Learning and Incorporating Top-Down Cues in Image Segmentation

Xuming He, Richard S. Zemel, and Debajyoti Ray

Department of Computer Science, University of Toronto
{hexm, zemel, debray}@cs.toronto.edu

Abstract. Bottom-up approaches, which rely mainly on continuity principles, are often insufficient to form accurate segments in natural images. In order to improve performance, recent methods have begun to incorporate top-down cues, or object information, into segmentation. In this paper, we propose an approach to utilizing category-based information in segmentation, through a formulation as an image labelling problem. Our approach exploits bottom-up image cues to create an over-segmented representation of an image. The segments are then merged by assigning labels that correspond to the object category. The model is trained on a database of images, and is designed to be modular: it learns a number of image contexts, which simplify training and extend the range of object classes and image database size that the system can handle. The learning method estimates model parameters by maximizing a lower bound of the data likelihood. We examine performance on three real-world image databases, and compare our system to a standard classifier and other conditional random field approaches, as well as a bottom-up segmentation method.

1 Introduction

Shortcomings in the standard bottom-up approach to image segmentation, together with evidence from studies of human vision [1], suggest that prior knowledge about objects facilitates segmentation. Incorporating top-down information faces several challenges: (1) the appearance of objects in a class varies greatly in natural images; (2) shape also varies considerably, and is often corrupted by occlusion; (3) if the number of classes is large, local features may be insufficient to discriminate the class. The images in Figure 1 illustrate some of these difficulties.

In this paper we describe a segmentation scheme that integrates bottom-up cues with information about multiple object categories. Bottom-up cues are used to produce an over-segmentation that is assumed to be consistent with object boundaries but breaks large objects into small pieces. The problem then becomes how to group those segments into larger regions. We propose to use the top-down category-based information to help merge those segments into object components. We define this merging problem as an *image labelling* problem: the aim is to assign labels to the segments so that the segments belonging to the

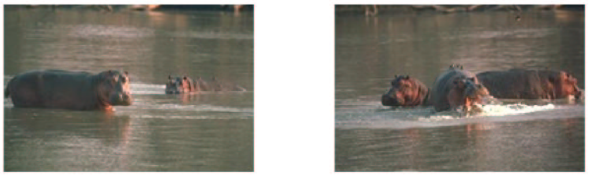

Fig. 1. Lighting and background effects create highly variable appearances of objects. The animal shapes also vary considerably, due to viewpoint changes, articulation, and occlusion, as shown in the hippo images. Discriminating classes based on local cues is often hard, as can be seen by comparing local patches of the two images.

same object category have the same labels. The labels are assigned jointly to an image, taking into account interactions between segments.

We adopt a learning approach to this labelling problem, learning the statistics of the correspondence between image features and labels, as well as the interactions between labels. We further decompose the problem by assigning images to contexts, and again use learning to define the contexts, and to find features that characterize the contexts. The resulting system produces a detailed segmentation of a test image into coherent regions, with a semantic label associated with each region in the image. The key contribution of this work is a modular, adaptive segmentation method that holds the potential for scaling up to large image databases and large numbers of object categories.

The rest of the paper is organized as follows. In Section 2 we describe related schemes for extending bottom-up cues for image segmentation to include top-down information. We then focus on the new combined approach in Section 3. Section 4 describes the learning and labeling algorithms. We compare our model with other approaches in Section 5.

2 Related Work

The primary methodological paradigm we employ is a discriminative learning approach, developed on a database of labeled images. A number of discriminative learning approaches have been developed utilizing labeled images for segmentation and related tasks. For example, conditional random field methods, originally defined for jointly labeling one-dimensional structures such as the parts-of-speech in a text string [2], have been extended to deal with two-dimensional images (e.g., [3]). In the domain of segmentation, Ren and Malik [4] propose a classification model using a number of low- and mid-level cues to define features of proposed segments, and training a classifier to discriminate good segments (based on human segmented natural images) from random ones. Our work aims to extend discriminative approaches to consider information about many different object classes.

Several recent segmentation approaches combine top-down knowledge with bottom-up information. These methods have generally focused on the figure-ground task, attempting to precisely delineate the boundaries of a single object in an image. One approach utilizes a deformable template to determine the boundary suggested by bottom-up cues [5], while another represents object knowledge as pairs of image fragments and their figure-ground labeling from a training set, and then segments a test image by covering it with a set of fragments whose appearances match the data and whose labeling is locally compatible [6]. These methods are highly class specific, working for a particular object type. A recent method extends the patch-based object knowledge to work with a wider variety of objects [7]. The approach proposed in this paper can be seen as attempting to incorporate more category-level rather than class-specific knowledge; the emphasis is on grouping image pixels into various categories across the whole image rather than a precise specification of a single figure-ground boundary.

