Data-Driven Scene Understanding with Adaptively Retrieved Exemplars

Xionghao Liu, Wei Yang, Liang Lin, and Qing Wang
Sun Yat-sen University

Zhaoquan Cai
Huizhou University

Jianhuang Lai
Sun Yat-sen University

This novel, data-driven framework for semantic scene understanding works without pixelwise annotation or classifier training using a probabilistic Expectation-Maximization formulation. It performs better than state-of-the-art methods in both semantic segmentation and image annotation.

Significant progress has been made in solving the task of semantic image understanding.[1,2] However, most methods usually build upon supervised learning with fully annotated data that is expensive and sometimes limited in large-scale scenarios.[3,4] To reduce the overload of data annotating, researchers have proposed several weakly supervised methods that can be trained with only image-level labels indicating the classes presented in the images.[5] Recently, data-driven approaches, which tend to leverage knowledge from auxiliary data in a weakly supervised fashion, have received increasing attention and demonstrate promising applications.[6,7] Following this trend, one interesting but challenging problem arises for scene understanding: how to parse raw images using the strength of numerous unsegmented but tagged images, because image-level tags can be achieved more easily.

To investigate this problem, we developed a unified framework: a novel probabilistic Expectation- Maximization (EM) formulation in which two mutually conditional steps perform iteratively, providing complementary information to each other in a self-driven manner. In addition, we can apply the proposed framework directly on new test images to perform multilabel image annotation. We evaluated our approach on several benchmarks and found it outperforms other state-of-the-art methods.

Related Work

Traditional efforts for scene understanding, such as Conditional Random Field (CRF),[1,2] Texton-Forest,[8] and Graph Grammar,[9] mainly focused on capturing scene appearances, structures, and spatial contexts by developing combinatorial models. These models were generally founded on supervised learning techniques and required manually prepared training data containing labels at the pixel level.

Several weakly supervised methods have been proposed to indicate the classes that are presented in the images with only image-level labels. For example, John M. Winn and Nebojsa Jojic[10] proposed learning object classes on the basis of unsupervised image segmentation. Ke Zhang and his colleagues[5] learned classification models for all scene labels by selecting representative training samples, and Alexander Vezhnevets and his colleagues[11] used multiple instance learning.

Some nonparametric approaches have also been studied that solve these problems by searching and matching with an auxiliary image database. For example, Ce Liu and his colleagues[6] discussed an efficient structure-aware matching algorithm to transfer labels from a database to the target image, but pixel-wise annotation was required for the auxiliary images.

Overview of the Two-Step Framework

In Step 1 (see Figure 1), we search for exemplar images (that is, reference images; see Figure 1a) from the auxiliary database (Figure 1b) that match the target image (Figure 1c). These references must share similar semantic concepts with the target. Moreover, we enforce the representation to be semantically meaningful—that is, the references that are selected must contain consistent tags. During the iteration process,

