A Hierarchical Conditional Random Field Model for Labeling and Segmenting Images of Street Scenes

Qixing Huang
Stanford University
huangqx@stanford.edu

Mei Han
Google Inc.
meihan@google.com

Bo Wu
Google Inc.
bowu@google.com

Sergey Ioffe
Google Inc.
sioffe@google.com

Abstract

Simultaneously segmenting and labeling images is a fundamental problem in Computer Vision. In this paper, we introduce a hierarchical CRF model to deal with the problem of labeling images of street scenes by several distinctive object classes. In addition to learning a CRF model from all the labeled images, we group images into clusters of similar images and learn a CRF model from each cluster separately. When labeling a new image, we pick the closest cluster and use the associated CRF model to label this image. Experimental results show that this hierarchical image labeling method is comparable to, and in many cases superior to, previous methods on benchmark data sets. In addition to segmentation and labeling results, we also showed how to apply the image labeling result to rerank Google similar images.

1. Introduction

Simultaneous segmenting and labeling images is a fundamental problem in computer vision. It is the core technology of image understanding, content based retrieval and object recognition. The goal is to assign every pixel of the image with an object class label. Most solutions fall into two general categories: parametric methods and nonparametric methods.

Parametric methods [2, 4, 7, 12, 14, 17, 18] usually involve optimizing a *Conditional Random Field* (CRF) model which evaluates the probability of assigning a particular label to each pixel, and the probability of assigning each pair of labels to neighboring pixels. A parametric method usually has a learning phase where the parameters of the CRF models are optimized from training examples, and an inference phase where the CRF model is applied to label a test image.

In contrast to parametric methods, nonparametric methods [10, 15] do not involve any training at all. The basic idea of these methods is to transfer labels from a retrieval set which contains semantically similar images. Nonparametric methods tend to be more scalable than parametric methods because it is easy for nonparametric methods to incorporate new training examples and class labels.

In this paper, we introduce a hierarchical two-stage CRF model which combines the ideas used in both parametric and nonparametric image labeling methods. In addition to learning a global CRF model from all the training images, we group training data into clusters of images with similar spatial object class layout and object appearance, and train a separate CRF model for each cluster. Given a test image, we first run the global CRF model to obtain initial pixel labels. We then find the cluster with most similar images, as shown in Fig. 1. Finally, we relabel the input image by the CRF model associated with this cluster. To effectively compare and extract similar images, we introduce a new image descriptor: the *label-based descriptor* which summarizes the semantic information of a labeled image.

Our approach is motivated by the emergence of large data sets of labeled images, such as Labelme data set [13]. The Labelme data set contains tens of thousands of labeled images. It provides sufficient instances to train classifiers for each type of images with similar spatial layout. In this paper, we focus on images of street scenes which are the most dominant ones in Labelme data set. However, there is no restriction on extending our approach to handling other types of images if more training data is available.

Experimental results show that the hierarchical two-stage CRF model is superior to the global CRF model learned from all training examples. Evaluations on benchmark data sets demonstrate that our approach is comparable, and in many cases, superior to state-of-the-art parametric and nonparametric approaches. In addition, we also show promising results of applying the label-based descriptor to compute images of similar spatial layout and re-rank similar image results from Google Image Search.

1.1. Related Work

Parametric methods. Image labeling by optimizing a CRF model has proven to be the state-of-the-art paramet-

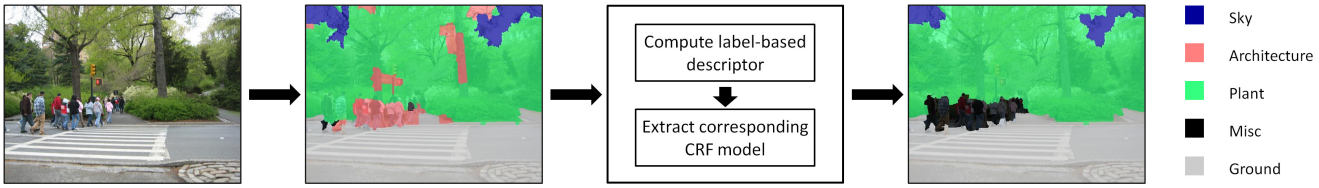

Figure 1: *The pipeline of our hierarchical two-stage CRF model. Given a test image, we first run the global CRF model trained by all training images to obtain initial pixel labels. Based on these pixel labels, we compute the label-based descriptor to find the closest image cluster. Finally, we relabel the test image using the CRF model associated with this cluster.*

ric image labeling method. Traditional CRF models [4, 14] combine unary energy terms, which evaluate the possibility of a single pixel taking a particular label, and pair-wise energy terms, which evaluate the probability of adjacent pixels taking different labels. Although these approaches work well in many cases, they still have their own limitations because these CRF models are only up to second-order and it is difficult to incorporate large-scale contextual information.

