Sparse Dictionaries for Semantic Segmentation

Lingling Tao[1], Fatih Porikli[2], and René Vidal[1]

[1] Center for Imaging Science, Johns Hopkins University, USA
[2] Australian National University & NICTA ICT, Australia

Abstract. A popular trend in semantic segmentation is to use top-down object information to improve bottom-up segmentation. For instance, the classification scores of the Bag of Features (BoF) model for image classification have been used to build a top-down categorization cost in a Conditional Random Field (CRF) model for semantic segmentation. Recent work shows that discriminative sparse dictionary learning (DSDL) can improve upon the unsupervised K-means dictionary learning method used in the BoF model due to the ability of DSDL to capture discriminative features from different classes. However, to the best of our knowledge, DSDL has not been used for building a top-down categorization cost for semantic segmentation. In this paper, we propose a CRF model that incorporates a DSDL based top-down cost for semantic segmentation. We show that the new CRF energy can be minimized using existing efficient discrete optimization techniques. Moreover, we propose a new method for jointly learning the CRF parameters, object classifiers and the visual dictionary. Our experiments demonstrate that by jointly learning these parameters, the feature representation becomes more discriminative and the segmentation performance improves with respect to that of state-of-the-art methods that use unsupervised K-means dictionary learning.

Keywords: discriminative sparse dictionary learning, conditional random fields, semantic segmentation.

1 Introduction

Semantic image segmentation is the problem of inferring an object class label for each pixel [17,12,16,33,27]. This is a fundamental problem in computer vision with many applications in scene understanding, automatic driving, surveillance, etc. However, this problem is significantly more complex than image classification, where one needs to find a single label for the image. This is because the joint labeling of all pixels involves reasoning about the image neighborhood structure, as well as capturing long-range interactions and high-level object class priors.

Prior Work. The most common approach to semantic segmentation is to model the image with a Conditional Random Field (CRF) model [17]. A CRF captures the fact that image regions corresponding to the same object class should have similar features, and regions that are similar to each other (in location or feature space) should be more likely to share the same label. In a second-order CRF model, the features coming from each region are usually modeled by the CRF

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 549–564, 2014.

unary potentials, which are based on appearance, context and semantic relations, while pairwise relationships are modeled by the CRF pairwise potentials, which are based on neighborhood similarity and co-occurrence information. For example, early works use patch/super-pixel/region based features such as a Bag of Features (BoF) representation of color, SIFT features [7,8], textonboost [24], co-occurrence statistics [8], relative location features [9], etc. Once the CRF model has been constructed, multi-label graph cuts [13] or other approximate graph inference algorithms can be used to efficiently find an optimal segmentation.

In spite of their success, a major disadvantage of second-order CRF models is that the features they use are too local to capture long-range interactions and object-level information. To address this issue, various methods have been proposed. One family of methods [3,15,33,22,27] uses other cues such as object detection scores, shape priors, motion information and scene information, to improve object segmentation. For instance, [15,22] combine object detection results with pixel-based CRF models; [33] further improves the algorithm by combining object detection results with shape priors and scene classification information for holistic scene understanding; and [27] uses exemplar-SVMs to get the detection results together with shape priors, and combines them with appearance models. Another family of methods uses more complex higher-order or hierarchical CRF models. For instance, [12] shows that the integration of higher-order robust P^N potentials improves over the second-order CRF formulation. Also [16] proposes a hierarchical CRF combining both segment-based and pixel-based CRF models using robust P^N potentials. However, a major drawback of these methods is that the CRF cliques need to be predefined. Hence they cannot capture global information about the entire object because the segmentation is unknown.

To address this issue, [26] proposes to augment the second-order CRF energy with a global, top-down categorization potential based on the BoF representation for image classification [6,18]. This potential is obtained as the sum of the scores of a multi-class SVM classifier applied to multiple BoF histograms per image, one per object class. Since each histogram depends on the unknown segmentation, during inference one effectively searches for a segmentation of the image that gives a good classification score for each histogram. While in this approach of [26] the visual words are learned independently from the classifiers, [10] shows how to extend this method by using a discriminative dictionary of visual words, which is learned jointly with the CRF parameters. Both approaches are, however, limited by the simplicity of the BoF framework. Recent work shows that discriminative sparse representations can improve over the basic BoF model for classification due to their ability to capture discriminative features from different classes. For instance, [20] proposes to learn a discriminative dictionary such that the classification scores based on the sparse representation are well separated; [32] shows that extracting sparse codes with a max-pooling scheme outperforms BoF for object and scene classification; [2] further improves classification performance by jointly learning the dictionary and the classifier parameters; and [1] presents a general formulation for supervised dictionary learning adapted to various tasks. However, these approaches have not been applied to semantic segmentation.

Paper Contributions. In this paper, we propose a novel framework for semantic segmentation based on a new CRF model with a top-down discriminative sparse dictionary learning cost. Our main contributions are the following:

1. A new categorization cost for semantic segmentation based on discriminative sparse dictionary learning. Although similar approaches have been explored in image classification tasks [20,32,2,1] and shown good performance, they have not been used to model top-down information in semantic labeling.
2. A new algorithm for jointly learning a sparse dictionary and the CRF parameters, which makes the learned dictionary more discriminative and specifically trained for the segmentation task. Prior work in this area either learned the dictionary beforehand or used energies that are linear on the dictionary and classifier parameters, which makes the learning problem amenable to structural SVMs [11] or latent structural SVMs [34]. In sharp contrast, we use a sparse dictionary learning cost, which makes the energy depend nonlinearly on the dictionary atoms. The learning problem we confront is, thus, significantly more difficult and requires the development of an ad-hoc learning method. Here, we propose a method based on stochastic gradient descent.
3. From a computational perspective, our approach is more scalable than that of [26]. This is because the approach in [26] is based on minimizing an energy involving the histogram intersection kernel, which requires the construction of graphs with many auxiliary variables. On the other hand, our learning scheme utilizes a stochastic gradient descent method, which requires fewer graph-cut inference computations for each training loop.