1070-986X/15/$31.00 © 2015 IEEE

Published by the IEEE Computer Society

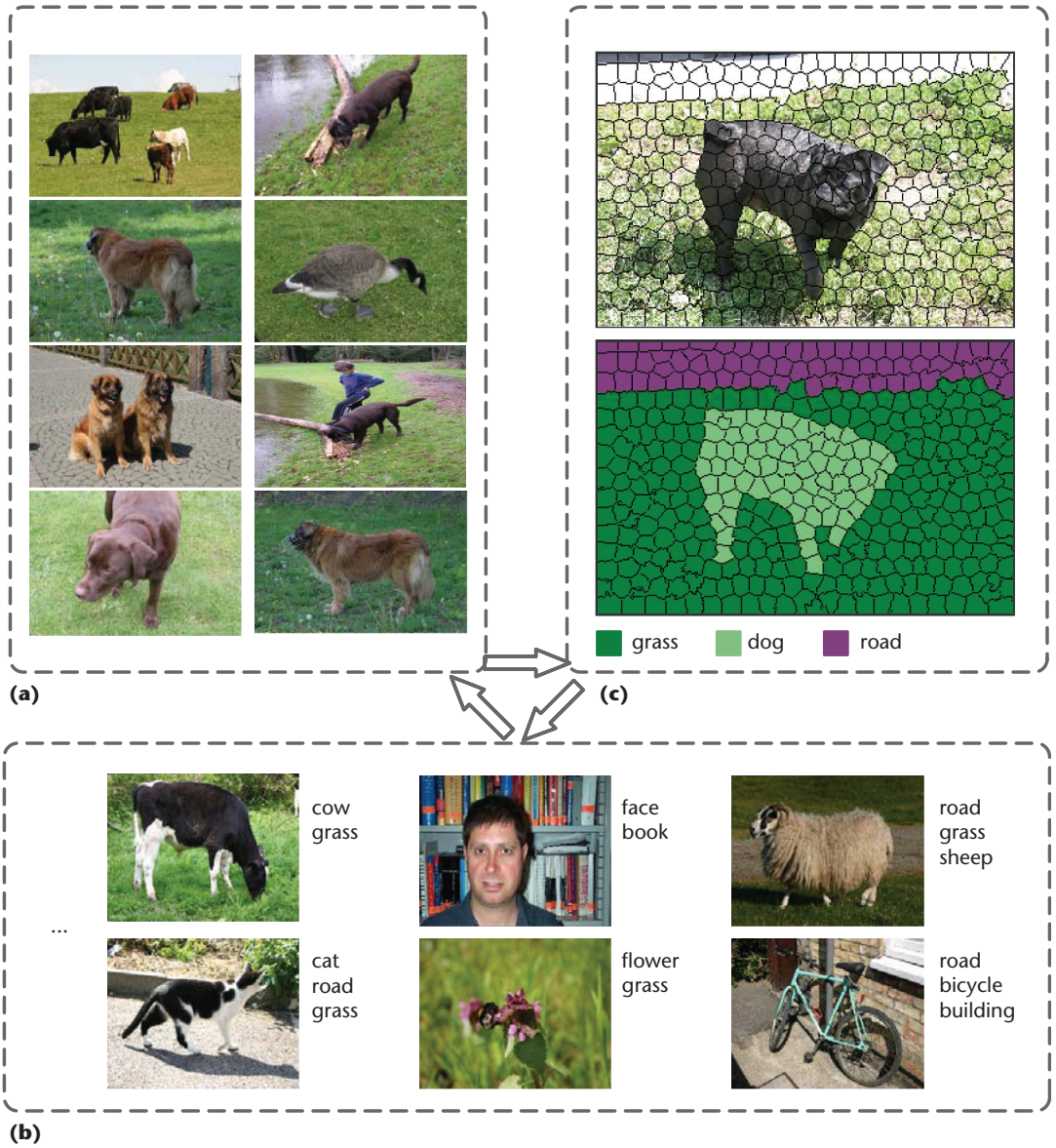


Figure 1. In our framework, we semantically segment the target image in a self-driven fashion. The algorithm iterates to retrieve (a) the exemplars matching the target from (b) the auxiliary data, and then (c) parses the target image using the strength of the selected exemplars.

we can take into account the tags of the target image, determined in the prior label assignment step (Step 2). We solve Step 1 using the proximal gradient method.

In Step 2, we assign labels to the pixels of the target by propagating semantics from the selected references. We create a graphical model in which the vertices are the superpixels from the target image and its references. Two types of edges are defined over the graph (which inspired by earlier work[12]): the inner edges connecting the adjacent vertices within the target, and the outer edges connecting the vertices of the target to those of its references. By aggregating the two types of edge connections, we can then derive the potentials into a Markov Random Fields (MRF) form, which can be quickly solved by the Graph Cuts algorithm.[2]

Problem Formulation

Here, we phrase the problem in a probabilistic formulation and then discuss the Expectation-Maximization (EM) inference framework for optimization.

Probability Model

Let $\Delta = \{I_k, L_k\}_{k=1}^{N}$ denote a set of images $\{I_k\}$ with image-level labels $\{L_k\}$. Each image I_k is represented as a set of superpixels $\{x_i^k\}_{i=1}^{n_k}$, where n_k is the number of superpixels in I_k.

Given the target image I_t, our task is to predict its image-level labels L_t, as well as to assign

each superpixel x_i^t a label $y_i^t \in L_t$. Let Y_t denote the whole label assignment—that is, $Y_t = \{y_i^t\}_{i=1}^{n_t}$; now we can define the joint probability distribution of target image I_t and the label assignment Y_t.

We also define a binary-valued correspondence variable $\alpha = \{\alpha_k\}_{k=1}^{N}$, such that $\alpha_k = 1$ if image I_k is selected as a reference for the target image. α is treated as a hidden variable.

The complete probability model is defined as

$$P(I_t, Y_t\alpha|\Delta) = P(I_t, Y_t|\alpha,\Delta)P(\alpha). \quad (1)$$

We further derive it by summing out α as

$$P(I_t, Y_t|\Delta) = \sum_{\alpha} P(I_t, Y_t|\alpha, \Delta)P(\alpha). \quad (2)$$

Then we derive the optimal label assignment Y_t by maximizing the probability

$$Y_t^* = \arg\max_{Y_t} P(I_t, Y_t|\Delta), \quad (3)$$

and solve it iteratively under an EM framework.

The EM Iterations

Radford M. Neal and Geoffrey E. Hinton showed that estimating Y_t^* from $P(I_t, Y_t|\Delta)$ is equivalent to minimizing the following energy function:[13]

$$\mathcal{L}(Q, Y_t) = -\sum_{\alpha} Q(\alpha) \ln P(I_t, Y_t, \alpha|\Delta) + \sum_{\alpha} Q(\alpha) \ln Q(\alpha), \quad (4)$$

where $Q(\alpha)$ is the posterior of the latent variable α.

Because the second term in Equation 4 is a constant, the optimization iterates in two steps: First, the E step minimizes the energy $L(Q, Y_t)$ with respect to $Q(\alpha)$ with Y_t fixed; second, the M step minimizes the energy $L(Q, Y_t)$ with respect to Y_t with $Q(\alpha)$ fixed.

Step 1. The E step: Approximating $Q(\alpha)$. The posterior of the latent variable $Q(\alpha)$ is defined as

$$Q(\alpha) = P(\alpha|I_t, Y_t, \Delta) = \frac{1}{Z}\exp\{-E_\alpha(\alpha, I_t, Y_t, \Delta)\}, \quad (5)$$

where Z is the normalization constant of the probability. The energy E_α evaluates the appearance and semantic consistency, which is specified as

$$E_\alpha(\alpha, I_t, Y_t, \Delta) = E_{Sc}(\alpha, I_t, \Delta) + \gamma E_{Sa}(\alpha, Y_t, \Delta). \quad (6)$$

where γ is the tradeoff parameter used to balance the appearance similarity and the semantic consistency.

The first term E_{Sc} measures the appearance similarity between I_t and images in Δ. It is defined as

$$E_{Sc} = \frac{1}{2}\|F(I_t) - B\alpha\|_2^2 + \beta\|\alpha\|_1, \quad (7)$$

where β is the tradeoff parameter used to balance the sparsity and the reconstruction error. $F(\cdot)$ is an m-dimensional global feature of an image, and $B \in R^{m\times N}$ is a matrix consisting of all the features of the image in Δ.

