Image Parsing with a Wide Range of Classes and Scene-Level Context

Marian George
Department of Computer Science
ETH Zurich, Switzerland

Abstract

This paper presents a nonparametric scene parsing approach that improves the overall accuracy, as well as the coverage of foreground classes in scene images. We first improve the label likelihood estimates at superpixels by merging likelihood scores from different probabilistic classifiers. This boosts the classification performance and enriches the representation of less-represented classes. Our second contribution consists of incorporating semantic context in the parsing process through global label costs. Our method does not rely on image retrieval sets but rather assigns a global likelihood estimate to each label, which is plugged into the overall energy function. We evaluate our system on two large-scale datasets, SIFTflow and LMSun. We achieve state-of-the-art performance on the SIFTflow dataset and near-record results on LMSun.

1. Introduction

Scene parsing is the assignment of semantic labels to each pixel in a scene image. There are various outdoor and indoor scenes (e.g., beach, highway, city street and airport) that image parsing algorithms try to label. Several systems [3, 7, 6, 9, 11, 15, 18, 19, 20, 24, 27, 28, 33, 36] have been designed to semantically classify each pixel in an image. Among the main challenges which face image parsing methods is that their recognition rate significantly varies among different types of classes. Background classes, which typically occupy a large proportion of the image's pixels, usually have uniform appearance and are recognised with a high rate (e.g., water, mountain, and building). Foreground classes, which typically occupy relatively few pixels in the image, have deformable shapes and can be occluded or arranged in different forms. Such classes (e.g., person, car, and sign) represent salient image regions that often capture the eye of a human observer. However, they frequently represent failure cases with recognition rates significantly lower than those of background classes.

Parametric scene parsing methods [16, 20, 27, 34] have achieved impressive performance on datasets with tens of labels. However, for relatively large datasets with hundreds of labels, it is more difficult to apply these methods due to expensive learning and optimisation requirements.

Recently, nonparametric image parsing methods have been proposed [6, 10, 25, 18, 29, 33] to efficiently handle the increasing number of scene categories and semantic labels. Nonparametric methods typically start by reducing the problem space from individual pixels to superpixels. First, an image set is retrieved, which contains the training images that are most visually similar to the query image. The number of candidate labels for a query image is restricted to the labels present in the retrieval set only. Second, classification likelihood scores of superpixels are obtained through visual features matching. Finally, context is enforced through minimizing an energy function which combines the data cost and knowledge about the classes co-occurences in neighboring superpixels.

A common challenge that faces nonparametric parsing methods is the image retrieval step. While image retrieval is useful for limiting the number of labels to consider, it is regarded as a very critical step in the pipeline [25, 33]. If the true labels are not included in the retrieved images, there is no chance to recover from this error. In [33], it is stated that most of the failure cases occur due to incorrect retrieval.

In this paper, we propose a novel nonparametric image parsing algorithm which targets better overall accuracy, with better recognition rates of less-represented classes. We design a system that is efficient and scalable to a continuously increasing number of labels. We make the following contributions:

1. We improve the likelihood scores of labels at superpixels through combining classifiers. Our system combines the output probabilities of multiple classification models to produce a more balanced score for each label at each superpixel. We learn the weights for combining the scores by applying likelihood normalization method on the training set in an automatic way.

2. We incorporate semantic context in a probabilistic framework. To avoid the elimination of relevant labels that cannot be recovered at later steps, we do not con-

struct a retrieval set. Instead, we use label costs learned from the global contextual correlation of labels in similar scenes to achieve better parsing results.

Our system achieves state-of-the-art per-pixel recognition rates on two large-scale datasets: SIFTflow [18] which contains 2688 images with 33 labels, and LMSun [29] which contains 45576 images with 232 labels.

2. Related Work

Several parametric and nonparametric scene parsing techniques have been proposed. Closely related to our method are the nonparametric systems which aim to achieve a wide coverage of semantic classes. The systems in [28, 33, 6] adopt different techniques for boosting the overall performance of nonparametric parsing. In [28], the authors combine region-parsing with per-exemplar SVM detector outputs. Per-exemplar detectors are used to transfer object masks into the test image for segmentation. Their system achieves impressive improvements in overall accuracy, but at the cost of expensive computational requirements. Calibrating the data terms requires batch offline training in a leave-one-out fashion, which is challenging to scale. [33] and [6] explicitly add superpixels of rare classes into the retrieval set to improve their representation. The authors of [33] filter the list of labels for a test image through an image retrieval step, and rare classes are enriched with more samples at query time. Our system differs in the superpixel classification technique, how we improve the recognition of rare classes, and how we apply semantic context. We promote the representation of foreground classes by merging classification costs of different contextual models, which produces more balanced label costs. We also avoid the bottleneck of image retrieval, and instead rely on global label costs in the inference step.