To the best of our knowledge, there is little work on using discriminative sparse dictionaries for semantic segmentation. This is arguably due to the complexity of jointly learning the dictionary and the CRF parameters. The only related works we are aware of are [35,31]. In [35], a sparse dictionary is used to build a sparse reconstruction weight matrix for all the super-pixels. Then a set of representative super-pixels for each class is learned based on the weight matrix, and classification is done by comparing reconstruction errors from each class. However, the atoms of the dictionary used in this model are all the data samples from one object class, thus there is no learning involved. On the other hand, in [31], a grid-based CRF is defined to model the top-down saliency of the image. The unary cost for each point on the grid is associated with the sparse representation of the SIFT descriptor at that point. A max-margin formulation and gradient descent optimization is then used to jointly learn the dictionary and the classifier. But this model gives only a binary segmentation on the grid, and requires fitting one dictionary per class, which could be computationally expensive for semantic segmentation tasks with a large number of classes.

Paper Outline. The rest of the paper is organized as follows. In §2 we review the basic CRF model and the CRF model with higher-order BoF potentials. In §3 we introduce higher-order potentials based on discriminative sparse dictionary learning. We describe how inference is done and propose a gradient descent method for jointly learning the dictionary and CRF parameters. In §4 we present some experimental results as well as a discussion of possible improvements.

2 Review of CRF Models for Semantic Segmentation

In this section, we describe how the semantic segmentation problem is formulated using a CRF model. In principle, the goal is to compute an object category label for each pixel in the image. In practice, however, the image is often over-segmented into super-pixels and the goal becomes to label each super-pixel. To that end, the image I is associated with a graph $G = (V, E)$, where V is the set of nodes and $E \subset V \times V$ is the set of edges. Each node $i \in V$ is a super-pixel and is associated with a label $x_i \in \{1, \dots, L\}$, where L is the number of object classes. Two nodes are connected by an edge if their super-pixels share a boundary.

To find a labeling $X = \{x_i\}_{i=1}^{|V|}$ for image I, rather than modeling the joint distribution of all labels $P(X)$, a CRF models the conditional distribution of the labels given the observations $P(X \mid I)$ with a Gibbs distribution of the form

$$P(X \mid I) \propto \exp\big(-E(X, I)\big), \tag{1}$$

where the energy function $E(X, I)$ is the sum of potentials from all cliques of G.

Second-Order CRF Model. In the basic second-Order CRF model, the energy function is given as

$$E(X, I) = \lambda_1 \sum_{i \in V} \phi_i^U(x_i, I) + \lambda_2 \sum_{(i,j) \in E} \phi_{ij}^P(x_i, x_j, I). \tag{2}$$

The unary potential $\phi^U(x_i, I)$ models the cost of assigning class label x_i to super-pixel i, while the pairwise potential $\phi_{ij}^P(x_i, x_j, I)$ models the cost of assigning a pair of labels (x_i, x_j) to a pair of neighboring super-pixels $(i, j) \in E$. Then, the best labeling is the one that maximizes the conditional probability, and thus minimizes the energy function. In this work, we will use different state-of-art choices for the unary and pairwise potentials, as described in the experiments.

Top-Down BoF Categorization Cost. As discussed before, the basic CRF model does not capture high-level information about an object class. To address this issue, [26] proposes a higher-order potential based on the BoF approach. The key idea is to represent an image I with L class-specific histograms $\{h_l(X)\}_{l=1}^{L}$, each one capturing the distribution of image features for one of the object classes. Let D be a dictionary of K visual words learned from all training images using K-means. Let $b_j \in \mathbb{R}^K$ be the encoding of feature descriptor f_j at the j-th interest point, i.e., $b_{jk} = 1$ if the j-th descriptor is associated with the k-th visual word, and $b_{jk} = 0$ otherwise. A BoF histogram for class l is constructed by accumulating b_j over interest points that belong to super-pixels with label l, that is

$$h_l(X) = \sum_{j \in S} b_j \delta(x_{s_j} = l), \tag{3}$$

where S is the set of all interest points in image I and $s_j \in V$ is the super-pixel containing interest point j. A top-down categorization cost is then defined by applying a classifier $\phi_l^O(\cdot)$ to this BoF histogram. To encourage the optimal segmentation to be such that the distribution of features within each segment resemble that of one of the object categories, the L categorization costs are integrated with the basic CRF model by defining the following energy

$$E(X,I) = \lambda_1 \sum_{i \in V} \phi_i^U(x_i, I) + \lambda_2 \sum_{(i,j) \in E} \phi_{ij}^P(x_i, x_j, I) + \sum_{l=1}^{L} \phi_l^O(h_l(X)). \qquad (4)$$

It is shown in [26] that if the classifiers ϕ_l^O are linear or intersection-kernel SVMs, the minimization of the energy can be done using extensions of graph cuts and that the CRF parameters can be learned by structural SVMs.

One drawback of the approach in [26] is that the dictionary is fixed and learned independently from the CRF parameters via K-means. To address this issue, [10] proposes to learn the dictionary of visual words jointly with the CRF parameters by defining a classifier for each visual word and augmenting the energy with a dictionary learning cost. Since the assignments of visual descriptors to visual words are unknown, these assignments become latent variables for the energy. The optimal segmentation and visual words assignments can be found via a combination of graph cuts and loopy belief propagation [21], and the dictionary and CRF parameters are then jointly learned by latent structural SVMs [34].

