Discriminative Maximum Margin Image Object Categorization with Exact Inference

Qinfeng Shi
ANU and NICTA
Australia
qinfeng.shi@anu.edu.au

Luping Zhou
ANU and NICTA
Australia

Li Cheng
TTI-Chicago
USA

Dale Schuurmans
Univ of Alberta
Canada

Abstract

Categorizing multiple objects in images is essentially a structured prediction problem: the label of an object is in general dependent on the labels of other objects in the image. We explicitly model object dependencies in a sparse graphical topology induced by the adjacency of objects in the image, which benefits inference, and then use maximum margin principle to learn the model discriminatively. Moreover, we propose a novel exact inference method, which is used in training to find the most violated constraint required by cutting plane method. A slightly modified inference method is used in testing when the target labels are unseen. Experiment results on both synthetic and real datasets demonstrate the improvement of the proposed approach over the state-of-the-art methods.

1. Introduction

A fundamental problem of image understanding is to categorize objects in images based on visual content. This is inherently a structured classification problem for object segments, which is usually obtained by image segmentation or human interaction. Recently, discriminative learning methods have been employed for this task, *e.g.* using conditional random fields (CRFs) [6–8], to exploit the spatial Markovian dependencies between neighboring pixels. A key requirement of these approaches however is an inference procedure for finding the best label assignment given a image content and the model. Previous approaches either ignore dependencies between the objects within an image, (which boils down to independent identically distributed (i.i.d) classification such as SVMs), or perform approximate inference either by message passing methods *e.g.* loopy belief propagation (LBP) [13, 14] or by graph-cuts [1, 5]. However, both graph-cut and LBP are approximate inference methods for multi-class problems in loopy graphs.

In this paper, we propose a discriminative approach for image object recognition based on the maximum margin principle [12]. This approach allows us to explicitly model the object-based nature of the problem by incorporating features and relations of segments rather than pixels. In particular, as an alternative of junction tree algorithm, we develop a novel *exact* inference algorithm for obtaining the global optimal assignment in our set-up, even in the multi-class scenario.

Related Work Recently, Corso et al. [2] have proposed the graph-shift algorithm and applied the CRF learning framework for image labeling, aiming to dynamically update the parent-child relationship in a hierarchical decomposition of a image. This is however still an approximate inference method. Zhu et al. [15] introduce an and/or graph, a special context sensitive grammar, and propose to infer both the label and the latent grammar graph from an image in an unsupervised manner. Meanwhile, Petrov and Klein [9] propose to discriminatively learn the log-linear hierarchical models and the grammars. As a trade-off, their objective function becomes non-convex due to the introduction of the latent variables, therefore there is no guarantee for global optimality.

However models which take into account dependence of *segments* have only recently been introduced. [10] modeled the problem with a fully connected CRF where nodes are the objects and the edges encode the co-ocurrence counts of labels in the training set. Learning is performed by approximate Maximum-Likelihood estimation through sampling, since exact inference is intractable. Performance improvement is reported when compared against independent predictions per object. This model was extended in [4] by incorporating *relative location* information, *e.g.* sky it is typically *above* ground.

2. Modeling Joint Categorization

Problem Formulation We assume that a segmentation of the original image is given. This can be done by several means, such as shown in [4]. We assume each segment has a latent label from a fixed dictionary of L labels $\mathcal{L} = \{l_1, \dots, l_L\}$. The goal is to infer the category of each segment. We model an arbitrary segmentation (a partition) of an image in N parts as a random vector $X = (X^1, \dots, X^N)$. A segmented image is therefore a specific realization of such random vector, which we denote by $x = (x^1, \dots, x^N)$. Since we assume a segmentation as given, this random vector is always observed. In practice each segment x^i will have a feature vector associated to it, which will be incorporated into our model. We denote an arbitrary joint labeling of the segmented image x by a random vector $Y = (Y^1 \dots, Y^N)$. A realization of Y is a specific joint labeling, which we denote by $y = (y^1, \dots, y^N)$, where $y^i \in \mathcal{L}$, $\forall i, 1 \leq i \leq N$.

