

Semantic Texton Forests for Image Categorization and Segmentation

Jamie Shotton[†]

Matthew Johnson^{*}

Roberto Cipolla^{*}

[†]Toshiba Corporate R&D Center
Kawasaki, Japan

^{*}Department of Engineering
University of Cambridge, UK

Abstract

We propose semantic texton forests, *efficient and powerful new low-level features*. These are ensembles of decision trees that act directly on image pixels, and therefore do not need the expensive computation of filter-bank responses or local descriptors. They are extremely fast to both train and test, especially compared with *k*-means clustering and nearest-neighbor assignment of feature descriptors. The nodes in the trees provide (i) *an implicit hierarchical clustering into semantic textons*, and (ii) *an explicit local classification estimate*. Our second contribution, the bag of semantic textons, combines a histogram of semantic textons over an image region with a region prior category distribution. The bag of semantic textons is computed over the whole image for categorization, and over local rectangular regions for segmentation. Including both histogram and region prior allows our segmentation algorithm to exploit both textural and semantic context. Our *third contribution is an image-level prior for segmentation that emphasizes those categories that the automatic categorization believes to be present*. We evaluate on two datasets including the very challenging VOC 2007 segmentation dataset. Our results significantly advance the state-of-the-art in segmentation accuracy, and furthermore, our use of efficient decision forests gives at least a five-fold increase in execution speed.

1. Introduction

This paper introduces *semantic texton forests*, and demonstrates their use for image categorization and semantic segmentation; see Figure 1. Our aim is to show that one can build powerful texton codebooks *without* computing expensive filter-banks or descriptors, and *without* performing costly *k*-means clustering and nearest-neighbor assignment. Semantic texton forests (STFs) fulfill both criteria. They are randomized decision forests that use only simple pixel comparisons on local image patches, performing both an implicit hierarchical clustering into semantic textons and an explicit local classification of the patch category. Our results show that STFs improve the state-of-the-art in *both*

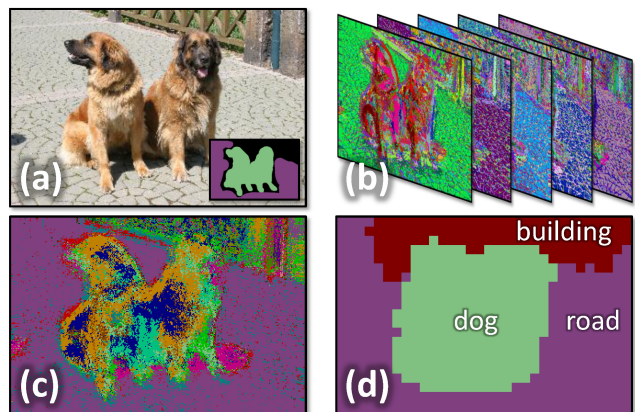


Figure 1. **Semantic texton forests.** (a) Test image, with ground truth in-set. Semantic texton forests very efficiently compute (b) a set of semantic textons per pixel and (c) a rough local segmentation prior. Our algorithm uses both textons and priors as features to give coherent semantic segmentation (d), and even finds the building unmarked in the ground truth. Colors show texton indices in (b), but categories corresponding to the ground truth in (c) and (d).

quantitative performance and execution speed.

We look at two applications of STFs: image categorization (inferring the object categories present in an image) and semantic segmentation (dividing the image into coherent regions and simultaneously categorizing each region). To these ends, we propose the *bag of semantic textons*. This is computed over a given image region, and extends the bag of words model [4] by combining a histogram of the hierarchical semantic textons with a region prior category distribution. By considering the image as a whole, we obtain a highly discriminative descriptor for categorization. For segmentation, we use many local rectangular regions and build a second randomized decision forest that achieves efficient and accurate segmentation.