Many researchers have considered variants of traditional CRF models to improve their performance. In [6], Kohli et al. proposed to use higher order potentials for improving the labeling consistency. Another line of research focuses on exploring the object class co-occurrence [2, 7, 12, 17, 18]. In particular, Ladický et al. [7] introduced a co-occurrence model that can be efficiently optimized using graph-cuts.

Our approach also falls into the category of parametric image labeling methods, but it has notable differences from previous approaches. Instead of improving the CRF model used in labeling, we try to divide training images into groups of visually and semantically similar images such that traditional CRF models could have better fits on each of them. Note that learning CRF models from clusters of similar images implicitly includes high-level statistics such as high-order potentials and object class co-occurrence.

Nonparametric methods. The key components of nonparametric methods are how to find the retrieval set which contains similar images, and how to build pixel-wise or superpixel-wise links between the input image and images in the retrieval set. In [10], Liu et al. introduced SIFT Flow to establish pixel-wise links. Since SIFT Flow works best when the retrieval set images are highly similar to the input image in spatial layout of object classes, Tighe and Lazebnik introduced a scalable approach that allows more variation between the layout of the input image and images in the retrieval set [15]. Moreover, both methods utilize a MRF model to obtain the final labeling result. The difference is that the approach of [15] works at super-pixel level which turns out to be more efficient than the approach of [10] which is pixel-wised.

Like most nonparametric methods, our approach also extracts information from images with similar spatial layout of object classes and object appearances. However, the fundamental difference is that we pre-compute classifiers for groups of similar images. This gives us freedom in designing suitable classifiers at the learning phase and saves the inference time.

2. Image Labeling Using Standard CRF

In this section, we describe the CRF model used for labeling images of street scenes. This CRF model serves as the building block for the hierarchical CRF model to be introduced in next Section. Our CRF model is similar to the ones used in [4] and [14]. However, we use different features for both the unary and pair-wise potentials which are more suitable for images of street scenes.

As there are many different objects in street scene images, it is quite challenging to classify all of them. In this paper, we choose five distinctive super-classes: sky, architecture, plant, road and misc. Each super-class contains several different objects. Please refer to Table 1 for details.

super-classes	objects
sky	sky, cloud
architecture	building, wall, hill, bridge, $\cdots$
plant	tree, grass, flower,$\cdots$
ground	road, street, sidewalk, water, earth,$\cdots$
misc	people, car, animal, bicycle, $\cdots$

Table 1: *Objects in each super-class.*

The motivation of introducing super-classes is threefold. First, the position, shape and appearance of these five super-classes primarily determine the semantic information of a street scene image. Second, maintaining a small set of super-classes reduces the training time which, on the other hand, enables us to incorporate more information for classification. Third, if necessary, one can still apply another layer of other classification method to distinguish the objects within each super-class.

The training and testing images used in this paper come from the Labelme data set [13]. We manually collect all the labeled images that were taken outdoor and contain at least two labels from the set of *sky, building, tree, street*. As

Figure 2: *We label image at the super-pixel level. (Left) Input image. (Right) Its super pixels. $\mathcal{N}(s)$ denotes the neighboring super-pixels (colored in green) of a super-pixel s (colored in black).*

many images from Labelme data sets are partially labeled, we only keep those images which have at least two different labels. This is because we need different labels within each image to extract contextual information for training. In total we have collected 3303 images. We randomly subdivide these images into a training set, a validation set and a testing set which contain, 1712 images, 301 images and 1100 images, respectively.

Similar to [4], we also over-segment each image and label it at the super-pixel level. We use the method introduced in [19] for computing super-pixels. With $\mathcal{I}_s$ we denote the set of superpixels of image I. We typically use 400 superpixels for one image. For each superpixel $s \in \mathcal{I}_s$, we compute a set of neighboring super-pixels $\mathcal{N}(s)$. $\mathcal{N}(s)$ includes two types of super-pixels: those that are adjacent to s and those that are not adjacent to s but in its neighborhood. The second type of neighboring super-pixels are used to incorporate contextual information at a larger scale (See Fig.2).