The core of our approach is an image labelling method, in which the objective is formulated as classifying all pixels of an image using some vocabulary of labels. Recent related methods employ class-specific detectors, and jointly make use of information across objects to form a parse tree of an image [8], or to simultaneously detect multiple objects from a common context [9]. Methods that utilize image caption information to learn associations between image features and keywords are also relevant [10]. The training information provided by captions is considerably weaker than the labeled pixels we utilize; one would expect this to lead to less precision in the test image labels. Finally, the discriminative multi-class learning method proposed in [11], which we compare to our method below, utilizes a similar objective and training information. Their approach involved numerous rounds of stochastic sampling for each training image, and required the labeling to apply to individual pixels. The learning method proposed here is considerably simpler, and operates at a higher level than individual pixels, lending it the potential of scaling up to larger object databases and images.

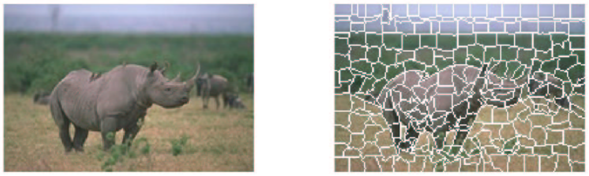

Fig. 2. An original image with 120x180 pixels becomes a 300 super-pixel image, where each contiguous region with a delineated boundary is a super-pixel.

3 Model Architecture

3.1 Super-pixel representation of images

The segmentation process requires that an image is labelled at a pixel level so that the segments fully cover the image. However, a label algorithm operating at the pixel level will typically be highly redundant, due to the similarity between neighboring pixels within each object category. A pixel level model will also be sensitive to, and limited by the resolution of an image. Instead, we build our model based on a higher level image representation than the pixel image, in which a small patch of similar pixels are grouped together to form a larger unit, a *super-pixel* [4]. Segmentation methods based on the bottom-up image cues can be utilized to generate such an image representation by over-segmenting the image into small but coherent regions. When the regions are small enough, their boundaries are usually consistent with the boundaries between object categories, and the potential error induced by such a decomposition will be relatively small. In this paper, we use a variant of the Normalized Cut segmentation algorithm [12], with a specific parameter setting to generate an over-segmentation of an image into super-pixels of a roughly consistent size, and build our approach on this superpixel representation.

The super-pixelization of an image can be viewed as a part of the bottom-up process in our system, while the labelling model discussed in the next section uses both top-down information and image cues to merge those super-pixels into segments with semantic meanings. Figure 2 shows an instance of super-pixel representation of image. Note that even if the size of a super-pixel is small, we significantly reduce the number of units to be labelled, which allows a compact model to be constructed without much sensitivity to the resolution of the image.

We also extract image features from the pixels grouped into super-pixels, providing a better description of input images for labelling. The resulting *image descriptor* of each super-pixel summarizes the statistics of the contained region with respect to features such as texture, edges, and color.

3.2 A Mixture of Conditional Random Fields

Our probabilistic model assigns labels to the super-pixels for a given input image by combining top-down category-based information with image cues. First, we introduce some notation. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$ be the input image, where S is a set of sites associated with the super-pixels and $\mathbf{x}_i$ is the image descriptor from the ith super-pixel. Each super-pixel $\mathbf{x}_i$ will be assigned a label $\mathbf{l}_i$ from a finite label set $\mathcal{L}$. The set of label variables $\{\mathbf{l}_i\}_{i \in S}$ for image $\mathbf{X}$ form a structural output $\mathbf{L}$.

We further decompose the labelling problem by assigning each image to a particular *context*; several recent approaches have demonstrated that the statistics of an image can be used to categorize the scene context (e.g., [13]). Suppose the images in a database can be grouped into several contexts. We denote the context set for the images in a database as $\mathcal{C}$, and c as the context variable for input image $\mathbf{X}$. Our model defines a conditional distribution over the output $\mathbf{L}$ given input $\mathbf{X}$:

$$P(\mathbf{L}|\mathbf{X}) = \sum_{c\in\mathcal{C}} P_M(\mathbf{L}|\mathbf{X}, c) P_G(c|\mathbf{X}) \tag{1}$$

where $P_M(\mathbf{L}|\mathbf{X}, c)$ is a conditional random field (CRF) for the context c, and $P_G(c|\mathbf{X})$ is a gating function which yields the probability distribution of context given the information from image $\mathbf{X}$. We refer to the model in Eqn. 1 as a Mixture of Conditional Random Fields (MoCRF). With CRFs as its mixture components, this model can be viewed as an extension of a mixture of experts model [14] by predicting a structural output from data. Below we describe the component CRF models in detail, followed by the gating function.

