Sample and Filter: Nonparametric Scene Parsing via Efficient Filtering

Mohammad Najafi[1,2] Sarah Taghavi Namin[1,2] Mathieu Salzmann[1,3] Lars Petersson[1,2]
[1]Australian National University (ANU) [2]NICTA* [3]CVLab, EPFL, Switzerland
{sarah.namin, mohammad.najafi, lars.petersson}@nicta.com.au mathieu.salzmann@epfl.ch

Abstract

Scene parsing has attracted a lot of attention in computer vision. While parametric models have proven effective for this task, they cannot easily incorporate new training data. By contrast, nonparametric approaches, which bypass any learning phase and directly transfer the labels from the training data to the query images, can readily exploit new labeled samples as they become available. Unfortunately, because of the computational cost of their label transfer procedures, state-of-the-art nonparametric methods typically filter out most training images to only keep a few relevant ones to label the query. As such, these methods throw away many images that still contain valuable information and generally obtain an unbalanced set of labeled samples. In this paper, we introduce a nonparametric approach to scene parsing that follows a sample-and-filter strategy. More specifically, we propose to sample labeled superpixels according to an image similarity score, which allows us to obtain a balanced set of samples. We then formulate label transfer as an efficient filtering procedure, which lets us exploit more labeled samples than existing techniques. Our experiments evidence the benefits of our approach over state-of-the-art nonparametric methods on two benchmark datasets.

1. Introduction

Scene parsing, also known as semantic segmentation, tackles the problem of assigning one class label to every pixel in an image (Fig. 1). The traditional approach to addressing this problem consists of having a separate training phase that learns a parametric model, which will then be applied to the test data [24, 14, 12, 10, 15, 13, 16, 27, 32, 9, 6, 23, 19, 25]. While effective, this approach doesn't account for the dynamic nature of our world, where images are constantly being acquired. Indeed, as new training data becomes available, these techniques need to re-train their model. Unfortunately, this process is generally very time-consuming; for example, training a state-of-the-art Convolutional Neural Network (CNN) can take several days.

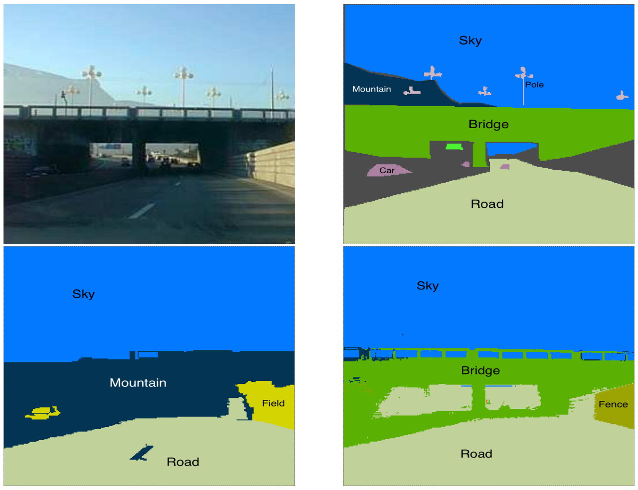


Figure 1: **Nonparametric Scene Parsing. Top left:** Query image; **Top right:** Ground-truth; **Bottom left:** Superparsing [28]; **Bottom right:** Our method.

Nonparametric methods have recently emerged as a solution to this drawback [17, 5, 11, 28, 21, 20, 26, 29]. Rather than training a model, these techniques aim at directly transferring the semantics of labeled images to the test data. As such, they can readily incorporate new labeled images as they become available.

Most nonparametric methods [17, 5, 28, 21, 20, 26, 29] follow a two-stage procedure: They first retrieve a set of images similar to the query image, and then transfer the labels of these retrieved images to the query. The retrieval step plays two important roles. First, it discards the labeled images that are irrelevant to the query. Second, by reducing the amount of data to take into account, it effectively speeds up the transfer step. While the benefits of the former point are unquestionable, the latter one is somewhat more dubious and mostly motivated by the relative lack of speed of the transfer step. Indeed, to remain fast, existing techniques typically throw away images which might still contain valuable information. This particularly causes problems when the classes are unbalanced, since the less-frequent classes might easily not even appear in the retrieved images.

* NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

In this paper, we introduce an approach to scene parsing that follows a sample-and-filter strategy. Specifically, instead of retrieving a fixed number of similar training images, we randomly sample the labeled superpixels from the training data according to an image-similarity score. We then formulate label transfer as a Gaussian filtering procedure, which computes the label of a query superpixel from the labels of the sampled superpixels. Thanks to the efficiency of our filtering procedure and to our sampling strategy, our approach lets us (i) make use of more labeled superpixels than existing retrieval-based techniques; and (ii) obtain a set of labeled samples that is more balanced in terms of class frequency.

We evaluate our method on two large-scale benchmark datasets, SIFTFlow [17] and LM-SUN [28]. Our experiments evidence the benefits of our approach in terms of both accuracy and computation time over state-of-the-art nonparametric scene parsing techniques.