The goal in image labeling is to associate each super-pixel s with a label $c_s \in \{sky, architecture, plant, ground, misc\}$. Each super-pixel has a vector of unary features $\mathbf{x}_s$, which includes color, positions and local gradient information. In addition, for each pair of neighboring super-pixels (s, s') where $s' \in \mathcal{N}(s)$, we define a vector of pairwise features $\mathbf{y}_{ss'}$. Then, computing all image labels involves minimizing the following objective function

$$E(\mathbf{c}, \theta) = \sum_{s \in I_s} (E_1(c_s; \mathbf{x}_s, \theta_1) + \sum_{s' \in \mathcal{N}(s)} E_2(c_{s'}, c_s; \mathbf{y}_{ss'}, \theta_2)). \quad (1)$$

where the unary term E_1 measures the consistency between the feature $\mathbf{x}_s$ of super-pixel s and its label c_s, the pair-wise term E_2 measures consistency between neighboring super-pixel labels c_s and $c_{s'}$, given pairwise feature $\mathbf{y}_{ss'}$. The model parameters are $\theta = (\theta_1, \theta_2, \lambda)$ (λ is defined in the term E_2, as shown in Eq. 2).

The objective $E(\mathbf{c}, \theta)$ is optimized using the efficient quad-relaxation technique described in [9]. The resulting labeling $\mathbf{c}$ implicitly defines a segmentation of the input image, with segment boundaries lying between each pair of adjacent super-pixels. In the remainder of this section, we will discuss the unary and pairwise energy terms in details.

2.1. Unary Energy Term

The unary energy term evaluates a classifier. The classifier takes the feature vector $\mathbf{x}_s$ of a super-pixel as input, and returns a probability distribution of labels for that super-pixel: $P(\mathbf{c}|\mathbf{x}, \theta_1)$. Same as in [14], we use JointBoost classifier [16]. Then, the unary energy of a label c_s is equal to its negative log-probability:

$$E_1(c_s, \mathbf{x}, \theta_1) = -\log P(c_s|\mathbf{x}_s, \theta_1).$$

Features. We use HSV color, image location of the super-pixel center and SIFT feature descriptors [8, 10] at scales 2^i where $4 \leq i \leq 7$ to form a basic 517-dimensional feature vector $\mathbf{x}_s$ per super-pixel s. Note that using multiple scales is to account for the varying size of the same object in different images.

Feature vectors. We could take this 517-dimensional feature vector into the JointBoost learning process. However, we found that a better strategy is to augment these feature vectors with those ones that are likely to separate each pair of different classes. A candidate feature vector which tends to separate two classes is the vector connecting two feature vectors with one from each class. Therefore, for each pair of classes c and c', we randomly pick N points $\mathbf{p}_i$ from class c and N points $\mathbf{q}_i$ from class c', and add the first $n (n = 15)$ eigenvectors of

$$M_{cc'} = \sum_{i=1}^{N} (\mathbf{p}_i - \mathbf{q}_i) \cdot (\mathbf{p}_i - \mathbf{q}_i)^T$$

as additional feature vectors. Experimental results show that adding these additional 150 feature vectors from 10 different pairs of classes (out of 5 classes) increases the pixel-wise classification accuracy by 4%.

Fig. 3 shows some classification result of applying the unary classifier. The unary classifier is able to obtain the outline of each object. However, there are still plenty of mis-classified pixels. This is because the unary term does not consider the consistency of labels across neighboring super-pixels and the spatial relationship between different objects. For example, in Fig. 3(a), the unary classifier misclassifies several super-pixels of the architecture class as the sky class. However, this issue can be resolved if we know that sky object is more coherent and a sky object is very unlikely to be under a building object.

2.2. Pairwise Energy Term

The goal of introducing the pair-wise energy term is to take contextual information into account. Similar to the unary energy term, the pairwise energy term also evaluates a

