Tell Me What You See and I will Show You Where It Is

Jia Xu[1] Alexander G. Schwing[2] Raquel Urtasun[2,3]
[1]University of Wisconsin-Madison [2]University of Toronto [3]TTI Chicago
jiaxu@cs.wisc.edu {aschwing, urtasun}@cs.toronto.edu

Abstract

We tackle the problem of weakly labeled semantic segmentation, where the only source of annotation are image tags encoding which classes are present in the scene. This is an extremely difficult problem as no pixel-wise labelings are available, not even at training time. In this paper, we show that this problem can be formalized as an instance of learning in a latent structured prediction framework, where the graphical model encodes the presence and absence of a class as well as the assignments of semantic labels to superpixels. As a consequence, we are able to leverage standard algorithms with good theoretical properties. We demonstrate the effectiveness of our approach using the challenging SIFT-flow dataset and show average per-class accuracy improvements of 7% over the state-of-the-art.

1. Introduction

Traditional approaches to semantic segmentation require a large collection of training images labeled at the pixel level. Despite the existence of crowd-sourcing systems such as Amazon Mechanical Turk (MTurk), densely labeling images is still a very expensive process, particularly since multiple annotators are typically employed to label each image. Furthermore, a quality control process is frequently required in order to sanitize the annotations.

Here, we are interested in leveraging weak annotations in order to reduce the labeling cost. In particular, we exploit image tags capturing which classes are present in the scene as our sole source of annotation (see Fig. 1 for an illustration). This is an interesting setting as tags are either readily available within most online photo collections or they can be easily obtained at a lesser cost than annotating semantic segmentation. This task is, however, very challenging, as an appearance model cannot be trained due to the fact that the assignment of superpixels to semantic labels is unknown, even at training time.

Several approaches have investigated this setting. In early work, Verbeek and Triggs [29] proposed the latent aspect model, which employs probabilistic latent semantic analysis (PLSA) to model each image as a finite mixture of latent classes also referred to as aspects. The authors extended this approach to capture spatial relationship via a Markov random field (MRF). This model was further extended in a series of papers by Vezhnevets *et al*. [30, 31, 32], for example, to leverage information between multiple images. However, the resulting optimization problem is very complex and non-smooth, making learning a very difficult task. As a consequence, several heuristics were employed to make the problem computationally tractable.

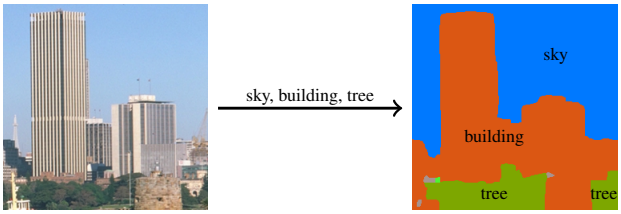

Figure 1. Our approach takes labels in the form of which classes are present in the scene during training, and learns a segmentation model, even though no annotations at the pixel-wise are available.

In this paper, we show that this problem can be formalized as the one of learning in a latent structured prediction framework, where the graphical model encodes the presence/absence of a class as well as the assignments of semantic labels to superpixels. As a consequence, we are able to leverage algorithms with good theoretical properties which have been developed for this more general setting. Under our model, different levels of supervision can be simply expressed by specifying which variables are latent and which are observed, without changing the learning and inference algorithms. We demonstrate the effectiveness of our approach using the challenging SIFT-flow dataset [14], showing improvements of 7% in terms of mean class accuracy over the state-of-the-art. In the remainder, we first review related work. We then present our weakly label segmentation framework, followed by an experimental evaluation and conclusions. Our code is available at http://pages.cs.wisc.edu/ jiaxu/projects/weak-label-seg/.

2. Related Work

Many different techniques have been proposed to handle the fully supervised setting where pixel-wise labels are available at training time. Amongst the most successful techniques are approaches based on bottom-up extraction of regions, which are then re-ranked to obtain the final segmentation [8, 4]. Another popular approach is to formulate segmentation as inference in a (conditional) Markov random field [3], particularly when seeking a full holistic scene interpretation [34, 12].

It is relatively easy for an annotator to provide information about which objects/classes are present in the scene. It is, however, significantly more tedious to carefully outline all visible objects. As a consequence, annotation time and cost can be significantly reduced by leveraging image tags, particularly as these annotations are readily available in many image collections.

There has been, however, little work in the weakly labeled setting due to the fact that it is significantly more challenging than the fully supervised task. One of the first approaches to learn a segmentation model given only image tags is the latent aspect model of [29], which leverages several appearance descriptors and the image location to learn a probabilistic latent semantic analysis (PLSA) model. The name 'aspect model' originates from the famous topic models for document classification and the 'aspects' refer to pixel class labels. Since these models do not capture the spatial 2D relationships commonly observed in images, PLSA was used as unary features in a Markov random field. Generalizations were subsequently introduced in a series of papers [30, 31, 32]. Contrasting the latent aspect model, these new approaches leverage label correlations between different images. However, they result in complex optimization problems which are non-convex, non-smooth and thus very difficult to optimize.

