H^{pos} \geq Z^{box}$ forces the model to assign at least one super-pixel to the bounding box class c. The matrix H^{neg}_c refers to label variables (for class c) of super-pixels which are partially inside the bounding boxes. A constraint of the form $H^{neg}_c \leq 0$ encodes that those 'negative' super-pixels should not take the bounding box class c. This is typically a reasonable constraint, as we assume our bounding box to be tight. However, due to the fact that our super-pixels suffer from under segmentation, we do not use negative constraints $H^{neg}_c \leq 0$ in our experimental evaluation. To make it robust to under segmentation, in practice we use super-pixels which are 80% inside the bounding boxes to define B^{pos}.

Unlabeled examples: We make use of unlabeled examples by simply incorporating them in the objective. Note that no constraints are added as no supervision is available.

Algorithm 1 Learning to Segment

1: **Input:** $X, \mathcal{S}$
2: Initialize: compute $Z(\mathcal{S})$, $B(\mathcal{S})$, $H \leftarrow BZ$;
3: **for** iter$= 1 \rightarrow$ max_iter **do**
4: Fix H solve for W independent of classes (1-vs-all linear SVM);
5: Fix W infer super-pixel labels H in parallel w.r.t images (small LP instances);
6: **end for**
7: **return** W, H;

3.3. Learning via Alternate Optimization

During learning, we jointly optimize for the feature weight matrix W encoding the appearance model, and the semantic labels H for all n super-pixels as specified by the program given in Eq. (3). Note that all forms of supervision considered in this paper can be incorporated via linear constraints. Nonetheless, Eq. (3) is generally a non-convex mixed integer programing problem, which is challenging to optimize. Investigating the program given in Eq. (3) more closely, we observe that our optimization problem is however bi-convex, *i.e.*, it is convex w.r.t. W if H is fixed, and convex w.r.t. H if W is fixed. Further, our constraints are linear and they only involve the super-pixel assignment matrix H. For optimization we therefore employ an alternating procedure, where we iterate the two steps of optimizing H and W for fixed values of W and H respectively. We refer the reader to Alg. 1 for an outline of the proposed learning algorithm.

It is easy to see that for a fixed class assignment matrix H, the resulting optimization task is equivalent to the fully supervised setting, where the labels are obtained from the current estimate of H. In our formulation this decomposes into C different 1-vs-all SVMs which can be trained in parallel.

When optimizing w.r.t. the assignment matrix H for a fixed appearance model W, we need to solve a constrained optimization problem where both the objective and the constraints are linear. In addition, H is required to be binary, resulting in an integer linear program (ILP). Such optimization problems are generally NP-hard. However, we will show that in our case we can decompose the problem into smaller tasks that can be optimally solved in parallel via an LP relaxation. This LP relaxation is guaranteed to retrieve an integer solution, and thus an optimal integral point.

Our objective in Eq. (3) is a min-max function with respect to H. Due to the dependence of μ^c, defined in the loss function given in Eq. (2), on the assignment matrix H, this problem is extremely challenging. However we found the solution obtained by simply dropping μ^c during learning to work very well in practice, as shown in the ex-

perimental section. In addition it drastically simplifies the loss-augmented inference required during learning. Solving for the assignment matrix H after dropping μ^c during learning is then equivalent to finding the maximum a-posteriori (MAP) prediction within the label space. To see this, note that picking the class with maximum $\mathbf{w}_c^T \mathbf{x}_p$ yields the smallest hinge-loss ξ. For instance, if $\mathbf{w}_c^T \mathbf{x}_p < 0$, setting $h_p^c = 0$ returns the smaller hinge-loss ξ. Similarly, if $\mathbf{w}_c^T \mathbf{x}_p > 0$, letting $h_p^c = 1$ yields a smaller loss ξ. Thus we want to pick a super-pixel class assignment H which maximizes the score while remaining feasible:

$$\begin{aligned} \max_{H} \quad & \text{tr}((X^T W)^T H) \\ \text{s.t.} \quad & H\mathbf{1}_C = \mathbf{1}_n, \quad H \in \{0,1\}^{n \times C}, \quad H \in \mathcal{S}. \end{aligned} \tag{8}$$

Hereby we combine the data into the matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$. The trace objective function is equivalent to a linear sum, hence we can divide it into a sum of super-pixel instances. The constraint $H\mathbf{1}_C = \mathbf{1}_n$ is also separable by super-pixel instances. But the weak supervision (including tags, semi-supervision, bounding boxes) constraints act within a single image. Hence, taking it all together, the entire constraint space is separable by images. This nice property permits to separate the ILP into much smaller sub-programs, which reason about each image independently and can be solved in parallel. We can further reduce the size of the individual ILPs by removing the variables referring to tags not relevant for a particular image.