2. Related Work

In recent years, scene parsing has attracted a lot of attention. In particular, many methods have proposed to tackle scene parsing by first learning a model from training data, and then applying this model to the unseen test data. A popular trend among these methods consists of learning a pixel classifier and use it as a unary potential in a Markov Random Field (MRF), which models the dependencies of the class labels of two or more pixels [24, 14, 12, 10, 15, 13, 16]. When it comes to the classifier itself, several directions have been proposed, such as boosting-based classifiers [24, 32, 9], or exemplar-based object detectors [27, 16]. With the recent advent of deep learning, several works have focused on developing CNNs to perform semantic segmentation [6, 23, 19, 25]. While effective, these approaches are parametric, and thus cannot incorporate new labeled data without a computationally expensive re-training procedure.

By contrast, nonparametric approaches do not learn any model, but instead transfer the labels of the training data to the query images. As a consequence, they can directly incorporate new labeled data. To the best of our knowledge, this idea was first introduced by Liu *et al*. [17], who made use of SIFTFlow [18] to transfer the labels from a small set of retrieved images to the query. Unfortunately, the computational cost of SIFTFlow significantly affected the speed of their approach. Instead, in [11], Gould & Zhang built on the efficient PatchMatch algorithm [3, 4], which allowed them to bypass the retrieval step and build a graph over the entire training set to perform label transfer. For the algorithm to remain tractable, however, the degree of the vertices in the graph had to be kept low, which, in turn, affected the labeling accuracy.

Superparsing, introduced by Tighe & Lazebnik [28], probably constitutes the most popular nonparametric approach to scene parsing. From a set of retrieved images, it produces a label for each query superpixel by combining the results of nearest-neighbor retrieval using multiple superpixel features in a naïve Bayes classifier. Inspired by [28], Eigen & Fergus [5] and Singh & Kosecka [26] proposed to learn weights for the different superpixel features; Myeong *et al*. [20, 21] incorporated pairwise and higher-order contextual relationships among the object categories into the Superparsing framework; Tung & Little [29] proposed to reason at the level of complete objects, obtained by an objectness criterion, instead of relying on superpixels. While all these modifications of Superparsing have indeed led to higher segmentation accuracy, they also come at a higher computational cost. Furthermore, and more importantly, all these methods, including Superparsing, make an initial strong decision to reject a large number of labeled images, many of which might still contain valuable information for the query.

By contrast, here, we introduce a sampling strategy to collect the relevant labeled superpixels, which lets us retrieve a balanced number of samples for each class. Thanks to this sampling procedure, and to our efficient filtering approach to label transfer, our algorithm yields accuracies that are competitive with the state-of-the-art methods, while being significantly faster.

3. Method

We now introduce our nonparametric approach to scene parsing. To this end, let $\mathcal{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_{N_t}\}$ denote the set of feature vectors $\mathbf{x}'_j$ representing the training superpixels, with corresponding ground-truth labels $\mathbf{Y}' = \{y'_1, y'_2, \dots, y'_{N_t}\}$, $y'_i \in \{1, \dots, L\}$. Our goal is to transfer these labels to a set of query superpixels encoded by their feature vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_q}\}$. As mentioned above, here, we follow a sample-and-filter approach, which first randomly samples a balanced set of relevant training superpixels, and then performs label transfer via efficient Gaussian filtering. In the remainder of this section, we present these two steps in detail.

3.1. Sampling Balanced Superpixels

It is undeniable that, as suggested by other nonparametric approaches [28, 5, 26, 29], many images from the training data are irrelevant to label the query image. Following this intuition and common practice, we therefore first rank the training images according to their similarity to the query image using the method explained in Section 3.1.1. At this stage, state-of-the-art nonparametric scene parsing algorithms [28, 5, 26, 29] simply discard the images beyond a pre-defined rank. This, however, typically discards many images with relevant information because of noise in the ranking process and because the pre-defined rank is usually chosen so as to keep few images. Furthermore, with this process, the number of retrieved superpixels belonging to each class is typically unbalanced.

By contrast, here, we propose to make use of the ranks to randomly sample training superpixels. To this end, we assign a dissimilarity value

$$d_j \in \left\{ \frac{1}{N_t}, \frac{1}{N_t - 1}, \ldots, 1 \right\} \tag{1}$$

to each training superpixel according to the rank of its corresponding image. Note that the superpixels in the image with the highest rank, *i.e.*, the image most similar to the query, will be assigned the lowest dissimilarity value. From these dissimilarity values, we compute a score for each superpixel as

$$p_j = \exp\left(-\frac{d_j^2}{\sigma_d} \right) \ , \ \ \forall\ j \in \{1, 2 \ldots, N_t\} \ . \tag{2}$$

We then use this score to randomly sample the superpixels using the method proposed in [30]. Ultimately, while superpixels with larger values p_j are more likely to be picked, this still potentially allows any superpixel to be selected.

Furthermore, and more importantly, since we randomly sample superpixels, and each superpixel is assigned a class label, we can enforce having a balanced set of training data by sampling the same number of superpixels for each class. Note that, in practice, this is not always possible, since some classes truly occur very rarely in the training data. This will be addressed in the label transfer step of our approach. Nevertheless, our sampling procedure produces a more balanced set of superpixels than the simple image retrieval strategy. Furthermore, thanks to our efficient filtering approach to label transfer, discussed below, we can exploit more labeled superpixels than state-of-the-art nonparametric scene parsing techniques.

3.1.1 Image Ranking

As mentioned above, our sampling strategy relies on an image ranking procedure that reflects the similarity between each training image and the query. This procedure works as follows. We extract three global image descriptors, *i.e.*, spatial pyramid of color histograms, GIST [22] and Histogram of Oriented Gradients (HOG) visual words [31], from each image in the training set and from the query. We then produce three rankings according to the similarity of each of these descriptors, using the χ^2 distance metric. The final rank of the images are then obtained by sorting their average ranks over these three rankings.

