Real-World Material Recognition for Scene Understanding

Sam Corbett-Davies
Department of Computer Science
Stanford University
scorbett@stanford.edu

Abstract

In this paper we address the problem of recognizing materials in consumer photographs. While material recognition isn't a new problem, the introduction of the OpenSurfaces dataset [1], allows it to be studied at a new scale. In particular, the dataset provides materials in a huge variety of real-world environments, with dramatic appearance and shading differences within each a material class. We propose a discriminative learning framework for the per-pixel classification of materials in an image. Huge appearance variation makes classifying some material classes extremely challenging - our method achieves only 34.5% classification accuracy. However, we show that even this weak material signal can be valuable for scene understanding. We use the output of our classifier as a new feature in a recent RGB-D scene understanding algorithm. We improve this state-of-the-art scene understanding method by 0.7%.

1. Introduction

The material of an object has the potential to be a very salient piece of information for the understanding of scenes. From a simple visual examination of an object, humans are often able to judge its weight, texture and function by estimating its material composition. Consider the bottles in Fig. 1. A good object detector would most likely label all three as "bottles". However, by judging each bottle's material composition, a human could sort the bottles by weight, determine which would break when dropped, or even determine which would be best to hold very hot liquid. Knowledge of such properties would be crucial for a personal robot, making material recognition a very interesting direction for scene understanding research. Despite this, material properties have been scarcely explored in the existing scene understanding literature.

This work is motivated by the introduction of new large-scale database of materials in real-world environments by Bell *et al.* called OpenSurfaces [1]. The dataset contains tens of thousands of Flickr images, with each surface segmented and annotated with surface normal, specularity, diffuse color, roughness, and scene context information including the associated object and scene type.

While other datasets have been developed for texture recognition [2, 3], material recognition "in the wild" (as opposed to under controlled conditions) has only recently been

Figure 1: An example of a situation where material recognition would be valuable. The three bottles, while similar in shape, have vastly different physical properties, which can be inferred from their material composition.

investigated, and no other dataset achieves the quantity of real-world scenes that OpenSurfaces does. For instance, in [2] Sharan *et al.* present a dataset of 100 Flickr images for each of 10 material categories. In contrast, OpenSurfaces has 25,000 scenes with over 110,000 segmented materials in 53 classes.

We hope that an understanding of the material composition of a scene will aid in the scene understanding problem, which requires the semantic labeling (ie sofa, chair, TV etc) of each pixel in an image to be determined. For this reason, our approach seeks to determine per-pixel material labels, while Sharan *et al.* [2] and subsequent works on their dataset [4] only determined the dominant material in an image.

Our method builds on the framework developed by Ladický and Torr for scene understanding [5, 6]. The underlying method is a Conditional Random Field (CRF) over the image with higher-order potentials to ensure segments are smoothly and consistently labeled, but we will only consider the unary potential in this paper. Materials are classified with a Random Forest that uses four features: SIFT [7], Color SIFT [8], Linear Binary Pattern (LBP) [9], and Textons [10]. These features are described in more detail in Section 3.1.

The rest of this paper is organized as follows: Section 2 surveys existing material recognition methods. Section 3 describes our machine learning approach, with the features detailed in Section 3.1 and the algorithm used described in 3.2. In Section 4 we present out results for material classification on the OpenSurfaces dataset, then in Section 5 we incorporate our classifier into an existing scene understand-

ing framework, improving its overall accuracy from 75.5% to 76.2%. Finally, we discuss our findings and draw conclusions in Section 6.

2. Related Works

The seminal work considering texture for computer vision was done by Dana and Ginneken [3], where they presented the CUReT dataset. Their thorough investigation of the visual properties of surface texture prompted early texture recognition approaches such as [10] (who developed the Texton feature for this purpose). However this dataset was generated in a controlled environment and only contained planar texture patches, and as a result it was far too easy to achieve good results on; [11] demonstrated over 95% classification accuracy on the CUReT dataset, but only 23% accuracy on the Flickr dataset of real world materials introduced by [2].