Importantly, our program has the additional property that the coefficient matrix for the constraints is totally unimodular. As a consequence we can solve each ILP exactly using a linear programming relaxation which is tight. This tightness is reflected in the following proposition. We refer the reader to the supplementary material for a complete proof.

Proposition 3.1. *Relaxing the integrality constraints in Eq. (8) and using a linear programming solver gives the integral optimal solution for our constraint set $\mathcal{S}$.*

Proof. (Sketch) The main idea of the proof is to show that the coefficient matrix of the linear programming relaxation is totally unimodular. By [13]: If the coefficient matrix A is totally unimodular and the right-hand side b is integral, then linear programs of the form $\{\min \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ have an integral optimum, for any cost vector $\mathbf{c}$. Hence, the LP relaxation gives the optimal integral solution. □

3.4. Inference

We experiment with two inference strategies. Given a learned appearance model W, our first strategy predicts using the standard 1-vs-all rule. We refer to this setting as "Ours (1-vs-all)." Our second strategy makes use of a tag predictor to create additional constraints for the test images.

Method	Supervision	per-class	per-pixel
Liu et al. [21]	full	24	76.7
Farabet et al. [10]	full	29.5	78.5
Farabet et al. [10] balanced	full	46.0	74.2
Eigen et al. [8]	full	32.5	77.1
Singh et al. [33]	full	33.8	79.2
Tighe et al. [35]	full	30.1	77.0
Tighe et al. [36]	full	39.3	78.6
Yang et al. [44]	full	48.7	79.8
Vezhnevets et al. [38]	weak (tags)	14	N/A
Vezhnevets et al. [39]	weak (tags)	22	51
Rubinstein et al. [29]	weak (tags)	29.5	63.3
Xu et al. [43]	weak (tags)	27.9	N/A
Ours (1-vs-all)	weak (tags)	**32.0**	**64.4**
Ours (ILT)	weak (tags)	**35.0**	**65.0**
Ours (1-vs-all + transductive)	weak (tags)	**40.0**	59.0
Ours (ILT + transductive)	weak (tags)	**41.4**	**62.7**

Table 1. Comparison to state-of-the-art on the SIFT-flow dataset. For [39], we report the per-pixel number from [37]. Note that our approach while only using tags as supervision and thus never observing a single pixel labeled, is able to perform almost as well as the state-of-the-art in the fully supervised setting. This is a remarkable fact. Furthermore we outperform the state-of-the-art in the weakly label case by more than 10%.

Method	Supervision	per-class	per-pixel
Shotton et al. [32]	full	67	72
Yao et al. [45]	full	79	86
Vezhnevets et al. [38]	weak (tags)	67	67
Liu et al. [22]	weak (tags)	N/A	**71**
Ours (ILT + transductive)	weak (tags)	**73**	70

Table 2. Comparison to state-of-the-art on the MSRC dataset.

Give the tag predictions we perform inference on the test set by minimizing Eq. (8) with the ILT constraints described in Eq. (4). Note that we do not employ the constraints provided in Eq. (5) as our tag classifier might be wrong. We refer to this setting as "Ours (ILT)."

3.5. Transductive Learning

In the standard setting, we learn the weights of the appearance model matrix W using the training set images. We also experiment with a transductive setting which exploits the test images as well. Note that the test set can be used by incorporating the images as unlabeled examples or by using a tag classifier and adding the constraints detailed in Eq. (4) for the test images. We refer to this setting as "Ours (1-vs-all + transductive)" and "Ours (ILT+transductive)" respectively.