The second term E_{Sa} in Equation 6 measures semantic consistency, defined as

$$E_{Sa} = \frac{1}{2}\sum_{i,j\in N} \mathcal{S}_{ij}\left\|\frac{\alpha_i}{\sqrt{A_{ii}}} - \frac{\alpha_j}{\sqrt{A_{jj}}}\right\|_2^2 + \lambda\,\alpha^T\mathcal{D}\alpha = \alpha^T\mathcal{L}\alpha + \lambda\,\alpha^T\mathcal{D}\alpha, \quad (8)$$

where S_{ij} measures the semantic similarity between $(I_i, I_j) \in \Delta$ as

$$\mathcal{S}_{ij} = \frac{|L_i \cap L_j|}{|L_i \cup L_j|} \quad (9)$$

and $\mathbf{A}$ in Equation 8 is a diagonal matrix, where $A_{ii} = \sum_{j=1}^{N} \mathcal{S}_{ij}$ and $\mathcal{L} = A^{-1/2}(A - \mathcal{S})A^{-1/2}$, in which $\mathcal{L}$ is the normalized Laplacian matrix.

Images with similar semantics should be encoded with similar activations. In other words, if two images have common labels, then the activations corresponding to this image pair should also be close to each other. The distance between their activation codes should be small.

$\mathcal{D}$ is a diagonal matrix where $\mathcal{D}_{kk}$ measures the semantic dissimilarity between $I_k \in \Delta$ and the target image I_t. Thus the second term $\alpha^T\mathcal{D}\alpha$ (note that $\alpha\mathcal{D}\alpha$ is convex, and convenient for optimization) penalizing the target I_t is reconstructed by images that are semantically dissimilar to I_t. We define the diagonal matrix $\mathcal{D}$ as

$$\mathcal{D}_{kk} = 1 - \frac{|L_t \cap L_k|}{|L_t \cup L_k|}, \quad (10)$$

where L_t are the latent labels of the target image, which are unknown at the beginning (we initialize L_t as the whole label set of the database), and can be determined from Y_t during later iterations.

Step 2. The M step: Estimating Y_t. The M step minimizes the following energy function with respect to Y_t:

$$E_M(Y_t) = -\sum_{\alpha} Q(\alpha) \ln P(I_t, Y_t, \alpha|\Delta), \quad (11)$$

However, summing out α for all possibilities demands very expensive computational costs, particularly to process a large number N of data. Instead, we seek a lower bound $E_M(Y_t)$. Assume that we can infer α^* with the maximized probability $Q(\alpha^*)$ by the E step. Then we can define the joint distribution of (I_t, Y_t) conditioned on $Q(\alpha^*)$, and we have

$$\sum_{\alpha} P(I_t, Y_t|\Delta; \alpha^*) > \sum_{\alpha} P(I_t, Y_t, \alpha|\Delta). \quad (12)$$

It is straightforward in the context of our task, because the cumulative density of assigning labels from good references (that is, given α^*) is higher than that with general cases. Thus, we set the lower bound as

$$E_M(Y_t) > -\sum_{\alpha} Q(\alpha) \ln P(I_t, Y_t, |\Delta; \alpha^*), \quad (13)$$

where $Q(\alpha)$ is fixed by the last E step. The energy to be minimized can be further simplified as

$$\hat{E}_M(Y_t) = -\ln P(I_t, Y_t|\Delta, \alpha^*). \quad (14)$$

Later, we will specify $-\ln P(I_t, Y_t|\Delta, \alpha^*)$ with a combinatorial graphical model.

Inference and Implementation

With the EM formulation, the inference algorithm iterates in two steps: first computing α^* in the E step for reference retrieval, then solving the optimal labeling Y_t^* with the selected references in the M step.

Adaptive Reference Retrieval

Maximizing $Q(\alpha)$ is equivalent to minimizing the energy defined in Equation 6 with regard to $\alpha^* = \arg\min_{\alpha} E_{\alpha}(\alpha, I_t, Y_t, \Delta)$. Notice that $E_{\alpha}(\alpha, I_t, Y_t, \Delta)$ can be regarded as a semantic-aware sparse representation, where we jointly model the appearance reconstruction with semantic consistency. Figure 2 intuitively illustrates this model, and it can be rewritten as

$$E_{\alpha} = \frac{1}{2}\|F(I_t) - B\alpha\|_2 + \beta\|\alpha\|_1 + \frac{1}{2}\gamma\,\alpha^T\Lambda\alpha, \quad (15)$$

where $\Lambda = 2(\mathcal{L} + \lambda\mathcal{D})$. The semantic associated terms in Equation 15 can be phrased in convex forms, so we can use the proximal gradient method to solve this problem efficiently. The optimization process is shown in Algorithm 1 (see Figure 3).

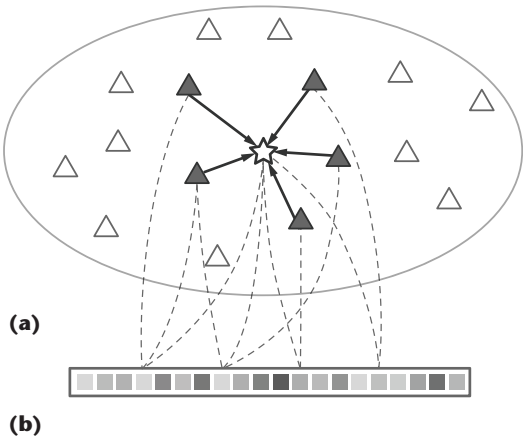


Figure 2. Illustration of semantic-aware sparse coding. (a) A star denotes the target image, and triangles denote each auxiliary image. The dark triangles represent the images selected as the references. (b) The grey squares represent various semantic labels that are introduced as constraints during the optimization. By the end of the process, the algorithm has selected a subset of auxiliary images as references for the target image.

Given the optimal α^*, we can simply select the references according to coding coefficients—for example, by a threshold. And we set $\alpha_k = 0$ if image I_k is not selected.