Inferring the correct segmentation depends on local image information that can often be ambiguous. The global statistics of the image, however, are more discriminative and may be sufficient to accurately estimate the image categorization. We therefore investigate how categorization can

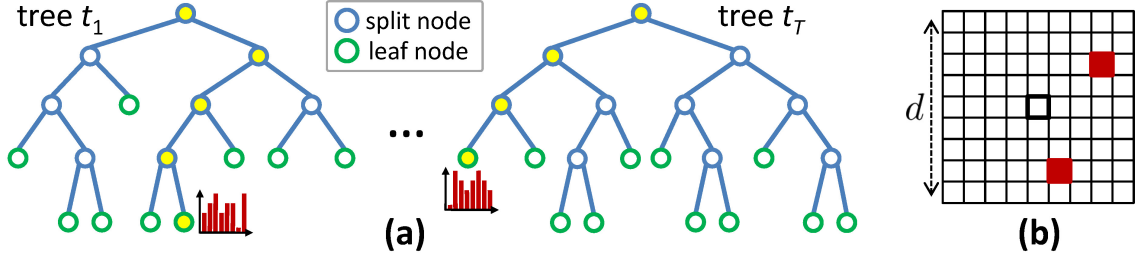


Figure 2. **(a) Decision forests.** A forest consists of T decision trees. A feature vector is classified by descending each tree. This gives, for each tree, a path from root to leaf, and a class distribution at the leaf. As an illustration, we highlight the root to leaf paths (yellow) and class distributions (red) for one input feature vector. This paper shows how to simultaneously exploit both the hierarchical clustering implicit in the tree structure and the node class distributions. **(b) Semantic texton forests features.** The split nodes in semantic texton forests use simple functions of raw image pixels within a $d \times d$ patch: either the raw value of a single pixel, or the sum, difference, or absolute difference of a pair of pixels (red).

act as an *image-level prior* to improve segmentation by emphasizing the categories most likely to be present.

To summarize, the main contributions of this work are:

(i) semantic texton forests which efficiently provide both a hierarchical clustering into semantic textons and a local classification; (ii) the bag of semantic textons model, and its applications in categorization and segmentation; and (iii) the use of the image-level prior to improve segmentation performance.

The paper is structured as follows. Section 2 gives a brief recap of randomized decision forests which form the basis of our new semantic texton forests, introduced in Section 3. These are used for image categorization in Section 4 and segmentation in Section 5. We evaluate and compare with related work in Section 6, and conclude in Section 7.

Related Work. Textons [11, 17, 31] and visual words [28] have proven powerful discrete image representations for categorization and segmentation [4, 27, 34, 37]. Filterbank responses (derivatives of Gaussians, wavelets, *etc.*) or invariant descriptors (*e.g.* SIFT [16]) are computed across a training set, either at sparse interest points (*e.g.* [19]) or more densely; recent results in [22] suggest that densely sampling visual words improves categorization performance. The collection of descriptors are then clustered to produce a codebook of visual words, typically with the simple but effective k -means, followed by nearest-neighbor assignment. Unfortunately, this three stage process is extremely slow and often the most time consuming part of the whole system, even with optimizations such as k d-trees, the triangle inequality [5], or hierarchical clusters [21, 26].

The recent work of Moosmann *et al.* [20] proposed a more efficient alternative, in which training examples are recursively divided using a randomized decision forest [1, 8] and where the splits in the decision trees are comparisons of a descriptor dimension to a threshold. Semantic texton forests, the main contribution of this paper, extend [20] in three ways: (i) we learn directly from image pixels, bypassing the expensive step of computing image descriptors; (ii) while [20] use the learned decision forest only for cluster-

ing, we also use it as a classifier, which enables us to use semantic context for image segmentation; and (iii) in addition to the leaf nodes used in [20], we include the branch nodes as hierarchical clusters. A related method, the *pyramid match kernel* [9], exploits a hierarchy in descriptor space, though it requires the computation of feature descriptors and is only applicable to kernel based classifiers. The pixel-based features we use are similar to those in [14], but our forests are trained to recognize object categories, not match particular feature points.

Other work has also looked at alternatives to k -means. Recent work [30] quantizes feature space into a hyper-grid, but requires descriptor computation and can result in very large visual word codebooks. Winder & Brown [33] learned the parameters of generic image descriptors for 3D matching, though did not address visual word clustering. Jurie & Triggs [12] proposed building codebooks using mean shift, but did not incorporate semantic information in the codebook generation.