With the focus of material recognition shifting to real-world examples, recent research has shown promising progress on this challenging task. [4] developed a number of new features for classifying materials, including computing HOG features [] along and across image edges. These features were quantized into visual words using k-means clustering, and a Bayesian model was learned to combine them into a classifier. This approach achieved 45% accuracy on the Flickr image dataset, but only allowed for a single material per image, so it is not directly applicable to scene understanding.

[12] improved on the performance of Liu *et al.* on the Flickr dataset, achieving 54% classification accuracy. This was done using the kernel descriptors developed in [13]. They develop more expressive features by realizing that popular image descriptors (SIFT, HOG) can be generalized as kernels over image patches. They also quantize their features into visual words, but using Large Margin Nearest Neighbor instead of k-means. Again, this approach predicts only the principal material the is present in an image.

3. Method

Our approach is built on the publicly available ALE framework for scene understanding [5], which uses a CRF model. In this paper we focus on only the unary potential, which is a discriminative algorithm for classifying the material of each pixel. The CRF model also contains higher-order potentials to smooth the pixel-wise classification result, but they are not explored in this project.

3.1. Features

Our learning algorithm determines the material of each pixel as a function of four feature vectors. These are SIFT, Color SIFT, Textons and LBP. Each feature is described in more detail below.

3.1.1 SIFT and Color SIFT

SIFT [7] was developed by Lowe over ten years ago and remains the most popular image descriptor in computer vision research. Only keypoints that persist at different scales are used (making the descriptor scale invariant), and the local image gradients around these points computed. By binning the gradient orientations relative to the dominant orientation, the descriptor is also rotation invariant. The result is a 128 dimensional feature vector.

Color SIFT [8] is an adaption of the SIFT descriptor that includes color invariants to make it more robust to colour changes in an image. While SIFT computes gradients in a grayscale image, CSIFT computes gradients in this color invariant space. We compute SIFT and CSIFT features at 4 scales using 8 gradient orientation bins.

3.1.2 Textons

Textons were one of the first feature descriptors developed explicitly for texture recognition [10]. The term texton was originally coined in 1981 [14] as the unit of "pre-attentive human texture perception". Leung *et al.* attempted to develop a computer representation of this concept by convolving an image with a bank of n filters (the original work used 48, we use 17), resulting in a n-dimensional vector of responses for each pixel. These vectors are quantized using k-means clustering into visual words, with each pixel mapped to its nearest neighbor to generate a texton map. In this work we used 150 clusters.

3.1.3 Local Binary Pattern (LBP)

The final feature we use was introduced by Ojala *et al.* in [9], and provides another way to represent the image structure around a pixel. For each pixel in the local region around a point, an 8-bit vector is computed to record which of the pixels' 8 neighbors has a smaller intensity. These vectors are collected into a histogram for the local region, which is the LBP feature vector. Again, we quantize the LBP vectors by performing k-means clustering on the vectors computed for all images.

3.2. Training

We use the Random Forests learning algorithm in this work, introduced by [15]. This algorithm has demonstrated good performance on tasks involving real-world data, including human pose classification [16] and edge detection [17]. The algorithm learns a set ("forest") of decision trees, each of which is trained on a random subset of the training set. At each node in a tree, the feature that best splits the data is chosen from a randomly selected subset of the features. The leaf nodes of each tree contain distributions over the class labels. At classification time the pixel feature

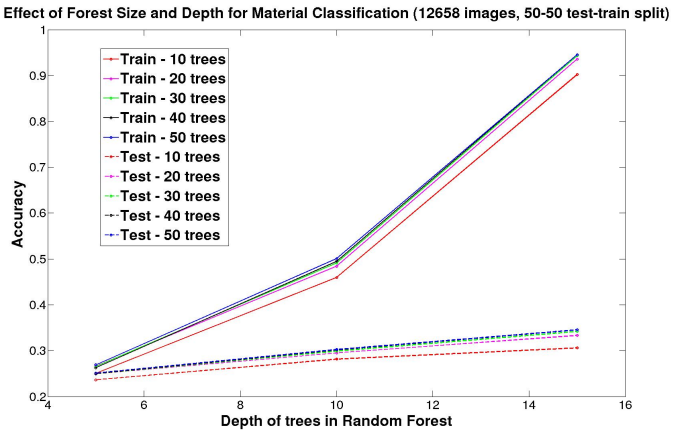

Figure 2: Training and test accuracy for forests of different sizes and depth. In general, accuracy on both training and test sets increases with forest size and tree depth, but tree depth is the more important parameter. Deeper forests tend to result in overfitting, with test error increasing only marginally with tree depth.