3.3 Context-dependent conditional random field

Given a context, the model captures the interactions between the labels of an image using a conditional random field of the labels $P_M(\mathbf{L}|\mathbf{X}, c)$. The random field is defined with respect to a graph G in which the label sites of neighboring super-pixels on the image plane are connected. We denote the neighbors of site i as $N(i)$.

The context-dependent CRF has three types of feature functions in its distribution, encoding the top-down contextual constraint of the labelling at three levels:

$$P_M(\mathbf{L}|\mathbf{X}, c) = \frac{1}{Z_c} \exp\{\sum_i f_a(\mathbf{l}_i, \mathbf{x}_i, c) + \sum_i \sum_{j\in N(i)} f_b(\mathbf{l}_i, \mathbf{l}_j, c) + f_c(\mathbf{L}, c)\}, \tag{2}$$

where $f_a(\mathbf{l}_i, \mathbf{x}_i, c)$ is a feature function describing the compatibility of the local image descriptor $\mathbf{x}_i$ at super-pixel i to a particular label variable $\mathbf{l}_i$; $f_b(\mathbf{l}_i, \mathbf{l}_j, c)$ accounts for pairwise interactions between labels of neighboring sites; and $f_c(\mathbf{L}, c)$ is a feature function for the global statistics of the label field $\mathbf{L}$ under context c. In our model, we implement those feature functions as follows:

(a). Local features $f_a(\mathbf{l}_i, \mathbf{x}_i, c)$. We utilize a classifier that independently predicts the label of every super-pixel to build the local feature function. The classifier provides a label distribution $\Phi_I(\mathbf{l}_i|\mathbf{x}_i, c)$ given input $\mathbf{x}_i$ and context c. The local feature $f_a(\mathbf{l}_i, \mathbf{x}_i, c)$ has the following form:

$$f_a(\mathbf{l}_i, \mathbf{x}_i, c, \gamma^c) = \alpha^c \sum_{k\in\mathcal{L}} \delta(\mathbf{l}_i = k) \log \Phi_I(\mathbf{l}_i = k|\mathbf{x}_i, c, \gamma^c), \tag{3}$$

where $\delta(x) = 1$ if x is true and 0 otherwise, α^c is a coefficient for modulating the entropy of the classifier output for context c, and γ^c represents the classifier parameters. The feature function describes the preference of different label configurations given the input. In this paper, we use a multilayer perceptron (MLP) as the classifier which takes color, edge magnitude and texture information from the ith super-pixel's descriptor as the input. Note that these feature functions may be able to find local image features that uniquely characterize a particular class, such as the combination of color, texture, and edges in a rhino's horn.

(b). Pairwise features $f_b(\mathbf{l}_i, \mathbf{l}_j, c)$. The pairwise feature functions exploit the local interactions between labels of neighboring super-pixels. We use a pairwise feature with a linear form in this model:

$$f_b(\mathbf{l}_i, \mathbf{l}_j, c) = \sum_{k \in \mathcal{L}} \sum_{k' \in \mathcal{L}} \delta(\mathbf{l}_i = k)\delta(\mathbf{l}_j = k') \log \Psi^c_{ij}(k, k'), \tag{4}$$

where Ψ^c_{ij} is a $|\mathcal{L}| \times |\mathcal{L}|$ compatibility matrix between label $\mathbf{l}_i$ and $\mathbf{l}_j$. The compatibility matrix incorporates both the statistics of neighboring label configurations and image descriptor information; it is defined as follows:

$$\Psi^c_{ij}(k, k') = \begin{cases} (1 - P^b_{ij}) \exp(\theta^c_{k,k'}) & k = k' \\ P^b_{ij} \exp(\theta^c_{k,k'}) & k \neq k' \end{cases} \tag{5}$$

where $\theta^c_{k,k'}$ is a scalar parameter for the compatibility of label values k, k' in context c. This formulation incorporates boundary information provided by a separate boundary classifier [15]: P^b_{ij} is the boundary probability between super-pixel i and j, which modulates the label pair compatibility, implementing the intuitive notion that the compatibility of labels of neighboring sites depends on the presence of a boundary between them. For example, one would expect that the likelihood of neighboring labels taking on the same value would decrease if there is a boundary between them, while the compatibility of taking on different values would decrease if no boundary exists. Therefore, $f_b(\mathbf{l}_i, \mathbf{l}_j, c)$ can be viewed as a data-dependent feature function specifying the regional context of labels.