The Model We cast this estimation problem as finding a discriminant function $f(x, y)$ such that for an image x_t with objects $(x_t^1, \dots, x_t^N)$, we assign the categories that receive the best score with respect to f,

$$y^* = \arg\max_y f(x_t, y). \tag{1}$$

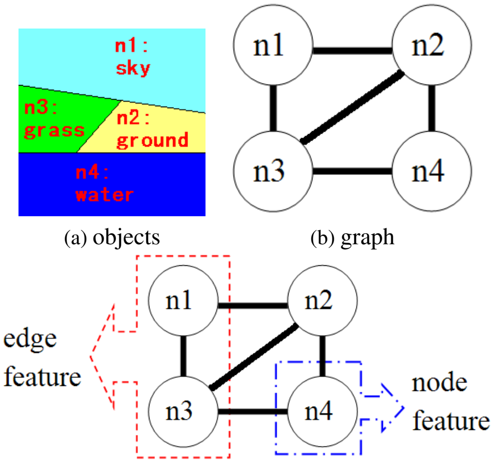
As in many learning algorithms, we consider functions that are linear in some feature representation

$$f(x, y; w) = \langle w, \phi(x, y) \rangle. \tag{2}$$

Here $\phi(x, y)$ is a feature map and w is the corresponding parameter vector. As in CRF, the feature map $\phi(x, y)$ can be decomposed into nodes and edges:

$$\phi(x, y) = \sum_i \phi_1(x, y^i) + \sum_{(i,j) \in \mathcal{A}} \phi_2(x, y^i, y^j). \tag{3}$$

Here ϕ_1 is the node feature or the intra-object feature and ϕ_2 is the edge feature or inter-object feature. More details of features are described in section 3.



(a) objects (b) graph

(c) features

Figure 1: An illustration of the image objects, graph and features. **(a)** contains 4 objects: sky,ground,grass and water. **(b)** is the induced graph. **(c)** shows the node and edge features: Node feature is used to encode the object characteristic, while the edge feature is to encode the interaction between objects. The figure is best viewed in color.

2.1. Maximum Margin Training

We now present a maximum margin training for categorizing image object. The set of labeled segments in one image is an instance. One of the advantages of this method is its ability to incorporate the cost function that the classifier is evaluated with. Let $\Delta(y_t, y^*)$ to be the cost of predicting y^* instead of the true label y_t. Typically one can choose $\Delta(y_t, y^*)$ to be the hamming loss. And we follow the general framework of [12] and look for model parameter w that separates the true label y_t from the other y with some margin depending on Δ additively.

$$\begin{aligned} &\min_{w,\xi} \quad \frac{\lambda \|w\|^2}{2} + \frac{1}{T} \sum_{t=1}^{T} \xi_t \\ &s.t. \quad \langle w, \phi(x_t, y_t) \rangle - \langle w, \phi(x_t, y) \rangle \geq \triangle(y_t, y) - \xi_t, \\ &\qquad \xi_t \geq 0, \forall t = 1, \dots T, \forall y \in \mathcal{Y}, \end{aligned} \tag{4}$$

where T is the number of training images and λ is the regularzation constant which is usually determined by model selection.

This problem (4) is intractable, since the configuration space $\mathcal{Y}$ is exponentially large. However, [12] shows that this problem can be approximately solved in polynomial time with good precision. The key idea is to find the most violated constraints for the current set of parameters and satisfy them up to some precision. In order to do this, one needs to find

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}} \Delta(y_t, y) + f(x_t, y; w), \tag{5}$$

using our inference method. Similarly, in the prediction phase when the true label y is not accessible, we would infer y^* by

