Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Sparse Reconstruction for Weakly Supervised Semantic Segmentation

Ke Zhang, Wei Zhang, Yingbin Zheng, Xiangyang Xue
School of Computer Science, Fudan University, China
{k_zhang,weizh,ybzh,xyxue}@fudan.edu.cn

Abstract

We propose a novel approach to semantic segmentation using weakly supervised labels. In traditional fully supervised methods, superpixel labels are available for training; however, it is not easy to obtain enough labeled superpixels to learn a satisfying model for semantic segmentation. By contrast, only image-level labels are necessary in weakly supervised methods, which makes them more practical in real applications. In this paper we develop a new way of evaluating classification models for semantic segmentation given weekly supervised labels. For a certain category, provided the classification model parameter, we firstly learn the basis superpixels by sparse reconstruction, and then evaluate the parameters by measuring the reconstruction errors among negative and positive superpixels. Based on Gaussian Mixture Models, we use Iterative Merging Update (IMU) algorithm to obtain the best parameters for the classification models. Experimental results on two real-world datasets show that the proposed approach outperforms the existing weakly supervised methods, and it also competes with state-of-the-art fully supervised methods.

1 Introduction

Semantic segmentation is an interesting and challenging problem in computer vision, which aims to identify the semantic label of pixels of images, i.e., assign each pixel in an image to one of pre-defined semantic categories. Semantic segmentation is usually considered as a supervised learning problem in contrast to low-level unsupervised segmentation which groups pixels into homogeneous regions based on visual features [Lu *et al.*, 2011; Munoz *et al.*, 2012].

In the past years there have been many attentions in the semantic segmentation task [Kohli *et al.*, 2009; Ladicky *et al.*, 2009; Shotton *et al.*, 2006; 2008; Yang *et al.*, 2007; Jain *et al.*, 2012; Lucchi *et al.*, 2012; Ladicky *et al.*, 2010]. Although these methods have shown promising results, most of them rely on a training set of annotated images, where the label for each pixel is known. However, in real-world applications, reliable pixel-level labeled images is rare, and manually labeling pixels is time-consuming and labor-intensive, so fully supervised methods cannot be widely applied in practice.

Recently, a few works have been proposed to address the *weakly supervised* semantic segmentation problem, where only the image-level annotations are available for training [Verbeek and Triggs, 2007; Vezhnevets and Buhmann, 2010; Vezhnevets *et al.*, 2011; 2012]. In general, the task of semantic segmentation includes several steps as follows: (1) Oversegment images into superpixles and extract visual features for each superpixel; (2) Train classifiers with the features of superpixles in training set; (3) Apply classifiers on testing images to get the results of semantic segmentation. Since there is no groundtruth label of superpixels in the training set, most existing weakly supervised methods for methods aim to maximize the likelihood between pixel label prediction and image-level label, or to maximize various kinds of potentials.

In this paper, we come up with a novel approach to weakly supervised semantic segmentation based on sparse subspace reconstruction and classification model evaluation. The overview of the proposed approach is shown in Fig.1. Firstly, Images are segmented into superpixels. For each category, we classify the positive and negative superpixels by a classification model with respect to this category, then take the positive superpixels as input, and learn the bases of the subspace for the corresponding category. Secondly, evaluation is conducted by comparing the difference between the reconstruction error of negative and positive superpixels. Thirdly, with a few random samplings of model parameters, we can efficiently identify the optimal parameter for classification model based on GMM (Gaussian Mixture Models), and then we can easily classify the superpixels of testing images and group adjacent the superpixels sharing the same label into one homogeneous region. The main contributions of this paper include:

- We develop a new way of evaluating classification models for semantic segmentation given weekly supervised labels. Based on the chosen classification model parameters, we get an approximate groundtruth for unavailable superpixel label and learn the positive basis superpixels which perform sparse reconstruction of category-specific subspace. By measuring the reconstruction errors among negative and positive superpixels, we provide a powerful criterion for evaluating the effectiveness