3.2. Label Transfer via Efficient Filtering

The sampling procedure of Section 3.1 produces a balanced set of N_s training superpixels encoded by feature vectors $\{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_{N_s}\}$. Our goal now is to transfer the labels of these superpixels to those of the query image. Here, we propose to formulate label transfer as an efficient Gaussian filtering operation.

To this end, let $\mathbf{q}'_j$ be the L-dimensional binary vector encoding the label of the j^{th} training superpixel as

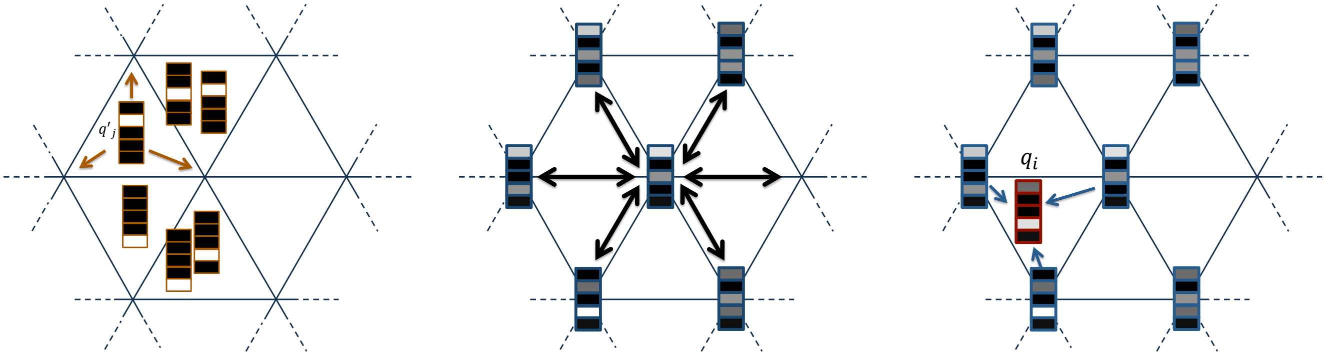
$$\mathbf{q}'_j(l) = \begin{cases} 1 & y'_j = l \\ 0 & \text{otherwise} \ , \end{cases} \tag{3}$$

where $\mathbf{q}'_j(l)$ indicates the l^{th} element of $\mathbf{q}'_j$. We then propose to estimate the label of the query superpixels as

$$\mathbf{q}_i = \sum_{j=1}^{N_s} k(\mathbf{x}_i, \mathbf{x}'_j)\mathbf{q}'_j \ , \ \ \forall\ i \in \{1, 2, \ldots, N_q\} \ , \tag{4}$$

where $k(\mathbf{x}_i, \mathbf{x}'_j)$ is a Gaussian kernel that encodes how similar two superpixels are in terms of their feature vectors $\mathbf{x}_i$ and $\mathbf{x}'_j$, and thus how strongly we believe that these two superpixels should have the same label. The specific form of kernel used in our experiments is given in Section 3.2.1.

Since Eq. 4 involves N_s summations for every query superpixel, the total computational complexity for a query image would be $O(N_s N_q)$. For large numbers of retrieved superpixels, which is what we advocate for here, this approach would thus be prohibitively costly. However, Eq. 4 corresponds to a Gaussian filtering operation, for which fast and accurate approximations have been proposed [1, 8, 2]. In particular, here, we make use of the permutohedral lattice-based formulation of [1]. This method relies on three steps, illustrated in Fig. 2. The first step is *Splatting*, which, in our case, consists of mapping the training data to the permutohedral lattice and computing the values at the vertices of the lattice. More specifically, the label vectors of the training superpixels are soft-assigned to the lattice vertices according to the barycentric coordinates of the feature vectors (*i.e.*, the value at a vertex is computed as a linear combination of its surrounding label vectors). In the *Blurring* step, which approximates the Gaussian filter locally, Gaussian blurring is performed on the vertices along each axis of the lattice. The blurring process is truncated, such that the value at each vertex is only affected by its direct neighbors. The last step is *Slicing*, which, in our case, consists of mapping the query superpixels to the lattice by computing the barycentric coordinates of their feature vectors. The label of a query point is obtained as a linear combination of the values at the vertices, using its barycentric coordinates. The first two steps, which only involve the training data, can be performed in $O(N_s)$. For each query superpixel, slicing can be done in constant time, *i.e.*, linearly dependent on the feature dimension, but not on N_s. Altogether, this therefore yields a total computational complexity of $O(N_s + N_q)$.



(a) Splatting

(b) Blurring

(c) Slicing

Figure 2: **Schematic view of the filtering process on the permutohedral lattice.** The block structures represent label vectors, and the gray-level intensities in each vector denote the likelihoods of different classes. Fig. 2a shows how the binary label vector $\mathbf{q}'_j$ (defined in Eq. 3) is mapped onto the lattice vertices. The blurring step is depicted in Fig. 2b, where Gaussian blurring is applied locally to the verrtices. Fig. 2c illustrates the slicing step, where a query data receives label information from the lattice vertices.