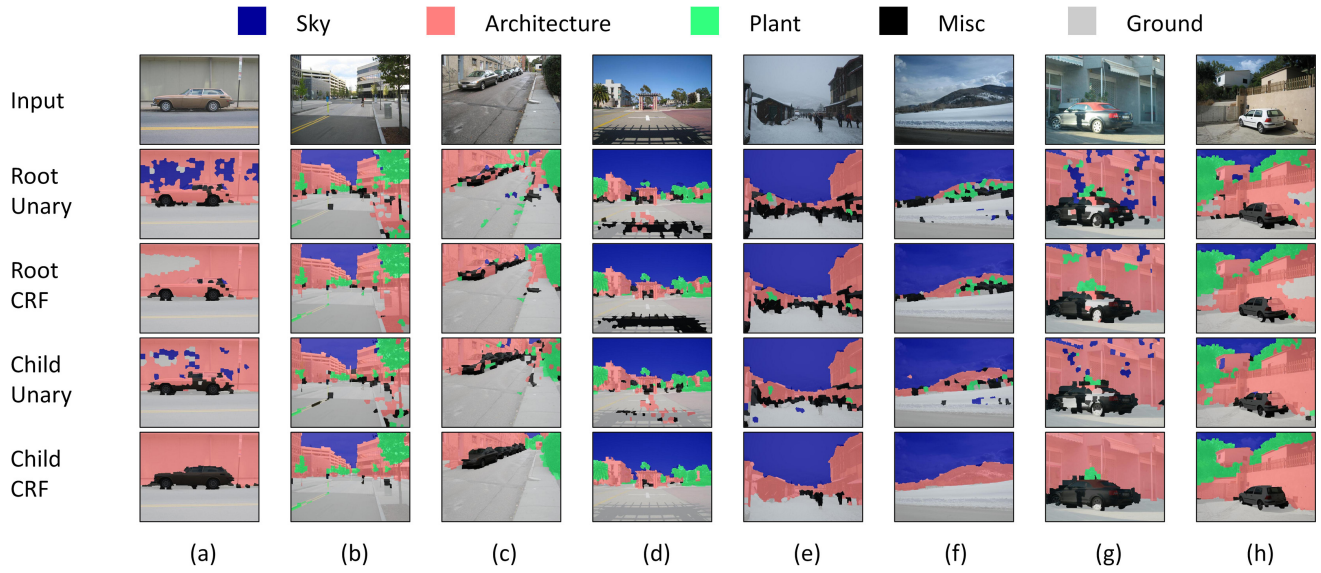

Figure 3: *Representative classification results on testing images from Labelme data set. The hierarchical CRF model yields more accurate and cleaner results than the standard CRF model on various scenes. (1st-row) Input images. (2nd-row to 5nd-row) Classification results using the global unary classifier, the global CRF model, the corresponding closest cluster unary classifier and the closest cluster CRF model, respectively.*

JointBoost classifier. The pairwise energy of a pair of labels c_s and $c_{s'}$ is equal to

$$E_2(c_s, c_{s'}, \mathbf{y}_{ss'}, \theta_2) = -\lambda \log P(c_s, c_{s'}|\mathbf{y}_{ss'}, \theta_2). \quad (2)$$

where λ controls the contribution of the pairwise term. We learn λ using the validation data set by trying different λ and picking the λ with the smallest testing error.

In our implementation, we define the pairwise feature $\mathbf{y}_{ss'} = (\mathbf{x}_s, \mathbf{x}_{s'})$. Again we use the technique described in the previous section to incorporate additional feature vectors for training.

The difference between our pairwise energy term and the one used in [14] is that we actually evaluate the complete distribution of labels of pairs of neighboring super-pixels. This enables us to incorporate the contextual information of different objects.

Fig 3 shows the comparison between using unary classifier and using CRF. It is clear that running the CRF with pairwise term results in much more coherent results.

3. Image Labeling Using Hierarchical CRF

The performance of a CRF model relies on the classification accuracy of the classifiers used to define both the unary and pairwise terms. One possibility of improving the classification accuracy is to use classifiers that are more powerful than Jointboost classifiers. However, these classifiers such as non-linear kernels usually drastically increase the training time. Moreover, they don't utilize the special structure existing in images of street scenes.

Our approach is motivated by the fact that images of street scenes can be divided into clusters of images with similar global layout and appearance. For example, images within one cluster may have sky on the top, buildings in the middle and roads on the bottom. Images within another cluster may have trees on the top and roads on the bottom. If we only take a look at images within each cluster, the object classes have roughly fixed spatial relationship and global appearance. In other words, the complexity and diversity of images within each cluster are reduced such that a standard CRF model is able to fit them very well.

Following the above discussion, we introduce a hierarchical two-stage CRF model for image labeling. In the learning phase, we first train a standard CRF model from all the training images. In the following, we will call this CRF model the global CRF model. Then we subdivide all the training images into clusters of images with similar global layout and appearance. We learn a separate CRF model for each cluster using the images within that cluster.

The key to make this two-stage CRF model work is to cluster images in a semantically meaningful way, which captures the distribution structure of street scene images. We introduce the label-based descriptor which summarizes the semantic information of labeled images, given the initial labeling from the global CRF model.

When applying this hierarchical CRF model to label a new image, we first run the global CRF model to obtain the initial pixel labels of this image. Based on these pixel labels, we then compute the corresponding label-based descriptor

and use it to find the closest cluster. Finally, we run the CRF model associated with that cluster to relabel the input image.

In the remainder of this section, we will introduce the label-based descriptor and how to use it for image clustering.