The usefulness of semantic context has been thoroughly explored in several visual recognition algorithms [6, 10, 11, 18, 23, 25, 33]. In the nonparametric scene parsing systems of [6, 25, 33], context has been used to improve the overall labeling performance in a feedback mechanism. In [6], initial labeling of superpixels of a query image is used to adapt the training set by conditioning on recognized background classes to improve the representation of rare classes. The goal is to improve the image retrieval set by adding back segments of *rare* classes. The system in [25] constructs a semantic global descriptor. Image retrieval is improved through combining the semantic descriptor with the visual descriptors. In [33], context is incorporated through building global and local context descriptors based on classification likelihood maps similar to [17]. Our method is different from these methods in that we do not use context at each superpixel in computing a global context descriptor, but instead we consider contextual knowledge over the image as a whole. We achieve contextually meaningful results through inferring label correlations in similar scene images. We also do not have a retrieval set which we aim to enrich. Instead, we formulate our global context in a probabilistic framework, where we compute label costs over the whole image. Also, our global context is performed online without any offline training. Another image parsing approach which does not rely on retrieval sets is [10], where image labeling is performed by transferring annotations from a graph of patch correspondences across image sets. This approach, however, requires large memory which is difficult to scale for large datasets like SIFTflow and LMSun.

Our approach is inspired from combining classifiers techniques [13] in machine learning, which have been shown to boost the strengths of single classifiers. Several fusion techniques have been successfully used in different areas of computer vision, like face detection [32], multi-label image annotation [22], object tracking [35], and character recognition [12]. However, the constituent classifiers and the mechanisms for combining them are quite different from our framework and the other techniques are only demonstrated on small datasets.

3. Baseline Parsing Pipeline

In this section, we present an overview of our baseline image parsing system, which consists of three steps: feature extraction (sec. 3.1), label likelihood estimation at superpixels (sec. 3.2), and inference (sec. 3.3).

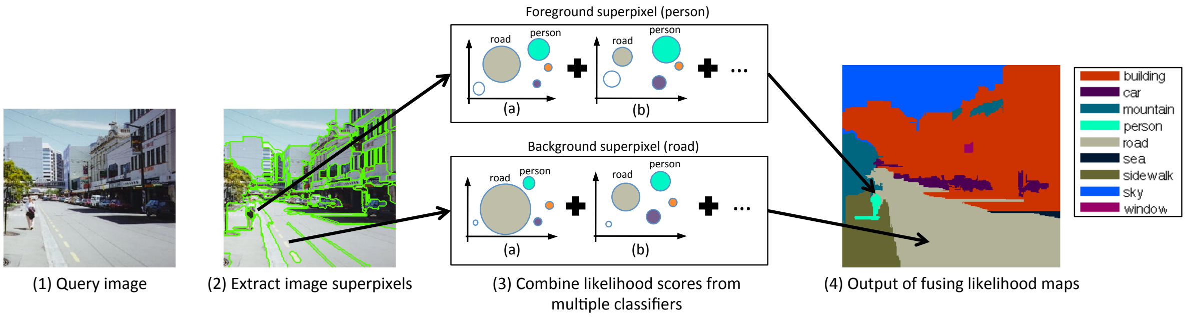
Following that, we present our contributions of improving likelihoods at superpixels and computing label costs for scene-level global context in sections 4 and 5 respectively.

3.1. Segmentation and Feature Extraction

To reduce the problem space, we divide the image into superpixels. We start by extracting superpixels from images using the efficient graph-based method of [8]. For each superpixel, we extract 20 types of local features to describe its shape, appearance, texture, color, and location, following the method of [29]. In addition to these features, we extract Fisher Vector (FV) [21] descriptors at each superpixel using the VLFeat library [31]. We compute 128-dimensional dense SIFT feature descriptors on 5 patch sizes (8, 12, 16, 24, 30). We build a dictionary of size 1024 words. We then extract the FV descriptors and apply PCA to reduce their size to 512 dimensions. Each superpixel is described by a 2202-dimensional feature vector.

3.2. Label Likelihood Estimation

We use the extracted features at the previous step to compute label likelihoods at each superpixel. Different from traditional methods, we do not restrict the potential labels for a test image. We instead compute the likelihood data term for each class label $c \in C$, where C is the total number of



(1) Query image

(2) Extract image superpixels

(3) Combine likelihood scores from multiple classifiers

(4) Output of fusing likelihood maps

Figure 1. Overview of the fusing classifiers approach. Likelihood scores from multiple models (3a) and (3b) are combined to produce the final likelihoods at superpixels. Likelihood scores of foreground classes (e.g. person) are boosted via our combination technique. The unbalanced (skewed) model in (3a) produces biased likelihoods towards background classes (e.g. road). This is reflected in the much larger score (bigger circle) for the road class when compared to the person class and other less-represented classes. For the balanced classifier in (3b), the scores are more balanced and less-represented classes get a higher chance (bigger circle) of being recognized.

classes in the dataset. The normalized cost $D(l_{s_i} = c|s_i)$ of assigning label c to superpixel s_i is given by:

$$D(l_{s_i} = c|s_i) = 1 - \frac{1}{1 + e^{-L_{unbal}(s_i,c)}}, \tag{1}$$

where $L_{unbal}(s_i, c)$ is the log-likelihood ratio score of label c, given by $L_{unbal}(s_i, c) = \frac{1}{2}log(P(s_i|c)/P(s_i|\bar{c}))$, where $\bar{c} = C \setminus c$ is the set of all labels except c, and $P(s_i|c)$ is the likelihood of superpixel s_i given c. We learn a boosted decision tree (BDT) [4] model to obtain the label likelihoods $L_{unbal}(s_i, c)$. For implementation, we use the publicly available boostDT [1] library. At this stage, we train the BDT model using all superpixels in the training set, which represent an unbalanced distribution of class labels C.