Another form of weak supervision are 2D bounding boxes. Grab-cut and its extensions have been widely used for interactive figure/ground segmentation [2, 20]. These methods learn Gaussian mixture models for the foreground and background, and a binary MRF encoding both appearance and smoothness is employed to perform the segmentation. As the energies employed are sub-modular, exact inference via graph-cuts is possible. Strokes are another popular way to provide weak annotations and are typically used with a human in the loop to correct mistakes. In [18], the deformable part-based model [7] is used with latent structured support vector machines to exploit weak labels in the form of bounding boxes. Recently, [5] showed that human labeling performance can be achieved when exploiting 3D information and weak annotations in the form of 3D bounding boxes.

A related problem is cosegmentation, where one is interested in segmenting objects which concurrently appear in a set of images [21]. Most previous methods focus on the setting where a single foreground object is present in all images [33, 16]. This setting has been extended to segment multiple objects by analyzing the subspace structure of multiple foreground objects [17], using a greedy procedure with submodular optimization [11], or by grouping image regions via spectral discriminative clustering [10].

The work most related to ours is [32], where the problem of weakly labeled segmentation from tags is formulated using a conditional random field (CRF), where nodes represent semantic classes at the superpixel level, unary potentials encode appearance and pairwise potentials encode smoothness. Their key contribution is a three step algorithm to learn the appearance model and the CRF weights. In particular, after every update of the CRF weights, an alternating optimization iterates between finding the pixel-wise labeling given the current model and updating the appearance model given an estimated labeling. The authors view optimization of the feature weights as a model selection procedure where every possible weight vector defines a different model. The optimization criteria employed is expected agreement, which is computed by partitioning the data into two parts which are encourage to agree in their predictions. As the cost function is non-differentiable, they resort to Bayesian optimization to select the next set of parameters. This makes learning extremely difficult and computationally expensive.

In contrast, in this paper we show that the problem of semantic segmentation from weakly labeled data in the form of class presence can be formulated as learning in a structured prediction framework with latent variables. As a consequence, well studied algorithms such as hidden conditional random fields (HCRFs) [19] or latent structured support vector machines (LSSVMs) [35] as well as efficient extensions [23] can be leveraged. This results in simpler optimization problems that can be optimized by algorithms possessing good theoretical guarantees.

3. Weakly Labeled Semantic Segmentation

In this paper we investigate how weak supervision can be used in order to perform semantic segmentation. In particular, we focus on the case where the supervision is given by means of a set of tags, describing which classes are present in the image. Towards this goal, we frame the problem as the one of learning in a graphical model encoding the presence and absence of each class as well as the semantic class of each superpixel.

3.1. Semantic segmentation from tags

More formally, let $y_i \in \{0, 1\}$ be a random variable describing whether the i-th class is present in the image, with $i \in \{1, \dots, C\}$ indexing the semantic classes. Further, let $h_j \in \{1, \dots, C\}$ be a random variable denoting the seman-

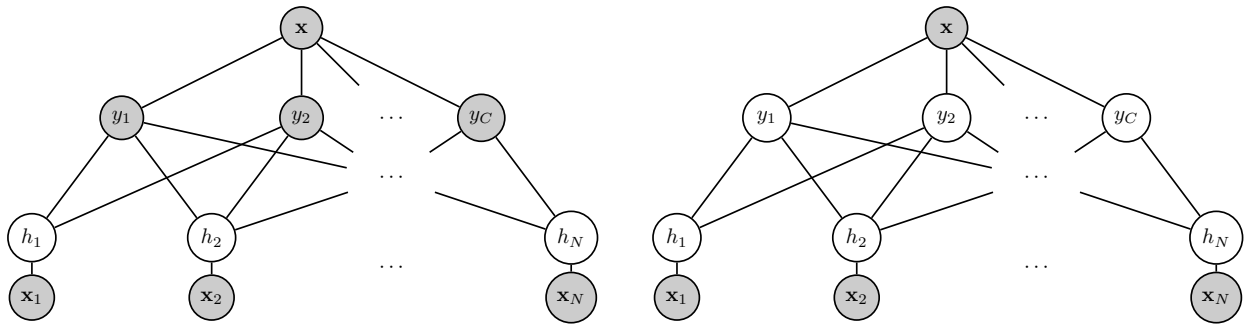

Figure 2. **Graphical Model:** (Left) Graphical model for learning as well as inference when the tags are provided at test time. (Right) Graphical model for inference when the tags are not provided at test time.