3 Proposed Discriminative Dictionary Learning CRF Cost

In this section, we present a discriminative sparse dictionary learning cost for semantic segmentation. As in [26,10], this cost is based on the construction of a classifier applied to a class-specific histogram. However, the key difference is that our histogram is a sum pooling over the sparse coefficients of all feature descriptors associated with a class. While histograms of this kind have been used for classification (see, e.g., [32]), the fundamental challenge when using them for segmentation is that the histograms depend on both the segmentation and the dictionary. In particular, the histograms depend nonlinearly on the dictionary, which makes learning methods based on latent structural SVMs no longer applicable. In what follows, we describe the details of the new categorization cost as well as how we solve the inference and learning problems.

Top-Down Sparse Dictionary Learning Cost. Let $D \in \mathbb{R}^{F \times K}$ be an unknown dictionary of K visual words, with each visual word normalized to unit norm. Each feature descriptor f_j is encoded with respect to D via sparse coding, which involves solving the following problem:

$$\alpha_j(D) = \operatorname*{argmin}_{\alpha} \{ \frac{1}{2} \| f_j - D\alpha \|^2 + \lambda \|\alpha\|_1 \}. \qquad (5)$$

Note the implicit nonlinear dependency of α on D. The sparse codes of all feature descriptors associated with class l are then used to construct a histogram

$$h_l(X,D) = \sum_{j \in S} \alpha_j(D) \delta(x_{s_j} = l) = \sum_{i \in V} \sum_{j \in S_i} \alpha_j(D) \delta(x_i = l), \qquad (6)$$

where S_i is the set of feature points that belong to super-pixel i. Note the dependency of h_l on both the segmentation X and the dictionary D. Finally, let $w_l \in \mathbb{R}^F$ be the parameters of a linear classifier for class l, where we remove the bias term to simplify the computation. Then the energy function in (4) becomes

$$E(X,I) = \lambda_1 \sum_{i\in V} \phi_i^U(x_i,I) + \lambda_2 \sum_{(i,j)\in E} \phi_{ij}^P(x_i,x_j,I) + \sum_{l=1}^{L} w_l^\top h_l(X,D). \tag{7}$$

Inference. Given an image I, the CRF parameters λ_1, λ_2, the classifier parameters $\{w_l\}_{l=1}^L$, and the dictionary D, our goal is to compute the labeling X^* that maximizes the conditional probability, i.e.,

$$X^* = \operatorname*{argmax}_X P(X \mid I) = \operatorname*{argmin}_X E(X,I). \tag{8}$$

To that end, notice that the top-down categorization term can be decomposed as a summation of unary potentials

$$\sum_{l=1}^{L} w_l^\top h_l(X,D) = \sum_{l=1}^{L} w_l^\top \sum_{i\in V}\sum_{j\in S_i} \alpha_j(D)\delta(x_i=l) = \sum_{i\in V} \overbrace{w_{x_i}^\top \sum_{j\in S_i} \alpha_j(D)}^{\psi_i^O(x_i,I)}. \tag{9}$$

Therefore, we can represent the cost function as

$$E(X,I) = \sum_{i\in V}\{\lambda_1\phi_i^U(x_i,I) + \psi_i^O(x_i,I)\} + \lambda_2 \sum_{(i,j)\in E} \phi_{ij}^P(x_i,x_j,I). \tag{10}$$

Since this energy is the sum of unary and pairwise potentials, it can be minimized using approximate inference algorithms, such as α expansion, $\alpha-\beta$ swap, etc.

Parameter and Dictionary Learning. Given a training set of images $\{I^n\}_{n=1}^N$ and their corresponding segmentations $\{X^n\}_{n=1}^N$, we now show how to learn the CRF parameters λ_1, λ_2, the classifier parameters $\{w_l\}_{l=1}^L$, and the dictionary D.

When D is known, we can approach the learning problem using the structural SVM framework [11]. To that end, we first rewrite the energy function as

$$E(X,I) = W^\top \Phi(X,I,D), \tag{11}$$

where

$$W = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ w_1 \\ \vdots \\ w_L \end{bmatrix} \quad \text{and} \quad \Phi(X,I,D) = \begin{bmatrix} \sum_{i\in V} \phi^U(x_i,I) \\ \sum_{(i,j)\in E} \phi_{ij}^P(x_i,x_j,I) \\ \sum_{i\in V}\sum_{j\in S_i} \alpha_j\delta(x_i=1) \\ \vdots \\ \sum_{i\in V}\sum_{j\in S_i} \alpha_j\delta(x_i=L) \end{bmatrix}. \tag{12}$$

We then seek a vector of parameters W of small norm such that the energy at the ground truth segmentation $E(X^n,I^n)$ is smaller than the energy at any other segmentation $E(\hat{X}^n,I^n)$ by a loss $\Delta(\hat{X}^n,X^n)$.[1] That is, we solve the problem

[1] We use a scaled Hamming loss $\Delta(\hat{X}^n,X^n) = \gamma\sum_{l=1}^L \frac{1}{N_l}\sum_{i\in V}\delta(\hat{x}_i^n = x_i^n)\delta(x_i^n = l)$.

$$\begin{aligned} \min_{W,\{\xi_n\}} \quad & \frac{1}{2}\|W\|^2 + \frac{C}{N}\sum_{n=1}^{N}\xi_n \\ \text{s.t.} \quad & \forall n \in \{1,\dots,N\}, \forall \hat{X}^n \\ & W^\top \Phi(\hat{X}^n, I^n, D) - W^\top \Phi(X^n, I^n, D) \geq \Delta(\hat{X}^n, X^n) - \xi_n, \end{aligned} \tag{13}$$

where $\{\xi_n\}$ are slack variables that account for the violation of the constraints.