3.2.1 Kernels

In this work, we define the kernel of Eq. 4 as

$$k(\mathbf{x}_i, \mathbf{x}'_j) = w_1 k_1(\mathbf{x}_i, \mathbf{x}'_j) + w_2 k_2(\mathbf{x}_i, \mathbf{x}'_j) \ , \tag{5}$$

where k_1 and k_2 are two Gaussian kernels defined below. Note that the algorithm described above translates easily to the two-kernel case by simply making use of two permutohedral lattices, and, for each query superpixel, combining the two predicted label vectors.

In practice, as a first kernel, we make use of a color-based Gaussian, expressed as

$$\begin{aligned} k_1(\mathbf{x}_i, \mathbf{x}'_j) = \exp\Bigg(& -\frac{\|\mathbf{c}_i - \mathbf{c}'_j\|^2}{\sigma_c^2} - \frac{|t_i - t'_j|^2}{\sigma_t^2} \\ & -\frac{|s_i - s'_j|^2}{\sigma_s^2} - \frac{|d_i - d'_j|^2}{\sigma_d^2} \Bigg) \ , \end{aligned} \tag{6}$$

where $\mathbf{c}$ is the vector of average RGB intensities of a superpixel, s is the standard deviation of the gray-level intensities in the superpixel, t is the minimum distance of the superpixel to the top of the image, and d is the dissimilarity value defined in Eq. 1. Note that we set $d_i = 0$ for the query superpixels.

The second kernel relies on the image gradient and is defined as

$$\begin{aligned} k_2(\mathbf{x}_i, \mathbf{x}'_j) = \exp\Bigg(& -\frac{\|\mathbf{h}_i - \mathbf{h}'_j\|^2}{\sigma_h^2} - \frac{|t_i - t'_j|^2}{\sigma_t^2} \\ & -\frac{|s_i - s'_j|^2}{\sigma_s^2} - \frac{|d_i - d'_j|^2}{\sigma_d^2} \Bigg) \ , \end{aligned} \tag{7}$$

where $\mathbf{h}$ is the 6-bin HOG descriptor of the superpixel.

In our experiments, the standard deviations σ_c, σ_t, σ_s, σ_d and σ_h, and the weights w_1 and w_2 were obtained using a validation set.

3.2.2 Handling Rare Classes

As mentioned in Section 3.1, while we aim at selecting a balanced set of training superpixels, having exactly an equal number for each class is not always possible, due to the insufficient number of superpixels in some *rare* classes. As a matter of fact, this problem occurs frequently in large-scale datasets, and would have a negative impact on the filtering procedure. Indeed, in Eq. 4, the contribution of a superpixel belonging to a rare class and highly similar to the query superpixel could easily be dominated by the combined contributions of superpixels from a common class, even if they are not too similar to the query.

To address this problem, we propose to modify the definition of $\mathbf{q}'_j$ in Eq. 3 as

$$\mathbf{q}'_j(l) = \begin{cases} \lambda(l) & y'_j = l \\ 0 & \text{otherwise} \ , \end{cases} \tag{8}$$

where $\lambda(l) = N_{max}/N(l)$, with N_{max} the maximum number of samples picked from any class, and $N(l)$ the number of samples picked from class l. The term λ approaches 1 for the frequent categories, whereas it increases the contribution of the superpixels belonging to rare classes in the filtering process. Note that, in the perfectly balanced case, all classes have again the same influence.

Algorithm 1: Sample & Filter Strategy for Nonparametric Label Transfer

```
Data: Query image + entire set of training images
Rank the training images based on their similarity to the query image   (Section 3.1.1)
Randomly sample training superpixels according to their dissimilarity values ($d_j$)   (Section 3.1)
for $i = 1$ to $N_q$ do
    $\mathbf{q}_i = \sum_{j=1}^{N_t} k(\mathbf{x}_i, \mathbf{x}'_j)\mathbf{q}'_j$ ;                    // Filtering the training superpixels
end
$\mathbf{u}_i = -\log(\tilde{\mathbf{q}}_i)$ ;        // Compute a unary term based on the normalized filtered labels
Compute the pixelwise location prior   (Section 3.3)
Perform inference in a dense pixel-wise CRF   (Section 3.3)
return Dense pixelwise labeling of the query image
```

3.3. MRF

The semantic information transferred to the query superpixels by our approach is of course prone to error. As is commonly done in nonparamatric scene parsing methods [28, 26, 20, 21, 5, 29, 11], we therefore make use of an MRF to further smooth these initial predictions. More precisely, our predictions act as unary terms in an MRF defined over the pixels of the query image, which thus prevents us from having to train a classifier.

Specifically, let $\tilde{\mathbf{q}}_i$ be the normalized version of the $\mathbf{q}_i$ obtained from Eq. 4. We then define the unary potential of each superpixel i as the negative logarithm of $\tilde{\mathbf{q}}_i$, and assign this unary potential to all the pixels within superpixel i. We further combine this unary with a location prior computed as a class histogram built for each pixel from the 15 top images in our ranking. We then make use of the fully-connected CRF model of Krähenbühl & Koltun [13], which relies on an efficient mean-field-based inference strategy to produce a pixelwise labeling of the query image.

The main steps of our nonparametric scene parsing approach are summarized in Algorithm 1.

4. Experiments

We evaluated our method on two large-scale datasets, SIFTFlow [17] and LM-SUN [28]. Below, we compare our results with those of state-of-the-art nonparametric scene parsing algorithms. In all our experiments, we obtained the superpixels using the same unsupervised segmentation method (graph-based segmentation [7]) as Superparsing.