3.1. Label-based Descriptor

In this section, we consider the problem of computing a compact representation, called label-based descriptor, for labeled images. By labeled image, we mean each pixel is labeled as one of the k object classes $\mathcal{L} = \{c_k\}$. Note that $k = 5$ in this paper.

The semantic information of an image is captured by the position, appearance and shape of each object in this image. Although it is easy to extract this semantic information from a labeled image, we have to summarize it in a compact way. Furthermore, as every image labeling method is subject to classification errors, another issue of designing label-based descriptor is how to make it robust against errors in pixel labels.

To encode the positional information of each object class in a given image I, we subdivide I into a uniform $n_p \times n_p$ grid. Within each grid cell g_{ij}, we evaluate the distribution p_{ijk} of each object class $c_k \in \mathcal{L}$. We collect all the cell coverage information into a vector $\mathbf{d}_I^p$ of length Kn_p^2. Picking the grid size value n_p is a tradeoff between descriptiveness and stability of this representation. A big n_p would make $\mathbf{d}_I^p$ capture the positional information more precisely, while a small n_p would make $\mathbf{d}_I^p$ less sensitive to image displacement and classification errors. For all the experiments listed in this paper, we set $n_p = 4$.

Similar to the positional information, we encode the appearance information by evaluating the mean color $\overline{\mathbf{c}}_{ijk} = (\overline{r}_{ijk}, \overline{g}_{ijk}, \overline{b}_{ijk})$ of each object class c_k within each cell g_{ij}. To stabilize the mean color statistics, we scale each mean color $\overline{\mathbf{c}}_{ijk}$ as $p_{ijk}\overline{\mathbf{c}}_{ijk}$. Again, all mean colors $\overline{\mathbf{c}}_{ijk}$ are collected into a vector $\mathbf{d}_I^c$ of length $3Kn_p^2$.

Finally, we write down the label-based descriptor of image I as $\mathbf{d}_I = (\mathbf{d}_I^p, w_c\mathbf{d}_I^c)$ where w_c weighs the importance of the appearance information. We set $w_c = 1$ by default. As we choose $K = 5$ in this paper, the dimension of a label-based descriptor is 320.

3.2. Image Clustering

We cluster the training examples based on their label-based descriptors. For partially labeled images, we run the root CRF to obtain labels for unlabeled pixels. Using the label-based descriptor, each training image is represented as a point in R^N where N is the dimension of the label-based descriptor.

Instead of clustering using the original label-based descriptors, we found that it is better to first reduce the dimensionality of label-based descriptors. Clustering in the projected space reduces the chance of obtaining clusters as isolated points. In our implementation, we use singular value decomposition to reduce the dimension of the label-based descriptors to M (M=2 in this paper). With $\overline{\mathbf{d}}_I$ we denote the projected label-based descriptor of each image I.

We employ the mean-shift clustering algorithm [1] to group images into clusters. Suppose the mean-shift clustering returns K clusters of images $\mathcal{C}_i$ where $1 \le i \le K$. For each cluster $\mathcal{C}_i$, we compute its barycenter $\mathbf{m}_i$ and variance σ_i as

$$\mathbf{m}_i = \frac{\sum_{I \in \mathcal{C}_i} \overline{\mathbf{d}}_I}{|\mathcal{C}_i|}, \quad \sigma_i = \rho\left(\frac{\sum_{I \in \mathcal{C}_i} (\overline{\mathbf{d}}_I - \mathbf{m}_i)(\overline{\mathbf{d}}_I - \mathbf{m}_i)^T}{|\mathcal{C}_i|}\right),$$

where $\rho(A)$ evaluates the maximum eigenvalue of a matrix A.

To ensure that each cluster includes sufficient number of training images, we enlarge each cluster $\mathcal{C}_i$ by including every image I whose association weight to $\mathcal{C}_i$

$$w(I, \mathcal{C}_i) = \exp\left(-\frac{\|\overline{\mathbf{d}}_I - \mathbf{m}_i\|^2}{2\sigma_i^2}\right) < \delta.$$

In this paper, we set $\delta = 0.1$.

For each cluster $\mathcal{C}_i$, we learn a CRF model from its enclosed training images. The association weight $w(I, \mathcal{C}_i)$ of each image I naturally describes how close this image is to cluster $\mathcal{C}_i$, so we weight the labeled instances by $w(I, \mathcal{C}_i)$ when learning the CRF model.