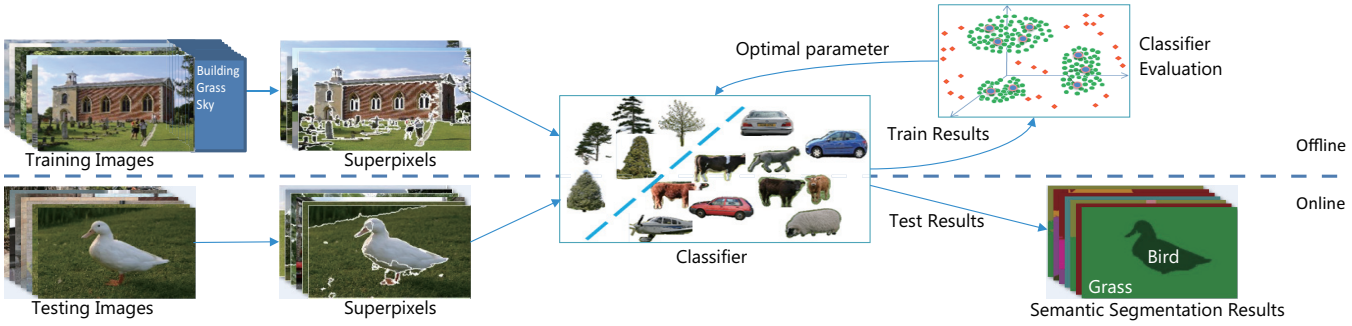

Figure 1: Overview of our framework. Images are firstly oversegmented into superpixels. For each category, we select a model to classify superpixels in training images. Based on the reconstruction error, we evaluate the basis superpixels and obtain the optimal classification model parameter. The optimal parameter is then used for classifying superpixels in test images.

of classification models. Experimental results on two challenging datasets show that our method outperforms previous weakly supervised methods, and it even competes with some fully supervised methods.

- Based on Gaussian Mixture Models, we also develop an Iterative Merging Update (IMU) algorithm to obtain the best parameters for the classification models. This algorithm aims to identify the model with best evaluation but minimal number of samplings, which is also known as exploration vs. exploitation trade-off.

The remainder of this paper is organized as follows: We review the related work in Section 2. We then elaborate the proposed method for sparse reconstruction weight learning, basis superpixels learning, and optimal model parameters estimating by Iterative Merging Update (IMU) algorithm in Section 3. Our experimental results on two real datasets are given in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

In [Shotton *et al.*, 2006], semantic segmentation was performed by defining a conditional random field (CRF) over image pixels with unary potentials which could be learnt by a boosted decision tree classifier over texture-layout filters. The successive research along this direction focused on improving the CRF structure, such as enabling inference for combining multiple segmentations [Jain *et al.*, 2012; Gonfaus *et al.*, 2010; Kohli *et al.*, 2009; Munoz *et al.*, 2010], integrating label co-occurrence statistics [Ladicky *et al.*, 2010], and introducing hierarchy with higher order potentials [Ladicky *et al.*, 2009]. [Vezhnevets *et al.*, 2012] introduced pairwise potentials among multi-feature images as components of CRF appearance model. Another direction was to develop faster and more accurate features. [Shotton *et al.*, 2008] obtained fast and powerful features via random decision forests that convert features to similar semantic texton histograms by combining node counts and category prior of each tree.

In weakly supervised semantic segmentation settings , no groundtruth is available for evaluating pixel-level classification. Most existing works make use of the global consistency between image label and superpixel label, together with local consistency between various kinds of potentials. Predictions of pixel-category are conducted by maximizing either or both consistencies. Towards less supervision, [Duygulu *et al.*, 2002] posed the problem as machine translation while [Verbeek and Triggs, 2007] solved this problem with aspect models and the spatial extensions. In [Vezhnevets and Buhmann, 2010], a multi-instance multi-task learning modification of semantic texton forest (STF) [Shotton *et al.*, 2008] was proposed. In [Vezhnevets *et al.*, 2011], multi-image model(MIM) was propsed to integrate observed training image features and labels together with latent superpixel labels into one network. [Vezhnevets *et al.*, 2012] defined a parametric family of structured models, and designed a Maximum Expected Agreement model selection principle to evaluate the model parameters.

Our method performs weakly supervised semantic segmentation by subspace reconstruction and model evaluation. As for subspace reconstruction, various algorithms have been implemented with different regularizers [Elhamifar and Vidal, 2009; Liu *et al.*, 2010; Wang *et al.*, 2011]. As for model evaluation, the criterion for evaluating model parameters introduced in [Vezhnevets *et al.*, 2012] can be easily affected by outliers. In contrast, our method goes from a different direction. No constraint is given on classification models employed. Based on the chosen classification model parameters, we get an approximate groundtruth for unavailable superpixel label and learn the positive basis superpixels which perform sparse reconstruction of category-specific subspace. Our method gives more promising results than the previous methods which use probabilistic inference merely with global and local consistency.