3.3. Smoothing and Inference

We formulate our optimization problem as that of maximum a posteriori (MAP) estimation of the final labeling L using Markov Random Field (MRF) inference. Using only the estimated likelihoods in the previous section to classify superpixels yields noisy classifications. Adding a smoothing term $V(l_{s_i}, l_{s_j})$ to the MRF energy function attempts to overcome that issue by punishing neighboring superpixels having semantically irrelevant labels. Our baseline attempts to minimize the following energy function:

$$E(L) = \sum_{s_i \in S} D(l_{s_i} = c|s_i) + \lambda \sum_{(i,j) \in A} V(l_{s_i}, l_{s_j}). \tag{2}$$

where A is the set of adjacent superpixel indices and $V(l_{s_i}, l_{s_j})$ is the penalty of assigning labels l_{s_i} and l_{s_j} to two neighboring pixels, computed from counts in the training set combined with the constant Potts model following the approach of [29]. λ is the smoothing constant. We perform inference using the α-expansion method with the code of [2, 14, 1].

In the next two sections, we present our main contributions of how we improve the superpixel classification step (section 4) and how we incorporate scene-level context to achieve better results (section 5).

4. Improving Superpixel Label Costs

While foreground objects are usually the most noticeable regions in a scene image, they are often misclassified by parsing algorithms. For example, in a city street scene, a human viewer would typically first notice the people, signs and cars before noticing the buildings and road. However, for scene parsing algorithms, foreground regions are often misclassified as being part of the surrounding background due to two main reasons. First, in the superpixel classification step, any classifier would naturally favor more dominant classes to minimize the overall training error. Second, in the MRF smoothing step, many of the superpixels which were correctly classified as foreground objects, are smoothed out by neighboring background pixels.

We propose to improve the label likelihood score at each superpixel to achieve a more accurate parsing output. We design different classifiers that offer complementary information about the data. All the designed models are then combined to derive a consensus decision. The overview of our fusing classifiers approach is shown in Figure 1. At test time, the label likelihood scores of all the BDT models are merged to produce the final scores at superpixels.

4.1. Fusing Classifiers

Our method is inspired from ensemble classifier techniques that train multiple classifiers and combine them to reach a better decision. Such techniques are specifically useful if the classifiers are different [13]. In other words, the error reduction is related to the uncorrelation between the trained models [30], i.e. the overall error is reduced if the classifiers misclassify different data points. Also, it has

[1] http://web.engr.illinois.edu/ dhoiem/software/

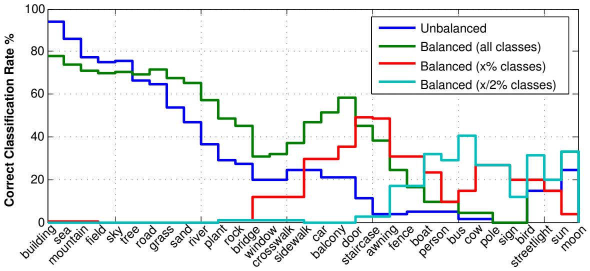


Figure 2. Classification rates (%) of individual classes for the different classification models trained on SIFTflow. Classes are ordered in descending order by the mean number of pixels they occupy (frequency) in scene images. Our goal is to decrease the correlation between the trained models.

been shown that partitioning the training set performs better than partitioning the feature space for large datasets [30].

We have observed that the classification error of a given class is related to the mean number of pixels it occupies in the scene images, as shown by the blue line in Figure 2. This agrees with the findings of previous methods [28, 33] that the classification error rate is related to the frequency of classes in the training set. However, we go beyond that by considering the frequency of the classes on the image level, which targets the problem of smoothing out the less-represented classes by a neighbouring background class.

To this end, we train three BDT models with the following training data criteria: (1) a balanced subsample of all classes C in the dataset, (2) a balanced subsample of classes occupying an average of less than $x\%$ of their images, and (3) a balanced subsample of classes occupying an average of less than $\lceil x/2 \rceil$ % of their images.

The motivation beyond these choices is to reduce the correlation between the trained BDT models as shown in Figure 2. While the unbalanced classifier mainly misclassifies the less-represented classes, the balanced classifiers recover some of these classes while making more mistakes on the more represented classes. By combining the likelihoods from all the classifiers, a better overall decision is reached that improves the overall coverage of classes (Figure 1). We observed that the addition of more classifiers did not improve the performance for any of our datasets.

The final cost of assigning a label c to a superpixel s_i can then be represented as the combination of the likelihood scores of all classifiers:

$$D(l_{s_i} = c|s_i) = 1 - \frac{1}{1 + e^{-L_{comb}(s_i,c)}} \tag{3}$$

where $L_{comb}(s_i, c)$ is the combined likelihood score obtained by the weighted sum of the scores from all classifiers:

$$L_{comb}(s_i, c) = \sum_{j=1,2,3,4} w_j(c)\, L_j(s_i, c), \tag{4}$$

where $L_j(s_i, c)$ is the score from the j^{th} classifier, and $w_j(c)$ is the normalized weight of the likelihood score of class c in the j^{th} classifier.