tic label associated with the j-th superpixel, and let x be the image evidence. We define $\mathbf{h} = (h_1, \ldots, h_N)$ to be the set of segmentation variables for all superpixels within one image, and $\mathbf{y} = (y_1, \ldots, y_C)$ the set of binary variables indicating for all classes their presence/absence. Note that we assume no training examples to be available for $\mathbf{h}$, and only $\mathbf{y}$ to be labeled. Employing the aforementioned notation, we define the probability for a given configuration $(\mathbf{y}, \mathbf{h})$ given an image x to be

$$p_\epsilon(\mathbf{y}, \mathbf{h} \mid x) = \frac{1}{Z_\epsilon(w)} \exp \frac{w^\top \phi(\mathbf{y}, \mathbf{h}, x)}{\epsilon},$$

where $Z_\epsilon(w)$ is the normalizing constant also known as the partition function. Note that the weights w are the parameters of the model and ϵ is a temperature parameter.

We define the potentials $\phi(\mathbf{y}, \mathbf{h}, x)$ to be the sum of unary terms encoding the likelihood of the tags, unary potentials encoding the appearance model for segmentation and pairwise potentials ensuring compatibility between both types of variables. Thus,

$$w^\top \phi(\mathbf{y}, \mathbf{h}, x) = \sum_i w_i^{pres\,T} \phi^{pres}(x, y_i) + \sum_j w_j^{ap\,T} \phi^{ap}(x, h_j) + \sum_{i,j} w_{i,j}^{co\,T} \phi^{co}(y_i, h_j). \quad (1)$$

Fig. 2 shows the graphical model encoding the dependencies introduced by this probabilistic model, with gray-colored nodes depicting observed variables. We note that this architecture is similar to the first two layers in the holistic model of [34], but we use different potentials and perform semantic segmentation in the weakly labeled setting. We now discuss the potentials employed in more details.

Appearance model: We utilize the superpixel features of [27], which include texture/SIFT, color, shape, location and GIST. This results in a 1690 dimensional feature vector, which we reduce to a 100 dimensional vector using PCA. To form the final feature, we append the superpixel location (*i.e.*, y-coordinate of its center) to form our final feature. Note that we learn a different set of weights for each class, yielding a $101 \cdot C$ dimensional feature vector.

Presence/Absence: We construct a 2D vector to encode the presence of each class. In the training, this potential is built from the ground truth, *i.e.*, $\phi^{pres}(y_i, x) = [1; -1]$ if class i is absent, while $\phi^{pres}(y_i, x) = [-1; 1]$ if class i is present. At test time, when this information is latent, this potential comes from an image level tag classifier. We refer the reader to the experimental section for more details about the specific form of this predictor. Note that typically one will use a predictor both at training and testing time, however, we have found the use of the oracle predictor at training to yield better results in practice. We hypothesize that this is due to the fact that in this setting the supervision is very weak.

Compatibility: The compatibility term encourages the consistency between the class presence variables and the superpixels, such that the information is propagated all the way to the segmentation. In particular, it penalizes configurations where a superpixel is labeled with a class that is inferred to be absent. Thus

$$\phi^{comp}(y_i, h_j) = \begin{cases} -\eta & \text{if } y_i = 0 \text{ and } h_j = i \\ 0 & \text{otherwise} \end{cases}$$

where η is a big number (10^5 in our experiments).

3.2. Learning in the Weakly Labeled Setting

During learning, we are interested in estimating a linear combination of features such that the distribution in Eq. (1) is able to discriminate between 'good' and 'bad' assignments for variables $\mathbf{y}$ and $\mathbf{h}$. To define 'good' we are given a training set of data samples. Contrasting the fully supervised setting where the training samples contain fully labeled configurations $(\mathbf{y}, \mathbf{h})$, the available data is only partly labeled. In particular, the training set $\mathcal{D}$ consists of $|\mathcal{D}|$ image-tag pairs $(\mathbf{y}, x)$, *i.e.*, $\mathcal{D} = \{(\mathbf{y}, x)_i\}_{i=1}^{|\mathcal{D}|}$.

During learning, a loss function $\ell(\hat{\mathbf{y}}, \mathbf{y})$ is commonly included to bias the algorithm, *i.e.*,

$$p_\epsilon^\ell(\hat{\mathbf{y}}, \hat{\mathbf{h}}, x) = \frac{1}{Z_\epsilon^\ell(w)} \exp \frac{w^\top \phi(\hat{\mathbf{y}}, \hat{\mathbf{h}}, x) + \ell(\hat{\mathbf{y}}, \mathbf{y})}{\epsilon}.$$

The intuition behind is to make the problem harder to generalize better. Thus, we want to find a weight vector w, which minimizes the sum of the negative (loss-augmented) marginal log-posterior of the training data $\mathcal{D}$ and a regularization term which originates from a prior distribution on w. The resulting program reads as follows

$$\min_w \frac{1}{2}\|w\|_2^2 - \sum_{(\mathbf{y},x)\in\mathcal{D}} \epsilon \ln \sum_{\hat{\mathbf{h}}} p_\epsilon^\ell(\mathbf{y}, \hat{\mathbf{h}} \mid x). \qquad (2)$$

Note that we marginalize over the unobserved superpixel variables $\mathbf{h}$ to obtain the likelihood of the observed data (i.e., class labels).