4.1. SIFTFlow

SIFTFlow [17] consists of 2,688 images taken from outdoor scenes and annotated with 33 different class labels. The standard partition of this dataset includes 2,488 training images and 200 test images. As noted in [28], this is a difficult dataset due to the large number of rare classes. For this dataset, we sampled a maximum of 2500 superpixels of each class. Note, however, that because of rarity, some classes had much fewer samples. Fig. 3 illustrates the class label distribution of the drawn samples.

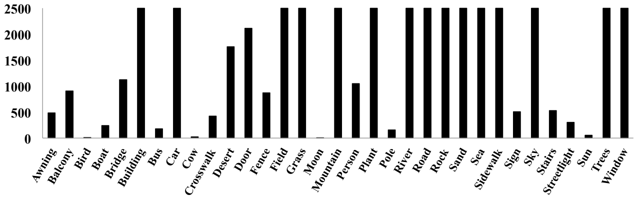


Figure 3: **Label distribution of the superpixels drawn from the training pool in SIFTFlow.** The number of samples was capped at 2500. Note, however, that some rare classes only have much fewer available samples, leading to a very imbalanced class distribution.

In Table 1, we compare our results with those of state-of-the-art nonparametric scene parsing methods in terms of per-pixel and average per-class accuracy. Our approach performs on par with the baselines in per-pixel accuracy, but outperforms most of them in per-class accuracy. This, we believe is due to the more balanced samples that we obtain. To verify this, we replaced our sampling strategy with a fixed retrieval set consisting of all the superpixels of the top 200 images in our ranking.[1] Running our filtering-based label transfer procedure on these superpixels resulted in 73.6% per-pixel accuracy and 22.2% per-class accuracy. As expected, while the effect on per-pixel accuracy is relatively small, the per-class accuracy decreases dramatically. This clearly evidences the importance of getting as balanced as possible a set of labeled superpixels. Fig. 4 provides a qualitative comparison of our results with those of Superparsing.

Note that, In Table 1, the highest per-class accuracy is achieved by [29]. This method, however, relies on an expensive procedure, thus requiring several minutes to process an image. By contrast, thanks to our efficient filtering approach, our algorithm only requires roughly 4 seconds, which outperforms all the baselines.

[1] We used 200 because it corresponds to the number of images retrieved by the baselines.