3 The Proposed Approach

Suppose that there are weakly labeled images and each image is oversegmented into several superpixels. For each image, the image-level label is provided while superpixel-level label is not available. For a specific category, if one image is labeled positive then there is at least one superpixel with the concerned label; otherwise, all superpixels included in the image are negative. Our goal is to determine the se-

mantic label for each superpixel, and then adjacent superpixels sharing the same label are fused as the whole one. Let $\mathcal{C} = \{c_1, \ldots, c_M\}$ be the semantic lexicon of M categories. We try to get a classification model for each category, and then can easily determine whether one superpixel belongs to this category or not. Suppose that classification models for the labels $c_m, m \in (1, \ldots, M)$ are parameterized by $\theta_m, m \in (1, \ldots, M)$, respectively. The task is to estimate the optimal parameters such that the classification models fit the weakly labeled images and perform well in prediction labels of superpixels. Since the labels of superpixels are unknown, the parameters of the classification models can not be learned in a straight way. Instead, we can firstly estimate the parameters at random, and then evaluate the random $guess$ by investigating whether the corresponding classifier performs well on the given data.

For a certain category c_m, all superpixels included in negative images are negative while the superpixels included in positive images can be either positive or negative. Once its classifier parameters θ_m are provided, the set of superpixels included in positive images can be divided into two subsets: positive or negative. Let $X = [\mathbf{x}_1, ..., \mathbf{x}_N]$ denote the set of all positive superpixels where each column is the visual feature of one superpixel. It should be pointed out that the above classification results might be incorrect because the parameters θ_m are provided at random.

3.1 Sparse Reconstruction Weight Learning

We extract low-level visual features for each superpixel, and the feature vector is often high-dimensional. However, high-dimensional data usually lie in a low-dimensional subspace with respect to some category the data belongs to, and one data point can be approximately reconstructed by some other points from the same subspace. For a certain category c_m, since $X = [\mathbf{x}_1, ..., \mathbf{x}_N]$ is the set of all positive superpixels, each superpixel can be reconstructed by the others in X, and intuitively, those more similar samples will contribute more in reconstructing it. The number of reconstructing points is much less than the number of all positive candidates in X, so the reconstruction is sparse. Denote $W \in \mathbb{R}^{N \times N}$ as the reconstruction matrix, which can be obtained by minimizing the cost function as follows:

$$\begin{aligned} &\min_{W} f(W) = (\|X - XW\|_F)^2 + \alpha(\|W\|_1)^2 \\ &s.t.\ \mathbf{1}^\top W = \mathbf{1}^\top,\ W_{ij} \in [0,1],\ (i,j = 1, \ldots, N) \end{aligned} \tag{1}$$

where $\mathbf{1}$ denotes an all-one vector, and α is the trade-off parameter. In the first term, $\|X - XW\|_F$ is the reconstruction error expressed in the Frobenius matrix norm. Denote the $j-$th column of X as $\mathbf{col}(X, j)$ which corresponds to the $j-$th superpixel. The $j-$th column of the product XW equals to the matrix X times the $j-$th column of W: $\mathbf{col}(XW, j) = X\mathbf{col}(W, j) = \sum_{i=1}^{N} W_{ij}\mathbf{col}(X, i)$; therefore, $(\|XW - X\|_F)^2 = \sum_{j=1}^{N} \| \sum_{i=1}^{N} W_{ij}\mathbf{col}(X, i) - \mathbf{col}(X, j)\|^2$. By constraining that $W_{j,j} = 0 (j = 1, \ldots, N)$, each superpixel can be estimated as a linear combination of other superpixels except itself, which also avoids the case that the optimal W collapses to the identity matrix. As for the second term, due to $W_{ij} \in [0, 1]$, minimizing the ℓ_1-$norm$ $\|W\|_1$ encourages $W_{i,j}$ to be zero if the reconstructing weight is too small such that W is sparse.

The cost function in Eq.(1) is convex and can be solved with the Augmented Lagrange Multiplier (ALM) method[Lin *et al.*, 2009]. However, W is an $N \times N$ matrix, and the computational cost is expensive for ALM to optimize the problem. As mentioned before, W is encouraged to be sparse, so we come up with a novel approach to solution by converting the original optimization of Eq.(1) with the complexity of $O(N^2)$ into N sub-problems each of which operates on a single column of W with the complexity of $O(N)$. Since these sub-problems are independent of each other, parallel computation can be employed to accelerate the optimization process. We firstly re-write Eq.(1) as follow:

$$\begin{aligned} f(W) = &\sum_{j=1}^{N} \mathbf{x}_j^\top \mathbf{x}_j - \sum_{j=1}^{N} 2\mathbf{x}_j^\top \sum_{i=1}^{N} \mathbf{x}_i W_{ij} \\ &+ \sum_{j=1}^{N}\sum_{d=1}^{D}(\sum_{i=1}^{N} \mathbf{x}_i(d) W_{ij})^2 + \alpha \left(\sum_{i,j=1}^{N} W_{ij} \right)^2 \end{aligned} \tag{2}$$