The problem in (13) is a quadratic optimization problem subject to a combinatorial number of linear constraints in W, one for each labeling $\hat{X}^n$. As shown in [11], this problem can be solved using a cutting plane method that alternates between two steps: given W one finds the most violated constraint by solving for $\bar{X}^n = \operatorname{argmin}_{\hat{X}}\{W^\top \Phi(\hat{X}, I^n, D) - \Delta(\hat{X}, X^n)\}$, and given a set of constraints $\bar{X}^n$ one solves for W with this constraint added.

Unfortunately, in our case both W and D are unknown. Moreover, the energy is not linear in D and its dependency on D is not explicit. As a result, the cutting plane method does not apply to our problem. Therefore, we propose an alternative approach inspired by recent work on image classification [1,2,31].

Let us first rewrite the optimization problem in (13) over both W and D as:

$$J(W, D) = \tag{14}$$
$$\frac{1}{2}\|W\|^2 + \frac{C}{N}\sum_{n=1}^{N}\Big[W^\top \Phi(X^n, I^n, D) - \min_{\hat{X}^n}\{W^\top \Phi(\hat{X}^n, I^n, D) - \Delta(\hat{X}^n, X^n)\}\Big].$$

The basic idea is to solve this problem by stochastic gradient descent and the key challenge is the computation of the gradient with respect to D. Let us denote the variables after the t-th iteration as D_t and W_t, and the most violated constraint as $\{\bar{X}_t^n\}$. We can easily compute the derivative of J with respect to W as:

$$\frac{\partial J}{\partial W}\Big|_{W_t, D_t} = W_t + \frac{C}{N}\sum_{n=1}^{N}(\Phi(X^n, I^n, D_t) - \Phi(\bar{X}_t^n, I^n, D_t)). \tag{15}$$

To compute the derivative of J with respect to D, notice that J depends implicitly on D through the sparse codes $\{\alpha_j\}$. Thus, we can compute $\partial J/\partial D$ using the chain rule, which requires computing $\partial J/\partial \alpha$ and $\partial \alpha/\partial D$.

Under certain assumptions, $\partial \alpha/\partial D$ can be computed as shown in [1,2,31]. Specifically, since $\mathbf{0}$ has to be a subgradient of the objective function in (5), the sparse representation α of feature descriptor f must satisfy

$$D^\top (D\alpha - f) = -\lambda \operatorname{sign}(\alpha). \tag{16}$$

Now, suppose that the support of α (denoted as Λ) does not change when there is a small perturbation of D and let $A = (D_\Lambda^\top D_\Lambda)^{-1}$, where D_Λ is a submatrix of D whose columns are indexed by Λ. After taking the derivative of (16) with respect to D we get:

$$\frac{\partial \alpha_{(k)}}{\partial D} = (f - D\alpha)A_{[k]} - (DA^\top)_{\langle k\rangle}\alpha^\top \quad \forall k \in \Lambda, \tag{17}$$

Algorithm 1. Parameter Learning for Semantic Labeling with Sparse Dictionaries

```
Initialize the parameter with $W_0$ and $D_0$
while iter $t \leq$ maxiter do
    Randomly select $Q$ images
    for $q = 1, \dots, Q$ do
        Compute sparse code $\alpha$ for $q$-th image using Eqn. (5)
        Find the most violated constraint $\bar{X}^q$ for this sample
    end for
    Compute the partial gradient of $W$ and $D$ corresponding to these $Q$ samples
    using Eqn. (15) and Eqn. (19). Denote them as $g_{Wt}$ and $g_{Dt}$ respectively.
    Gradient Descent: $W_{t+1} = W_t - \tau_t g_{Wt}, D_{t+1} = D_t - \tau_t g_{Dt}$
    $D_{t+1}$=normalize($D_t$)
    $t++$
end while
```

where (k), $[k]$, and $\langle k \rangle$ denote the k-th entry, row, and column, respectively.

Given the set of images $\{I^n\}_{n=1}^N$ with the corresponding set of feature points $\{S^n\}_{n=1}^N$, one can apply the chain rule to compute $\frac{\partial J}{\partial D}$. Denote $z_j^n = \frac{\partial J}{\partial \alpha_j^n}$ as the partial derivative of J with respect to the sparse codes α_j^n of feature point j in image I^n, then

$$z_j^n = \frac{\partial J}{\partial \alpha_j^n}\Big|_{W_t, D_t} = w_{x_{s_j}^n, t} - w_{\hat{x}_{s_j,t}^n, t}, \tag{18}$$

where $x_{s_j}^n, \hat{x}_{s_j,t}^n$ denote the ground-truth label and the computed label of feature point f_j^n at iteration t respectively. According to the chain rule, we have

$$\begin{aligned}
\frac{\partial J}{\partial D} &= \sum_{n=1}^{N} \sum_{j \in S^n} \frac{\partial J}{\partial \alpha_j^n}^{\top} \frac{\partial \alpha_j^n}{\partial D} = \sum_{n=1}^{N} \sum_{j \in S^n} \sum_{k \in \Lambda_j^n} \frac{\partial J}{\partial \alpha_j^n(k)}^{\top} \frac{\partial \alpha_j^n(k)}{\partial D} \\
&= \sum_{n=1}^{N} \sum_{j \in S^n} \sum_{k \in \Lambda_j^n} z_j^n(k) \{ (f_j^n - D\alpha_j^n) A_{j[k]}^n - (DA_j^{\top})_{\langle k \rangle} {\alpha_j^n}^{\top} \} \\
&= \sum_{n=1}^{N} \sum_{j \in S^n} (f_j^n - D\alpha_j^n)(A_j^n z_j^n)^{\top} - D{A_j^n}^{\top} z_j^n {\alpha_j^n}^{\top},
\end{aligned} \tag{19}$$

where $A_j^n = (D_{\Lambda_j^n}^{\top} D_{\Lambda_j^n})^{-1}$. For simplicity, we removed the sub-script t from all the variables that change through iterations.