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}} f(x, y; w). \tag{6}$$

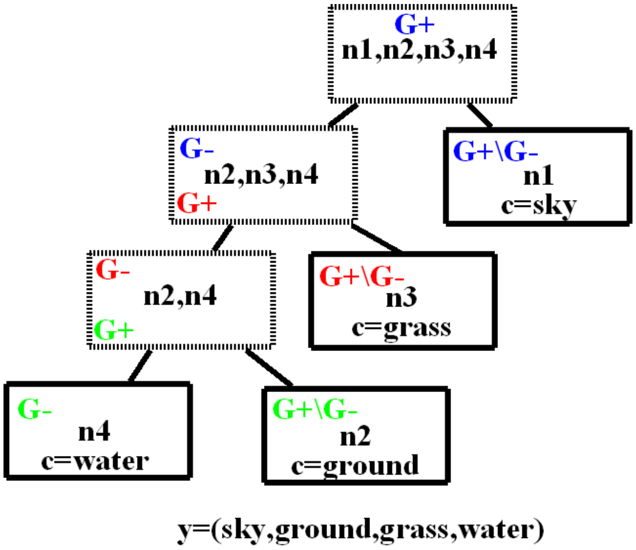


Figure 2: An exemplar illustration of performing backtracking on the graph of Figure 1. The boxes with solid boundary are the subgraphs, of which the classes are assigned. The top layer box represented by blue G_+ decomposes to blue G_- and $G_+\backslash -$. And then the blue G_- is considered as a new G_+ in red color and it starts decomposing again to red G_- and $G_+\backslash -$.

2.2. Exact Inference

We now propose a novel exact inference algorithm for solving (5).

Let G_∞ be the full induced graph of an image x_t, $\mathcal{G}$ be the set of all possible subgraphs of G_∞. The goal is to find the associated y on nodes that give the highest $S(x_t, y) = \Delta(y_t, y) + f(x_t, y; w)$ for any x_t. Clearly enumerating all possible y for all nodes is intractable, because there are exponential many y. Alternatively, we can decompose $S(x_t, y)$ to subgraphs. We find the best labels for small subgraphs and then use them to find the best labels for larger subgraphs iteratively until we are done for the full graph. By this procedure, many non-optimal configuration will be discarded. Before going to more detail, we shall introduce some definition and notions. The **boundary nodes** of a graph G are the nodes adjacent to any node that is not in G. For any G, we associate a special label called key label to G. The key label has to be on one of the boundary nodes of G. When there is only one boundary node, the key label uniquely determines the label of the boundary node.

Given a subgraph G_+ and assume its key label is c_+, the recursion form is:

$$S(G_+, c_+) = \max_{G_-\subset G_+, c_-, c_\pm\in\mathcal{L}} \{S(G_-, c_-) + g(G_+\backslash G_-, c_\pm, G_-, c_-)\}, \qquad (7)$$

where G_- is a subgraph of G_+, c_- is the boundary label of G_-, and $c_\pm$ is the label of $G_+\backslash G_-$. Here we restrict that all nodes in $G_+\backslash G_-$ have to share the same label $c_\pm$. We further restrict $|G_+| = |G_-|+1$ to reduce the computational load while retaining the optimal assignment unchanged.

Algorithm 1 Exact inference for a graph with size N

```
Input: Graph $G_\infty$, set of categories $\mathcal{L}$
Output: score $S$, optimal label $y^*$
Initialize table $B1, B2, B3$ for forward computing:
for $i = 1$ to $N$ do
    for $G_+ \in \mathcal{G}$ s.t. $|G_+| = i$ do
        for $c_+ = l_1$ to $l_L$ do
            $(B1(G_+, c_+), B2(G_+, c_+), B3(G_+, c_+)) \leftarrow (G_-^*, c_-^*, c_\pm^*) = \operatorname{argmax}_{G_- \subset G_+, |G_+| = |G_-|+1, c_-, c_\pm \in \mathcal{L}} \{S(G_-, c_-) + g(G_+\backslash G_-, c_\pm, G_-, c_-)\}$
            $S(G_+, c_+) = S(G_-^*, c_-^*) + g(G_+\backslash G_-^*, c_\pm^*, G_-^*, c_-^*)$
        end for
    end for
end for
Back tracking:
$c_+ \leftarrow \operatorname{argmax}_c S(G_\infty, c)$, $G_+ \leftarrow G_\infty$
repeat
    $G_-, c_-, c_\pm \leftarrow (B1(G_+, c_+), B2(G_+, c_+), B3(G_+, c_+))$
    $(G_+, c_+) \leftarrow (G_-, c_-)$
    $y^*(G_+\backslash G_-) \leftarrow c_\pm$
until $G_- = \emptyset$
```