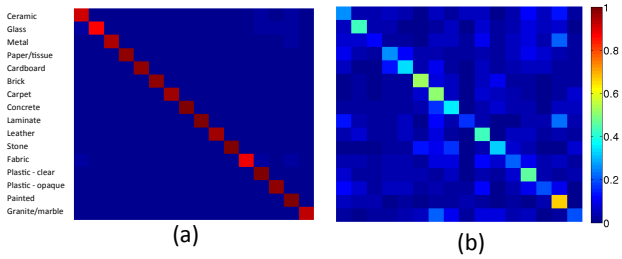

Figure 3: Confusion matrices for a Random Forest with 50 trees of depth 15. Columns correspond to predicted labels and rows to ground truth labels, with the cells colored depending on the proportion of classifications in each ground truth/prediction pair. (a) Confusion matrix for training set, (b) confusion matrix for test set. Overfitting is evident.

vector is classified by each tree, which votes on the label distribution of the pixel. These distributions are averaged to find the final classification probability. A single tree often suffers from high variance, but this can be decreased with only a small increase in bias by adding trees to the forest. In this project we experiment with forests of different sizes and depths during our analysis of the learning algorithm.

While there are 53 material labels in the dataset, a number of them are badly defined (such as the "multiple materials" and "I don't know" classes), some are not commonly present in scene understanding datasets ("animal fur" and "skin"), and some contain very few examples. We narrowed these classes down to 16 with the highest quality examples. 1000 images per category were randomly selected, each cropped tight around the material of interest. Material samples that were very small (fewer than 30 pixels on either side) were rejected, leaving 12568 samples remaining, half forming the training set and the other half left for the testing set. Features were evaluated at each pixel, but were subsampled in a 5x5 grid for training.

To give the reader an idea of the size of the training set, the training algorithm initially failed because the number of data points (ie features $\times$ *subsampled* image pixels) overflowed a 32-bit integer. However, there is likely to be a lot of redundancy in these data points, as pixels from the same material image will be very similar (especially if the material lacks visual texture).

4. Results

We trained Random Forests containing 10, 20, 30, 40 and 50 trees, each 5, 10 or 15 nodes deep. Training Random Forests is approximately linear in the number of trees but exponential in the depth of the tree. Training the largest forest (50 trees to a depth of 15) took approximately a week on a 24 core machine, while training a *single tree* of depth 20 took over 24 hours (before we gave up). Consequently, computation constraints prevented us from trying larger trees.

Figure 3 shows the classification confusion matrix for the most accurate forest on the test set (50 trees 15 deep). This algorithm achieved a test set accuracy of 34.5%, while the training accuracy was 94.6%. The large disparity between the two accuracies suggests that overfitting is present.

Fig. 2 shows how training and testing accuracy changed with forest size and depth. Using forests with a depth of 5 both training and testing accuracy are low for forests of all sizes, indicating the algorithm is limited by high bias. Using deeper trees causes both training and test accuracy to increase, but training accuracy increases much faster. This is a curious observation; the algorithm is generalizing better, *even while* it increasingly overfits the training data. Essentially what is happening is that the majority of the variance added by increasing the depth of the trees is contributing to overfitting, but some is extracting valuable information from the features and improving the overall performance of the classifier. In addition, this observation may be partially caused by others terms in the CRF smoothing the Random Forest output and mitigating some of the increased variance introduced by deeper trees. We would have liked to use even more data to try and decrease this overfitting, but the aforementioned computation constraints prevented this. Figure 3 shows that increasing the number of trees in the forest has a much smaller but equally consistent positive effect on accuracy.