(c). Global features $f_c(\mathbf{L}, c)$. The global feature function provide a coarse level constraint for the label configuration of the random field. In our model, the global features constrain the overall image label distribution to conform to a typical, average label distribution that characterizes the relative proportion of the various labels in a specific context. Assuming this average label distribution is $\mu^c = (\mu^c_1, ..., \mu^c_{|\mathcal{L}|})$ for a given context c, we define a global feature that maximizes the match between the actual label distribution and the distribution μ^c:

$$f_c(\mathbf{L}, c) = \beta^c \sum_i \sum_{k \in \mathcal{L}} \delta(\mathbf{l}_i = k) \log \mu^c_k, \tag{6}$$

where β^c is the weighting coefficient. This feature function is equivalent to the negative Kullback-Leibler divergence between the image label distribution and the target distribution for the given context. Note that this feature provides a global bias to the single node potential in the conditional random field.

3.4 Gating function $P_G(c|\mathbf{X})$

The gating function is specified by a context classifier which generates a distribution of context c given an input image. The inputs to the classifier are the aggregate statistics of the image descriptors, including color, edge density and texture information. We use a multilayer perceptron as the context classifier in this model.

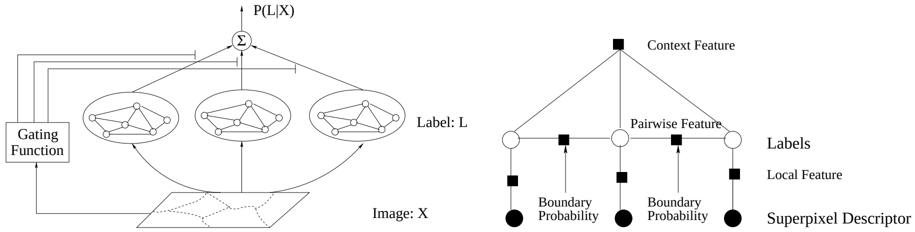

Fig. 3. Graphical model representation. **Left**: The superpixel descriptors are input to context-specific processing, with the gating function modulating the relevance of each context to a given image. **Right**: The context-specific processing combines local information based on super-pixel descriptor and specific label compatibility; pairwise interactions between labels of neighboring sites, modulated by the boundary probability; and global bias provided by the context-specific average label distribution.

3.5 Model summary

To summarize, our model has the following form:

$$P(\mathbf{L}|\mathbf{X}) = \sum_c \frac{P_G(c|\mathbf{X})}{Z_c} \exp\{\sum_{i,j} \mathbf{l}_i^T \log \Psi_{ij}^c \mathbf{l}_j + \alpha^c \sum_i \mathbf{l}_i^T \log \Phi_I + \beta^c \sum_i \mathbf{l}_i^T \log \mu^c\} \tag{7}$$

where the label variable $\mathbf{l}_i$ is represented as a vector with $|\mathcal{L}|$ elements, in which the kth element is 1 and the other elements are 0 when $\mathbf{l}_i = k$. Figure 3 provides an overview of the main components of the model. Note that the final label distribution can readily be used to define a segmentation of the image into coherent regions, where a segment corresponds to each contiguous group of pixels that are assigned the same label.

4 Image Labeling and Parameter Estimation

4.1 Inference and learning criterion

Given a new image $\mathbf{X}$, we predict its labelling based on the Maximum Posterior Marginals (MPM) criterion:

$$\mathbf{l}_i^* = \arg\max_{\mathbf{l}_i \in \mathcal{L}} \sum_{c \in \mathcal{C}} P_M(\mathbf{l}_i|\mathbf{X}, c) P_G(c|\mathbf{X}), \tag{8}$$

where the marginal label distributions of each super-pixel, $P_M(\mathbf{l}_i|\mathbf{X}, c)$, are computed by applying loopy belief propagation to every context-dependent CRF.

Given a set of labeled image data $\mathcal{X} = \{(\mathbf{L}^n, \mathbf{X}^n)\}$, we estimate the model's parameters based on the Conditional Maximum Likelihood criterion, that is,

$$\hat{\Theta} = \arg\max_{\Theta} \sum_n \log P(\mathbf{L}^n|\mathbf{X}^n), \tag{9}$$

where Θ denotes all the parameters in the model. Treating the context variable c as missing data, we could apply the EM algorithm to the learning problem. However, due to the partition functions in the mixture components, the posterior distribution $q(c|\mathbf{L}^n, \mathbf{X}^n)$ is intractable. Instead, we define a new cost function which is a lower-bound of the conditional data likelihood:

$$Q = \sum_n \sum_c P_G(c|\mathbf{X}^n) \log P_M(\mathbf{L}^n|\mathbf{X}^n, c). \tag{10}$$

Note that $Q \leq \sum_n \log[\sum_c P_G(c|\mathbf{X}^n) P_M(\mathbf{L}^n|\mathbf{X}^n, c)] = \sum_n \log P(\mathbf{L}^n|\mathbf{X}^n)$.