where $\mathbf{x}_i(d)$ denotes the $d-$th element of $\mathbf{x}_i$. According to Cauchy-Schwarz Inequality $(\sum_{i=1}^{n} a_i b_i)^2 \leq (\sum_{i=1}^{n} a_i^2)(\sum_{i=1}^{n} b_i^2)$, we get:

$$\begin{aligned} (\sum_{i=1}^{N} \mathbf{x}_i(d) W_{ij})^2 &\leq \sum_{i=1}^{N} \frac{(\mathbf{x}_i(d) W_{ij})^2}{T_{ijd}} \\ \left(\sum_{i,j=1}^{N} W_{ij} \right)^2 &\leq \sum_{i=1,j=1}^{N} \frac{(W_{ij})^2}{Q_{ij}} \end{aligned} \tag{3}$$

where $T_{ijd} \in (0, 1), Q_{ij} \in (0, 1)$ and $\sum_{i=1}^{N} T_{ijd} = 1$, $\sum_{i,j=1}^{N} Q_{ij} = 1$. Thus, we obtain the upper bound of our cost function as follows:

$$\begin{aligned} f(W) \leq \sum_{j=1}^{N} \Bigg\{ &\mathbf{x}_j^\top \mathbf{x}_j + \sum_{i=1}^{N} \Bigg\{ -2(\mathbf{x}_i^\top \mathbf{x}_j) W_{ij} \\ &+ \left(\sum_{d=1}^{D} \frac{((\mathbf{x}_i(d))^2}{T_{ijd}} + \frac{\alpha}{Q_{ij}} \right) (W_{ij})^2 \Bigg\} \Bigg\} \end{aligned} \tag{4}$$

The equalities in Eq.(3) and Eq.(4) hold if and only if

$$\begin{aligned} T_{ijd} &= \frac{(\mathbf{x}_i(d) W_{ij})^2}{\sum_{i=1}^{N} (\mathbf{x}_i(d) W_{ij})^2}; \\ Q_{ij} &= \frac{(W_{ij})^2}{\sum_{i,j=1}^{N} (W_{ij})^2}; \end{aligned} \tag{5}$$

Therefore, under the condition of Eq.(5), the original optimization problem is equivalent to minimizing the right side of Eq.(4), which can be furthermore divided into N independent QP (Quadratic Programming) sub-problems:

$$\begin{aligned} &\min_{W_{\cdot j}} \frac{1}{2} W_{\cdot j}^\top \Lambda_j W_{\cdot j} + B_j^\top W_{\cdot j} \\ &s.t.\ W_{\cdot j} \succeq 0,\ \mathbf{1}^\top W_{\cdot j} = 1; \end{aligned} \tag{6}$$

where $W_{\cdot j}$ denotes the $j-$th column of W whose element is non-negative. $\Lambda_j \in R^{N\times N}$ is a diagonal matrix with the $i-$th element on the diagonal equal to $2\big(\sum_{d=1}^{D}\frac{(\mathbf{x}_i(d))^2}{T_{ijd}}+\frac{\alpha}{Q_{ij}}\big)$. $B_j \in R^{N\times 1}$ is a vector with the $i-$th element equal to $-2\mathbf{x}_i^\top\mathbf{x}_j$. Such quadratic programming problem can be easily solved via the existing software solvers such as CVXOPT[1], MOSEK[2], and TOMLAB[3]. By iteratively solving the optimization problem in a flip-flop manner, i.e., updating T_{ijd}, Q_{ij} with Eq.(5) and updating W_{ij} with Eq.(6) alternatively until convergence, we obtain the optimal reconstruction weight matrix W.

3.2 Learning Basis Superpixels

Once the sparse reconstruction weight matrix W has been learned, we select the basis superpixels spanning the subspace corresponding to the current category c_m. Actually, the weight matrix W can be viewed from two perspectives: i)$column\ view$: Each column of W corresponds to one reconstructed superpixel; ii)$row\ view$: Each row of W illustrates the contributions of one superpixel to reconstructing others in the category c_m. Based on the $column\ view$, we learn the reconstruction weight matrix in above subsection. Now, we focus on the $row\ view$ to choose the basis superpixels.

Note that the weight matrix W is sparse, and from the $row\ view$ of W, the larger weights imply that the corresponding superpixel play an important role in reconstructing others and in spanning the subspace. Therefore, we calculate the sum of each row $Sum(i)=\sum_{j=1}^{N}W_{ij}$ and select the superpixels with the largest $Sum(i)$'s as the bases. The pseudo-code for learning basis superpixels is shown in Algorithm 1.