4. Experimental Evaluation

4.1. Dataset and Super-pixel/feature Extraction

Dataset: To illustrate the performance of our model, we conduct a rigorous evaluation on the Siftflow data set [21], which has been widely studied [21, 10, 8, 33, 35, 36, 44, 38, 39, 43]. The Sift-Flow data contains $m = 2688$ images and

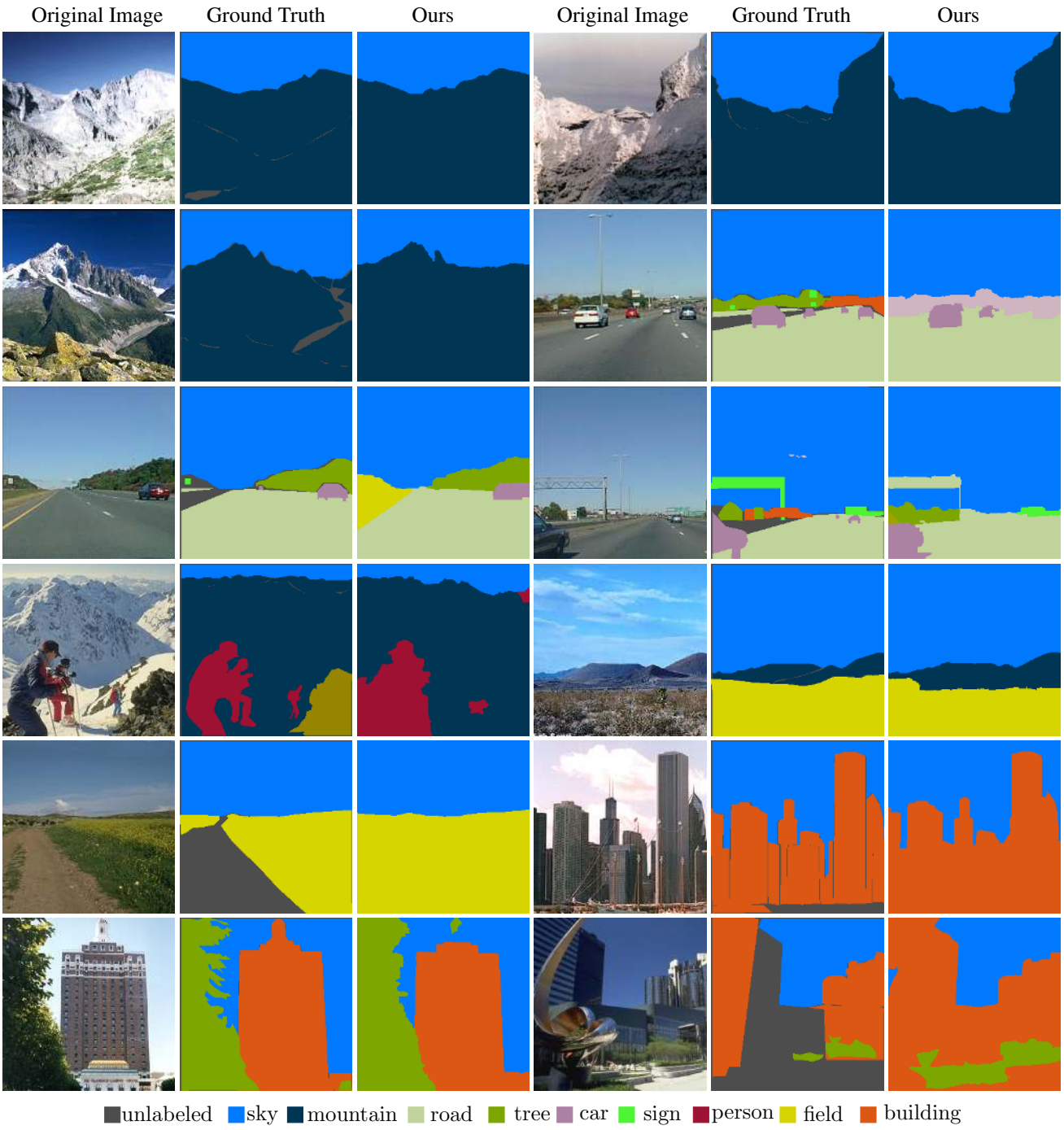

Figure 2. Sample results from "Ours(ILT+transductive)". Note gray regions in the second and fifth column are not labeled in ground truth. **Best viewed in color.**

$C = 33$ classes in total. The data set is very challenging due to its large scale and its heavily tailed class frequency distribution. A few 'stuff' classes like 'sky,' 'road,' 'sea,' and 'tree' are very common, while the 'things' classes like 'sun,' 'person,' and 'bus' are very rare. We use the standard split of 2488 training images and 200 testing images provided by [21], and randomly sampled 20% of the training set to tune our parameter λ. To further evaluate the capacity of our algorithm, we also test it on the MSRC dataset [32], which has $m = 591$ images and $C = 21$ classes. We report both the per-class and per-pixel accuracy.