4.2 A modular training approach

Given the cost function in Eqn. 10, we can compute its gradient and estimate all the parameters using a gradient ascent method. However, training all parameters together becomes difficult in practice when we have a large label set, and large image database. In this work, we propose a modular approach to estimate the parameters, such that many components are learned separately and are then merged into the full system in a consistent way. This learning procedure may not produce an optimal system ultimately, but the approach leads to a more efficient learning process, capable of scaling up to large datasets.

The learning procedure is carried out as follows: (1). We cluster the training data, where each training image is represented by its aggregate label distribution, and define each cluster as a context. The clustering divides the training data into subsets, such that each image corresponds to a specific context. (2). Given this division of training data, we can train the gating function that predicts which context an image is in given its image features. (3). Within each subset, we estimate the parameters $\{\gamma^c\}$ of each context-dependent image classifier to independently predict the label distribution given the super-pixel descriptors as input. (4). Finally, we combine these components and jointly learn the remaining parameters in the model (the coefficients $\{\alpha^c, \beta^c\}$ and the compatibility parameters θ^c) by maximizing the cost function in Eqn. 10.

More specifically, in step 1, the clustering method is based on a mixture of unigram model for the labels: $P_u(\mathbf{L}) = \sum_c \prod_i P_u(\mathbf{l}_i|c)P_u(c)$, which we learn using the EM algorithm on the training data set. The conditional probability $P_u(\mathbf{l}_i|c)$ acts as the cluster center, or the prototype label distribution in context c, and is thus used as μ^c in the global feature function. In step 2, given the mixture of unigram model, we can compute the cluster responsibility of every image. Those responsibilities are used as training targets for the gating function $P_G(c|\mathbf{X})$. Step 3 can occur in parallel with step 2, as by sampling the responsibilities, we can form the context-dependent subsets from the training data, and learn the parameters γ^c of the local feature functions on the appropriate subsets.

Finally, in step 4, after parameters of the local and global feature functions as well as the gating function have been learned, we merge them into the model and optimize the remaining parameters with respect to the cost function. Note that the context-dependent CRFs are log-linear models with parameters $\{\theta^c, \alpha^c, \beta^c\}$, which can be estimated by gradient ascent:

$$\mathit{\Delta}\theta^c \propto P_G(c|\mathbf{X}^n) \sum_n \sum_{i,j \in N(i)} (\mathbf{l}_i^n \mathbf{l}_j^{nT} - \langle \mathbf{l}_i \mathbf{l}_j^T \rangle_{P_M(\mathbf{l}_i, \mathbf{l}_j | \mathbf{X}^n, c)}) \tag{11}$$

$$\mathit{\Delta}\alpha^c \propto P_G(c|\mathbf{X}^n) \sum_n \sum_i (\mathbf{l}_i^{nT} - \langle \mathbf{l}_i^T \rangle_{P_M(\mathbf{l}_i | \mathbf{X}^n, c)}) \log \mathit{\Phi}_I(\mathbf{l}_i | \mathbf{x}_i^n, c) \tag{12}$$

$$\mathit{\Delta}\beta^c \propto P_G(c|\mathbf{X}^n) \sum_n \sum_i (\mathbf{l}_i^{nT} - \langle \mathbf{l}_i^T \rangle_{P_M(\mathbf{l}_i | \mathbf{X}^n, c)}) \log \mu^c. \tag{13}$$

To avoid overfitting, we add a Gaussian prior on the parameters, which is equivalent to weight decay during learning. As the CRFs are defined on loopy graphs with intractable partition functions, the marginal distributions of the label variables in the gradient updates cannot be computed exactly. In this work, we approximate them by applying the loopy belief propagation algorithm. An alternative approach is to apply contrastive divergence [16] to each component CRF. The empirical results show that both of these approaches obtain similar and satisfactory performance in our model; below we report results using loopy belief propagation.

5 Experimental Evaluation

5.1 Data sets

We applied our model to three different real data sets. In order to compare our method with an alternative approach, we utilized the two datasets used in our mCRF work [11], and used the same training and testing split as in that work. The first dataset is the Sowerby database, including a set of color images of outdoor scenes and their associated labels. The data set has a total of 104 images with 7 labels: 'sky', 'vegetation', 'road marking', 'road surface', 'building', 'street objects' and 'cars'. 60 of these images are used for training and the remaining 44 for testing. The second dataset is a 100-image subset of the Corel image database, consisting of African and Arctic wildlife natural scenes. It also has 7 classes: 'rhino/hippo', 'polar bear', 'vegetation', 'sky', 'water', 'snow' and 'ground'; and has a train/test split of 60/40.