2. Randomized Decision Forests

We begin with a brief review of randomized decision forests [1, 8]. As illustrated in Figure 2(a), a decision forest is an ensemble of T decision trees. Associated with each node n in the tree is a learned class distribution $P(c|n)$. A decision tree works by recursively branching left or right down the tree according to a learned binary function of the feature vector, until a leaf node l is reached. The whole forest achieves an accurate and robust classification by averaging the class distributions over the leaf nodes $L = (l_1, \dots, l_T)$ reached for all T trees:

$$P(c|L) = \frac{1}{T} \sum_{t=1}^T P(c|l_t). \quad (1)$$

Existing work has shown the power of decision forests as either classifiers [2, 14, 18] or a fast means of clustering descriptors [20]. In this work we show how to simultaneously exploit *both* classification and clustering. Furthermore, we generalize [20] to use the tree hierarchies as hierarchical clusters.

Randomized Learning. We use the extremely randomized trees algorithm [8] to learn binary forests. Each tree is trained separately on a small random subset $I' \subseteq I$ of the training data I . Learning proceeds recursively, splitting the training data I_n at node n into left and right subsets I_l and I_r according to a threshold t of some split function f of the feature vector \mathbf{v} :

$$I_l = \{i \in I_n \mid f(\mathbf{v}_i) < t\}, \quad (2)$$

$$I_r = I_n \setminus I_l. \quad (3)$$

At each split node, several candidates for function f and threshold t are generated randomly, and the one that maximizes the expected gain in information about the node categories is chosen [14]:

$$\Delta E = -\frac{|I_l|}{|I_n|} E(I_l) - \frac{|I_r|}{|I_n|} E(I_r), \quad (4)$$

where $E(I)$ is the Shannon entropy of the classes in the set of examples I . The recursive training continues to a maximum depth D or until no further information gain is possible. The class distributions $P(c|n)$ are estimated empirically as a histogram of the class labels c_i of the training examples i that reached node n .

Implementation Details. The amount of training data may be significantly biased towards certain classes in some datasets. A classifier learned on this data will have a corresponding prior preference for those classes. To normalize for this bias, we weight each training example by the inverse class frequency as $w_i = \xi_{c_i}$ with $\xi_c = (\sum_{i \in I} [c = c_i])^{-1}$. The classifiers trained using this weighting tend to give a better class average performance. After training, an improved estimate of the class distributions is obtained using *all* training data I , not just the random subset I' . We found this to improve the generalization of the classifiers slightly, especially for classes with few training examples.

Advantages. Using ensembles of trees trained on only small random subsets of the data helps to speed up training time and reduce over-fitting [1]. The trees are fast to learn and extremely fast to evaluate since only a small portion of the tree is traversed for each data point.

3. Semantic Texton Forests

Semantic texton forests (STFs) are randomized decision forests used for both clustering and classification. The split functions f in STFs act on small image patches \mathbf{p} of size $d \times d$ pixels, illustrated in Figure 2(b). These functions can be (i) the value $p_{x,y,b}$ of a single pixel (x, y) in color channel b , or (ii) the sum $p_{x_1,y_1,b_1} + p_{x_2,y_2,b_2}$, (iii) difference $p_{x_1,y_1,b_1} - p_{x_2,y_2,b_2}$, or (iv) absolute difference $|p_{x_1,y_1,b_1} - p_{x_2,y_2,b_2}|$ of a pair of pixels (x_1, y_1) and (x_2, y_2) from possibly different color channels b_1 and b_2 . We visualize some learned semantic textons in Figure 3.