Fig 1 shows the pipeline of image labeling using the hierarchical CRF model, Given an input image I, we first optimize the global CRF to obtain initial pixel labels. We then compute its label-based descriptor using these initial

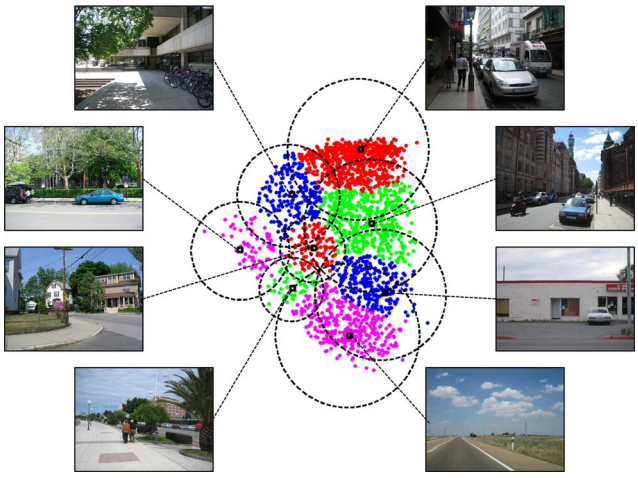

Figure 4: *1702 training images are partitioned into 8 clusters based on label descriptors.*

(a)

	Per.	Misc	Sky	Arch.	Plant	Ground
Misc	15.9	72.3	0.0	5.2	7.4	15.1
Sky	14.2	0.0	97.9	1.5	0.6	0.0
Arch.	30.9	2.3	4.2	81.4	6.9	5.2
Plant	20.0	0.3	2.0	2.6	87.5	7.6
Ground	19.0	3.5	0.3	6.9	3.2	86.1

(b)

	Unary	CRF
He et al.	82.4	89.5
Shotton et al.	85.6	88.6
Global	83.2	86.8
Hierarchical	86.1	90.7

(c)

	Sky	Vertical	Ground
Sky	0.84/0.78	0.16/0.22	0.0/0.0
Vertical	0.06/0.09	0.91/0.89	0.03/0.02
Ground	0.0/0.0	0.07/0.1	0.93/0.90

(d)

	Per.	Misc	Sky	Arch.	Plant	Ground
Misc	3.1	71.8	0.1	9.2	4.4	14.5
Sky	25.0	0.7	92.2	5.7	0.8	0.4
Arch.	48.3	4.3	3.2	83.1	4.2	5.2
Plant	4.8	0.3	3.3	13.3	75.5	7.6
Ground	18.8	6.2	0.4	4.8	3.4	85.2

(e)

	Per.	Misc	Sky	Arch.	Plant	Ground
Misc	4.9	65.6/76.5	0.9/0.5	14.9/10.2	3.5/3.5	15.1/9.3
Sky	14.5	0.4/0.2	93.3/94.0	3.0/2.7	2.6/2.7	0.7/0.4
Arch.	41.5	4.8/3.5	2.4/2.3	81.1/84.0	7.2/5.8	4.5/4.4
Plant	12.3	2.4/2.4	3.3/3.3	11.3/9.3	79.1/81.1	3.9/3.9
Ground	26.8	6.3/4.9	0.8/0.8	3.9/3.4	3.2/3.1	85.8/87.8

Table 2: *Statistics of our method on various data sets. (a) Confusion matrix of the hierarchical CRF model on the Group 3, 7 and 17 of the MSRC data set [14]. (b) Comparison of classification accuracy with He et al. [4] and Shotton et al. [14] on the Sowerby data set. (c) Confusion matrices of our method (Left) and surface context [5] (Right). (d) Confusion matrix of the hierarchical CRF model on category street, insidecity, highway and tallbuilding of the SIFTFLOW data set [15]. (e) Confusion matrices of the standard CRF model (Left) and the hierarchical CRF model (Right) on 1100 testing images from Labelme data set.*

pixel labels and find its corresponding cluster $\mathcal{C}_i$ that has the biggest association weight $w(I, \mathcal{C}_i)$. Finally, we re-label the input image by running the CRF model associated with cluster $\mathcal{C}_i$. Fig 4 demonstrates the clusters generated from our training images.

A critical issue in mean-shift clustering is to set the parameter σ. σ controls the granularity of the clustering. Using a small number of clusters would make the CRF model of each cluster under-fitted while using a large number of clusters would make the CRF model of each cluster over-fitted. Thus, we compute σ such that it results in clusters that maximize the classification accuracy of the hierarchical CRF model. In our implementation, we choose 8 candidate σs that are uniformly sampled between $\frac{d}{16}$ and $\frac{d}{2}$ where d is the diameter of the projected label-based descriptors of all training images. We pick σ as the one that leads to the highest classification accuracy. In our experiments, the optimal value of $\sigma = \frac{d}{8}$.