To explore the scaling potential of our approach, we defined a third dataset by expanding this Corel dataset to include 305 manually labelled images with 11 classes: 'rhino/hippo', 'tiger', 'horse','polar bear', 'wolf/leopard', 'vegetation', 'sky', 'water', 'snow', 'ground' and 'fence'. The training set includes 229 randomly selected images and the remaining 76 are used for testing. We call this extended Corel data set CorelB, and refer to the smaller one as CorelA in the following sections.

Again, for comparison purposes, we use the same set of basic image features as in [11], including color, edge and texture information. For the color information, we transform the RGB values into CIE Lab* color space, which is perceptually uniform. The edge and texture are extracted by a set of filter-banks including a difference-of-Gaussian filter at 3 different scales, and quadrature pairs of oriented even- and odd-symmetric filters at 4 orientations $(0; \pi/4; \pi/2; 3\pi/4)$ and 3 scales. We also include the vertical and horizontal position of each pixel. Thus each pixel is represented by a 32 dimensional image feature vector. For super-pixels, we compute the normalized histograms of those image features extracted from the pixels in each super-pixel.

5.2 Model specification

We use the normalized cut segmentation algorithm to build the super-pixel representation of the images, in which the segmentation algorithm is tuned to generate more than 300 segments for each image. Segments smaller than a minimum size (6 pixels) are merged into the neighboring super-pixels. This yields approximately 300 super-pixels per image on average. The boundary information is extracted using the algorithm in [15]. To avoid underflow, we convert the raw output of boundary probability into interval $[0.1, 0.9]$ by an affine transform.

The number of contexts in our experiments is specified based on the complexity of data set. For Sowerby and CorelA data sets, we use 2 contexts in clustering, and for CorelB, we use 4 contexts. The model selection issue is not explored here, and is left to future work.

The gating function is a MLP with 25 hidden units. It takes the normalized histograms of the image features in each image as input. We use 20 bins for each image feature. To avoid overfitting, the MLP is trained with Gaussian priors on weights. The local classifiers are also MLPs with 30 hidden units, using the histograms of the image features in each super-pixel as input. They are trained with cross-validation.

We compare our approach with a simple pixel-wise classifier and a CRF model. These comparisons provide insight into the utility of the pairwise compatibilities (CRF vs. classifier) and the contexts (MoCRF vs. CRF). The pixel-wise classifier is a MLP with one hidden layer, taking image features from a 3×3 window centered at each pixel and predicting the pixel's label. The CRF uses context-independent local feature and pairwise feature functions. The feature functions have the same form as our model. The distribution of label configuration $\mathbf{L}$ defined by the CRF has the following form:

$$P_{CRF}(\mathbf{L}|\mathbf{X}) = \frac{1}{Z}\exp\{\sum_{i,j}\mathbf{l}_i^T \log \Psi_{ij}\mathbf{l}_j + \alpha\sum_{i}\mathbf{l}_i^T \log \Phi_I(\mathbf{l}_i|\mathbf{x}_i)\} \qquad (14)$$

where Φ_I is a local classifier trained separately on all the data and Ψ_{ij} is the compatibility function including boundary information. We trained the CRF model using the pseudo-likelihood algorithm, and tested its performance using the same MPM criterion where the marginal distribution is calculated by the loopy belief propagation algorithm.

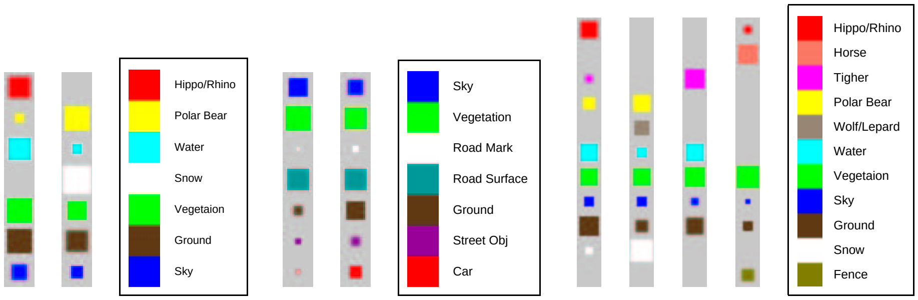

Fig. 4. The learned prototype label distribution for each of the three datasets: CorelA, Sowerby, and CorelB, is shown, with its associated key. See text for discussion.

5.3 Results

We clustered the training images in each dataset as described above, yielding 2 clusters for the CorelA and Sowerby datasets, and 4 clusters for CorelB. In Fig. 4, we visualize the typical label distributions of the contexts from all three datasets. Note that these distributions usually have semantic meaning which is easy to interpret. For instance, the contexts in CorelA dataset represent the tropical and arctic environments, while the Sowerby dataset contexts are rural and suburban areas. CorelB dataset has 'tropic','field','jungle' and 'arctic' as its contexts. Given the context settings, we trained a context classifier as the gating function for each dataset. To evaluate those context classifiers, we use the largest cluster responsibility as the target context, and compute the accuracy of the classifier output. Based on that metric, the context classifiers we trained achieve 82%, 92% and 85% accuracy on Sowerby, CorelA and CorelB, respectively.