4.2. Normalized Weight Learning

We learn the weights $\mathbf{w} \equiv [w_j(c)]$ of all classes C in offline settings using the training set. We compute the weights separately for each classifier. The weight $\tilde{w}_j(c)$ of class c for the j^{th} classifier is computed as the average ratio of the sum of all likelihoods of class c, to the sum of all likelihoods of all classes $c_i \in C \backslash c$ of all superpixels $s_i \in S$:

$$\tilde{w}_j(c) = \frac{|C_j|}{C} \frac{\sum_{s_i \in S} L_j(s_i, c)}{\sum_{s_i \in S} \sum_{c_i \in C \backslash c} L_j(s_i, c_i)} \tag{5}$$

where $|C_j|$ is the number of classes covered by the j^{th} classifier and not covered by any other classifier with a smaller number of classes.

The normalized weight $w_j(c)$ of class c can then be computed as: $w_j(c) = \tilde{w}_j(c) / \sum_{j=1,2,3,4} \tilde{w}_j(c)$. Normalizing the output likelihoods in this manner gives a better chance for all classifiers to be considered in the result with an emphasis on less-represented classes. In sec. 6, we show the superior performance of our fusion scheme to other traditional fusion mechanisms: averaging and median rule.

5. Scene-Level Global Context

When exploiting scene parsing problems, it is useful to incorporate the semantics of the scene in the labeling pipeline. For example, if we know that a given scene is a beach scene, we will expect to find labels like sea, sand, and sky with a much higher probability than expecting to find labels like car, building, or fence. We use the initial labeling results of a test image in estimating the likelihoods of all labels $c \in C$ (sec. 5.1). The likelihoods are estimated globally over an image, i.e. there is a unique cost per label per image. We then plug the global label costs into a second MRF inference step to produce better results (sec. 5.2).

Our approach, unlike previous methods, does not limit the number of labels to those present in the retrieval set but instead uses the set to compute the likelihood of class labels in a k-nn fashion. The likelihoods are normalized by counts over the whole dataset and smoothed to give a chance to labels not in the retrieval set. We also employ the likelihoods in MRF optimization, not for filtering the number of labels.

5.1. Context-Aware Global Label Costs

We propose to incorporate semantic context through using label statistics instead of global visual features. The intuition behind such choice is that ranking by global visual features often fails to retrieve similar images on the scene level [29, 33]. For example, a highway scene could

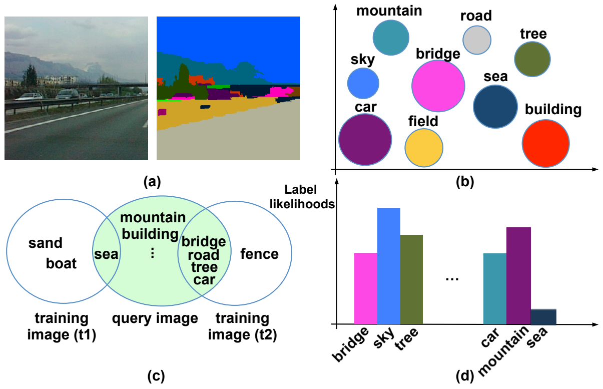


Figure 3. Scene-level global context. (a) The initial labeling of a query image is used to (b) assign weights to the unique classes in the image. A class with a bigger weight is represented by a larger circle. (c) Training images are ranked by the *weighted* size of intersection of their class labels with the query. (d) Global label likelihoods are computed through label counts in the top-ranked images.

be confused with a beach scene with road pixels misclassified as sand. However, ranking by label statistics, given a relatively good initial labeling, retrieves more semantically similar images that aim to remove outlier labels (e.g., sea pixels in street scene), and recover missing labels in a scene.

For a given test image I, minimizing the energy function in equation 2 produces an initial labeling L of the superpixels in the image. If C is the total number of classes in the dataset, let $T \subset C$ be the set of unique labels which appear in L, i.e. $T = \{t \,|\exists s_i : l_{s_i} = t\}$, where s_i is a superpixel with index i in the test image, and l_{s_i} is the label of s_i. We exploit semantic context in a probabilistic framework, where we model the conditional distribution $P(c|T)$ over class labeling C given the initial global labeling of an image T. We compute $P(c|T) \forall c \in C$ in a K-nn fashion:

$$P(c|T) = \frac{(1 + n(c, K_T))/n(c, S)}{(1 + n(\bar{c}, K_T))/|S|}, \tag{6}$$

where K_T is the K-neighborhood of initial labeling T, $n(c, X)$ is the number of superpixels with label c in X, $n(\bar{c}, X)$ is the number of superpixels with all labels except c in X, and $|S|$ is the total number of superpixels in the training set. We normalize the likelihoods and add a smoothing constant of value 1.