Super-pixel segmentation: For each image, we compute the Ultrametric Contour Map (UCM) [1], and threshold it

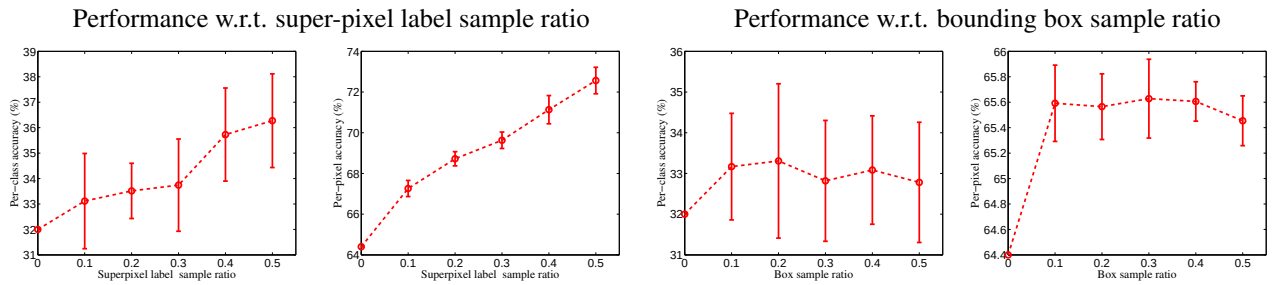

Figure 3. Per-class (first and third column) and per-pixel accuracy (second and forth column) with respect to super-pixel label (first two columns) and bounding box (last two columns) sample ratio.

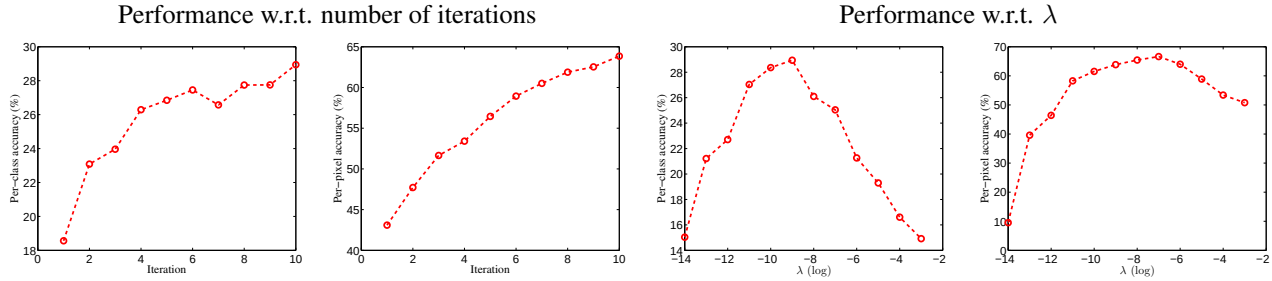

Figure 4. Per-class (first and third column) and per-pixel accuracy (second and forth column) from "Ours(1-vs-all)" with respect to number of iterations (first two columns) and λ (last two columns). The accuracy is reported on the validation set.

at 0.4 to extract super-pixels. UCM produces a few tiny super-pixels, which create noise during learning. To alleviate this issue we adopt a local search algorithm [26] which merges similar adjacent tiny super-pixels. On average this procedure results in 14 regions per image. We use the same super-pixel segmentation for all the reported experiments.

Super-pixel feature extraction: For each super-pixel, we first extract R-CNN [11] features within the bounding box as well as within the masked box. These two sets of features – 8192 dimensional in total – capture the local context and the shape of the super-pixel. To take into account global context and super-pixel size/location, we further replace the bounding box with the whole image to compute additional features. That is, we compute R-CNN features for the whole image, as well as the masked image. This gives another 8192 dimensional feature vector. After concatenation, we obtained a $d = 16884$ dimensional feature vector $\mathbf{x}_p$ for each super-pixel p.