The performance of MoCRF is first evaluated according to the label error metric on the pixel level, i.e., the percentage of incorrectly labelled pixels. We compared the performance of MoCRF to a simple pixel-wise classifier (P_Class), the super-pixel classifier in MoCRF considered alone (S_Class), and the CRF model over three datasets. We also include the performance of mCRF on the Sowerby and CorelA datasets [11]. The correct classification rates on the test sets of three datasets are shown in Figure 5A.

We can see that the super-pixel based classifiers alone provide a significant improvement over the pixel-wise classifiers. Built on the the same bottom-up cues, our model also has better performance over the super-pixel classifier and the conventional CRF model. Furthermore, it provides a slighter better performance than the mCRF model [11]. Note that our MoCRF model has a much simpler structure than the mCRF model: for the Sowerby and CorelA datasets, MoCRF has approximately 300 label variables, (equal to the number of super-pixels), no hidden variables, and approximately 120 parameters for training excluding the classifiers; while mCRF has about 2×10^4 label variables, 10^3 hidden variables and 10^3 free parameters. Learning is therefore quite slow in mCRF, and the model has poor scaling properties. Thus, although we only match this

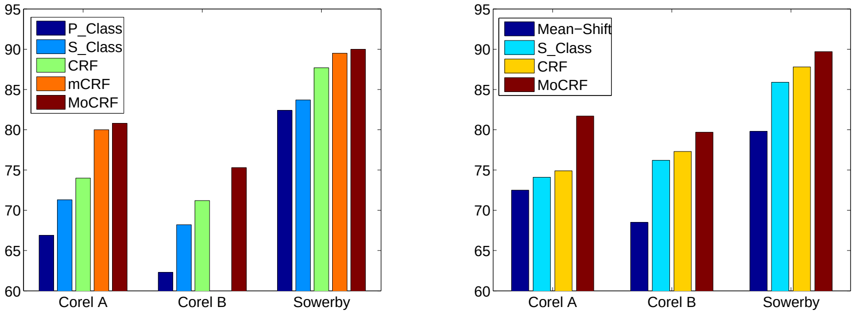

Fig. 5. A (left): Classification rates; B (right): Segmentation accuracy for the models.

earlier model in terms of classification accuracy, our model can be applied to the problems with a considerably larger set of labels and larger image sizes.

We compare the performance of the pixel-wise classifier, our model, and Mean-Shift segmentation in Figure 5B. We tune the parameters of Mean-Shift such that it generates the best results according to the manual labeling for a small set of randomly chosen images. The performance is measured according to a second metric used for evaluation, a segmentation metric which computes the percentage of pixel pairs that are correctly segmented. To reduce the computational burden, we randomly sampled 10% pixels from each image to estimate the accuracy. Again, we can see that our model obtains better results by adding top-down category information, and multi-level contextual constraints.

We also show the outputs of these methods on some test images in Figure 6. The figure shows the approaches based solely on low-level cues can be fooled, such that some single objects in the images are split. MoCRF works much better on those images by integrating the super-pixel representation and mixture of CRF framework. Note that the super-pixelization will cause some errors which cannot be corrected by the top-down information. Also, the model cannot use global spatial configuration to correct errors since no geometric information is included in the global feature functions.

6 Discussion

In this paper we have presented a discriminative framework that integrates bottom-up and top-down cues for image segmentation. We adopt a labelling approach to provide some purchase on the segmentation problem. A chief contribution of our model with respect to segmentation is the resulting extension of top-down cues to include a considerably wider range of object classes than earlier methods. The proposed framework is modular, in that images in a database are classified as to their context, and separate processes are learned for the different contexts. This modularity presents some promise of the system extending to