Instead of summing over all the image samples, in our algorithm, we use stochastic gradient descent, i.e., at each iteration we select a small subset of sample images and compute the gradient based on this subset only. The detailed algorithm is described in Algorithm 1.

Since the problem of jointly learning D and W is non-convex, it is very important to have a good initialization for Algorithm 1. We compute D_0 by applying the sparse dictionary learning algorithm of [19] to all feature descriptors $\{f_j\}$.

We then compute W_0 as $[\lambda_1, \lambda_2, \lambda_3 w_1, \ldots, \lambda_3 w_L]$, where $\{w_l\}_{l=1}^{L}$ are the parameters of a multi-class linear SVM classifier (without bias term) trained on the histograms $\{h_l(X^n, D_0)\}$, and λ_1, λ_2, λ_3 are the parameters of the model

$$E(X, I) = \lambda_1 \sum_{i \in V} \phi^U(x_i, I) + \lambda_2 \sum_{(i,j) \in E} \phi_{ij}^P(x_i, x_j, I) + \lambda_3 \sum_{l=1}^{L} w_l^\top h_l(X, D_0) \quad (20)$$

trained on the segmentations $\{X^n\}$ using standard structural SVM learning.

4 Experimental Results

Datasets. We evaluate our algorithm on three datasets: the Graz-02 dataset, the PASCAL VOC 2010 dataset and the MSRC21 dataset. The Graz-02 Dataset [23] contains 900 images of size 480×640. Each image is labeled with 4 categories: bike, pedestrian, car and background. In our experiments, we use 450 images for training and the other 450 for testing. The PASCAL VOC 2010 dataset [5] contains 1928 images labeled with 20 object classes and a background class. Following [14], since there is no publicly available groundtruth for the test data, we split the training/validation dataset and use 600 images among them for training, 364 for validation and 964 for testing. The MSRC21 dataset [25] consists of 591 color images of size 320×213 and corresponding ground-truth labeling for 21 classes. The standard train-validation-test split is used as described in [25].

Metric. We evaluate our algorithm using two performance metrics: accuracy and intersection-union metric (VOC measure). We compute the per-class accuracy as the percentage of pixels that are classified correctly for each object class, and report the 'average' accuracy (the mean of the per-class percentages) and the 'global' accuracy (the percentage of pixels from all classes that are classified correctly). We compute the VOC measure for each object class as $\frac{\#TP}{\#TP + \#FP + \#FN}$, where $\#TP$, $\#FP$ and $\#FN$ are the number of true positives, false positives and false negatives, respectively, and report the mean VOC measure over all classes.

Top-Down Term. Since this framework is general, it can be applied with different unary, pairwise and top-down terms with different features. In our experiments, we used three different methods to extract feature points and compute object-level histograms. In the first method (TP1), we extract sparse SIFT features for each image at detected interest points, similar to [26,10]. In this case, each super-pixel region can contain 0, 1 or more feature points, and we use the absolute value of the sparse code for our top-down term. In the second method (TP2), we extract one SIFT feature at the center of each super-pixel region, to capture the texture of the whole region. In the third method (TP3), we compute the vectorized average TextonBoost scores of all pixels in each super-pixel as feature points. In the last two methods, each super-pixel is associated with only one feature point. The first two methods are used for the Graz02 dataset, while the third method is used for both the PASCAL VOC and MSRC21 datasets.

Unary Potentials. We use different unary potentials for different datasets. For the Graz-02 dataset, we use the same unary potentials as in [26,10] in order to make our results comparable. Specifically, we first create super-pixels by over-segmenting each image using the Quick Shift algorithm [30]. Then we extract dense SIFT features on each image, and compute the BoF representation for each super-pixel region. We then train an SVM with a χ^2-RBF kernel using LibSVM [4]. For each super-pixel, we apply the SVM classifier to the associated histogram and compute the logarithm of the output probability as the unary potential. For the PASCAL VOC and MSRC21 datasets, we use the pixel-wise unaries based on TextonBoost classifier provided by [14]. The super-pixel unary potentials are then computed by first taking the logarithm of the probabilities and then averaging over all pixels inside each super-pixel.

Pairwise Potentials. For all datasets, we use a contrast sensitive cost $\frac{B_{ij}}{1+\|C_i-C_j\|}$ [10] as pairwise potentials, where B_{ij} is the length of shared boundary between super-pixel i and j, and C_i is the mean color of super-pixel i.