To get the neighborhood K_T, we rank the training images by their distance to the query image. The distance between two images is computed as the *weighted* size of intersection of their class labels, intuitively reflecting that the neighbors of T are images with many shared labels with those in T. We assign a different weight to each class in T in such a way to favor less-represented classes.

As shown in Figure 3, our algorithm works in three steps. It starts by (1) assigning a weight ω_t to each class $t \in T$, which is inversely proportional to the number of superpixels in the test image with label t: $\omega_t = 1 - \frac{n(t,I)}{|I|}$, where $n(t, I)$ is the number of superpixels in the test image with label $l_{s_i} = t$, and $|I|$ is the total number of superpixels in the image. Then, (2) training images are ranked by the weighted size of intersection of their class labels with the test image. Finally, (3) the global label likelihood $L_{global}(c) = P(c|T)$ of each label $c \in C$ is computed using equation 6.

Computing the label costs is done online for a query image without any batch offline training. Our method improves the overall accuracy by using only the ground truth labels of training images without any global visual features.

5.2. Inference with Label Costs

Once we obtained the likelihoods $L_{global}(c)$ of each class $c \in C$, we can define a label cost $H(c) = -log(L_{global}(c))$. Our final energy function becomes:

$$E(L) = \sum_{s_i \in S} D(l_{s_i} = c|s_i) + \lambda \sum_{(i,j) \in A} V(l_{s_i}, l_{s_j}) + \sum_{c \in C} H(c).\delta(c), \tag{7}$$

where $\delta(c)$ is the indicator function of label c:

$$\delta(c) = \begin{cases} 1 & \exists s_i : l_{s_i} = c \\ 0 & otherwise \end{cases}$$

We solve equation 7 using α-expansion with the extension method of [5] to optimize label costs. Optimizing the energy function in equation 7 effectively minimizes the number of unique labels in a test image to those which have low label costs, i.e. which are most relevant to the scene.

6. Experiments

We ran our experiments on two large-scale datasets: SIFTflow [18] and LMSun [29]. SIFTflow has 2,488 training images and 200 test images. All images are of outdoor scenes of size 256x256 with 33 labels. LMSun contains both indoor and outdoor scenes, with a total of 45,676 training images and 500 test images. Image sizes vary from 256x256 to 800x600 pixels with 232 labels.

We use the same evaluation metrics and train/test splits as previous methods. We report the per-pixel accuracy (the percentage of pixels of test images that were correctly labeled), and per-class recognition rate (the average of per-pixel accuracies of all classes). We evaluate the following variants of our system: (i) *baseline*, as described in sec. 3, (ii) *baseline (with balanced BDT)*, which is the baseline approach using a balanced classifier, (iii) *baseline + FC (NL fusion)*, which is the baseline in addition to the fusing classifiers with normalized-likelihood (NL) weights in sec. 4, and (iv) *full*, which is baseline + fusing classifiers + global costs. To show the effectiveness of our fusion method (sec. 4.2), we report the results of (v) *baseline + FC (average* *fusion)*, which is fusing classifiers by averaging their likelihoods, and (vi) *baseline + FC (median fusion)*, which is fusing classifiers by taking the median of their likelihoods. We also report results of (vii) *full (without FV)*, which is full system without using the Fisher Vector features.

We fix $x = 5$ (sec.4.1), a value that was obtained through empirical evaluation on a small subset of the training set.

6.1. Results

We compare our results with state-of-the-art methods on SIFTflow in Table 1. We have set $K = 64$ top-ranked training images for computing the global context likelihoods (sec. 5.1). Our full system achieves 81.7% per-pixel accuracy, and 50.1% per-class accuracy, which outperforms the state-of-the-art method of [33] (79.8% / 48.7%). Results show that our fusing classifiers step significantly boosts the coverage of foreground classes, where the per-class accuracy increases by around 15% over the baseline method. Our semantic context (sec. 5) improves both the per-pixel and per-class accuracies through optimizing for fewer labels which are more semantically meaningful. Fisher Vectors improved the recognition by around 3%. In Figure 6, we show examples of parsing results on the SIFTflow dataset.

Method	Per-pixel	Per-class
Liu et al. [18]	76.7	N/A
Farabet et al. [7]	78.5	29.5
Farabet et al. [7] balanced	74.2	46.0
Eigen and Fergus [6]	77.1	32.5
Singh and Kosecka [25]	79.2	33.8
Tighe and Lazebnick [29]	77.0	30.1
Tighe and Lazebnick [28]	78.6	39.2
Yang et al. [33]	79.8	48.7
Baseline	78.3	33.2
Baseline (with balanced BDT)	76.2	45.5
Baseline + FC (NL fusion)	80.5	48.2
Baseline + FC (average fusion)	78.6	46.3
Baseline + FC (median fusion)	77.3	46.8
Full without Fisher Vectors	77.5	47.0
Full	**81.7**	**50.1**

Table 1. Comparison with state-of-the-art per-pixel and per-class accuracies (%) on the SIFTflow dataset.