The aforementioned program generalizes a few well-known settings. Letting $\epsilon = 0$, we obtain structured support vector machines with latent variables as introduced by [35], while setting $\epsilon = 1$ yields the hidden conditional random field of [19]. In case of fully observed data we obtain the conditional random field framework of [13] or the structured support vector machine of [25, 28] when employing $\epsilon = 1$ and $\epsilon = 0$ respectively.

The weakly labeled setting is significantly more difficult to solve for general graphical models. The additional challenge besides summations over exponentially sized sets $\mathbf{h}$ and $\mathbf{y}$, is the non-convexity of the objective given in Eq. (2) resulting from the partition function. We note, however, that the cost function of the program given in Eq. (2) is a difference of terms, each being convex in the parameters w. We exploit this fact and employ the concave-convex procedure (CCCP) [36], which is a generalization of expectation maximization (EM) to minimize Eq. (2).

CCCP is an iterative approach. At each iteration we linearize the concave part at the current iterate w and solve the remaining convex objective augmented by a linear term to update the weight vector w. Importantly, this approach is guaranteed to converge to a stationary point [24]. To linearize the concave part, we are required to compute an expectation of the feature vector $\phi(\mathbf{y}, \mathbf{h}, x)$ w.r.t. a distribution over the unobserved variables $\mathbf{h}$. More formally this expectation is defined as

$$E_{p(\hat{\mathbf{h}}|x)}\left[\phi(\mathbf{y}, \hat{\mathbf{h}}, x)\right] = \sum_{\hat{\mathbf{h}}} p(\hat{\mathbf{h}} \mid x)\phi(\mathbf{y}, \hat{\mathbf{h}}, x).$$

Given this expectation, we solve a fully supervised objective with modified empirical means. Note that the derivation naturally results in a two-step approach where we first compute a distribution over the unobserved variables $\mathbf{h}$ to obtain the expectation, before using this information to solve the fully supervised learning problem. The procedure is summarized in Fig. 3.

Structured prediction with latent variables
Iterate between

1. The latent variable prediction problem:

$$\forall x \text{ compute} \quad E_{p(\hat{\mathbf{h}}|x)}\left[\phi(\mathbf{y}, \hat{\mathbf{h}}, x)\right]$$

2. Solving the parameter update task

$$\min_w \frac{1}{2}\|w\|_2^2 + \sum_{(\mathbf{y},x)\in\mathcal{D}} \left(\epsilon \ln Z_\epsilon^\ell(w) - w^\top E_{p(\hat{\mathbf{h}}|x)}\left[\phi(\mathbf{y}, \hat{\mathbf{h}}, x)\right]\right)$$

Figure 3. Latent Structured Prediction via CCCP

For the first step it is crucial to notice that in our graphical model we can trivially solve the 'latent variable prediction problem' given the bi-partite model of the weakly labeled segmentation task. Assuming the ground truth tags $\mathbf{y}$ to be known (see Fig. 2), the model decomposes into unaries over superpixels, and inference can be efficiently and exactly solved to yield a distribution $p(\hat{\mathbf{h}} \mid x)$. For the second step we need to solve a fully supervised learning task. We refer the reader to [23] for an efficient way to optimize this cost function.

3.3. Loss function

The distribution of class presence as well as the distribution of pixel-wise labelings follows a power law distribution (*i.e.*, many classes occur very rarely). In order to take this into account we derive a loss function which employs the statistics of class presence at the image level. As the segmentation metric is average per-class accuracy, our loss gives more importance for mistakes in classes that appear very rarely. In particular, for each class i, we count how many training images contain this class, and then normalize this frequency vector $\mathbf{t}$ to sum to 1. The loss function $\ell(\hat{\mathbf{y}}, \mathbf{y})$ is then defined to decompose into a sum of unary terms, *i.e.*, $\ell(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i\in\{1,\dots,C\}} \ell_i(\hat{y}_i, y_i)$ with

$$\ell_i(\hat{y}_i, y_i) = \begin{cases} \frac{1}{t_i} & \text{if } y_i \neq \hat{y}_i \text{ and } y_i = 0 \\ t_i & \text{if } y_i \neq \hat{y}_i \text{ and } y_i = 1 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where y_i is the ground truth label, and $\hat{y}_i$ is the prediction for the i-th class. Note that our loss function is only defined on the class presence variables $\mathbf{y}$ that are observed during training.