Aggregated Label Assignment

Given the references determined by α^*, we propagate their semantic labels to I_t by constructing a combinatorial graph. We extract superpixels from both I_t and the references as graph vertices, and we connect them with probabilistic edges incorporating their affinities, as Figure 4 illustrates.

Two types of edges are considered over the graph: the inner edges ω connecting the spatial neighboring superpixels within the target (red wavy lines in Figure 4), and the outer edges ξ connecting the superpixels of the target to those of its references (straight green lines in Figure 4). Each superpixel of the target connects with the q most similar superpixels of each reference.

We define $-\ln P(I_t, Y_t|\Delta, \alpha^*)$ in Equation 14 on the graphical model as

Input: Target image feature $F(I_t)$, codebook B, semantic constraints Λ, and the threshold σ for stop.

Output: Semantically sparse coding coefficient α^*.

Initial: To initialize α^* randomly, let $k = 1$. Denote $g(\alpha) = \frac{1}{2}\left\| F(I_t) - B\alpha \right\|_2 + \frac{1}{2}\gamma \alpha^T \Lambda \alpha$, so Equation 15 can be reformulated as $E_\alpha = g(\alpha) + B\left\|\alpha\right\|_1$.

1: **while** $\|\alpha^{k+1} - \alpha^k\|_2 > \sigma$ **do**

2: Compute the gradient of $g(\alpha^k)$ at α^k,

$\nabla g(\alpha^k) = \boldsymbol{B}^T(\boldsymbol{B}\alpha^k - F(I_t)) + \gamma\Lambda\alpha^k$

3: $\mathbf{z}_L^* = \arg\min_{\mathbf{z}}(\mathbf{z} - \alpha^{k\,T}\nabla g(\alpha^k) + \beta)\|\mathbf{z}\|_1 + \frac{U}{2}\|\mathbf{z} - \alpha^k\|_2$,

where $U>0$ is a parameter.

4: Iteratively U increase by a constant factor until the condition

$g(\mathbf{z}_L^*) \le M_g^L(\alpha^k, \mathbf{z}_L^*) := g(\alpha^k) + \nabla g(\alpha^k)^T(\mathbf{z}_L^* - \alpha^k) + \frac{L}{2}\|\mathbf{z}_L^* - \alpha^k\|_2$

is met,

else return to step 3.

5: Update: $\alpha^{k+1} = \alpha^k + \nu_k(\mathbf{z}_L^* - \alpha^k)$, where $\nu_k \in (0,1]$.

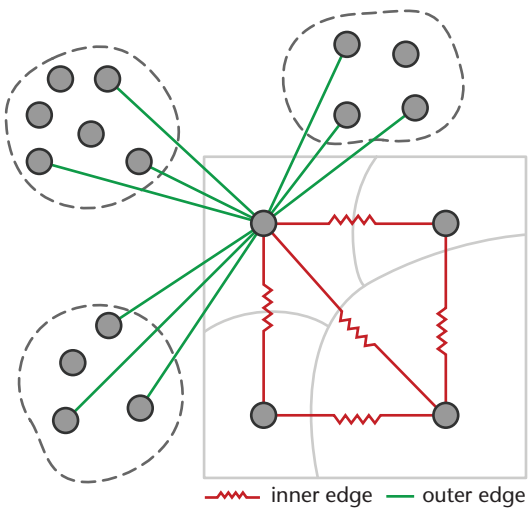
6: $k := k + 1$

7: **end while**

8: $\alpha^* = \alpha^k$

Figure 3. Algorithm 1: Adaptive reference retrieval. To solve the similarity term with semantic consistency, the term was fixed using the proximal gradient method.

Figure 4. Illustration of the combinatorial graphical model. The dark circles represent the superpixels; the four dark circles over the square region are extracted from the target image while the others are extracted from references that are denoted by dashed regions.



$$-\ln P(I_t, Y_t|\Delta, \alpha^*) = \sum_{i=1}^{n_t}\psi(y_i^t|\alpha^*, \Delta) + \sum_{(x_i^t, x_j^t)\in\omega}\phi(y_i^t, y_j^t, x_i^t, x_j^t), \quad (16)$$

where ω is the inner edges. The optimization of Equation 14 becomes a tractable graphical model optimization problem.

To derive the potentials of assigning labels to one vertex of the target $\psi(y_i^t|\alpha^*, \Delta)$ in Equation 16, we propose the *semantic- based superpixel density prior*, which is defined as

$$\psi(y_i^t|\alpha^*, \Delta) = \sum_{k=1}^{N}\alpha_k^*\rho(x_i^t, I_k)\delta(y_i^t \in L^k), \quad (17)$$

where $\rho(x_i^t, I_k)$ denotes the density of superpixel x_i^t in image I_k, which is defined as

$$\rho(x_i^t, I_k) = \frac{1}{N_\xi}\sum_{(x_i^t, x_j^k)\in\xi}\left\|\int(x_i^t) - \int x_j^k\right\|_2, \quad (18)$$

where ξ denotes outer edges, N_ξ is the number of outer edges, and $f(\cdot)$ is the feature vector of a superpixel. This density measures the similarity between the superpixel x_i^t in the target and its neighboring superpixels connected by outer edges in the reference image I_k, so it implicitly exhibits the probability of x_i^t sharing the same labels with its reference I_k.

The pairwise potentials—that is, $\phi(y_i^t, y_j^t, x_i^t, x_j^t)$ in Equation 16—encourages smoothness between neighboring superpixels within the target:

$$\phi(y_i^t, y_j^t, x_i^t, x_j^t) = \left\|f(x_i^t) - f(x_j^t)\right\|_2 \delta(y_i^t \neq y_j^t), \quad (19)$$

where $\delta(\cdot)$ is the indicator function.

Input: Target $I_t = \{x_i^t\}_{i=1}^{n_t}$, and auxiliary $\Delta = \{I_k, L_k\}_{k=1}^{N}$.