Table 2 compares the performance of the same variants of our system with the state-of-the-art methods on the large-scale LMSun dataset. LMSun is more challenging than SIFTflow in terms of the number of images, the number of classes, and the presence of both indoor and outdoor scenes. Accordingly, we use a larger value of $K = 200$ in equation 6. Our method achieves near record performance in per-pixel accuracy (61.2%), while placing second in per-class accuracy. The effectiveness of the fusing classifiers technique is shown in the improvement of both per-pixel (by 3%) and per-class (by 4.5%) accuracies over the baseline system. The global context step improves the class coverage by around 2%. Figure 7 shows the output of our scene parsing system on some images from LMSun.

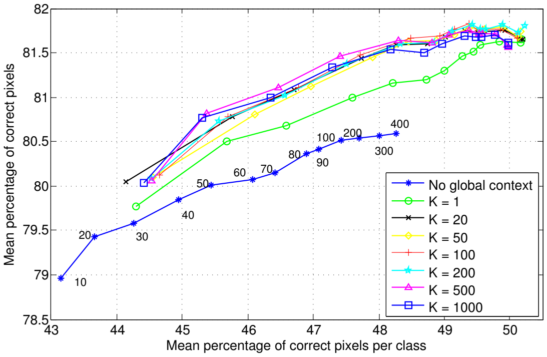


Figure 4. Analysis of the performance when varying the number of trees for training the BDT model, at different values of top K images for the global context step on the SIFTflow dataset. The y-axis shows the per-pixel accuracies (%) and the x-axis show the per-class accuracies (%) for different numbers of trees.

Method	Per-pixel	Per-class
Tighe and Lazebnick [29]	54.9	7.1
Tighe and Lazebnick [28]	**61.4**	15.2
Yang et al. [33]	60.6	**18.0**
Baseline	57.3	9.5
Baseline (with balanced BDT)	45.4	13.8
Baseline + FC (NL fusion)	60.0	14.2
Baseline + FC (average fusion)	60.5	11.4
Baseline + FC (median fusion)	59.2	14.7
Full without Fisher Vectors	58.2	13.6
Full	61.2	16.0

Table 2. Comparison with state-of-the-art per-pixel and per-class accuracies (%) on the LMSun dataset.

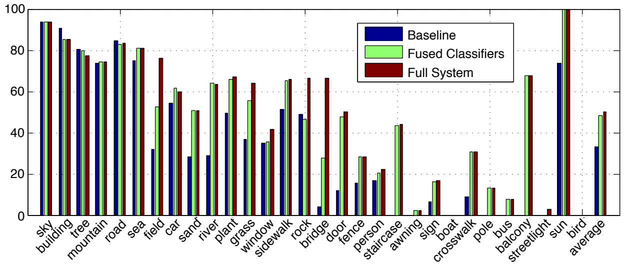


Figure 5. Classification rates (%) of individual classes for the baseline, fused classifiers, and the full system on SIFTflow. Classes are sorted from most frequent to least frequent.

We next analyze the performance of our system when varying the number of trees T for training the BDT model (sec. 4.1), and the number of top training images K in the global label costs (sec. 5.1). Figure 4 shows the per-pixel accuracy (on the y-axis) and the per-class accuracy (on the x-axis) as a function of T for a variety of Ks. Increasing the value of T generally produces better classifica-