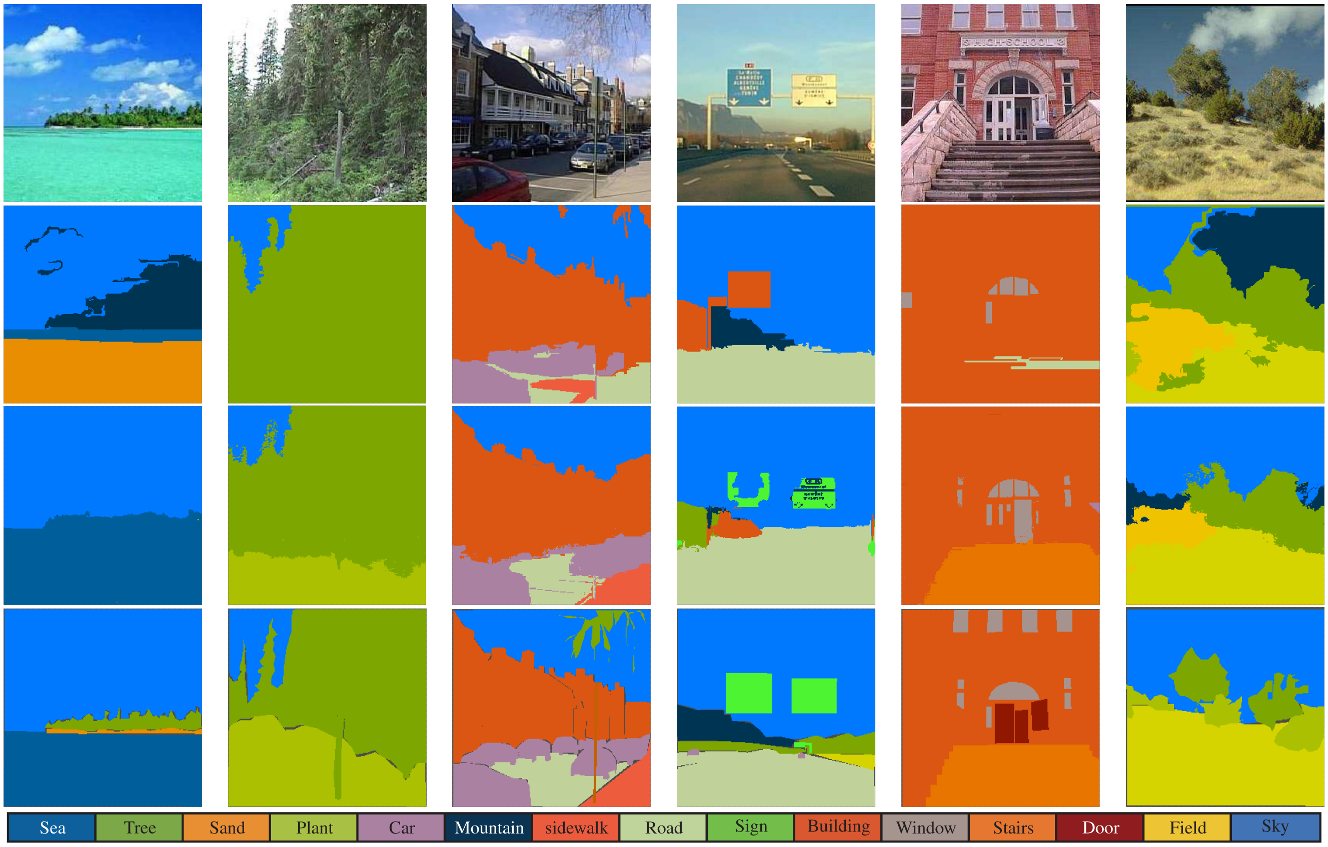


Figure 4: **Qualitative comparison of our results with those of Superparsing [28] on SIFTFlow. 1st row:** Query image; **2nd row:** Superparsing; **3rd row:** Our approach; **4th row:** Ground-truth.

Table 1: Comparison of our approach (Sample & Filter) with the state-of-the-art nonparametric methods on SIFTFlow. We report the per-pixel and average per-class accuracies, as well as the average time to process one image. For the baselines, a $>$ indicates that the reported runtimes do not include the entire processing time.

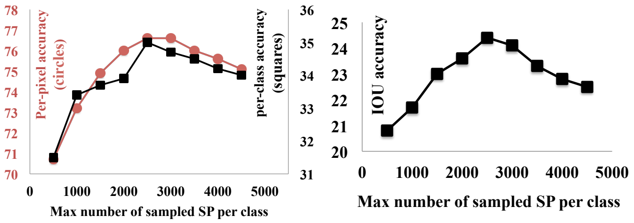
	per-pixel	**per-class**	**runtime**
Sample & Filter	74.5	35.5	2.0s
Sample & Filter (with MRF)	76.6	35.0	4.2s
Superparsing (with MRF) [28]	76.2	29.1	$>$5.9s
Eigen *et al*. (with MRF) [5]	77.1	32.5	$>$16.6s
Myeong *et al*. (with MRF) [20]	77.1	32.3	$>$23s
SIFTFlow [18]	76.7	-	$>$25mins
WAKNN (with MRF) [26]	79.2	33.8	$>$70s
CollageParsing (with MRF) [29]	77.1	41.1	2mins

Note that our runtimes were obtained on a standard desktop with an Intel 3.07GHz six-core processor and 12 GB RAM. Our algorithm was implemented mostly in Matlab, with the exception of the filtering step, which was built upon the C++ code of [13]. This leaves room for speed improvement. While we do not know the exact setup of the baselines, we believe that, since we used an ordinary platform, the runtime comparison remains fair.

Table 2: Comparison of our approach (Sample & Filter) with Superparsing using an ideal image ranking on SIFTFlow.

	per-pixel	**per-class**
Sample & Filter (with MRF)	83.1	44.3
Superparsing (with MRF) [28]	80.2	33.6

To further evaluate the potential of our approach, and following the analysis performed in [28], we performed an additional experiment based on an *ideal* image ranking strategy. To this end, and following [28], the retrieval was achieved using histograms of ground-truth class labels, both for the training and test images. The idea here is to try and evaluate the best possible performance of our approach. The results of this experiment are reported in Table 2, where we compare our approach with the results of [28] obtained in the same ideal setting. These results indicate that, given a better image similarity measure, our method has the potential to achieve higher accuracy than Superparsing, especially in terms of per-class accuracy.



(a) Per-pixel and per-class

(b) IoU

Figure 5: **Influence of N_s.** We report the per-class, per-pixel and IoU accuracies as a function of the maximum number of superpixels sampled from each class. Note that our approach yields good results for a large range of values.

To study the influence of the number of superpixels sampled from each class on our results, we ran our approach with N_s ranging from 500 to 5000. In Fig. 5a, we report the per-pixel and per-class accuracies as a function of N_s, which shows that our approach yields good results in the range 2000-3500. In Fig. 5b, we report the Intersection over Union (IoU) of our results w.r.t. ground-truth as a function of N_s. As a comparison, the IoU of Superparsing [28], computed from their results available online, is 21.1. This shows that our approach (IoU = 24.4 at $N_s = 2500$) also outperforms this baseline according to this error metric. Note that the results of the other baselines are not publicly available.

Our approach handles the class imbalance problem by varying the impact of different classes in the filtering process according to their frequency in the training samples. Treating all classes equally (replacing Eq. 8 with Eq. 3) in our method yields per-pixel and per-class accuracies of (77.2%, 24.4%). This shows that our strategy significantly improves the per-class accuracy at only a negligible cost in terms of per-pixel accuracy.

We further performed an ablation study to study the influence of different parameters in our model. For instance, replacing our filtering process with a KNN classifier gave accuracies of (75.1%, 23.5%), which evidences the benefits of our filtering-based approach. Furthermore, removing t_i, s_i, or d_i from the kernels led to accuracies of (76.2%, 28.9%), (75%, 33%) and (64.4%, 36.7%), respectively, which indicates that all these features are beneficial.

Fig. 6 shows a failure case of our method. This figure depicts a query image followed by the top six images in the similarity ranking, the result of our algorithm and the ground-truth. In this case, the image ranking strategy retrieved a semantically irrelevant group of images. As suggested by Table 2, improving the image similarity metric would address this problem.