Define $g(G_+\backslash G_-, c_\pm, G_-, c_-)$ as the incremental function to associate the subgraph $G_+\backslash G_-$ and its label $c_\pm$ with the subgraph G_- and its key label c_-. This function will be used to measure the score increment given $(c_\pm, G_-, c_-)$. Given these definitions, we can express $g(.)$ as

$$\begin{aligned} & g(G_+\backslash G_-, c_\pm, G_-, c_-) \\ &= f(x_t, c_{)} + \Delta(y_t(G_+\backslash G_-), c_\pm) \\ &= \langle w_2, \phi_2(x_t, c_-, c_\pm)\rangle + \langle w_1, \phi_1(x_t, c_\pm)\rangle \\ &\quad + \sum_{v\in(G_+\backslash G_-)} \Delta(y_t(v), c_\pm), \end{aligned}$$

where $y(G)$ denotes the label of nodes in G, and $y(v)$ denotes the label of node v.

Algorithm 1 provides the detailed pseudo-code of our inference algorithm, which is able to find the same optimal assignment in (5) one would obtain by naively enumerating possible y. (6) can then be solved as a special case by removing Δ. Basically it has two steps: forward computing and back tracking. The back tracking step is used to retrieve the best subgraphs G_-, $G_+\backslash G_-$ and labels c_-, $c_\pm$. Figure 2 illustrate such step for the 4 objects image in figure 1.

We begin with $G_+ = \{n_1, n_2, n_3, n_4\}$ and look for the best $G_-, c_-, c_\pm$ in tables $B1, B2, B3$. $c_\pm$ determines the $y(n_1)$ to be sky in this case. Then consider $G_+ = \{n_2, n_3, n_4\}$ and look up the tables, we get $y(n_3)$ is grass. Keeping doing so we obtain y over all nodes. As we can see in algorithm 1, most computation is in the forward computing part. The complexity of our inference is exponential to the tree-width (maximal clique size). This is the same as junction tree algorithm. However, introducing some heuristic can considerably reduce the complexity to polynomial with a price that the inference may not be exact. Our attempts on speeding up by heuristic lead to poorer experiment result. Our conjuncture is cutting plane method relies on exact or very good approximated inference. Even occasional poor approximation will cause early stop of the optimization as we observed in our experiment.

3. Experiments

Datasets We conducted experiments on three datasets: a synthetic dataset and two well-known datasets: the MSRC object categorization dataset[1] and the Corel dataset[2].

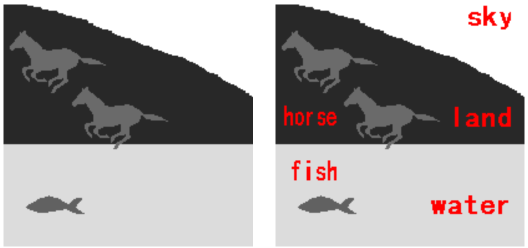
Comparison Methods We compared our approach (called SVM-DP) with three state-of-the-art methods:

SVMs multi-class SVMs [3]. Here we use the LIBSVM package [3] for both linear kernel (SVM-L) and RBF kernel (SVM-RBF);

MRF markov random field (MRF) that utilizes graph-cuts [1, 5] for inference;

CRF conditional random field (CRF) for categorization in [4]. Our implementation of [4] uses exact inference instead of sampling to compute the derivative of log-partition function. This is because it is much easier to implement exact inference than to perfectly reproduce the sampling scheme of [4]. Aiming at a fair comparison, all structured algorithms use the same node and edge features with an exception that the implementation of [4] adds context matrices into the existing edge potential.

Features We assume ϕ_1 is composed by a tensor product of instance and label feature functions, given by $\phi_1(x^i, y^i) = \varphi_1(x^i) \otimes y_i$ where $\varphi_1(x^i)$ is the raw node feature depending only on the observed image segment x^i. Similarly $\phi_2(x, y^i, y^j) = \varphi_2(x^i, x^j) \otimes y^{ij}$, where $\varphi_2(x^i, x^j)$ is the raw edge feature depending only on the observation as well, and $y^{ij} := [y^i \ y^j]$. φ_1 and φ_2 are assembled from



(a) raw image (b) grandtruth

Figure 3: Sample images from the synthetic dataset.