Output: Label of each superpixel $Y_t = \{y_i^t\}_{i=1}^{n_t}$.

Initial: L_t^1 contains all labels, and $n = 1$.

while $L_t^{n+1} \neq L_t^n$ **do**
  Minimize E_α defined in Equation 15 using Algorithm 1.
  Sort $\alpha*$ in descending order, select the images corresponding to the p-first nonzero coefficients, as a set B.
  for all x_i^t in I_t **do**
    for all image I_k in B **do**
      Select the q-most similar superpixels $O_{x_i^t}^k = \{x_j^k\}_{j=1}^q$
      Construct $O_{x_i^t} = \cup_k O_{x_i^t}^k$
    end for
    Add (x_i^t, x_j^k) to ω for all $x_j^k \in O_{x_i^t}$.
    Add (x_i^t, x_j^t) to ξ for all neighbors $\{x_j^t\}$ of $x_i^t, i \neq j$
  end for
  Minimize Equation 16. Optimize the latent label Y_t^* using alpha-beta swap algorithms of graph cuts.
  Update L_t^{n+1} as the unique set of Y_t^*.
  $n := n + 1$
end while

Figure 5. Algorithm 2 presents the overall procedure of our framework. Our framework is an iterative solution that uses adaptive reference retrieval and aggregated label assignments.

Thus the approximate solutions of Equation 16 can be found using alpha-beta swap algorithms of graph cuts. The sketch of our framework is shown in Algorithm 2 (see Figure 5).

Image Annotation

We propose a simple method to transfer n labels to a test image I_t from the query's K nearest neighbors in the training set. For a given test image I_t, we determine the sparse reconstruction coefficient vector α by solving the problem in Equation 15, where we set $\lambda = 0$, and set the other parameters ($q, p, \beta,$ and γ) as the same as described later under "Implementation Details." We denote the optimal sparse coefficient solution as $\hat{\alpha}$ and its top K largest value as $\hat{\pi} \in R^{K\times 1}$ corresponding to image label indicator $\mathbf{l}_i \in R^C$, $i = 1, 2, ..., K$. We can then obtain the label vector probability of the test image as

$$\mathbf{z}_t = \sum_{i=1}^{K} \hat{\pi}_i \mathbf{l}_i, \tag{20}$$

where $\hat{\pi}_i$ is the ith component of vector $\hat{\pi}$. The labels corresponding to the top few largest values in $\mathbf{z}_t$ are considered as the final annotations of the test image.

We compared two annotation methods:

- *weighted*: weighting the annotation with the sparse reconstruction coefficient $\hat{\pi}_i$, and
- *unweighted*: setting $\hat{\pi}_i = 1, i = 1, \cdots, K$ in manually,

and found that the sparse coefficient α is extremely useful for image annotation.

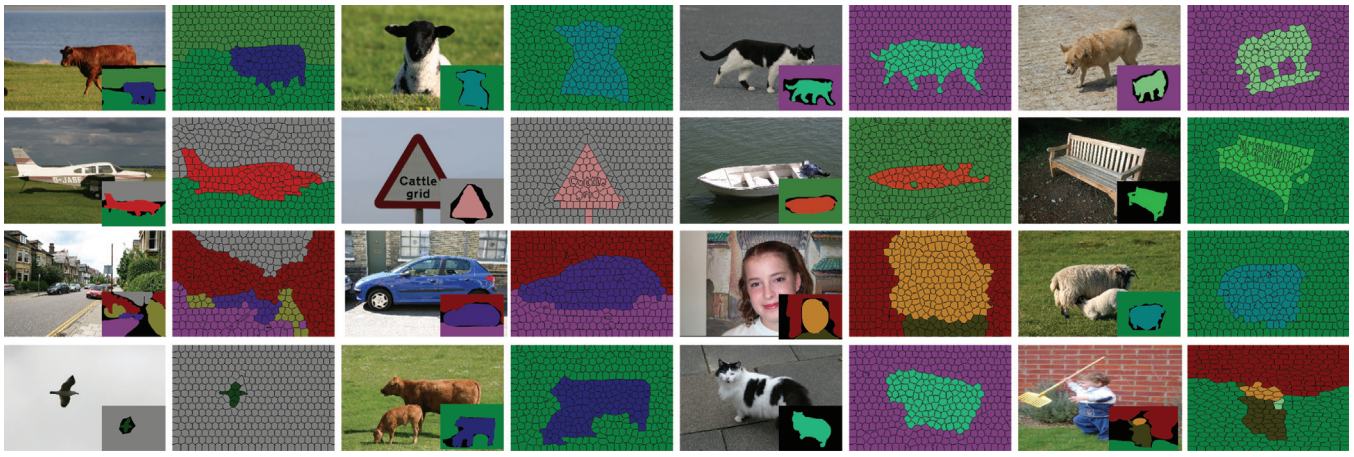
We also compared our proposed method with classical image annotation approaches and found that our propagation process is robust and less sensitive to image noise owing to the semantic constraints in the image retrieval step. We also found that we can retrieve images by jointly matching appearances as well as semantics. Finally, the proposed algorithm is scalable to a large scale.