There are two heuristics that could accelerate the speed of running the hierarchical CRF model. First, as the global CRF model is only used to find the corresponding cluster of each input image, we can use fewer stumps for Jointboost classifiers of both the unary term and the pairwise term. Experimental results show that reducing the number of stumps of both classifiers by $\frac{3}{4}$ only reduces the pixel-wise classification accuracy by 0.05%. Second, when optimizing the CRF model, one can start from the labeled result of the global CRF model. In average, this saves the total running time of optimizing the hierarchical CRF by 15%.

4. Experimental Results

We have evaluated the performance of both the standard CRF model and the hierarchical CRF model on the 1100 testing images described above. Table 2(e) shows the confusion matrices of both methods and Fig. 3 shows some representative results. It is clear that the hierarchical CRF model is superior to the standard CRF model. The pixel-wise classification accuracy of the hierarchical CRF model is 85.7% while that of the standard CRF model is 83.2%.

Although the hierarchical CRF model improves the classification accuracy of all the classes, the improvement on the misc class is significantly larger than improvements on the other four classes. This is because the misc class is more complex than the other four classes in appearance, shape and spatial positions. The standard CRF model, although performs well on the other four classes, is not discriminative enough to classify the misc class. However, looking at the misc class within images of each cluster, since these images already have similar global appearance, the variance in shape, appearance and spatial position appears to be small. Thus, the performance of the CRF model associated with each cluster is much better than that of the standard CRF

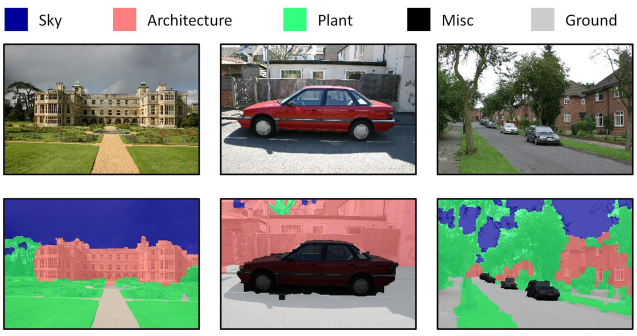

Figure 5: *Example results of our method on the MSRC data sets.*

model.

Evaluation. We have compared our results with those of He et al [4] and those of Shotton et al.[14] on the Sowerby data set used in [4]. As shown in Table. 2(b), the standard CRF model is slightly worse than their methods. This is because the standard CRF model is trained from a wide range of images while the images in the Sowerby data set has restricted global appearance. However, the hierarchical CRF model, which learns a CRF model from the most similar cluster of images, turns out to be better than their methods.

We have also tested our method on the MSRC data set. In this experiment, we only tested three groups of images which are related to images of street scenes (See Fig. 5 for example results). Table 2(a) shows the confusion matrix of the hierarchical CRF model. On these three groups of images, the pixel-wise classification accuracy of our method is 84.5%, which is very competitive to the performance of Shotton et al. [14].

Moreover, we have compared our method with the surface context method [5] which segments an image into three classes: sky, vertical and ground. To make this comparison, we combine the plant class and the arch class as the vertical class. In addition, we include the misc class into the ground class. On the benchmark data set provides in [5], we improved the pixel-wise classification accuracy from [5] by 3.7% (See Table 2(c) for details).

Finally, we evaluated our method on the SIFTFLOW data set [15]. Table 2(d) shows the confusion matrix of our method. Compared with the nonparametric method introduced in [15], our method shows similar results on sky and road classes, and better results on misc, plant and architecture classes (See Fig. 6 for selected results).

Timing. Using our Matlab implementation, labeling a test image of size 450×600 takes about 30 seconds on a machine with 2.2GHZ CPU. On average, computing the superpixels takes 8 seconds, computing the descriptors takes 10 seconds, and solving the CRF takes about 6 seconds each.

5. Application to Computing Similar Images

In this section, we show one application of the label-based descriptor for image comparison. The label descriptor defined in Section. 3 does not consider the shape of each object, which is highly sensible to human eyes. Thus, we augment the label-based descriptor to take this information into account.

To capture the shape information, we evaluate the orientation distribution of the boundaries between each pair of different objects. For stability concern, we use a coarse grid $n_e \times n_e$ ($n_e = 2$ in this paper) and use n_b ($n_b = 4$ in this paper) bins for orientation. For efficiency, we count on the pairs of adjacent superpixels with different object labels. The edge orientation of each such pair is perpendicular to the centroid of the superpixels. All the counts are collected into a vector $\mathbf{l}_b$ of size $\frac{K(K-1)}{2} n_b \times n_e^2 = 160$. The dimension of an augmented label-based descriptor becomes 480. We weight the edge part as 0.01 by default.