Implementation Details. We use the VL_feat toolbox [29] for preprocessing. We use vl_quickshift to generate super-pixels and set the parameter that controls super-pixel size to $\tau = 8$. When extracting dense SIFT features to construct the unaries, we use the vl_dsift function with spatial bin size set to 12. To define the top-down cost, when computing sparse SIFT features (TP1), we apply the vl_sift function with default settings, while for TP2, we set the position for SIFT features to be the center position of each super-pixel, and the spatial bin size to 8. For initializing the linear classifiers $w_1, \dots, w_L$, we use the Matlab Structural SVM toolbox [28]. For initializing the dictionary and computing sparse representations, we use the sparse coding toolbox provided by [19], where λ is set to 0.1, and the dictionary is of size 400 for SIFT feature points, and 50 for TextonBoost based feature points. The parameter C in our Max-Margin formulation is set to 1000. The scale γ of the hamming loss is set to 1000. For gradient descent, we use an initial step size τ_0 =1e-6. We run 100 iterations for Graz02, and 600 iterations for PASCAL VOC and MSRC21. For PASCAL VOC and MSRC21, we use the validation data to train our parameters, while the unary potentials from [14] are computed based on training data. For Graz02, both unary potentials and model parameters are computed based on training data.

4.1 Graz-02 Dataset

Results. Tables 1 and 2 show the VOC measure and per-class accuracy, respectively, on the Graz-02 dataset. Since we randomly sampled super-pixels to compute the unary potentials for this dataset, we run the experiment 5 times and calculate the mean and variance of the result (reported in parenthesis). In the tables, U+P refers to the basic CRF model described by Eqn. (2), and TP1 and TP2 refer to the first two methods for extracting top-down feature points. Notice that the U+P result is computed by our implementation, while results from [26,10] are taken from the original paper. To show that these results are comparable, we observe that in [10], their U+P implementation gives an average of 50.82% in VOC measure metric, and 80.36% in average per-class accuracy, which means our method and [10] are built based on comparable baselines.

Table 1. VOC measure on Graz-02 Dataset

	U+P	[26]	[10]	Ours-TP1	Ours-TP2
BG	79.4 (0.8)	82.3	78.0	86.4 (0.1)	87.2 (0.1)
Bike	44.3 (0.3)	46.2	55.6	52.8 (0.1)	52.5 (0.1)
Car	40.6 (1.5)	36.5	41.5	44.1 (0.3)	48.4 (0.6)
Human	37.9 (1.2)	39.0	37.3	41.2 (0.8)	44.1 (0.6)
Mean	50.6 (0.3)	51.0	53.1	56.1 (0.1)	58.0 (0.1)

Table 2. Accuracy on Graz-02 Dataset

	U+P	[26]	[10]	Ours-TP1	Ours-TP2
BG	81.6 (0.3)	86.4	75.9	90.6 (0.1)	91.2 (0.1)
Bike	85.9 (0.1)	73.0	84.9	77.8 (1.7)	76.3 (0.5)
Car	78.9 (0.8)	68.7	76.7	66.3 (6.6)	68.2 (1.6)
Human	80.0 (1.4)	71.3	79.8	66.7 (5.5)	70.0 (1.2)
Mean	81.6 (0.1)	74.9	79.3	75.4 (0.6)	76.4 (0.1)
Global	81.72 (0.2)	N/A	N/A	87.6 (0.1)	88.1 (0.1)

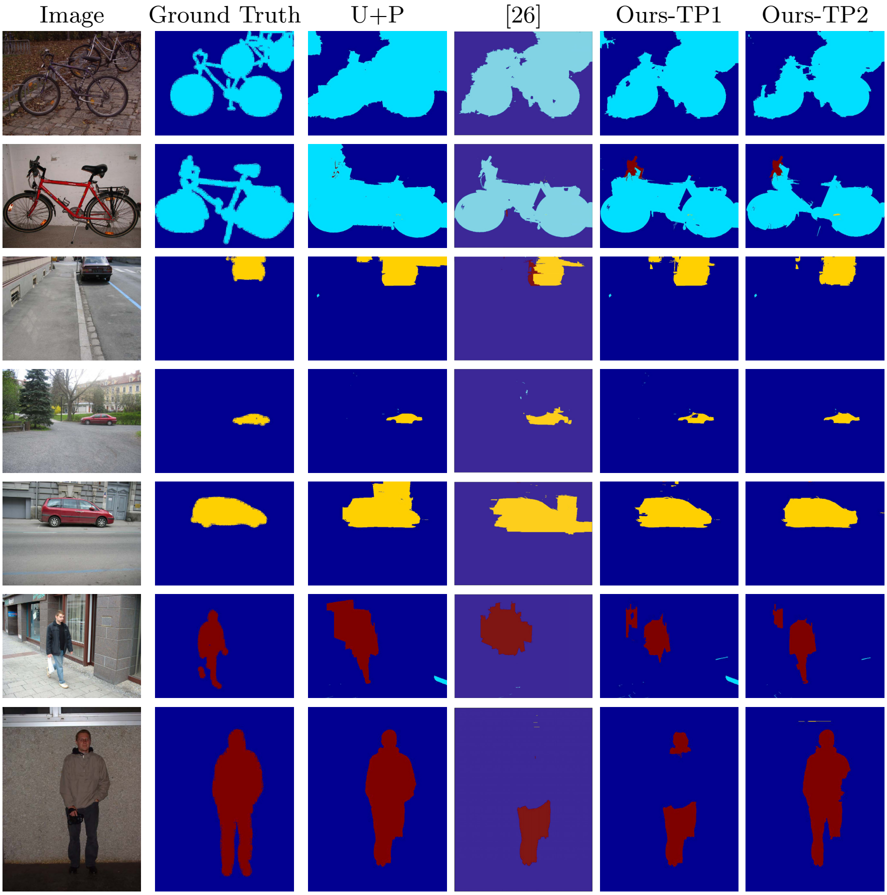

Fig. 1. Example segmentation results for the Graz02 dataset using different methods. The background, bikes, cars and humans are color coded as blue, cyan, yellow and red respectively.