To textonize an image, a $d \times d$ patch centered at each pixel is passed down the STF resulting in semantic texton

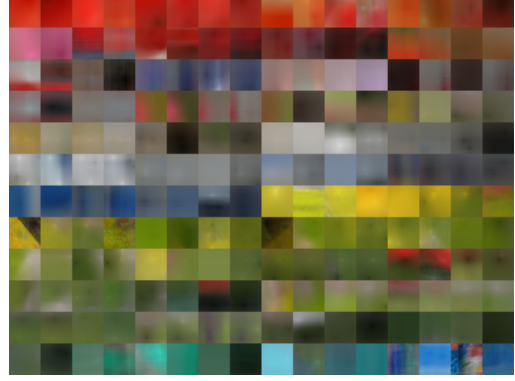


Figure 3. **Semantic textons.** A visualization of leaf nodes from one tree (distance $d = 21$ pixels). Each patch is the average of all patches in the training images assigned to a particular leaf node l . Features evident include color, horizontal, vertical and diagonal edges, blobs, ridges and corners. Associated with each semantic texton is a learned distribution $P(c|l)$, used to give the rough local segmentation of Figure 1(c).

leaf nodes $L = (l_1, \dots, l_T)$ and the averaged class distribution $P(c|L)$. Examples are shown in Figure 1. A pixel-level classification based on the local distributions $P(c|L)$ gives poor but still surprisingly good performance (see Section 6.1). However, by pooling the statistics of semantic textons L and distributions $P(c|L)$ over an image region, the *bag of semantic textons* presented below in Section 3.1 forms a much more powerful feature for image categorization and semantic segmentation.

Learning Invariances. Although using raw pixels as features is much faster than first computing descriptors or filter-bank responses, one risks losing their inherent invariances. To avoid this, we augment the training data with image copies that are artificially transformed geometrically and photometrically [14]. This allows one to *learn* the right degree of invariance required for a particular problem. In our experiments we explored rotation, scaling, and left-right flipping as geometric transformations, and affine photometric transformations.

Implementation Details. An STF can be trained using (i) pixel-level supervision, (ii) weak supervision, in which the members of the set of classes present in the whole image are used as training labels for all pixels, or (iii) no supervision, where the split function that most evenly divides the data is used. In the unsupervised case, the STF forest acts only as a hierarchical clusterer, not a classifier. We examine the effect of the level of supervision in Section 6. We found the CIELab color space to generalize better than RGB, and it is used in all experiments. Training examples are taken on a regular grid (every 5×5 pixels) in the training images, excluding a narrow band of $\frac{d}{2}$ pixels around the image border to avoid artifacts; at test time, the image is extended to ensure a smooth estimate of the semantic textons near the border.

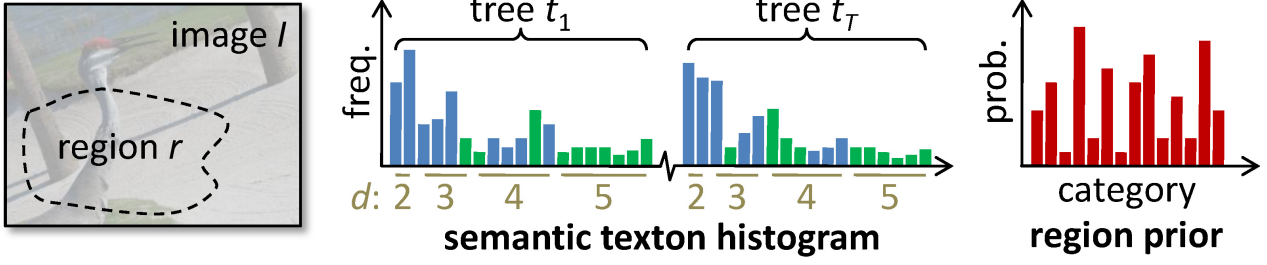


Figure 4. **Bags of semantic textons.** Within a region r of image I we generate the semantic texton histogram and region prior. The histogram incorporates the implicit hierarchy of clusters in the STF, containing both STF leaf nodes (green) and split nodes (blue). The depth d of the nodes in the STF is shown. The STFs need not be to full depth, and empty bins in the histogram are not shown as the histogram is stored sparsely. The region prior is computed as the average of the individual leaf node class distributions $P(c|l)$.

3.1. Bags of Semantic Textons

A popular and powerful method for categorizing images and detecting objects is the bag of words model [4, 28, 37]. A histogram of visual words is created over the whole image or a region of interest [3], either discarding spatial layout or using a spatial hierarchy [13]. The histogram is used as input to a classifier to recognize object categories. We propose the localized bag of semantic textons (BoST), illustrated in Figure 4. This extends the bag of words with low-level semantic information, as follows.