Algorithm 1: Learning Basis Superpixels

Input:
$X=[\mathbf{x}_1,...,\mathbf{x}_N]$;
r: the Number of Basis Superpixels;
Output:
$[\widetilde{\mathbf{x}}_1,...,\widetilde{\mathbf{x}}_r]$: Basis Superpixels Spanning the Subspace;
begin
  Initialization: $W\leftarrow W_0$;
  while *not convergence* **do**
    1.Update T, Q with Eq.(5);
    2.Solve QP Problem in Eq.(6);
  $Sum(i)=\sum_{j=1}^{N}W_{ij}$, (i=1,...,N);
  Select r Superpixels with Large $Sum(i)$'s as Bases;

For a certain category c_m, its classification model is parameterized by θ_m, and the learned basis superpixels are $\widetilde{X}=[\widetilde{\mathbf{x}}_1,...,\widetilde{\mathbf{x}}_r]$. If the classification model parameters are appropriately estimated and the learned basis superpixels are satisfying, the distance between negative superpixels and the subspace spanned by $\widetilde{X}=[\widetilde{\mathbf{x}}_1,...,\widetilde{\mathbf{x}}_r]$ should be large. Let $\mathbb{X}_{neg}$ denote all superpixels included in negative images, and each column of $\mathbb{X}_{neg}$ corresponds to one superpixel which is definitely negative. We define the distance between negative superpixels and the subspace spanned by $\widetilde{X}$ as the minimal negative-by-positive reconstruction error $min_{\mathbb{W}_{neg}}\|\mathbb{X}_{neg}-\widetilde{X}\mathbb{W}_{neg}\|_F$, where $\mathbb{W}_{neg}$ is the reconstruction weight matrix of negative superpixel by the learned bases. Since the learned bases $\widetilde{X}$ depend on θ_m, the above reconstruction error depends on θ_m as well, and the more appropriate the parameter θ_m is, the larger the negative-by-positive reconstruction error becomes. Therefore, it can be looked on as one criterion to evaluate the estimation of θ_m, denoted as $Score(\theta_m)=min_{\mathbb{W}_{neg}}\|\mathbb{X}_{neg}-\widetilde{X}\mathbb{W}_{neg}\|_F$.

Algorithm 2: Iterative Merging Update

Input:
Sampling pairs θ_m and $Score(\theta_m)$ as $<\Theta,\Psi>$:
$\Theta=\{\theta_m^1,\theta_m^2,...,\theta_m^K\}$,
$\Psi=\{Score(\theta_m^1),Score(\theta_m^2),...,Score(\theta_m^K)\}$
Output: $\theta_m^*=\mathrm{argmax}_{\theta_m}\,Score(\theta_m)$
begin
  $\theta_m^*=\mathrm{argmax}\,\{Score(\theta_m^1),...,Score(\theta_m^K)\}$
  Fitting sampling data with Conditional GMM:
  $f_{\Psi|\Theta}(y|x)=\sum_{j=1}^{C}w_j(x)N(y|m_j(x),\sigma_j^2)$;
  for $(i=1;i<K;i++)$ **do**
    1.Use KL-divergence to compute the difference between two components of GMM;
    2.Find pair (j_1,j_2) with minimal KL-divergence;
    3.Merging pair (j_1,j_2) with the following rules:

$$\begin{aligned} w&=w_{j_1}+w_{j_2},\\ m&=\frac{w_{j_1}}{w}m_{j_1}+\frac{w_{j_2}}{w}m_{j_2},\\ \sigma&=\frac{w_{j_1}}{w}\sigma_{j_1}+\frac{w_{j_2}}{w}\sigma_{j_2}+\\ &\quad\frac{w_{j_1}w_{j_2}}{w^2}(m_{j_1}-m_{j_2})(m_{j_1}-m_{j_2})^\top; \end{aligned}$$

    4. $\hat{\theta}=\arg_x m(x)$, where $m(x)$ is the centroid of new component after merging pair (j_1,j_2);
    5.Update $\theta_m^*=\underset{\theta}{\mathrm{argmax}}(Score(\theta_m^*),Score(\hat{\theta}))$

3.3 Optimal Model Parameters Estimating

Now we provide a powerful criterion $Score(\theta_m)$ for evaluating the effectiveness of classification models. The larger $Score(\theta_m)$ is, the better classification model we get. How can we find the θ_m that maximizes $Score(\theta_m)$? It is hard to compute the gradient of $Score(\theta_m)$ with respect to θ_m, thus direct solution of maximizing $Score(\theta_m)$ is intractable, and enumeration of the entire solution space is also time-consuming. Here, inspired by [Sung, 2004], we use an Iterative Merging Update (IMU) algorithm based on Gaussian Mixture Model (GMM) which fits the distribution of $Score(\theta_m)$.