4.2. Evaluation on Various Weak Superpvisons

Training with tags only: We first evaluate our algorithm on the standard weakly supervised semantic segmentation setting. During training, we are only given image level tags, and at test time, we infer pixel-wise labels without tags. We first investigate the feature weights we learned using Alg. 1 via the 1-vs-all inference approach. For each super-pixel, we simply pick the class with maximum potential from $\mathbf{x}^T W$. As shown in Tab. 1, this simple approach tagged "Ours(1-vs-all)" achieves 32.0% per-class accuracy and a 64.4% per-pixel accuracy, outperforming the state-of-the-art. Motivated by recent work [38, 43], we trained a 1-vs-all linear SVM ILT classifier with R-CNN features extracted from the whole image. We then feed the classifier output into our inference detailed in Eq. (8), and predict the labels for super-pixels for the testing images. We refer to this setting via "Ours (ILT)." It further improves the per-class accuracy to 35.0%, and the per-pixel metric to 65.0%.

Training with tags (transductive): As provided in Tab. 1, using the transductive setting we achieved a 41.4% per-class accuracy, which outperforms the state-of-the-art by 11.9%. We also note that using this transductive setting without the tag classifier achieves 40.0% per-class accuracy, which further demonstrates that ILT are very helpful in inferring pixel-wise labels. As shown in Tab. 2, the resulting performance on MSRC in perClass/perPixel accuracy is 73%/70% using cross validation. In contrast [38] reports 67%/67%. We present qualitative results in Fig. 2. The presented approach performs well for 'stuff' segments like 'sky,' 'mountain,' and 'road' which have a fairly reliable super-pixel segmentation and a discriminative appearance model. We are also able to obtain correct labels for 'things' classes (*e.g.*, 'cars' and 'person').

Training with semi-supervision: We next evaluate the semi-supervised setting where a subset of the super-pixels is labeled in addition. We focus on the 1-vs-all inference

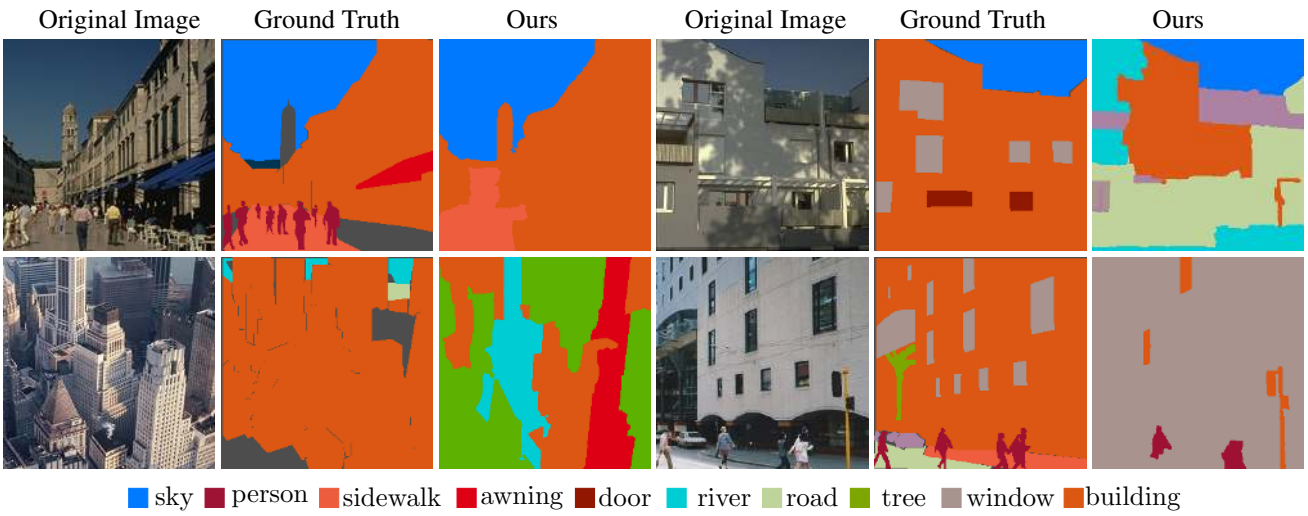

Figure 5. Failure cases. **Best viewed in color.**

setting. Strokes are simulated by randomly labeling superpixes from the ground truth using a sampling ratio of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Due to the randomness we repeat our experiments 10 times and report the mean and standard deviation in the first two plots of Fig. 3. Note that both the per-class and the per-pixel accuracies improve consistently when more super-pixels are observed. Furthermore, the per-pixel accuracy increases almost linearly with the sampling ratio.