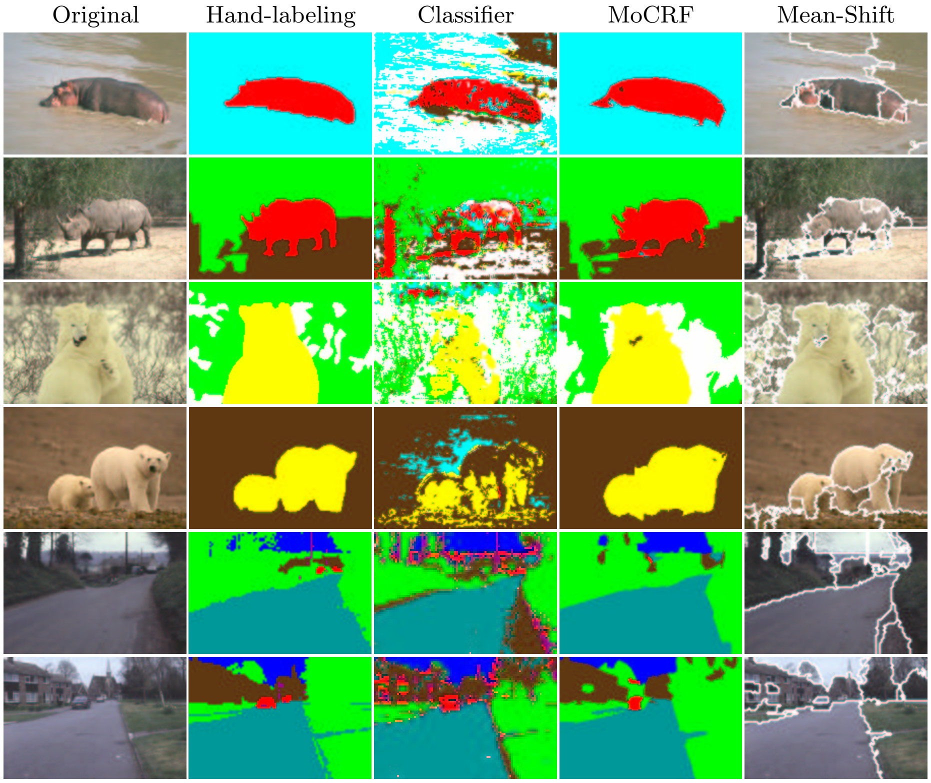

Fig. 6. Some labeling results for the Corel (4 top rows) and Sowerby (2 bottom rows) datasets, using the pixel-wise classifier, CRF, MoCRF, and Mean Shift segmentation. The color keys for the labels are the same as Fig. 4.

large databases of images. While the top-down cues can be learned in a context-specific manner, the system integrates these with bottom-up cues, which are utilized in several ways: to define super-pixels in an image; to determine probabilities of local boundaries between super-pixels, which are used to constrain and guide labelling; and to enable context classification.

The results of applying our method to three different image datasets suggest that this integrated approach may extend to a variety of image types and databases. The labeling system consistently out-performs alternative approaches, such as a standard classifier and a standard CRF. Its performance matches that of an existing method, which operates at the pixel level and entails a considerably more involved training procedure, one which is unlikely to scale to larger images and image databases. Relative to a standard segmentation method, the segmentations produced by our method are more accurate, even when the standard method is optimized for a given test image. A relatively weak component in our model appears to be the gating function, as the images whose contexts are incorrectly classified contain a disproportionate number of label errors. We are currently evaluating other methods of summarizing the statistics of an image in order to facilitate more accurate context classification. Finally, a limitation of our model concerns its reliance on detailed training data. However, a growing effort to label images (e.g., [17]) should lead to a rapid growth in the volume of available labeled images.

Acknowledgments

We thank BAE Systems for letting us use their Sowerby Image Database. Funded by grants from Communications and Information Technology Ontario and NSERC.

References

1. Peterson, M., Gibson, B.: Shape recognition contributions to figure-ground organization in three-dimensional displays. Cognitive Psychology **25** (1993) 383–429.
2. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th ICML. (2001).
3. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: ICCV. (2003).
4. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV. (2003).
5. Liu, L., Sclaroff, S.: Region segmentation via deformable model-guided split and merge. In: ICCV. (2001).
6. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: Proceedings IEEE Workshop of Perceptual Organization in Computer Vision. (2004).
7. Yu, S., Shi, J.: Object-specific figure-ground segregation. In: CVPR. (2003).
8. Tu, Z., Chen, X., Yuille, A., Zhu, S.C.: Image parsing: Unifying segmentation, detection, and object recognition. International Journal of Computer Vision **63** (2005) 113–140.
9. Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the trees: A graphical model relating features, objects and scenes. In: NIPS-04. (2004).
10. Carbonetto, P., de Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: ECCV. (2004).
11. He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale conditional random fields for image labelling. In: CVPR. (2004).
12. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. PAMI **22** (2000) 888–905.
13. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: Computation in neural systems **14** (2003) 391–412.
14. Jacobs, R.A., Jordan, M.I., Nowlan, S., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3** (1991) 1–12.
15. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color and texture cues. IEEE Trans. PAMI. **26** (2003) 530–549.
16. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation **14** (2002) 1771–1800.
17. Russell, B., Torralba, A., Murphy, K., Freeman, W.: LabelMe: A database and web-based tool for image annotation (2005).