φ_1 We extract a texton feature vector [11] from each patch, hence every pixel is represented by a texton vector. The node feature for an object is the empirical mean of the texton vector of pixels. The raw node feature $\varphi_1(x^i) = [1 \ \varphi_1(x^i)]$.

φ_2 We use the mean of the boosted texton probability density [11] of all interior and boundary pixels of the objects as their edge feature. The raw edge feature $\varphi_2(x^i, x^j) = [1 \ \varphi_2(x^i) \ \varphi_2(x^j)]$.

3.1. Synthetic Dataset

To show that our approach is capable of dealing with contextual information in images, we build a synthetic image dataset containing 100 images and 5 object categories: *sky*, *land*, *water*, *horse*, *fish*. A gallery of sample images are displayed in Figure 3.

In this dataset, the two object categories, namely horses and fish appear in random positions while satisfying the following contextual constraints: fish always stay in the water, and horses stays on land most of time but occasionally touch the water. There are also the three background categories; namely sky, land and water. Each has a different but fixed intensity value, hence these categories are easy separated by almost any reasonable classifier. The challenge comes from the horse and the fish categories, as they are random samples drawn from Gaussian distributions with the *same* variance (0.01) but slightly *different* mean values (100 for horse and 101 for fish. The gray level range is $[0, 255]$).

We use 20-fold cross-validation, and for simplicity of demonstration, we use a simple node feature, namely the empirical mean of pixel intensity of each object. The edge feature is the absolute difference of two adjacent objects' pixel intensities tensored with the classes. In this dataset, we notice that even using very small portion of data for training, all the comparison algorithms can achieve reasonable performance, hence we use only 5% for training and 95% for testing. Table 1 compares the accuracies of the various methods. Our proposed approach is able to achieve the highest average accuracy, 97.06%, on test data.

[1] http://research.microsoft.com/en-us/projects/objectclassrecognition/ .

[2] http://www.cs.toronto.edu/∼hexm/data/corel_subset.mat.

[3] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

SVM-L	SVM-RBF	MRF	CRF	SVM-DP
78.12	76.81	78.75	$>$ 10 days	**97.06**

Table 1: A comparison of accuracies on the synthetic dataset using 20-folds cross-validation. The exact inference in fully connected graph in CRF is very expensive when the tree-width is big. As result, CRF can't finish running after 10 days.

T vs. P	sky	land	water	horse	fish
sky	1.00	0.00	0.00	0.00	0.00
land	0.00	1.00	0.00	0.00	0.00
water	0.05	0.00	1.00	0.00	0.00
horse	0.00	0.00	0.00	0.55	0.45
fish	0.00	0.00	0.00	0.36	0.64

Table 2: Confusion matrix (True vs. Predict) of SVM-L on synthetic dataset.

T vs. P	sky	land	water	horse	fish
sky	1.00	0.00	0.00	0.00	0.00
land	0.00	1.00	0.00	0.00	0.00
water	0.05	0.00	0.91	0.04	0.00
horse	0.00	0.00	0.00	0.99	0.01
fish	0.00	0.04	0.00	0.01	0.95

Table 3: Confusion matrix on synthetic dataset using SVM-DP (our approach).

The sky, land and water are relatively easy to classify because their intensity values are distinct. In fact, SVMs do not make any mistakes in predicting these three categories, as shown in Table 2. However, the fish and horse categories turn out to be very difficult for SVMs: Table 2 shows that 45% of horses are misclassified as fish and 36% of fish are misclassified as horses. On the other hand, the structured learning algorithms, including MRF, CRF and the proposed approach, are all capable of capturing the characteristics of the objects and the relationships between pairs of objects, which helps achieve reasonable predictions, even when the node features are less informative. Therefore, they outperform SVMs in this task. For example, Table 3 shows our approach does very well in horse and fish. Note that a small error rate occurs in water category. This is due to small training size — we only use 5 images to train. Our experiment shows that by increasing the training size, the error on water will vanish.