5. Scene Understanding

Our reason for developing this material classification algorithm is to improve scene understanding. To achieve this we use the output of our most accurate Random Forest as additional features to a recent RGB-D scene understanding

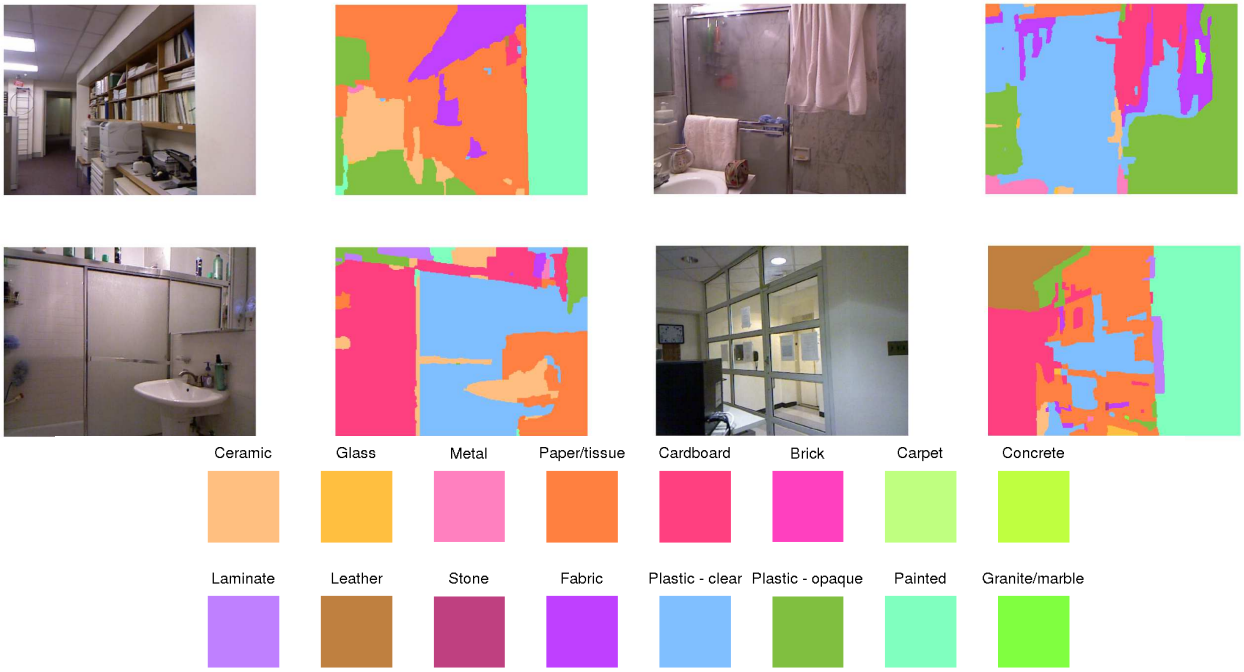

Figure 4: Examples of per-pixel material classification on images in the NYU-V1 scene understanding dataset. Some materials are classified correctly, but as expected the classifier struggles with textureless materials (often mistaking cardboard, painted surfaces, opaque plastic and laminate for one another).

algorithm presented by Ren *et al.* [18]. This scene understanding approach determines the semantic labels of superpixels in a tree of segmentation layers using a linear SVM. Our Random Forest outputs a per-pixel probability distribution over the 16 material types for that pixel, which can be thought of as a 16 dimensional feature vector. We augmented the existing feature set with these material features and retrained the algorithm on the NYU-V1 RGB-D dataset [19].

Figure 5 shows the classification accuracy at each layer of the segmentation tree, and when all layer are combined. By introducing material features we improve the test set accuracy of the semantic labeling from 75.46% to 76.15%. For comparison, this improvement is approximately the same as was achieved in [18] by introducing a novel way to combine segmentation layers. Although this is a small improvement, it illustrates the value of even relatively poor material labels for improving scene understanding. Figure 4 shows the most likely material label for a number of NYU-V1 images, as determined by our classifier. The effect of the higher-order CRF terms can be seen, as superpixels in each image take the same label.