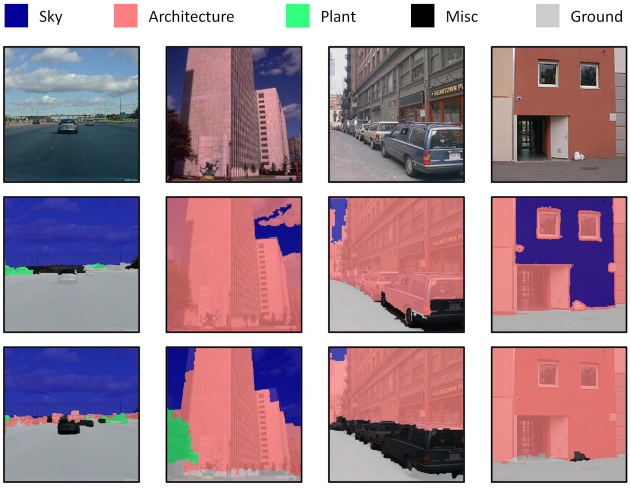

Figure 6: *Comparison between our method (Third row) and the SuperParsing method [15] (Second row) on the SIFT-Flow data set.*

An application of the label-based descriptor is refining image search result. Taking Google Similar Image Search for example, it returns about 400 similar images for a query image. However, in many cases these images are not necessarily similar in appearance to the query image. We rerank these images by distances of their label descriptors to the label-based descriptor of the query image. As shown in Fig. 7, the re-ranking result obtained using label-based descriptor is significantly better than the original rank provided by Google Similar Image Search.

Another possibility of reranking these images is to use the gist descriptor [11]. However, the gist descriptor can only find very similar images. This behavior has been pointed out in [3] where a query image is searched within several millions of images to ensure that the gist descriptor could return similar images. Our label-based descriptor, which extracts image semantics, is able to find similar images in a wide range. We believe that this descriptor is beneficial to several applications such as image completion, similar images browsing and image tag transfering.

6. Conclusion

In this paper, we present an approach to segment images of street scenes into regions of sky, architecture, plant, road and misc. We introduced a novel hierarchical two-stage CRF model based on learning the CRF models from clusters of images with similar object appearance and spatial object class layout. For image clustering, we introduced the label-based descriptor which summarizes the semantic informa-

Figure 7: *Application of the label descriptor to reranking Google similar images. Google similar image returns about 400 images for a query image. We run both the label-based descriptor and the Gist descriptor ro rerank these images. Note that the label descriptor can find more semantically similar images than the Gist descriptor which only finds very similar images.*

tion of a labeled image. We have evaluated our approach on benchmark data sets. The results are comparable to, and in many cases, superior to the state-of-the-art methods.

References

[1] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002. 1957

[2] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, June 2008. 1953, 1954

[3] J. Hays and A. a. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics*, 26(3):4, July 2007. 1959

[4] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages 695–702, 2004. 1953, 1954, 1955, 1958, 1959

[5] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *International Conference of Computer Vision (ICCV)*, volume 1, pages 654 – 661. IEEE, October 2005. 1958, 1959

[6] P. Kohli, L. Ladický, and P. H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *International Journal of Computer Vision*, 82(3):302–324, Jan. 2009. 1954

[7] L. Ladicky, C. Russell, and P. Kohli. Graph Cut based Inference with Co-occurrence Statistics. *ECCV 2010*, pages 1–14, 2010. 1953, 1954

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006. 1955

[9] M. Leordeanu and M. Hebert. Efficient MAP approximation for dense energy functions. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 545–552, New York, New York, USA, 2006. ACM Press. 1955

[10] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, pages 1972–1979, 2009. 1953, 1954, 1955

[11] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. 1959

[12] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in Context. *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct. 2007. 1953, 1954

[13] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, Oct. 2007. 1953, 1954

[14] J. Shotton, J. Winn, and C. Rother. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, 3951:1–15, 2006. 1953, 1954, 1955, 1956, 1958, 1959

[15] J. Tighe and S. Lazebnik. SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. *ECCV 2010*, pages V: 352–365, 2010. 1953, 1954, 1958, 1959

[16] A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 3(v):762–769, 2004. 1955

[17] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. *Proceedings Ninth IEEE International Conference on Computer Vision*, 1:273–280 vol.1, 2003. 1953, 1954

[18] T. Toyoda and O. Hasegawa. Random field model for integration of local information and global information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:1483–1489, August 2008. 1953, 1954

[19] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and Supervoxels in an Energy Optimization Framework. *in European Conference on Computer Vision (ECCV)*, pages 211–224, 2010. 1955