Experiments

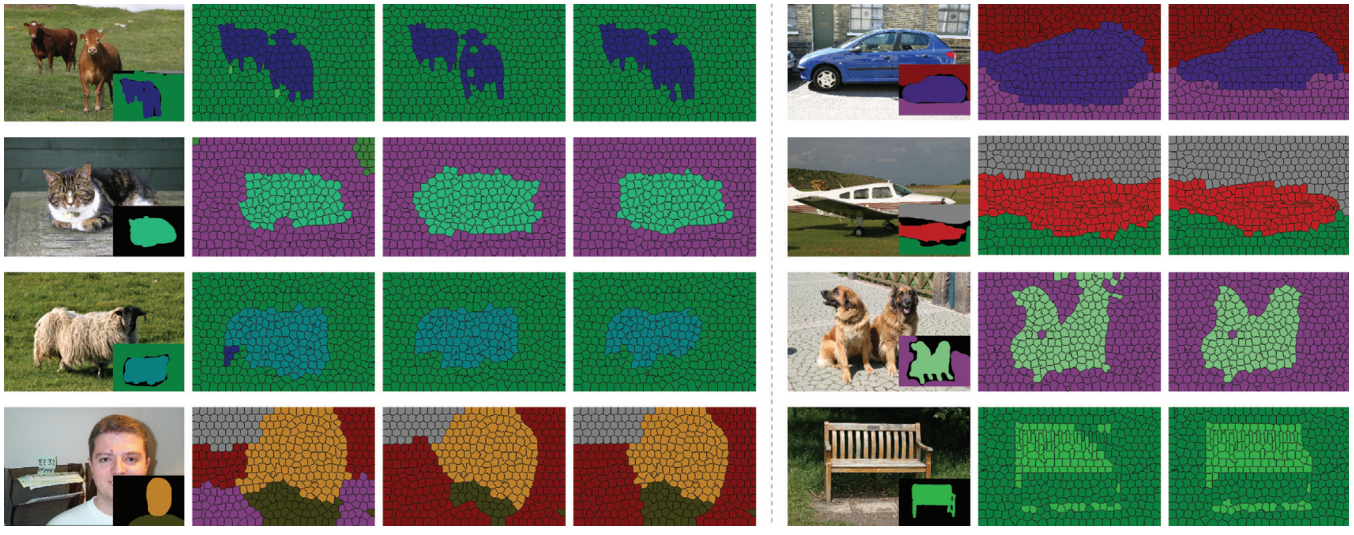
We conducted extensive experiments to validate the performance of our method. We also conducted an empirical study on the effectiveness of the proposed EM iterations.

Implementation Details

Five parameters are required to be set in our framework. We set $q = 20$ to construct the



(a)



(b)

Figure 6. Semantic segmentation experiments using the MSRC dataset: (a) some final results and (b) some intermediate results. The original image and its ground truth are shown on the left, and the semantic segmentation result using our method is on the right.

q-nearest graph, and set $p = 10$ to retrieve 10 images as reference for each test image. In the experiment, we also set $\lambda = 1$ empirically. The other parameters β and γ are introduced later.

Dataset

To verify the effectiveness of our method, we compared it with state-of-the-art methods by conducting experiments on two challenging datasets, MSRC[1] and VOC 2007.[14] We used the standard average per-class measure (*average accuracy*) to evaluate performance. For each test image, we used the training set as the auxiliary data for our framework.

Experiment Series 1: Image Semantic Segmentation

Here, we discuss our analysis of the parameters and the tests we performed on the MSRC and VOC 2007 datasets.

Parameter analysis. Specifically, we focused on the effects of β and γ, which control the influence of the appearance term (also called the sparse term) and the semantic constraint term, respectively, in Equation 15. These two parameters are crucial to our results. The range of β and γ were both set to {0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30}. We used the semantic segmentation results (see Figure 6) to tune parameters.

We used the MSRC dataset to fine-tune the parameters. The results of changing the parameter values are presented in Figure 7, from which we can observe the following conclusions:

- When β and γ increase from small values to large values, performance apparently varies. This shows that the sparse term and semantic constraint term greatly impact performance.
- The mean average precision reached the peak points (0.71) when $\beta = 0.1$ and $\gamma = 0.2$ on MSRC. These values lie in the middle range, showing that precision does not increase monotonically when β and γ increase. In the following experiments, we adopt the best parameter settings on all datasets.

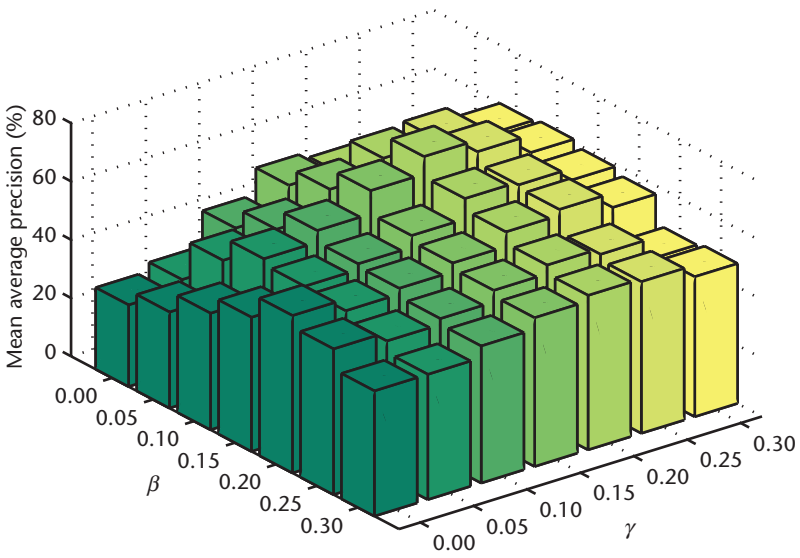


Figure 7. Parameter tuning results of parameters β and γ for the MSRC dataset.

Experiments on the MSRC dataset. Given this insight, we compared the proposed method with two state-of-the-art algorithms, MIM[11] and the method developed by Ke Zhang and his colleagues.[5]

Table 1 shows that, on average, our algorithm outperformed the others. Benefitting from the semantic constraints incorporated in our approach, we achieved significant improvements for certain difficult classes—for example, Chair and Cat. Figure 6a presents several visualized results with the corresponding ground-truths, and, because of space limitations here, more semantic segmentation results are available in supplementary material (see http://vision.sysu.edu.cn/projects/scene_parsing).