3.4. Inference

The configuration with the minimum energy or the highest probability $p(\mathbf{y}, \mathbf{h} \mid x)$, also known as the maximum a

posteriori (MAP) estimate, can be computed by solving the following problem

$$(\mathbf{y}^*, \mathbf{h}^*) = \arg\max_{\mathbf{y},\mathbf{h}} w^\top \phi(\mathbf{y}, \mathbf{h}, x) \quad (4)$$

given an image x. This is an NP-hard task since the optimization is equivalent to an integer linear program. Fortunately, linear programming (LP) relaxations have proven very effective. We employ a message passing approach to leveraging the graphical model structure. In particular, we use distributed convex belief propagation (dcBP) [22], which has convergence guarantees. Note that this is not the case for other message passing algorithms such as loopy belief propagation.

4. Experimental Evaluation

We perform our experiments using the SIFT-flow segmentation dataset [14], which contains 2688 images and $C = 33$ classes. This dataset is very challenging due to the large number of classes (4.43 classes per image) as well as the fact that their frequency is distributed with a power-law. As shown in the first line of Table 2, a few 'stuff' classes like sky, sea and tree are very common, while the 'object' classes like person, bus and sun are very rare. We use the standard dataset split (2488 training images and 200 testing images) provided by [14].

Following [32] we report mean per-class accuracy as our metric. This metric gives the same importance to each class, independently of their frequency. We construct our superpixels using the ultrametric contour map of [1], which respects boundaries well even when a small number of superpixels is used. In our experiments, we set the boundary probability threshold to be 0.14, which results in 19 segments per image on average.

In our experiments we exploit two settings. In the first case we follow the standard weakly labeled setting, in which only image level tags are given for training and no annotations are given at the pixel-level. During testing, no source of annotation is provided. Learning in this setting corresponds to the graphical model in Fig. 2 (left), while inference is shown on Fig. 2 (right). In the second setting we assume that tags are given both at training and test time, and thus the graphical model in Fig. 2 (left) depicts both learning and inference. This is a natural setting when employing image collections where tags are readily available.

Our first experiment utilizes tags only at training. We utilize an image-tag classifier which leverages deep learning in order to construct $\phi^{press}(x, y_i)$ at test time. In particular, we first extract a 4096 dimensional feature vector for each image from the second to last layer of a convolutional neural network (CNN) pre-trained on ImageNet [6]. We use the publicly available implementation of [9] to compute the features, and a linear SVM per class to form the final potential. We refer to this setting as "Ours (CNN-Tag)."

Method	Supervision	Per-Class (%)
Tighe et al. [26]	full	39.2
Tighe et al. [27]	full	30.1
Liu et al. [14]	full	24
Vezhnevets et al. [31]	weak	14
Vezhnevets et al. [32]	weak	21
Ours (CNN-Tag)	weak	**27.9**
Ours (Truth-Tag)	weak	**44.7**

Table 1. Comparison to state-of-the-art on the SIFT-flow dataset. We outperformed the state-of-the-art in the weakly supervised setting by 7%.

Comparison to the state-of-the-art: Tab. 1 compares our approach to state-of-the-art weakly labeled approaches. For reference, we also include the state-of-the-art when pixel-wise labels are available at training (fully labeled setting). We would like to emphasize that our approach outperforms significantly (7% higher) all weakly labeled approaches. Furthermore, we even outperform the fully supervised method developed by Liu *et al*. [14]. The per-class rates for each class are provided in Tab. 2. We observe that our approach performs well for classes which have very distinctive and consistent appearance, *e.g*., sand, sun, staircases. We missed a few classes, *e.g*., bus, crosswalk, bird, due to their largely varying appearance and small training set size.

Quality of image-tag prediction: Our CNN-Tag predictor predicts tags with an accuracy of 93.7%, which is measured as the mean of the diagonal of the confusion matrix. The last row of Table 2 shows the performance of the tag predictor for each class. Interestingly, tag prediction errors do not correlate well with segmentation errors, e.g., crosswalk and bird tags are predicted with very high accuracy, but segmentation accuracy is very low for both classes.

Qualitative results: Fig. 4 and Fig. 5 show success and failure cases respectively. Typical failure modes are due to under-segmentation when creating the superpixels as well as dealing with classes where different instances have very different appearance, *e.g*., due to viewpoint changes.