[1]CVXOPT: http://abel.ee.ucla.edu/cvxopt/index.html
[2]MOSEK: http://www.mosek.com
[3]TOMLAB: http://www.tomlab.com

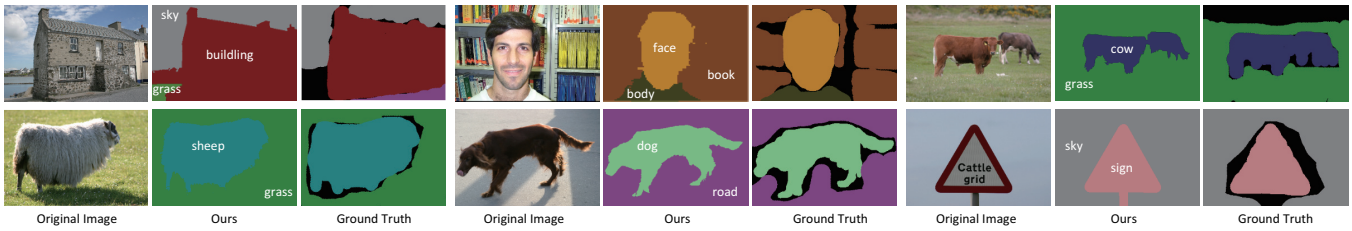

Figure 2: Some example results for semantic segmentation by our method in comparison with the ground-truth on MSRC.

In the parameter space, we randomly sample K different θ_m's and calculate the corresponding $Score(\theta_m)$. Thus, we get K sample pairs $< \theta_m^k, Score(\theta_m^k) >, (k = 1, .., K)$. Let the set of θ_m^k is denoted as Θ, and the set of $Score(\theta_m^k)$ as Ψ. Each element in Θ and Ψ can be considered as random variable, and the joint probability density can be defined by GMM as follows:

$$\begin{aligned} f_{\Theta,\Psi}(x,y) &= \sum_{j=1}^{C} \pi_j N(x,y|\mu_j,\Sigma_j) \\ \sum_{j=1}^{C} \pi_j = 1,\ \mu_j &= \begin{bmatrix} \mu_{j\Theta} \\ \mu_{j\Psi} \end{bmatrix},\ \Sigma_j = \begin{bmatrix} \Sigma_{j\Theta\Theta} & \Sigma_{j\Theta\Psi} \\ \Sigma_{j\Psi\Theta} & \Sigma_{j\Psi\Psi} \end{bmatrix} \end{aligned} \tag{7}$$

where C is the number of GMM components and π_j is the weight of the j-th component. μ_j is the joint mean vector and Σ_j is the joint co-variance matrix for the j-th GMM component. We obtain the conditional PDF (Probability Density Function) of $\Psi|\Theta$ as

$$f_{\Psi|\Theta}(y|x) = \sum_{j=1}^{C} w_j(x) N(y|m_j(x), \sigma_j^2) \tag{8}$$

where

$$\begin{aligned} w_j(x) &= \frac{\pi_j N(x|\mu_{j\Theta}, \Sigma_{j\Theta\Theta})}{\sum_{j=1}^{C} \pi_j N(x|\mu_{j\Theta}, \Sigma_{j\Theta\Theta})} \\ m_j(x) &= \mu_{j\Psi} + \Sigma_{j\Psi\Theta}\Sigma_{j\Theta\Theta}^{-1}(x - \mu_{j\Theta}) \\ \sigma_j &= \Sigma_{j\Psi\Psi} - \Sigma_{j\Psi\Theta}\Sigma_{j\Theta\Theta}^{-1}\Sigma_{j\Theta\Psi} \end{aligned} \tag{9}$$

In Iterative Merging Update (IMU) algorithm, we initialize the GMM with K components. In each iteration, we calculate KL-divergences between components and update the GMM by merging the most similar pair of components. At the centroid of new component after merging pair, the conditional probability density of $Score(\theta_m)$ achieves the local peak and thus we add the corresponding θ_m to the set of parameter candidates. The detail of IMU Algorithm is shown in Algorithm 2. IMU algorithm reduces the initial model to a finite Gaussian mixture of C components, where C ranges from K to 1.

4 Experiments

We conduct experiments on two real-world image datasets: MSRC [Shotton *et al.*, 2006], and VOC2007 [Everingham *et al.*, 2007].On both datasets, we use EDISON system[Comaniciu and Meer, 2002] for the low-level segmentation. Popular visual descriptors including texture (i.e., the output of filter banks), color statistics (histogram or moments) and SIFT [Lowe, 2004] are used for each superpixel. Max-margin classification models are employed in our experiments.