Given for each pixel i the leaf nodes $L_i = (l_1, \dots, l_T)_i$ and inferred class distribution $P(c|L_i)$, one can compute over image region r (i) a non-normalized histogram $H_r(n)$ that concatenates the occurrences of tree nodes n across the different trees [20], and (ii) a prior over the region given by the average class distribution $P(c|r) = \sum_{i \in r} P(c|L_i)$. In contrast to [20], we include both leaf nodes l and split nodes n in the histogram, such that $H_r(n) = \sum_{n' \in \text{child}(n)} H_r(n')$. The histogram therefore uses the hierarchy of clusters implicit in each tree. Each $P(c|L_i)$ is already averaged across trees, and hence there is a single region prior $P(c|r)$ for the whole forest.

Our results in Section 6 show that the histograms and region priors are complementary, and that the hierarchical clusters are better than the leaf node clusters alone. For categorization (Section 4), we use BoSTs where the region is the whole image. For segmentation (Section 5), we use a learned combination of BoSTs over many local rectangular regions to model layout and context.

Implementation Details. The counts of tree root nodes hold no useful information and are not included in the histograms. The histograms are sparse near the leaves, and can be stored efficiently since the histogram counts at split nodes can be quickly computed on-the-fly. If region r is rectangular, the histograms and class distributions can be calculated very quickly using integral histograms [24].

4. Image Categorization

The task of image categorization is to determine those categories (e.g. dog images, beach images, indoor images) to which an image belongs. Previous approaches use global

image information [23], bags of words [7] or textons [34]. We propose a new image categorization algorithm that exploits the hierarchy of semantic textons and the node prior distributions $P(c|n)$.

For this we use a non-linear support vector machine (SVM). This depends on a kernel function K that defines the similarity measure between images. To take advantage of the hierarchy in the STF, we adapt the pyramid match kernel [9] to act on a pair of BoST histograms computed across the whole image.

Consider first the BoST histogram computed for just one tree in the STF. The kernel function (based on [9]) is then

$$K(P, Q) = \frac{1}{\sqrt{Z}} \tilde{K}(P, Q), \quad (5)$$

where Z is a normalization term for images of different sizes computed as $Z = \tilde{K}(P, P) \tilde{K}(Q, Q)$, and \tilde{K} is the actual matching function, computed over levels of the tree as

$$\tilde{K}(P, Q) = \sum_{d=1}^D \frac{1}{2^{D-d+1}} (\mathcal{I}_d - \mathcal{I}_{d+1}). \quad (6)$$

using the histogram intersection \mathcal{I}

$$\mathcal{I}_d = \sum_j \min(P_d[j], Q_d[j]), \quad (7)$$

where D is the depth of the tree, P and Q are the hierarchical histograms, and P_d and Q_d are the portions of the histograms at depth d , with j indexing over all nodes at depth d . There are no nodes at depth $D + 1$, hence $\mathcal{I}_{D+1} = 0$. If the tree is not full depth, missing nodes j are simply assigned $P_d[j] = Q_d[j] = 0$.

The kernel over all trees in the STF is calculated as $K = \sum_t \gamma_t K_t$ with mixture weights γ_t . Similarly to [37], we found $\gamma_t = \frac{1}{T}$ to result in the best categorization results. This method is very effective, but can be improved by using the learned ‘prior’ distributions $P(c|n)$ in the STF. We build a 1-vs-others SVM kernel K_c per category, in which the count for node n in the BoST histogram is weighted by the value $P(c|n)$. This helps balance the categories, by selectively down-weighting those that cover large image areas

(e.g. grass, water) and thus have inappropriately strong influence on the pyramid match, masking the signal of smaller classes (e.g. cat, bird).

In Section 6.2, we show the improvement that the pyramid match kernel on the hierarchy of semantic textons gives over a radial basis function on histograms of just leaf nodes. We also obtain an improvement using the per-category kernels K_c instead of a global kernel K . Finally, we show how this categorization can act as an image-level prior for segmentation in Section 5.1.