Training with bounding boxes: Next we evaluate the bounding box setting. Again we focus on the 1-vs-all inference setting. We consider bounding boxes for 'things' classes: {bird, boat, bus, car, cow, door, moon, person, pole, sign, streetlight, sun, window} and simulate this setting by randomly picking bounding boxes around connected segments from ground truth annotations. Due to the randomness we repeat our experiments 10 times and report the mean and standard deviation in the last two plots of Fig. 3. We improve the per-class accuracy by 1% even with only 10% of 'things' boxes, which translates to approximately 0.05 boxes per image. We also observe the improvement is less significant, when more boxes are added. This is due to the quality of our super pixels, which suffer from under segmentation.

4.3. Model Behavior

Number of iterations: During training, we iterate between learning the feature weights and inferring super-pixel labels. To study how the performance changes with respect to the number of iterations, we evaluate the weights learned after each iteration using the 1-vs-all rule, and plot the accuracy on the validation set in Fig. 4. We observe that the per-class/per-pixel accuracy quickly improves for the first 4 iterations, and starts to converge after 8 iterations. In all our experiments, we report the performance after 10 iterations.

Performance with respect to λ: To evaluate how our algorithm behaves w.r.t. λ, we plot the accuracy of "Ours(1-vs-all)" for all λ in $\{2^{-14}, ..., 2^{-3}\}$ in Fig. 4. We used a weighted sum of the per-pixel and per-class accuracy to find the best lambda on the validation set. All experiments use this fixed value of $\lambda = 2^{-9}$.

Running time: As we discussed in the optimization, both tasks of learning W and inferring H can be parallelized. On our machine with 12 threads, each iteration takes about $1 \sim 2$ minutes, resulting in less than 20 minutes for training on the full Siftfow dataset. Inference takes $< 0.01s$ per image after super-pixel segmentation and feature extraction.

Failure cases: We present failure cases in Fig. 5. Super-pixel under segmentation is a common failure mode, where small 'things' segments (top left in Fig. 5) are challenging to obtain by UCM. Extreme shading changes (top right in Fig. 5) pose challenges just like cluttered textures (bottom left in Fig. 5). As shown on the right hand side of Fig. 5 our model may also get confuse classes that co-occur frequently. For instance, we accidentally labeled building segments as window, and vice versa. This is expected as only tags are used for learning.

5. Conclusion

In this paper, we introduced a unified semantic segmentation approach to handle weak supervision in the form of tags, partial labels and bounding boxes. Our approach is efficient in both training and testing. We demonstrated the effectiveness of our approach on the challenging Siftflow dataset and show that the presented method outperforms the state-of-the-art by more than 10%. Our method provides a natural way to make use of readily available weak labeled data at a large scale, and hence offers a potential to build a base of visual knowledge [6] using for example data from the Internet.

Acknowledgments: We thank NVIDIA Corporation for the donation of GPUs used in this research. This work was partially funded by NSF RI 1116584 and ONR-N00014-14-1-0232.

References

[1] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale Combinatorial Grouping. In *Proc. CVPR*, 2014. 2, 6

[2] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 82(2):113–132, 2009. 1, 2

[3] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *Proc. ICCV*, 2001. 1, 2

[4] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Proc. CVPR*, 2010. 2

[5] L. C. Chen, S. Fidler, A. Yuille, and R. Urtasun. Beat the MTurkers: Automatic Image Labeling from Weak 3D Supervision. In *Proc. CVPR*, 2014. 2

[6] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Proc. ICCV*, 2013. 8

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009. 1

[8] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *Proc. CVPR*, 2012. 1, 2, 5

[9] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *PAMI*, 36(2), 2014. 2

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers. In *Proc. ICML*, 2012. 1, 5

[11] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic segmentation. In *Proc. CVPR*, 2014. 2, 7

[12] L. Grady. Random walks for image segmentation. *PAMI*, 28(11):1768–1783, 2006. 1, 2

[13] L. Grady. Minimal Surfaces Extend Shortest Path Segmentation Methods to 3D. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):321–334, 2010. 5