Tags given at both training and testing: In our second setting, tags are given both at training and testing. Note that the training procedure here is identical to the previous setting. However, at test time our image level class potentials are built from observed ground truth tags. We denote this setting as "Ours (Truth-Tag)." As shown in Tab. 1, we

%	sky	tree	building	mountain	road	car	sidewalk	sea	window	person	plant	rock	river	grass	door	field	sign	streetlight	sand	fence	pole	bridge	boat	awning	staircase	sun	balcony	crosswalk	bus	bird	**averg.**
Tag freq.	85.4	50.1	45.8	37.9	31.7	23.8	17.0	14.1	13.4	12.7	12.4	10.2	9.8	9.3	9.3	8.9	8.1	8.1	5.8	5.5	3.5	3.4	3.2	2.9	2.4	2.2	1.8	1.4	1.2	0.3	
CNN-Tag	5.8	25.1	18.5	38.0	9.1	6.8	1.6	13.5	8.7	16.7	15.2	60.3	63.2	53.0	48.6	76.7	38.7	26.3	91.1	81.2	40.8	20.1	77.3	56.3	81.8	100.0	43.9	0.5	61.3	0.0	**27.9**
Truth-Tag	12.3	27.9	23.3	33.0	10.0	14.2	4.5	18.8	10.8	22.0	37.1	83.0	64.6	63.1	49.3	81.4	41.7	22.0	87.6	81.3	36.9	39.5	74.9	44.5	79.5	100.0	37.6	23.7	58.5	58.5	**44.7**
CNN-ILT	93.0	81.5	86.5	82.5	91.0	94.5	90.0	97.5	93.0	89.5	86.0	91.0	92.5	89.5	93.5	92.5	91.5	93.0	95.5	93.5	95.5	94.5	96.5	95.0	98.0	100.0	99.0	99.0	98.0	99.0	**93.7**

Table 2. **Accuracy for each class:** First row shows tag frequency (percentage of images) for each class. Rows 2 and 3 show segmentation accuracy for each class when a CNN tag predictor or the ground truth tags are used respectively. The last row shows the accuracy of our image tag predictor for each class.

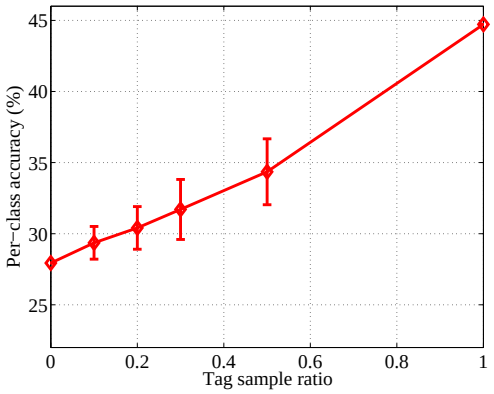

Figure 7. Per-Class accuracy as a function of the percentage of ground-truth tags available at test time.

almost double the per-class accuracy of the previous setting. Surprisingly, we outperformed all fully labeled approaches while not requiring any example to be labeled at the pixel-level. Fig. 6 depicts qualitative results for this setting. When image level tags are given, our approach is able to identify more challenging classes, *e.g.*, buildings.

Partial tags given at test time: We further evaluate our model when only a subset of the tags are provided. For each run, we randomly sample a small portion of ground truth (GT) tags, and predict the remaining ones via our CNN tag classifier. The combined potentials are then fed into our model for inference. We conduct our experiments using four different sample ratios $\{0.1, 0.2, 0.3, 0.5\}$. For each setting, we repeat our procedure 10 times and report the mean and standard deviation. As shown in Fig. 7, our approach gradually improves when more GT tags are given.

5. Conclusion

We have presented an approach to semantic segmentation which is able to exploit weak labels in the form of image tags when no pixel-wise labeling are available. We have shown that this problem can be formulated as structured prediction in a graphical model with latent variables. Unlike existing approaches, this allowed us to leverage standard algorithms with good theoretical guarantees. We have demonstrated the effectiveness of our approach and showed improvements of 7% over the state-of-the-art in this task. Our novel view of the problem can be used to incorporate other types of supervision without changing the learning or inference algorithms. In the future we plan to exploit other annotations such as the type of scene or bounding boxes as well as other forms of learning such as active learning [15] to further reduce the need of supervision.

Acknowledgments: We thank Sanja Fidler and Vikas Singh for helpful discussions. This work was partially funded by NSF RI 1116584 and ONR-N00014-13-1-0721.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 2011. 5

[2] 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. 2

[3] G. Cardinal, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony Potentials for Joint Classification and Segmentation. In *Proc. CVPR*, 2010. 2

[4] J. Carreira, F. Li, and C. Sminchisescu. Object Recognition by Sequential Figure-Ground Ranking. *IJCV*, 2011. 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] 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. 5

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 2

[8] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using region. In *Proc. CVPR*, 2009. 2