5. Semantic Segmentation

To demonstrate the power of the BOSTs as features for segmentation, we adapt the TextonBoost algorithm [27]. The goal is to segment an image into coherent regions and simultaneously infer the class label of each region (see Figure 6). In [27], a boosting algorithm selected features based on localized counts of textons to model patterns of texture, layout and context. The context modeled in [27] was ‘textural’, for example: sheep often stand on something green. We adapt the rectangle count features of [27] to act on both the semantic texton histograms and the BOST region priors. The addition of region priors allows us to model context based on *semantics* [25], not just texture. Continuing the example, our new model can capture the notion that sheep often stand on grass.

The segmentation algorithm works as follows. For speed we use a second randomized decision forest in place of boosting. We train this *segmentation forest* to act at image pixels i , using pixels on a regular grid as training examples. At test time, the segmentation forest is applied at each pixel i densely or, for more speed, on a grid. The most likely class in the averaged category distribution (1) gives the final segmentation for each pixel. The split node functions f compute either the count $H_{r+i}(n = n')$ of semantic texton n' , or the probability $P(c = c' | r + i)$ of class c' , within rectangle r translated relative to pixel i . By translating rectangle r relative to the pixel i being classified, and by allowing r to be a large distance away from i (up to half the image size), such features can exploit texture, layout and context information (see [27] for a detailed explanation). Our extension to these features exploits semantic context by using the region prior probabilities $P(c | r + i)$ inferred by the semantic textons. We show the benefit this brings in Section 6.3.

5.1. Image-Level Prior

We could embed the above segmentation forest in a conditional random field model to achieve more coherent results or to refine the grid segmentation to a per-pixel segmentation [10, 27]. Instead, we decided to investigate a simpler and more efficient approach using the powerful image categorizer we built in Section 4. For each test image we separately run the categorization and segmentation algorithms. This gives an image-level prior (ILP)

distribution $P(c)$ and a per-pixel segmentation distribution $P(c|i)$ respectively. We use the ILP to emphasize the likely categories and discourage unlikely categories, by multiplying the (quasi-independent) distributions as $P'(c|i) = P(c|i)P(c)^\alpha$, using parameter α to soften the prior. We show in Section 6.3 and Figure 6 how the addition of the ILP gives a considerable improvement to the resulting segmentations. Li & Fei-Fei [15] proposed a related idea that uses scene categorization as priors for object detection.

6. Experiments

We performed experiments on two challenging datasets:

	classes	images train	images test
MSRC [27]	21	276	256
VOC 2007 (Seg) [6]	21	422	210

We use the standard train/test splits, and the hand-labeled ground truth to train the classifiers. Image categorization performance is measured as mean average precision [6]. Segmentation performance is measured as both the category average accuracy (the average proportion of pixels correct in each category) and the global accuracy (total proportion of pixels correct). The category average is fairer and more rigorous, as it normalizes for category bias in the test set. Training and test times are reported using an unoptimized C# implementation on a single 2.7GHz core.

6.1. Semantic Texton Forests

Before presenting in-depth results for categorization and segmentation, let us look briefly at the STFs themselves. In Figure 1, we visualize the inferred leaf nodes $L = (l_1, \dots, l_T)$ for each pixel i and the most likely category $c_i^* = \arg \max_{c_i} P(c_i | L)$. Observe that the textons in each tree capture different aspects of the underlying texture and that even at such a low level the distribution $P(c | L)$ contains significant semantic information. Table 1 gives a naïve segmentation baseline on the MSRC dataset by comparing c_i^* to the ground truth.

	Global	Average
supervised	49.7%	34.5%
weakly supervised	14.8%	24.1%

Table 1. Naïve segmentation baseline on MSRC.

Clearly, this segmentation is poor, especially when trained in a weakly supervised manner, since only very local appearance and no context is used. Even so, the signal is remarkably strong for such simple features (random chance is under 5%). We show below how using semantic textons as features in higher level classifiers greatly improves these numbers, even with weakly supervised or unsupervised STFs.