Discussion. From Table 1 we can see that our method outperforms both our baseline U+P and other state-of-art methods (except for the bike category). However, the per-class accuracy in Table 2 is not improved except for the Background category. This is understandable since our goal is to reduce the false negative rate as well as the false positive rate, while the accuracy metric focuses on the true positive rate exclusively. Note that for the Car and Human category, the VOC measure is improved by around 7% while the accuracy decreases by around 10%. This implies that a lot of false positives are removed, i.e. less background pixels are labeled as object. That is also why we observe improvement in both the accuracy and VOC measures for the Background class. Notice also that the performance for the Bike class decreases for our method. Our conjecture is that in the annotations of Graz02 the pixels inside the wheel are labeled as bike, while most of them are background except for the spokes. This leads to decreased performance, since some of the pixels inside the wheel are classified as background. We would expect better results with more detailed annotations.

We show some qualitative results in Fig. 1. As we can see, although more foreground object pixels are labeled as background, the segmentation is more accurate at the boundaries and fewer superpixels from the background are labeled as other class. For example, for the Bike category, our method can remove false positives in the triangle area (row 2 in Fig. 1).

To further understand the effect of jointly learning the dictionary and the CRF parameters, we run experiments where only the weights W is updated, while the dictionary D is fixed. In this case, we achieve an average VOC measure of 50.1% for TP1 and 51.0% for TP2, which seems to suggest that for this dataset, updating the dictionary leads to majority of the improvement.

4.2 PASCAL VOC2010

Results. Fig. 2 shows the per-class VOC measure obtained by the baseline method (U+P) and our proposed method on the PASCAL VOC2010 dataset using the third feature extraction method (TP3) to construct the top-down categorization cost. In addition, Table 3 shows the average VOC measure obtained by both methods together with the results of [25,14,33] for comparison. The grid-CRF method refers to the one used by [25], while its performance is reported in [14]. Notice that the dense-CRF in [14] models each pixel as a node of the graph, and the work in [33] uses also detection scores. On the other hand, our method adopts a super-pixel based CRF instead of a dense pixel based CRF and does not use any detection information directly. Therefore, it is more fair to compare our results with those of the grid-CRF method in [25].

Discussion. As expected, our U+P baseline performs as good as the grid-CRF model, since they have similar graph size. Our method with jointly updating dictionary and CRF leads to a 1.4% improvement in VOC measure and the performance is comparable with more complex methods [14,33]. As we can see in Fig. 2, for most of the object classes, we obtain an improvement of up to 5%.

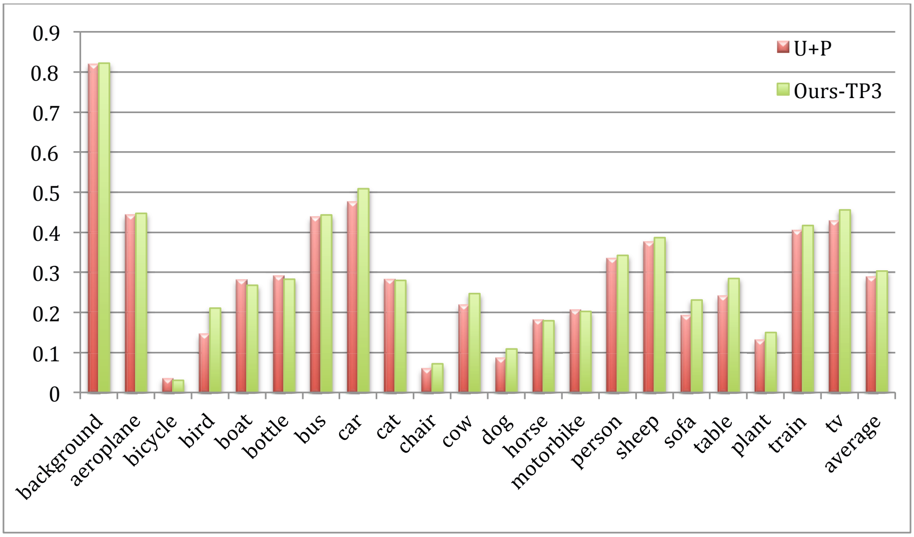

Fig. 2. VOC measure on VOC 2010 dataset using baseline U+P and our method

Table 3. Results on VOC2010 Dataset

	grid-CRF [25]	dense-CRF [14]	[33]	U+P	Ours-TP3
VOC measure	28.3	30.2	31.2	28.9	30.3

4.3 MSRC21

Results. Table 4 gives the mean and global accuracy obtained by the baseline method (U+P) and our proposed method on the MSRC21 dataset using the TextonBoost based unary potential and top-down terms, as for the PASCAL VOC2010 dataset. The results of [24,15,14,33] are also reported for comparison, while the performance of [15] on MSRC21 dataset is reported in [33]. We also show some qualitative results in Fig. 3.

Discussion. The global accuracy given by our algorithm is slightly worse than that of other methods. However, the mean accuracy is on par with the performance of the dense-CRF model [14], and is only 1% less than the performance of [33]. As explained before, [33] combines both scene information and object information, while our method only uses TextonBoost feature. This suggests that our algorithm gives comparable result while using simpler models. Finally, while our results are just marginally better than those of the U+P baseline, when looking at the example segmentations in in Fig. 3 we observe that our methods gives qualitatively better segmentations.