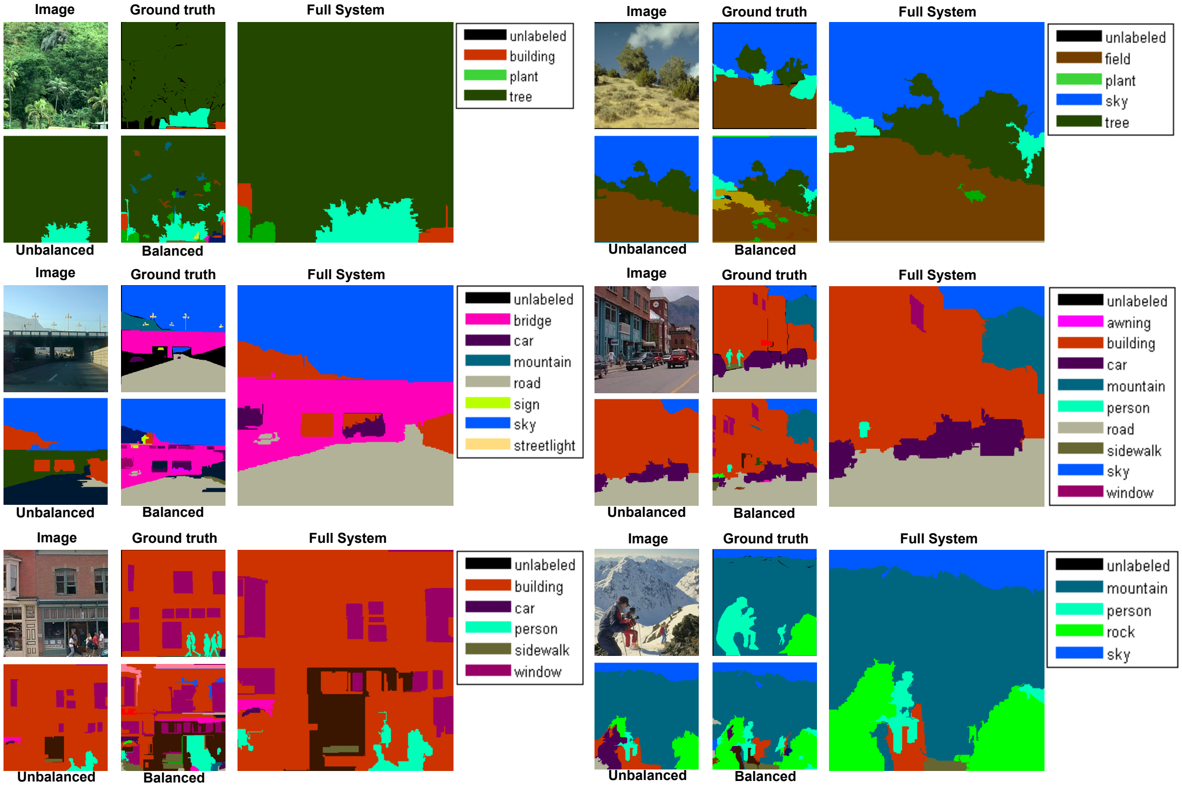


Figure 6. Examples of parsing results on the SIFTflow dataset (best viewed in color). Top left is the original image, on its right is the ground truth labeling, bottom left is the output from the baseline, on its right the output of the balanced classifier. Finally, the output of the full system is on the far right (third column). The unbalanced classifier often misses the foreground classes by oversmoothing the results. The balanced classifier performs better with foreground classes, but yields more noisy classification. The full system combines the benefits of both classifiers, improving both the overall accuracy and the coverage of foreground classes (e.g., building, bridge, window, and person)

tion models that better describe the training data. At $T \geq 400$, performance levels off. As shown, our global label costs consistently improve the performance over the baseline method with no global context. Using more training images (higher K) improves the performance through considering more semantically-relevant scene images. However, performance starts to decrease for very high values of K (e.g., $K = 1000$) as more noisy images start to be added.

Figure 5 shows the per-class recognition rate for the baseline, combined classifiers, and the full system on SIFTflow. Our fusing classifiers technique produces more balanced likelihood scores that cover a wider range of classes. The semantic context step removes outlier labels and recovers missing labels, which improves the recognition rates of both common and rare classes. Recovered classes include field, grass, bridge, and sign. Failure cases include extremely rare classes, e.g. cow, bird, desert, and moon.

6.2. Running Time

We analyzed the runtime performance for both SIFTflow and LMSun (without feature extraction) on a four-core 2.84GHz CPU with 32GB of RAM without code optimization. For the SIFTflow dataset, training the classifier takes an average of 15 minutes per class. We run the training process in parallel. The training time highly depends on the feature dimensionality. At test time, superpixel classification is efficient, with an average of 1 second per image. Computing global label costs takes 3 seconds. Finally, MRF inference takes less than one second. We run MRF inference twice for the full pipeline. LMSun is much larger than SIFTflow. It takes 3 hours for training the classifier, less than a minute for superpixel classification per image, less than 1 minute for MRF inference, and $\sim$2 minutes for global label cost computation.

6.3. Discussion

Our scene parsing method is generally scalable as it does not require any offline training in a batch fashion. However, the time required for training a BDT classifier increases linearly with increasing the number of data points. This is challenging with large datasets like LMSun. Randomly subsampling the dataset has a negative impact on the overall