Except where otherwise stated, we used STFs with the following parameters, hand-optimized on the MSRC validation set: distance $d = 21$, $T = 5$ trees, maximum depth $D = 10$, 500 feature tests and 10 threshold tests per split,

and $\frac{1}{4}$ of the data per tree, resulting in approximately 500 leaves per tree. Training the STF on the MSRC dataset took only 15 minutes.

6.2. Image Categorization

We performed an experiment on the MSRC data to investigate our SVM categorization algorithm. The mean average precisions (AP) in Table 2 compare our modified pyramid match kernel (PMK) to a radial basis function (RBF) kernel, and compare the global kernel K to the per-category kernels K_c . In the baseline results with the RBF kernel, only the leaf nodes of the STF are used, separately per tree, using term frequency/inverse document frequency to normalize the histogram. The PMK results use the entire BOST which for the per-category kernels K_c are weighted by the prior node distributions $P(c|n)$. Note that the mean AP is a much harder metric and gives lower numbers than recall precision or AuC; the best result in the table shows very accurate categorization.

	Global kernel K	Per-category kernel K_c
RBF	49.9	52.5
PMK	76.3	78.3

Table 2. **Image categorization results.** (Mean AP).

As can be seen, the pyramid match kernel considerably improves on the RBF kernel. By training a per-category kernel, a small but noticeable improvement is obtained. For the image-level prior for segmentation, we thus use the PMK with per-category kernels. In Figure 5 we plot the global kernel performance against the number T of STF trees, and see that categorization performance increases with more trees though eventually levels out.

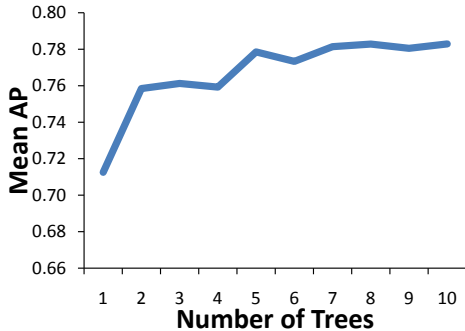


Figure 5. **Categorization accuracy vs number of STF trees.**

6.3. Semantic Segmentation

MSRC Dataset [27]. We first examine the influence of different aspects of our system on segmentation accuracy. We trained segmentation forests using (a) the histogram $H_r(l)$ of just leaf nodes l , (b) the histogram $H_r(n)$ of *all* tree nodes n , (c) just the region priors $P(c|r)$, (d) the full model using all nodes and region priors, (e) the full model trained

without random transformations, (f) all nodes using an unsupervised STF (no region priors are available), and (g) all nodes using a weakly-supervised STF (only image labels). The category average accuracies are given in Table 3 with and without the image-level prior (ILP).

	Without ILP	With ILP
(a) only leaves	61.3%	64.1%
(b) all nodes	63.5%	65.5%
(c) only region priors	62.1%	66.1%
(d) full model	63.4%	66.9%
(e) no transformations	60.4%	64.4%
(f) unsupervised STF	59.5%	64.2%
(g) weakly supervised STF	61.6%	64.6%

Table 3. **Comparative segmentation results on MSRC.**

There are several conclusions to draw. (1) In all cases the ILP improves results. (2) The hierarchy of clusters in the STF gives a noticeable improvement. (3) The region priors alone perform remarkably well. Comparing to the segmentation result using only the STF leaf distributions (34.5%) this shows the power of the localized BOSTs that exploit semantic context. (4) Each aspect of the BOST adds to the model. While, without the ILP, score (b) is slightly better than the full model (d), adding in the ILP shows how the region priors and textons work together.¹ (5) Random transformations of the training images improve performance by adding invariance. (6) Performance increases with more supervision, but even unsupervised STFs allow good segmentations.

Given this insight, we compare against [27] and [32]. We use the same train/test split as [27] (though not [32]). The results are summarized in Figure 6. Across the whole challenging dataset, using the full model with ILP achieved a class average performance of 66.9%, a significant improvement on both the 57.7% of [27] and the 64% of [32]. The global accuracy also improves slightly on [27]. The image-level prior improves performance for all but three classes, but even without it, results are still highly competitive with other methods. Our use of balanced training has resulted in more consistent performance across classes, and significant improvements for certain difficult classes: cow, sheep, bird, chair, and cat. We do not use a Markov or conditional random field, which would likely further improve our performance [27].