[14] M. Guillaumin, D. Kttel, and V. Ferrari. Imagenet auto-annotation with segmentation propagation. *IJCV*, 110(3):328–348, 2014. 1

[15] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010. 1, 2

[16] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. ECCV*, 2014. 2

[17] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *Proc. CVPR*, 2012. 2

[18] G. Kim and E. P. Xing. On multiple foreground cosegmentation. In *Proc. CVPR*, 2012. 2

[19] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Graph Cut based Inference with Co-occurrence Statistics. In *Proc. ECCV*, 2010. 1, 2

[20] 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. 1, 2

[21] C. Liu, J. Yuen, and A. Torralba. Nonparametric Scene Parsing via Label Transfer. *PAMI*, 2011. 1, 2, 5, 6

[22] S. Liu, S. Yan, T. Zhang, C. Xu, J. Liu, and H. Lu. Weakly supervised graph propagation towards collective image parsing. *IEEE Transactions on Multimedia*, 14(2):361–373, 2012. 2, 5

[23] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *Proc. CVPR*, 2011. 2

[24] L. Mukherjee, V. Singh, J. Xu, and M. D. Collins. Analyzing the subspace structure of related images: Concurrent segmentation of image sets. In *Proc. ECCV*, 2012. 2

[25] M. Pandey and S. Lazebnik. Scene Recognition and Weakly Supervised Object Localization with Deformable Part-Based Models. In *Proc. ICCV*, 2011. 2

[26] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *Proc. CVPR*, 2014. 7

[27] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *Siggraph*, 2004. 1, 2

[28] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *Proc. CVPR*, 2006. 2

[29] M. Rubinstein, C. Liu, and W. T. Freeman. Annotation propagation in large image databases via dense image correspondence. In *Proc. ECCV*, 2012. 2, 5

[30] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction with Latent Variables for General Graphical Models. In *Proc. ICML*, 2012. 1

[31] A. G. Schwing and R. Urtasun. Fully Connected Deep Structured Networks. http://arxiv.org/abs/1503.02351, 2015. 1

[32] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008. 5, 6

[33] G. Singh and J. Kosecka. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *Proc. CVPR*, 2013. 1, 2, 5

[34] H. O. Song, R. B. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. One-bit object detection: On learning to localize objects with minimal supervision. In *Proc. ICML*, 2014. 2

[35] J. Tighe and S. Lazebnik. Superparsing - Scalable Nonparametric Image Parsing with Superpixels. *IJCV*, 2013. 1, 2, 5

[36] J. Tighe and S. Lazebnik. Scene Parsing with Object Instances and Occlusion Ordering. In *Proc. CVPR*, 2014. 1, 2, 5

[37] A. Vezhnevets. *Weakly Supervised Semantic Segmentation of Natural Images*. PhD thesis, ETH Zurich, 2012. 1, 2, 5

[38] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with a multi image model. In *Proc. ICCV*, 2011. 1, 2, 5, 7

[39] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly Supervised Structured Output Learning for Semantic Segmentation. In *Proc. CVPR*, 2012. 1, 2, 3, 5

[40] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation Revisited: Models and Optimization. In *Proc. ECCV*, 2010. 2

[41] W. Xia, C. Domokos, J. Dong, L.-F. Cheong, and S. Yan. Semantic segmentation without annotating segments. In *Proc. ICCV*, 2013. 1, 2

[42] J. Xu, M. D. Collins, and V. Singh. Incorporating User Interaction and Topological Constraints within Contour Completion via Discrete Calculus. In *Proc. CVPR*, 2013. 2

[43] J. Xu, A. G. Schwing, and R. Urtasun. Tell me what you see and i will show you where it is. In *Proc. CVPR*, 2014. 1, 2, 3, 5, 7

[44] J. Yang, B. L. Price, S. Cohen, and M. Yang. Context Driven Scene Parsing with Attention to Rare Classes. In *Proc. CVPR*, 2014. 1, 2, 5

[45] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. CVPR*, 2012. 1, 2, 5

[46] B. Zhao, J. T. Kwok, and C. Zhang. Maximum Margin Clustering with Multivariate Loss Function. In *Proc. ICDM*, 2009. 3

[47] B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. In *Proc. ICML*, 2008. 3