Experiments on the VOC 2007 dataset. Few performance results using the VOC 2007 dataset have been reported, due to the 20 extremely challenging categories it contains. We compared our method with the Weakly Supervised STF[8] (Semantic Texton Forests) by running the code provided by the author. We also compared our method with that of Ke Zhang and his colleagues.[5] Results are reported in Table 2, with our method outperforming the others by 3 percent.[5]

It takes about 8 seconds per image with an unoptimized Matlab implementation for semantic segmentation on a 64-bit system with a Core-4 3.6 GHz CPU and 4 GBytes of memory (1 second to extract features, 5 seconds to do sparse coding with semantic constraints, and 2 seconds for optimization by GraphCuts).

Moreover, we validated the effectiveness of the proposed EM iterations from two aspects. First, we plotted the energy E_α in each iteration, which is the energy of semantic-aware sparse coding defined in Equation 15 (see Figure 8). Figure 6b shows some intermediate results with the EM iterations, empirically supporting the effectiveness of the iterations. (Generally, the iteration is complete after two or three steps because the average number of labels for each image is 3 in both the MSRC and the VOC 2007 datasets.)

Experiment Series 2: Image Annotation on a Test Image

Here, we analyze our method and compare it with others.

Benchmarks and metrics. We implemented three popular algorithms, MAHR (Multi-Label Hypothesis Reuse),[15] MLkNN (Multi-Label k Nearest Neighbors),[16] and ML-LOC (Multi-Label Label Correlations Locally),[17] as benchmark baselines for the image annotation task. We evaluated and compared these algorithms over two datasets, MSRC and VOC 2007, each of which was randomly and evenly split into training and testing subsets. We measured image annotation performance by mean average precision, which is widely used for evaluating the performances of ranking related tasks.

Table 1. Accuracy (%) of our method for each category on the MSRC dataset, in comparison with other algorithms. The most accurate result in each category appears in bold.

Method	Building	Grass	Tree	Cow	Sheep	Sky	Airplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	Average*
Multi-Image Model (MIM)[11]	12	83	70	**81**	**93**	**84**	**91**	55	**97**	**87**	92	**82**	69	**51**	61	59	66	53	44	9	**58**	67
The method developed by Ke Zhang and his colleagues [5]	**63**	**93**	**92**	62	75	78	79	64	95	79	**93**	62	76	32	**95**	48	**83**	63	38	**68**	15	69
Our method	45	73	65	79	81	66	71	**87**	75	84	73	73	**94**	**51**	89	**85**	42	**83**	**81**	66	32	**71**

*Average accuracy over all categories.

Table 2. Accuracy (%) of our method for each category on the VOC 2007 dataset, in comparison with other algorithms. The most accurate result in each category appears in bold.

Method	Airplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Diningtable	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	TVmonitor	Average*
Weakly Supervised STF[8]	14	8	11	0	**17**	**46**	5	13	4	0	**30**	29	12	**18**	40	6	17	17	14	9	16
The method developed by Ke Zhang and his colleagues [5]	48	**20**	**26**	**25**	3	7	**23**	**13**	**38**	**19**	15	39	17	**18**	25	**47**	9	**41**	17	33	24
Our method	**68**	14	12	16	4	27	18	12	28	16	7	**46**	**36**	11	**78**	18	**29**	11	**47**	**41**	**27**

*Average accuracy over all categories.

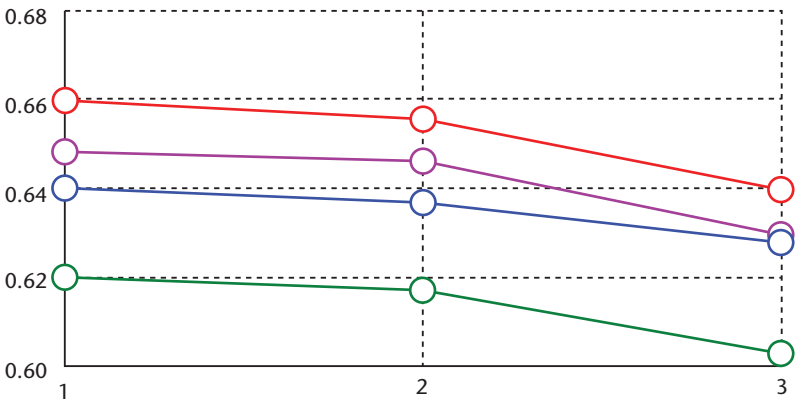


Figure 8. Illustration of the decreasing energy E_α with regard to time. The x-axis indicates the number of iterations, and the y-axis shows the energy E_α of Equation 15. The results were randomly selected from the test set.

MLkNN and ML-LOC are state-of-the-art multilabel annotation algorithms in the literature. They have been reported to outperform most other multilabel annotating algorithms, such as RankSVM.[18]

Results. The weighted method outperformed the unweighted one, as Table 3 shows. We found that the sparse coefficient α is useful for improving image annotation performance and for image semantic segmentation, as we did the image retrieval by jointly matching their appearance as well as their semantics. The larger α_i is, the greater semantic similarity there is between the test image and image I_i (that is, sharing the more common labels).

Table 3. Comparisons of mean average precision (%) of image label annotations on two different datasets. The most accurate result in each category appears in bold.

Dataset	Annotation method				
	MAHR	MLkNN	ML-LOC	Our method, unweighted	Our method, weighted
MSRC	49.5	70.8	77.3	76.1	**84.7**
VOC 2007	34.0	47.6	48.9	45.8	**57.5**

Figure 9. Some example results on image annotation from the MSRC (left) and VOC 2007 dataset (right).

The weighted method we have proposed outperforms the three classical methods listed in Table 3. Some example image annotation results from the MSRC and VOC 2007 dataset are shown in Figure 9. In the figure, we display only the top two or three labels for VOC 2007 and MSRC, since the average number of labels for each image in VOC 2007 and MSRC is two and three, respectively.