These results used our novel learned and extremely fast STFs, without needing any slow hand-designed filter-banks or descriptors. Extracting the semantic textons at every pixel takes an average of only 275 milliseconds per image, categorization takes 190 ms, and evaluating the segmentation forest only 140 ms. For comparison [27] took over 6 seconds per test image, and [32] took an average of over 2

¹This effect may be due to segmentation forest (b) being overconfident: looking at the 5 most likely classes inferred for each pixel, (b) achieves 87.6% while (d) achieves a better 88.0%.

seconds per image for feature extraction and between 0.3 to 2 seconds for estimating the segmentation. Our algorithm is well over 5 times faster *and* improves quantitative results. Minor optimizations have subsequently led to a real-time system that runs at over 8 frames per second.

VOC 2007 Segmentation Dataset [6]. This very new dataset contains 21 extremely challenging categories including background. We trained an STF, a segmentation forest, and an ILP on this data, using the ‘trainval’ split and keeping parameters as for MSRC. The results in Figure 7 compare with [6]. Our algorithm performs over twice as well as the only *segmentation* entry (Brookes), and the addition of the ILP further improves performance by 4%. The actual winner of the segmentation challenge, the TKK algorithm, used *segmentation-by-detection* that fills in detected object bounding boxes by category. To see if our algorithm could use a *detection*-level prior DLP (identical to the ILP but using the detected bounding boxes and varying with image position) we took the TKK entry output as the DLP. Our algorithm gave a large 12% improvement over the TKK *segmentation-by-detection*, highlighting the power of STFs as features for segmentation.

7. Conclusions

This paper presented semantic texton forests as efficient texton codebooks. These do not depend on local descriptors or expensive k -means clustering, and when supervised during training they can infer a distribution over categories at each pixel. We showed how bags of semantic textons enabled state-of-the-art performance on challenging datasets for image categorization and semantic segmentation, and how the use of an inferred image-level prior significantly improves segmentation results. The substantial gains of our method over traditional textons are training and testing efficiency and improved quantitative performance.

The main limitation of our system is the large dimensionality of the bag of semantic textons. This necessitates a trade-off between the memory usage of the semantic texton integral images and the training time if they are computed on-the-fly. Using just the region priors is however more memory efficient.

As future work, we are interested in reducing the level of supervision needed for the segmentation forests, perhaps using latent topic models [32]. We would like to build STFs on local image *volumes*, either 3D medical images or video sequences [36]. Our system could benefit from further knowledge of the individual objects present [35]. Finally, efficient sparse visual words may be possible by combining efficient detectors [29] with our semantic texton forests.

Acknowledgements. We thank John Winn, Bjorn Stenger, Kenichi Maeda, Osamu Yamaguchi, and Ville Viitaniemi for their valuable help and discussions.

References

- [1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.
- [2] A. Bosch, A. Zisserman, and X. Muñoz. Image classification using random forests and ferns. In *ICCV*, 2007.
- [3] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007.
- [4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.
- [5] C. Elkan. Using the triangle inequality to accelerate k -means. In *Proc. Int. Conf. on Machine Learning*, pages 147–153, 2003.
- [6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC Challenge 2007. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [7] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.
- [8] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.
- [9] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.
- [10] X. He, R. Zemel, and M. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, volume 2, pages 695–702, June 2004.
- [11] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, Mar. 1981.
- [12] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, volume 1, pages 604–610, 2005.
- [13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
- [14] V. Lepetit, P. Lager, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR*, pages 2:775–781, 2005.
- [15] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007.
- [16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov. 2004.
- [17] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, June 2001.
- [18] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *CVPR*, 2005.
- [19] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, Oct. 2004.
- [20] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2006.
- [21] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.
- [22] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.
- [23] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Visual Perception, Progress in Brain Research*, 155(1):23–26, 2006.
- [24] F. M. Porikli. Integral Histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829–836, 2005.
- [25] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.
- [26] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, Minneapolis, June 2007.
- [27] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, To appear 2007.
- [28] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 2: 1470–1477, 2003.