Methods	Train	Test
[Shotton *et al.*, 2008]	-	64.6
[Vezhnevets *et al.*, 2011]	83	67
Ours	**87**	**69**

Table 1: Accuracies (%) of semantic segmentation of our method on both training and test subset of MSRC dataset, in comparison with state-of-art weakly supervised methods. The results of the state-of-art were reported in the related literatures.

The MSRC image dataset contains 591 images of resolution 320×213 pixels, accompanied with a hand labeled object segmentation of 21 object categories[Shotton *et al.*, 2006]. Pixels on the boundaries of objects are usually labeled as background and not taken into consideration in these segmentations. For fair comparison, we use the same train/test splits as in [Shotton *et al.*, 2008]. Some example results for semantic segmentation by our method in comparison with the ground-truth are shown in Fig.2. Table 1 gives the average accuracies of semantic segmentation of our method on both training and test subset, compared with the other two well-known weakly supervised methods. The results of the state-of-the-art were reported in [Shotton *et al.*, 2008] and [Vezhnevets *et al.*, 2011]. As can be seen, the performance of our method is better than that of the others. Table 2 shows the results of our method for individual labels on MSRC dataset, in comparison with other competitive algorithms (including fully supervised and weakly supervised ones). Our method outperforms state-of-the-art weakly supervised methods on average, and it is even competitive with those fully supervised methods in many cases.

PASCAL VOC 2007 dataset [Everingham *et al.*, 2007] was used for the PASCAL Visual Object Category segmentation contest 2007. It contains 5011 training and 4952 testing images where only the bounding boxes of the objects present in the image are marked, and 20 object classes are given for the task of classification, detection, and segmentation. Rather on the 5011 annotated training images with bounding box indicating object location and rough boundary, we conduct ex-

	Methods	building	grass	tree	cow	sheep	sky	aeroplane	water	face	car	bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat	average
Fully Supervised	[Shotton *et al.*, 2006]	62	**98**	86	58	50	83	60	53	74	63	75	63	35	19	92	15	86	54	19	62	07	58
	[Yang *et al.*, 2007]	63	**98**	90	66	54	86	63	71	83	71	80	71	38	23	88	23	88	33	34	43	**32**	62
	[Shotton *et al.*, 2008]	49	88	79	**97**	**97**	78	82	54	**87**	74	72	74	36	24	93	51	78	75	35	**66**	18	67
	[Ladicky *et al.*, 2009]	**80**	**96**	86	74	87	**99**	74	**87**	86	87	82	**97**	**95**	30	86	31	**95**	51	**69**	**66**	09	75
	[Csurka and Perronnin, 2011]	75	93	78	70	79	88	66	63	75	76	81	74	44	25	75	24	79	54	55	43	18	64
	[Lucchi *et al.*, 2012]	59	90	**92**	82	83	94	**91**	80	85	**88**	**96**	89	73	**48**	**96**	**62**	81	**87**	33	44	30	**76**
Weakly Supervised	[Verbeek and Triggs, 2007]	45	64	71	75	74	86	81	47	1	73	55	**88**	6	6	63	18	80	27	26	55	8	50
	[Vezhnevets and Buhmann, 2010]	7	96	18	32	6	**99**	0	46	**97**	54	74	54	14	9	82	1	28	47	5	0	0	37
	[Vezhnevets *et al.*, 2011]	5	80	58	**81**	**97**	87	**99**	63	91	**86**	**98**	82	67	**46**	59	45	66	**64**	**45**	33	**54**	67
	Ours	**63**	93	**92**	62	75	78	79	**64**	95	79	93	62	**76**	32	**95**	**48**	**83**	63	38	**68**	15	**69**

Table 2: Accuracies (%)of our method for individual labels on MSRC dataset, in comparison with other algorithms (fully supervised or weakly supervised). The last column is the average accuracy over all labels.

	Methods	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	average
Fully Supervised	Brookes	6	0	0	0	0	9	5	10	1	2	11	0	6	6	29	2	2	0	11	1	6
	[Shotton *et al.*, 2008]	66	6	15	6	15	32	19	7	7	**13**	44	31	**44**	27	39	35	12	7	39	23	24
	[Ladicky *et al.*, 2009]	27	**33**	**44**	11	14	**36**	30	**31**	27	6	**50**	28	24	38	**52**	29	28	12	45	46	30
	[Csurka and Perronnin, 2011]	**73**	12	26	**21**	**20**	0	17	**31**	**34**	6	26	41	7	31	34	30	11	28	5	50	25
	TKK	19	21	5	16	3	1	**78**	1	3	1	23	**69**	**44**	**42**	0	**65**	**30**	**35**	**89**	**71**	**31**
Weakly Supervised	[Shotton *et al.*, 2008][1]	14	8	11	0	**17**	**46**	5	**13**	4	0	**30**	29	12	**18**	**40**	6	**17**	17	14	9	16
	Ours	**48**	**20**	**26**	**25**	3	7	**23**	**13**	**38**	**19**	15	**39**	**17**	**18**	25	**47**	9	**41**	**17**	**33**	**24**

Table 3: Accuracies (%) of our method for individual labels on VOC2007 dataset, in comparison with other algorithms (fully supervised or weakly supervised). The last column is the average accuracy over all labels.

periments on the segmentation set with the 'train-val' split including 422 training-validation images and 210 test images, which are well segmented and thus are suitable for evaluation of the segmentation task.