Compared with traditional supervised learning methods, our framework is more flexible for real applications such as online image retrieval. In future work, we plan to improve our algorithm's efficiency by utilizing parallel implementation and validate our approach on larger-scale datasets. **MM**

Acknowledgments

The corresponding author of this work is Qing Wang. This work was supported by the National Natural Science Foundation of China (nos. 61170193, 61370185, and 61173084), Guangdong Science and Technology Program (no. 2012B031500006), Guangdong Natural Science Foundation (no. S2012020011081), and Special Project on Integration of Industry, Education and Research of Guangdong Province (nos. 2012B091100148 and 2012B091000101). This work is partially supported by the Hong Kong Scholar program.

References

1. J. Shotton et al., "Textonboost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation," *Proc. 9th European Conf. Computer Vision, Part* 1 (ECCV), 2006, pp. 1–15.
2. L. Ladicky et al., "Graph Cut Based Inference with Co-Occurrence Statistics," *Proc. 11th European Conf. Computer Vision: Part V* (ECCV), 2010, pp. 239–253.
3. L. Lin et al., "A Stochastic Graph Grammar for Compositional Object Representation and Recognition," *Pattern Recognition*, vol. 42, no. 7, 2009, pp. 1297–1307.
4. L. Lin et al., "Representing and Recognizing Objects with Massive Local Image Patches," *Pattern Recognition*, vol. 45, no. 1, 2012, pp. 231–240.
5. K. Zhang et al., "Sparse Reconstruction for Weakly Supervised Semantic Segmentation," *Proc. 23rd Int'l Joint Conf. Artificial Intelligence* (IJCAI), 2013, pp. 1889–1895.
6. C. Liu, J. Yuen, and A. Torralba, "Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2009, pp. 1972–1979.

7. P. Luo et al., "Joint Semantic Segmentation by Searching for Compatible-Competitive References," *Proc. 20th ACM Int'l Conf. Multimedia*, 2012, pp. 777–780.
8. J. Shotton, M. Johnson, and R. Cipolla, "Semantic Texton Forests for Image Categorization and Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2008, pp. 1–8.
9. L. Lin et al., "Discriminatively Trained And-Or Graph Models for Object Shape Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, 2014, pp. 959–972.
10. J. Winn and N. Jojic, "Locus: Learning Object Classes with Unsupervised Segmentation," *Proc. 10th IEEE Int'l Conf. Computer Vision, vol.* 1 (ICCV), 2005, pp. 756–763.
11. A. Vezhnevets, V. Ferrari, and J.M. Buhmann, "Weakly Supervised Semantic Segmentation with a Multi-Image Model," *Proc. 2011 IEEE Int'l Conf. Computer Vision* (ICCV), 2011, pp. 643–650.
12. L. Lin, X. Liu, and S.-C. Zhu, "Layered Graph Matching with Composite Cluster Sampling," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, 2010, pp. 1426–1442.
13. R.M. Neal and G.E. Hinton, "A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models*, Springer, 1998, pp. 355–368.
14. M. Everingham et al., "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," 2007; http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html.
15. S.-J. Huang, Y. Yu, and Z.-H. Zhou, "Multi-Label Hypothesis Reuse," *Proc. 18th ACM SIGKDD Conf. Knowledge Discovery and Data Mining* (KDD), 2012, pp. 525–533.
16. M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A Lazy Learning Approach to Multi-Label Learning," *Pattern Recognition*, vol. 40, no. 7, 2007, pp. 2038–2048.
17. S.-J. Huang and Z.-H. Zhou, "Multi-Label Learning by Exploiting Label Correlations Locally," *Proc. 26th AAAI Conf. Artificial Intelligence* (AAAI), 2012, pp. 949–955.
18. A. Elisseeff and J. Weston, "A Kernel Method for Multi-Labelled Classification," *Advances in Neural Information Processing Systems*, vol. 14, 2001, pp. 681–687.

Xionghao Liu is a master's student majoring in pattern recognition and computer vision at Sun Yat-sen University, Guangzhou, P.R. China. His research interests include computer vision and machine learning. He received his BE degrees in automation from Sun Yat-sen University. Contact him at lxiongh@126.com.

Wei Yang is a PhD student in the Department of Electronic Engineering at the Chinese University of Hong Kong. His research interests include computer vision and machine learning. Yang received his master's degree in computer software and theory from Sun Yat-sen University. Contact him at platero.yang@gmail.com.

Liang Lin is a professor in the School of Advanced Computing at Sun Yat-sen University. His research focuses on new models, algorithms, and systems for intelligent processing and understanding of visual data. He has published more than 60 papers in top-tier academic journals and conferences and served as associate editor for the journals *Neurocomputing* and *The Visual Computer*. Contact him at linliang@ieee.org.

Qing Wang is an associate professor of computer science at Sun Yat-sen University. His research focuses on human-computer interaction, user experience, collaborative software, and Web usability, with special interest in utilizing browser history in collaboration. Wang received his PhD in computer science from Sun Yat-sen University. He is a member of SIGCHI. Contact him at ericwangqing@gmail.com.

Zhaoquan Cai is a professor of computer science at Huizhou University, China. His research interests include computer networks, intelligent computing, and database systems. Cai received his PhD in computer science from Huazhong University of Science and Technology. Contact him at caizhaoquan@139.com.

Jianhuang Lai is a professor and dean of the School of Information Science and Technology at Sun Yat-sen University. His research focuses on image processing, pattern recognition, multimedia communication, and wavelets and their applications. Lai received his PhD in mathematics from Sun Yat-sen University. He serves as a Standing Member of the Image and Graphics Association of China and a Standing Director of the Image and Graphics Association of Guangdong. Contact him at stsljh@mail.sysu.edu.cn.