Our method scales linearly with the number of labeled images due to the initial KNN retrieval step. Note that this could be sped up by using an approximate NN scheme. The remaining steps scale linearly with the number of sampled superpixels, as discussed in Section 3.2.

Table 3: Comparison of our approach (Sample & Filter) with Superparsing on LM-SUN.

	per-pixel	per-class	runtime (excluding retrieval)
Sample & Filter	54.6	6.7	3.7s
Sample & Filter (with MRF)	55.1	6.6	6.0s
Superparsing [28]	50.6	7.1	18.3s
Superparsing (with MRF) [28]	54.4	6.8	18.3s

Table 4: Comparison of our approach (Sample & Filter) with Superparsing using an ideal image ranking on LM-SUN.

	per-pixel	per-class
Sample & Filter (with MRF)	69.3	15
Superparsing [28] (with MRF)	66	13.2

4.2. LM-SUN

The LM-Sun dataset [28] is one the most challenging benchmarks available for scene parsing. It includes 45,676 images, among which, following the standard partition, 500 images are taken as test data. The ground-truth annotations of this dataset are comprised of 232 different categories. In this case, we sampled a maximum of 25,000 superpixels per class.

In Table 3, we compare our results with those of [28], which constitutes the state-of-the-art on this dataset. To the best of our knowledge, Superparsing [28] is the only nonparametric approach that has been evaluated on this large-scale dataset. As a matter of fact, the scale of this dataset causes most nonparametric method to be intractable. By contrast, our efficient algorithm can still yield state-of-the-art accuracies in a reasonable time. In particular, our *Sample & Filter* procedure takes 3.7 seconds per image on average, versus 13.1 seconds for Superparsing to transfer the labels. Furthermore, for each query image, our algorithm performs filtering on 367,080 superpixels on average, which is about 10 times larger than the 35,600 superpixels (200 retrieved images, each containing approximately 178 superpixels) analyzed by Superparsing. In other words, not only is our approach faster than Superparsing, but it can also exploit more labeled data. Fig. 7 provides a qualitative comparison of our results with those of Superparsing.

As in the previous section, we conducted an additional experiment using an ideal image ranking by making use of histograms of ground-truth annotations. Table 4 provides the results of this experiment. Note that, again, our approach has higher potential for improvement given a better image similarity measure.

5. Conclusion

In this paper, we have introduced a nonparametric approach to scene parsing based on the concept of sampling

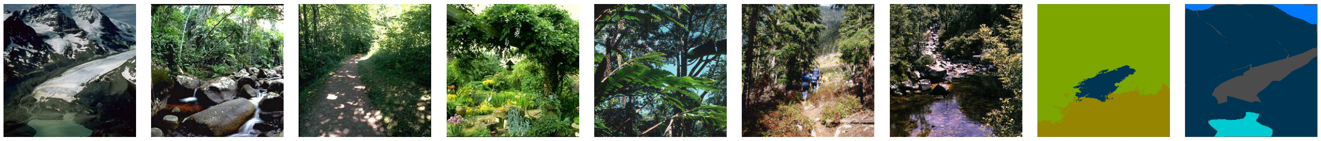


Figure 6: **Failure case.** The first image is the query and the next six images are the top ranked training images. The last two images denote our results and the ground-truth, respectively. Note that the top images in the ranking are semantically irrelevant, which leads to inaccurate labeling. As suggested by Table 2, however, a better image ranking would yield a significant improvement of our results.