	Segmentation Level						
	1	2	3	4	5	6	Combined
Ren et al.	69.55%	70.94%	71.96%	71.35%	68.14%	63.35%	75.46%
With Material Features	70.87%	72.47%	73.04%	72.40%	68.93%	63.85%	76.15%

Figure 5: Semantic labeling accuracy for the scene understanding approach by Ren *et al.*, showing how adding our material features improve scene understanding. We improve accuracy on each segmentation layer by an average of 1.04%, and by 0.7% when these layers are combined.

6. Discussion and Conclusion

On face value our results might seem disappointing, but it must be stressed how challenging it is, even for humans, to determine material exclusively from local appearance features. For example, consider the three material samples from the OpenSurfaces dataset shown in Fig. 6. There is nothing discriminative about their appearance - without knowledge of the object they belong to or the context the exist in, we cannot determine their material types. This is why our algorithm does much better on stone and brick (which has distinctive appearances) than on plastic and laminate.

In general, it is not clear how much a human's ability to recognize materials in objects is a result of our amazing object recognition ability combined with strong priors on the material composition of objects. For example, we know that most coffee mugs are ceramic, so we might not be recognizing the ceramic material so much as recognizing the coffee

(a) (b) (c)

Figure 6: Cropped samples of 3 different materials from the OpenSurfaces dataset, illustrating how challenging it is to determine material type visually, without the use of context. (a) is opaque plastic, (b) is laminate and (c) is painted.

cup and making an inference about its material. While I believe there is room for improvement in appearance-based material classification (for example by introducing some of the features used in [12]), I believe such improvements will be marginal, and real progress towards solving this problem will require material and object type to be considered jointly in a scene understanding framework.

The development of such a framework for material and object recognition would allow prompt exciting advances in robotics. For example, a robot could be taught to avoid, or move cautiously near, objects that could be sharp (ie metal) or fragile (ie glass or ceramic). This would be a step towards the safe collaboration between robots and humans in a kitchen or home.

Our algorithm's performance can be evaluated by considering the Flickr image dataset, for which state-of-the art accuracy is 54% [12]. However, the OpenSurfaces dataset is qualitatively much more challenging because the Flickr dataset contains hand-picked images of single objects, so there is no clutter or shadowing from other objects that is present in the OpenSurfaces images. In addition, because material classification was done image-wise rather than pixel-wise, it is possible that this approach learned a primitive object detector to aid classification. The OpenSurfaces dataset is very new, so we don't know of any methods directly comparable to ours.

In conclusion, this paper has presented a material classification algorithm trained on a large scale, real-world dataset. Our final test accuracy was 34.5%, and we show that augmenting an existing scene understanding method with this material information improves its accuracy by 0.7%.

References

[1] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Opensurfaces: A richly annotated catalog of surface appearance," in *SIGGRAPH Conf. Proc.*, vol. 32, no. 4, 2013. 1

[2] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009. 1, 2

[3] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics (TOG)*, vol. 18, no. 1, pp. 1–34, 1999. 1, 2

[4] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring features in a bayesian framework for material recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 239–246. 1, 2

[5] P. Kohli and P. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009. 1, 2

[6] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, "Graph cut based inference with co-occurrence statistics," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 239–253. 1

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. 1, 2

[8] A. E. Abdel-Hakim and A. A. Farag, "Csift: A sift descriptor with color invariant characteristics," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 1978–1983. 1, 2

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002. 1, 2

[10] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001. 1, 2

[11] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 11, pp. 2032–2047, 2009. 2

[12] D. Hu, L. Bo, and X. Ren, "Toward robust material recognition for everyday objects." in *BMVC*, 2011, pp. 1–11. 2, 5

[13] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in *Advances in Neural Information Processing Systems*, 2010, pp. 244–252. 2

[14] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, Mar. 1981. 2

[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. 2

[16] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013. 2

[17] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *ICCV*, 2013. 2

[18] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2759–2766. 4

[19] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 746–760. 4