Table 4. Results on MSRC21 Dataset

	Shotton et al. [24]	HCRF + Coocc. [15]	Dense CRF [14]	Yao et al.[33]	U+P	Ours-TP3
Mean Accuracy	67	77.8	78.3	79.3	77.7	78.4
Global Accuracy	72	86.5	86.0	86.2	84.3	84.5

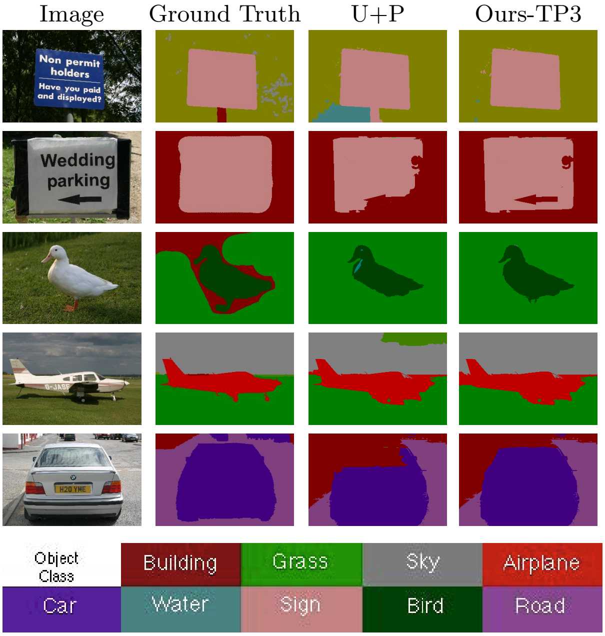

Fig. 3. Example segmentation results for the MSRC21 dataset using the U+P baseline and our proposed method

5 Conclusion

In this paper, we presented a new semantic segmentation framework that incorporates a top-down object categorization cost based on a discriminative sparse representation of each object. We proposed an optimization framework to jointly learn the sparse dictionary and the CRF parameters, so that the dictionary is specifically trained for the segmentation task. Experimental results showed that our algorithm outperforms the basic CRF model and the top-down model with BoF representation, suggesting that a jointly learned dictionary can help to improve segmentation performance compared with a pre-learned BoF dictionary.

Acknowledgements. We thank Florent Couzinié-Devy for interesting discussions about the gradient computation. The first and last author were supported in part by grants NSF 1218709, ONR N000141310116 and ERC VideoWorld. Part of the work was conducted when the first two authors were at Mitsubishi Electric Research Laboratories (MERL). This part was funded by MERL only.

References

1. Bach, F., Mairal, J., Ponce, J.: Task-driven dictionary learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(4), 791–804 (2012)
2. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
3. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. Journal of Computer Vision 88(2), 303–338 (2010)
6. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005 (2005)
7. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: IEEE Int. Conf. on Computer Vision (2009)
8. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: IEEE Conf. on Computer Vision and Pattern Recognition (2008)
9. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. International Journal of Computer Vision 80(3), 300–316 (2008)
10. Jain, A., Zappella, L., McClure, P., Vidal, R.: Visual dictionary learning for joint object categorization and segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 718–731. Springer, Heidelberg (2012)
11. Joachims, T., Finley, T., Yu, C.N.J.: Cutting-plane training of structural SVMs. Machine Learning 77(1), 27–59 (2009)
12. Kohli, P., Ladicky, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: IEEE Conf. on Computer Vision and Pattern Recognition (2008)
13. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. on Pattern Analysis and Machine Intelligence 26(2), 147–159 (2004)
14. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: Neural Information Processing Systems, pp. 109–117 (2011)
15. Ladický, Ľ., Sturgess, P., Alahari, K., Russell, C., Torr, P.H.S.: What, where and how many? Combining object detectors and cRFs. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 424–437. Springer, Heidelberg (2010)
16. Ladicky, L., Russell, C., Kohli, P., Torr, P.: Associative hierarchical CRFs for object class image segmentation. In: IEEE Int. Conf. on Computer Vision (2009)
17. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML (2001)
18. Laptev, I.: On space-time interest points. International Journal of Computer Vision 64(2-3), 107–123 (2005)

19. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Neural Information Processing Systems, pp. 801–808 (2007)
20. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. IEEE Conference on Computer Vision and Pattern Recognition (2008)
21. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: Proceedings of Uncertainty in AI, pp. 467–475 (1999)
22. Naikal, N., Singaraju, D., Sastry, S.S.: Using models of objects with deformable parts for joint categorization and segmentation of objects. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part II. LNCS, vol. 7725, pp. 79–93. Springer, Heidelberg (2013)
23. Opelt, A., Pinz, A.: The TU Graz-02 database (2002), `http://www.emt.tugraz.at/~pinz/data/GRAZ02/`
24. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: IEEE Conf. on Computer Vision and Pattern Recognition (2008)
25. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int. Journal of Computer Vision 81(1), 2–23 (2009)
26. Singaraju, D., Vidal, R.: Using global bag of features models in random fields for joint categorization and segmentation of objects. In: IEEE Conference on Computer Vision and Pattern Recognition (2011)
27. Tighe, J., Lazebnik, S.: Finding things: Image parsing with regions and perexemplar detectors. In: IEEE Conf. on Computer Vision and Pattern Recognition (2013)
28. Vedaldi, A.: A MATLAB wrapper of SVM^{s}truct (2011), `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.html`
29. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), `http://www.vlfeat.org/`
30. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 705–718. Springer, Heidelberg (2008)
31. Yang, J., Yang, M.: Top-down visual saliency via joint crf and dictionary learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
32. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
33. Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: IEEE Conf. on Computer Vision and Pattern Recognition (2012)
34. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 1169–1176. ACM, New York (2009)
35. Zhang, K., Zhang, W., Zheng, Y., Xue, X.: Sparse reconstruction for weakly supervised semantic segmentation. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 1889–1895 (2013)