3.2. Real Datasets

We conducted separate sets of experiments on two real world image datasets. The first is the MSRC Object categorization dataset, where we specifically used the scenery 1 portion that contains 30 images and 7 classes: *building*, *grass*, *tree*, *cow*, *horse*, *sheep* and *sky*. The second is the Corel dataset, which has 100 images and 7 classes: *hippo*, *polar bear*, *water*, *snow*, *vegetation*, *ground*, and *sky*.

Here each image contains one or multiple objects. The boundary (or area) of the object is obtained using a segmentation procedure, and our task is to recognize the category of each object. After random permutation of the images, each dataset is split to have roughly 60% for training and 40% for testing. We remove the areas which are less than 1.5% of the whole image, since these are too small to be meaningful for recognition.

Data	SVM-L	SVM-RBF	MRF	CRF	SVM-DP
MSRC	86.21	82.76	86.21	93.10	**96.55**
Corel	58.62	65.52	87.93	86.21	**98.28**

Table 4: Accuracy comparison on Real dataset.

T vs. P	hippo	polar	water	snow	vege	ground	sky
hippo	1.00	0.00	0.00	0.00	0.00	0.00	0.00
polar	0.00	0.86	0.14	0.00	0.00	0.00	0.00
water	0.00	0.00	1.00	0.00	0.00	0.00	0.00
snow	0.00	0.00	0.00	1.00	0.00	0.00	0.00
vege	0.00	0.00	0.00	0.00	1.00	0.00	0.00
ground	0.00	0.00	0.00	0.00	0.00	1.00	0.00
sky	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table 5: Confusion matrix on the Corel dataset using SVM-DP (our approach).

Table 4 displays the test accuracy on both the MRSC and Corel datasets. Our approach (SVM-DP) clearly outperforms the others. As expected, the structured learning algorithms again outperform the *i.i.d* methods (SVMs), by leveraging the edge features ϕ_2 of the induced graph and making decision jointly. Polar bear is sometimes confused with water (see Table 5) due to their texton features appear very similar in feature space.

4. Outlook and Discussion

In this paper, we propose a principled discriminative method for image object recognition, developed around the large margin principle. In particular, we developed a novel exact inference algorithm that can obtain the optimal assignment of (5) and (6), even in the multi-class scenario. Experiments on synthetic and real-world datasets demonstrate the excellency of the proposed approach.

For future work, a natural extension is to incorporate pixel-wise image segmentation into the proposed framework. Also designing good heuristic procedures to speed up the inference in large tree-width without compromising the performance is worth investigating.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004. 1, 4

[2] J. Corso, A. Yuille, and Z. Tu. Graph-shifts: Natural image labeling by dynamic hierarchical computing. In *CVPR*, pages 1–8, 2008. 1

[3] K. Crammer, Y. Singer, N. Cristianini, J. Shawe-taylor, and B. Williamson. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:2001, 2001. 4

[4] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*. MIT Press, 2008. 1, 2, 4