[9] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013. 5

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

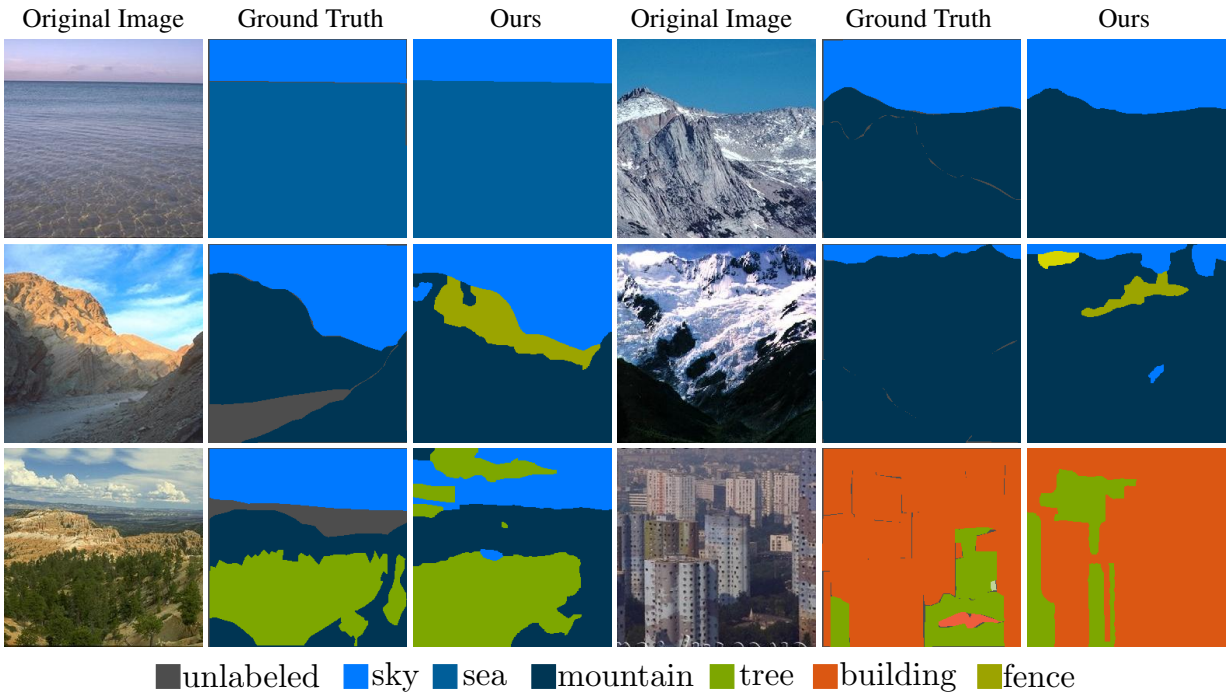

Figure 4. Sample results when tags are predicted at test time using a convolutional net. **Best viewed in color.**

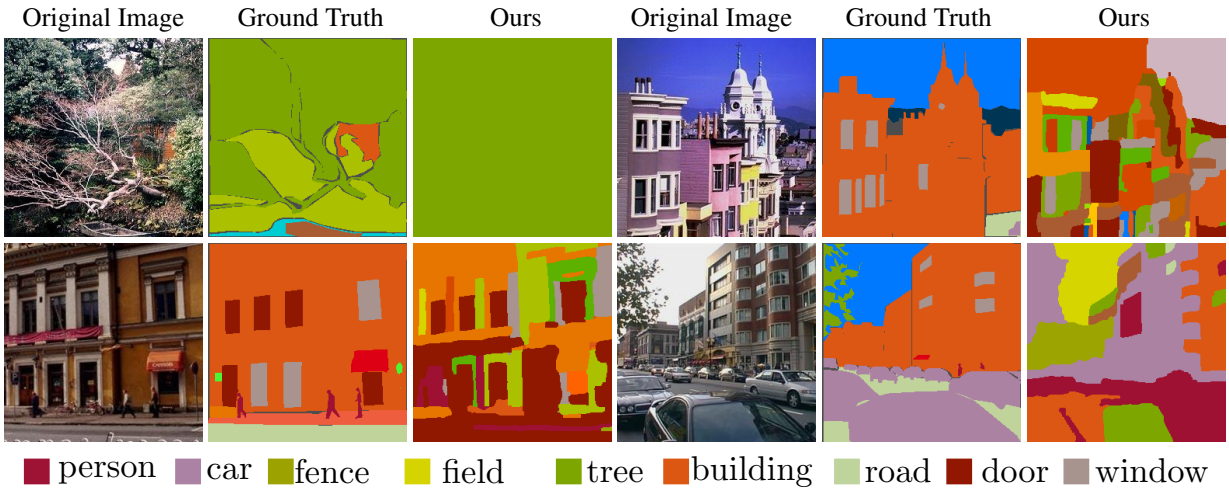

Figure 5. Failure cases when tags are predicted using a convolutional net at test time. **Best viewed in color.**