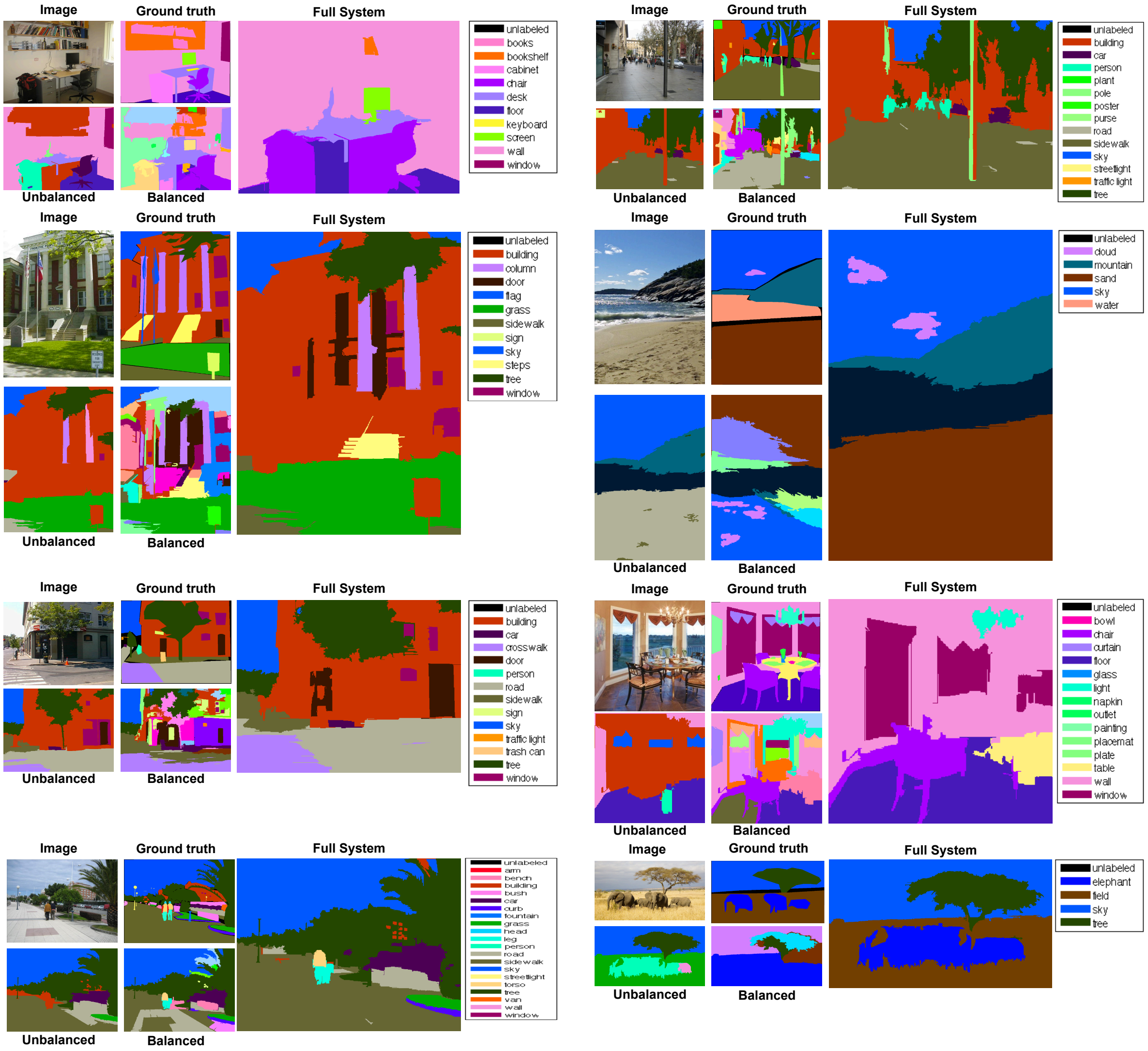


Figure 7. Examples of parsing results on the LMSun dataset (best viewed in color). The layout of the results is the same as in Fig. 6. Foreground classes (e.g. screen, sidewalk, person, torso, pole, cloud, table, light, and elephant) are successfully recognized by our system.

precision of the classification results. We plan to investigate alternative approaches like [26] of mining discriminative data points that better describe each class. Our system still faces challenges in trying to recognize very less-represented classes in the dataset (e.g., bird, cow, and moon). This could be handled via better contextual models per query image.

7. Conclusion

In this work, we have presented a novel scene parsing algorithm that improves the overall labeling accuracy, without smoothing away foreground classes which are important for human observers. Through combining likelihood scores from different classification models, we have successfully boosted the strengths of individual models, thus improving both the per-pixel, as well as the per-class accuracies. To avoid eliminating correct labels through image retrieval, we have encoded global context into the parsing process in a probabilistic framework. We have extended the energy function to include global label costs that achieve more semantically meaningful parsing output. Experiments have shown the superior performance of our system on the SIFT-flow dataset and comparable performance to state-of-the-art methods on the LMSun dataset.

Acknowledgements

We thank Friedemann Mattern and Christian Floerkemeier for the useful discussions, and the CVPR reviewers for their insightful feedback.

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.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[3] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[4] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.

[5] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 96(1):1–27, 2012.

[6] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *ECCV*, 2008.

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*, 2012.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[9] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[10] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classifcation models: Combining models for holistic scene understanding. In *NIPS*, 2008.

[11] G. Heitz and D. Koller. Learning spatial context: using stuff to find things. In *CVPR*, 2008.

[12] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *PAMI*, 16(1):66–75, 1994.

[13] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *PAMI*, 20(3):226–239, 1998.

[14] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[15] P. Kontschieder, S. Rota Bulo, M. Pelillo, and H. Bischof. Structured labels in random forests for semantic labelling and object detection. *PAMI*, 36(10):2104–2116, 2014.

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

[17] L.-J. Li, H. Su, E. P. Xing, and L. Fei-fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

[18] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 33(12):2368–2382, 2011.

[19] D. Munoz, J. A. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.

[20] H. J. Myeong, Y. Chang, and K. M. Lee. Learning object relationships via graph-based context model. In *CVPR*, 2012.

[21] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010.

[22] X. Qi and Y. Han. Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, 40(2):728–741, 2007.

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

[24] 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*, 2006.

[25] G. Singh and J. Košecká. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *CVPR*, 2013.

[26] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[27] P. Sturgess, K. Alahari, L. Ladický, and P. H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009.

[28] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.

[29] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. *IJCV*, 101(2):329–349, 2013.

[30] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–404, 1996.

[31] A. Vedaldi and B. Fulkerson. VLfeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[32] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[33] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014.

[34] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

[35] Z. Yin, F. Porikli, and R. Collins. Likelihood map fusion for visual object tracking. In *WACV*, 2008.

[36] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *ECCV*, 2010.