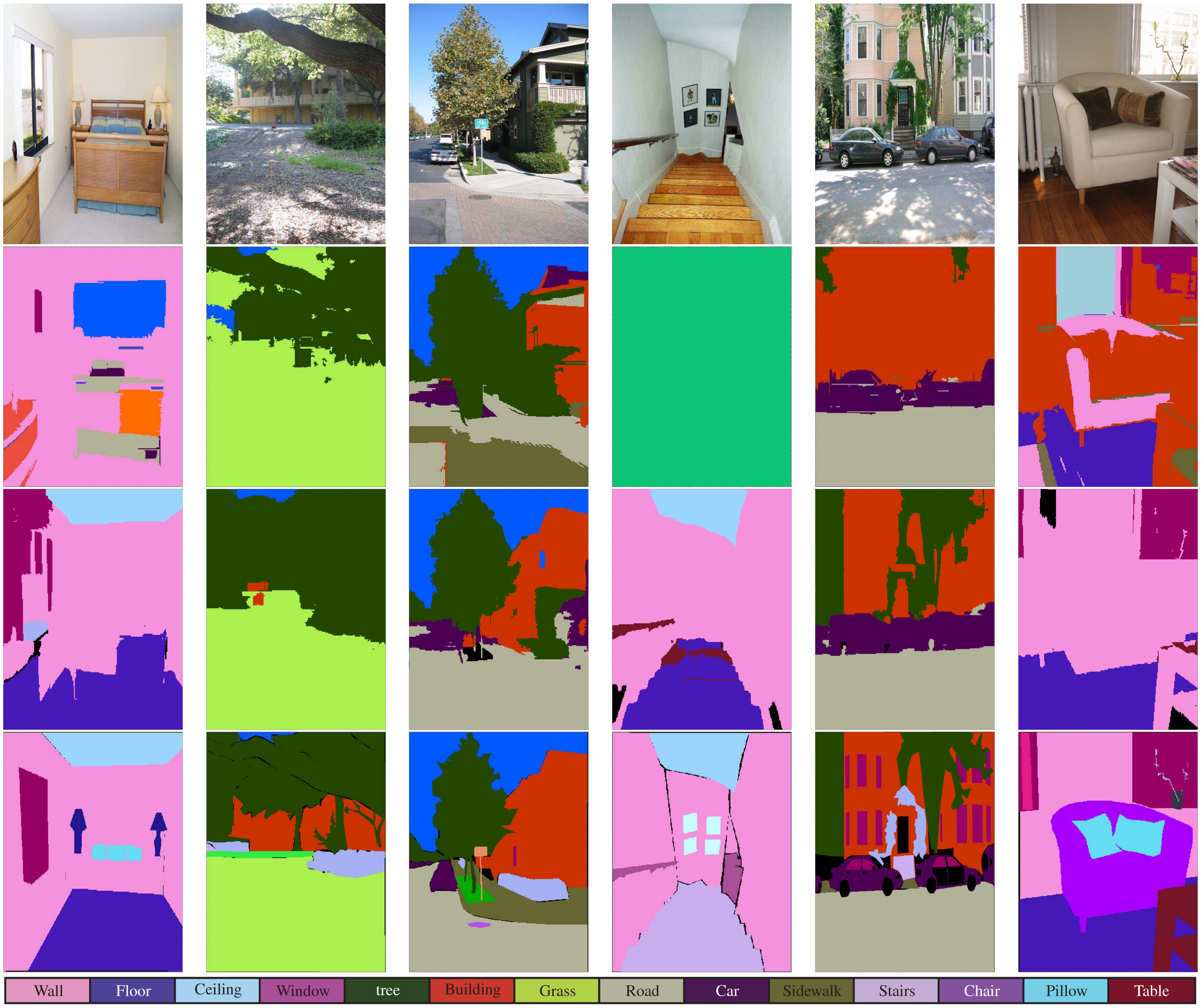


Figure 7: **Qualitative comparison of our results with those of Superparsing [28] on LM-SUN. 1st row:** Query image; **2nd row:** Superparsing; **3rd row:** Our approach; **4th row:** Ground-truth.

and filtering. Instead of using a fixed retrieval set of images, our approach samples labeled superpixels, thus allowing us to obtain a more balanced set of data. This, in conjunction with our efficient filtering-based label transfer procedure, has proven effective at handling large-scale datasets. In particular, our approach has achieved accuracies that are competitive with the state-of-the-art nonparametric methods, while being faster than them. In the future, we intend to study better image similarity metrics, which, as evidenced by our analysis, has potential to further boost our accuracy.

References

[1] A. Adams, J. Baek, and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, 2010. 3

[2] A. Adams, N. Gelfand, J. Dolson, and M. Levoy. Gaussian kd-trees for fast high-dimensional filtering. *ACM Transactions on Graphics*, 28(3):21:1–21:12, 2009. 3

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3), 2009. 2

[4] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *ECCV*, 2010. 2

[5] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *CVPR*, 2012. 1, 2, 5, 6

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*, 2012. 1, 2

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 5

[8] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics*, 30(4):69:1–69:12, 2011. 3

[9] M. George. Image parsing with a wide range of classes and scene-level context. In *CVPR*, 2015. 1, 2

[10] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 1, 2

[11] S. Gould and Y. Zhang. Patchmatchgraph: Building a graph of dense patch correspondences for label transfer. In *ECCV*, 2012. 1, 2, 5

[12] P. Kohli, L. Ladický, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009. 1, 2

[13] P. Krähenbühl and V. Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013. 1, 2, 5, 6

[14] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical random fields. *PAMI*, 36(6):1056–1077, 2014. 1, 2

[15] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Inference methods for crfs with co-occurrence statistics. *IJCV*, 2013. 1, 2

[16] B. Liu and X. He. Multiclass semantic video segmentation with object-level active inference. In *CVPR*, 2015. 1, 2

[17] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 33(12):2368–2382, 2011. 1, 2, 5

[18] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(12):2368–2382, 2011. 2, 6

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2

[20] H. Myeong, J. Y. Chang, and K. M. Lee. Learning object relationships via graph-based context model. In *CVPR*, 2012. 1, 2, 5, 6

[21] H. Myeong and K. M. Lee. Tensor-based high-order semantic relation transfer for semantic scene segmentation. In *CVPR*, 2013. 1, 2, 5

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 3

[23] A. Sharma, O. Tuzel, and D. W. Jacobs. Deep hierarchical parsing for semantic segmentation. In *CVPR*, 2015. 1, 2

[24] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for mulit-class object recognition and segmentation. In *ECCV*, 2006. 1, 2

[25] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao. Integrating parametric and non-parametric models for scene labeling. In *CVPR*, 2015. 1, 2

[26] G. Singh and J. Kosecka. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *CVPR*, 2013. 1, 2, 5, 6

[27] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 1, 2

[28] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. *IJCV*, 101(2):329–349, 2013. 1, 2, 5, 6, 7, 8

[29] F. Tung and J. J. Little. Collageparsing: Nonparametric scene parsing by adaptive overlapping windows. In *ECCV*, 2014. 1, 2, 5, 6

[30] C. K. Wong and M. C. Easton. An efficient method for weighted sampling without replacement. siam journal of computing. *SIAM Journal o Computing*, 9(1):111–113, 1980. 3

[31] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 3

[32] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014. 1, 2