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

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

[13] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for segmenting and labeling sequence data. In *Proc. ICML*, 2001. 4

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

[15] W. Luo, A. Schwing, and R. Urtasun. Latent structured active learning. In *NIPS*, 2013. 6

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

[17] 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

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

[19] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden-state Conditional Random Fields. *PAMI*, 2007. 2, 4

[20] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": inter-

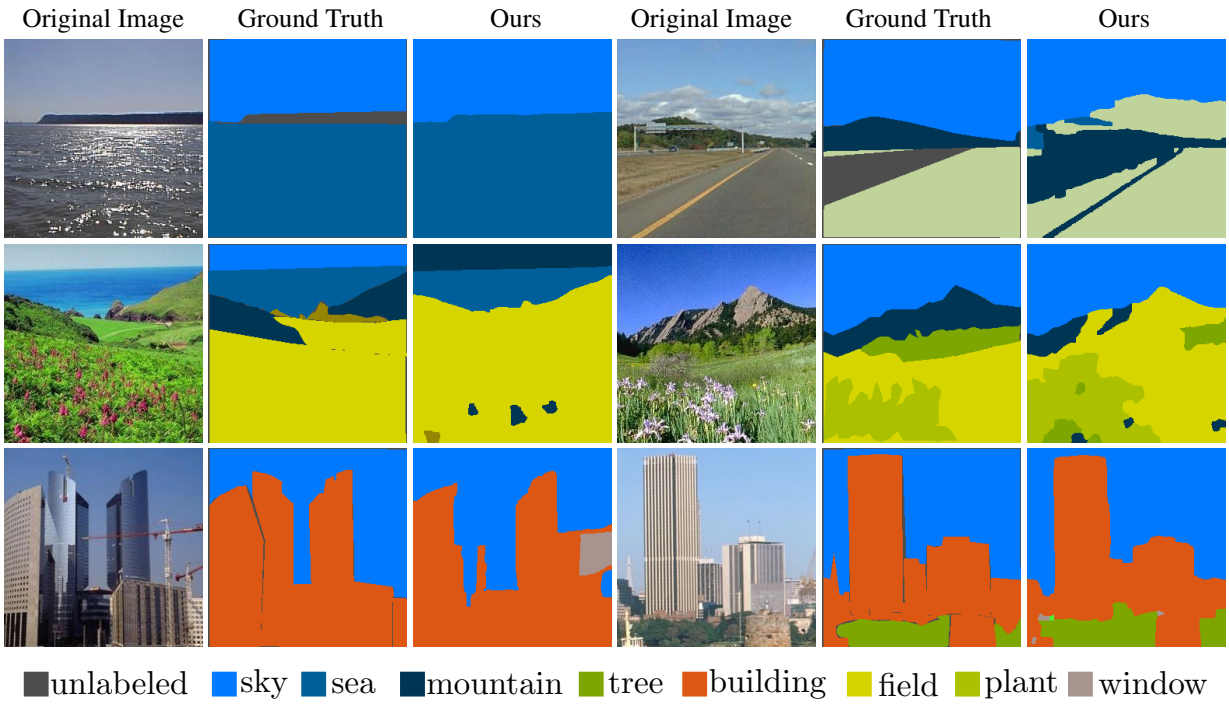

Figure 6. Sample results when ground truth tags are given at test time. **Best viewed in color.**

active foreground extraction using iterated graph cuts. *Siggraph*, 2004. 2

[21] 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

[22] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Distributed Message Passing for Large Scale Graphical Models. In *Proc. CVPR*, 2011. 5

[23] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction with Latent Variables for General Graphical Models. In *Proc. ICML*, 2012. 2, 4

[24] B. Sriperumbudur and G. Lanckriet. On the convergence of the concave-convex procedure. In *Proc. NIPS*, 2009. 4

[25] B. Taskar, C. Guestrin, and D. Koller. Max-Margin Markov Networks. In *Proc. NIPS*, 2003. 4

[26] J. Tighe and S. Lazebnik. Finding Things: Image Parsing with Regions and Per-Exemplar Detectors. In *Proc. CVPR*, 2013. 5

[27] J. Tighe and S. Lazebnik. Superparsing - Scalable Nonparametric Image Parsing with Superpixels. *IJCV*, 2013. 3, 5

[28] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large Margin Methods for Structured and Interdependent Output Variables. *JMLR*, 2005. 4

[29] J. Verbeek and B. Triggs. Region classification with Markov field aspect models. In *Proc. CVPR*, 2007. 1, 2

[30] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *Proc. CVPR*, 2010. 1, 2

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

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

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

[34] 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. 2, 3

[35] C.-N. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proc. ICML*, 2009. 2, 4

[36] A. L. Yuille and A. Rangarajan. The Concave-Convex Procedure (CCCP). *Neural Computation*, 2003. 4