The experimental results of our method compared with other related works are given in Table 3. The last column of 3 shows that the average accuracy of our method is better than that of the other weakly supervised method [Shotton *et al.*, 2008] and is comparable to those fully supervised ones. Our method performs far better than the only segmentation entry (Brookes)[Everingham *et al.*, 2007]. Although our method uses much fewer training images than TKK[Everingham *et al.*, 2007] which is trained by 422 training-validation images as well as a large number of annotated images with semantic bounding boxes from 5011 training sample, our method still get a comparable result to state-of-the-art methods.

5 Conclusions

We perform semantic segmentation in a weakly supervised framework where only image-level labels are available. Sparse reconstruction is used to learn the basis superpixels spanning the subspace for a certain category. If the classification model parameters are appropriately estimated and the learned basis superpixels are satisfying, the distance between negative superpixels and the subspace should be large. We introduce one criterion to evaluate the estimation of classifier parameter. Based on Gaussian Mixture Models, we select the best parameters for the classifiers employed. The proposed method is a general framework for evaluating various classification models, and it is suitable for any parametric model.

[1]The result of weakly supervised STF is not provided in the paper [Shotton *et al.*, 2008], here we show the result obtained by running the code provided by the authors.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. We would also like to thank Mr. Bilei Zhu for the fruitful discussions. This work was supported in part by the Shanghai Leading Academic Discipline Project (No.B114), the STCSM's Programs (No. 12XD1400900), the National High Technology Research and Development Program of China (No.2011AA100701), and the 973 Program (No.2010CB327906).

References

[Comaniciu and Meer, 2002] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002.

[Csurka and Perronnin, 2011] G. Csurka and F. Perronnin. An efficient approach to semantic segmentation. *IJCV*, 95(2):198–212, 2011.

[Duygulu *et al.*, 2002] P. Duygulu, K. Barnard, J. De Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *ECCV*, 2002.

[Elhamifar and Vidal, 2009] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009.

[Everingham *et al.*, 2007] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007. In *http://www.pascalnetwork.org/challenges/VOC/voc2007*, 2007.

[Gonfaus *et al.*, 2010] J.M. Gonfaus, X. Boix, J. Van De Weijer, A.D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *CVPR*, 2010.

[Jain *et al.*, 2012] A. Jain, L. Zappella, P. McClure, and R. Vidal. Visual dictionary learning for joint object categorization and segmentation. *ECCV*, 2012.

[Kohli *et al.*, 2009] P. Kohli, L. Ladickỳ, and P.H.S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009.

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

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

[Lin *et al.*, 2009] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Technical report, UILU-ENG-09-2215,*, 2009.

[Liu *et al.*, 2010] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

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

[Lu *et al.*, 2011] Yao Lu, Wei Zhang, Hong Lu, and Xiangyang Xue. Salient object detection using concavity context. In *ICCV*, 2011.

[Lucchi *et al.*, 2012] A. Lucchi, Y. Li, K. Smith, and P. Fua. Structured image segmentation using kernelized features. *ECCV*, 2012.

[Munoz *et al.*, 2010] Daniel Munoz, J Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.

[Munoz *et al.*, 2012] Daniel Munoz, James Andrew Bagnell, and Martial Hebert. Co-inference for multi-modal scene analysis. In *ECCV*. 2012.

[Shotton *et al.*, 2006] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, 2006.

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

[Sung, 2004] H.G. Sung. *Gaussian mixture regression and classification*. PhD thesis, RICE UNIVERSITY, 2004.

[Verbeek and Triggs, 2007] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007.

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

[Vezhnevets *et al.*, 2011] A. Vezhnevets, V. Ferrari, and J.M. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *ICCV*, 2011.

[Vezhnevets *et al.*, 2012] A. Vezhnevets, V. Ferrari, and J.M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, 2012.

[Wang *et al.*, 2011] S. Wang, X. Yuan, T. Yao, S. Yan, and J. Shen. Efficient subspace segmentation via quadratic programming. *AAAI*, pages 519–524, 2011.

[Yang *et al.*, 2007] L. Yang, P. Meer, and D.J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *CVPR*, 2007.