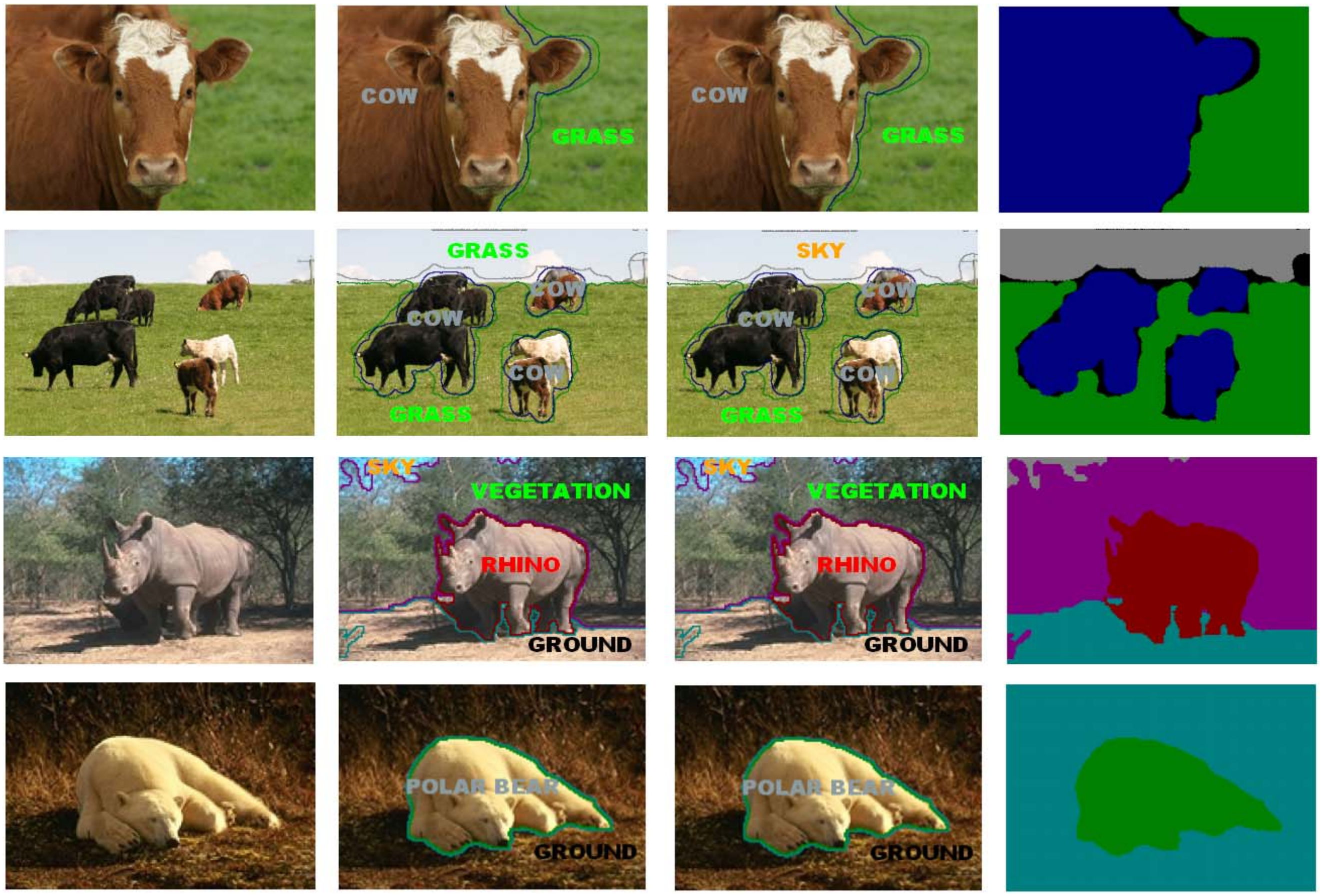


Figure 4: Examples of images from real datasets. **First Column:** The raw images. **Second Column:** The Categorization result of CRF. **Third Column:** The Categorization result of SVM-DP. **Fourth Column:** The ground truth images. Top two images are from MSRC, and the bottom two images are from Corel. Note that CRF misclassifies the sky as grass on the second image in the second row. The figure is best viewed in color.

[5] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Soc., Series B*, 51(2):271–279, 1989. 1, 4

[6] X. He, R. S. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, pages 695–702, 2004. 1

[7] X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, pages 338–351, 2006.

[8] S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 68(2):179–201, 2006. 1

[9] S. Petrov and D. Klein. Discriminative log-linear grammars with latent variables. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *NIPS*, pages 1153–1160. MIT Press, Cambridge, MA, 2008. 1

[10] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 1

[11] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, pages 1–15, 2006. 4

[12] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005. 1, 2

[13] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, UC Berkeley, Dept. of Statistics, September 2003. http://www.eecs.berkeley.edu/ wainwrig/Papers/WaiJorVariational03.pdf. 1

[14] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, pages 239–269, 2003. 1

[15] L. Zhu, Y. Chen, and A. Yuille. Unsupervised learning of a probabilistic grammar for object detection and parsing. In *NIPS*. MIT Press, 2007. 1