USING SEMANTIC FEATURES FOR SCENE CLASSIFICATION: HOW GOOD DO THEY NEED TO BE?

Matthew Boutell [1], Anustup Choudhury[2], Jiebo Luo[3], and Christopher M. Brown[2]

[1]Dept. of Comp. Sci. & Soft. Eng.
Rose-Hulman Inst. of Technology
boutell@rose-hulman.edu

[2] Department of Computer Science
University of Rochester
{anustup, brown}@cs.rochester.edu

[3]Research and Development Labs
Eastman Kodak Company
jiebo.luo@ kodak.com

ABSTRACT

Semantic scene classification is a useful, yet challenging problem in image understanding. Most existing systems are based on low-level features, such as color or texture, and succeed to some extent. Intuitively, semantic features, such as sky, water, or foliage, which can be detected automatically, should help close the so-called semantic gap and lead to higher scene classification accuracy. To answer the question of how accurate the detectors themselves need to be, we adopt a generally applicable scene classification scheme that combines semantic features and their spatial layout as encoded implicitly using a block-based method. Our scene classification results show that although our current detectors collectively are still inadequate to outperform low-level features under the same scheme, semantic features hold promise as simulated detectors can achieve superior classification accuracy once their own accuracies reach above a nontrivial 90%.

1. INTRODUCTION

Semantic scene classification categorizes images into classes such as beach, field, or street. This is useful for many applications. For example, knowing a scene class facilitates intelligent image enhancement and automatic annotation of images in a digital album. However, classification is a difficult problem in and of itself. Most approaches in the literature use low-level features, such as color, edges, or texture, and work with limited success. Certain classifiers, such as support vector machines, can improve accuracy, e.g. [11], but are constrained by the expressiveness of the features chosen for the problem. This is the so-called "semantic gap".

Increasingly, the research community has been pursuing the clear alternative of semantic features, corresponding to objects or materials occurring in the scene, such as sky, water, buildings, or foliage. Intuitively, semantic features should help bridge the semantic gap. The main question is not *if*, but *when*, i.e., how good do the semantic features need to be?

Recent advances in object and material detection have made such an approach possible. Indeed, researchers have begun to utilize *semantic* features directly for non-classification tasks, such as image retrieval [9][10], even though the detectors themselves are faulty and can sometimes fail.

However, few have used semantic features *directly* for image classification [8]. In this study, we intend to answer the question posed earlier, in the context of outdoor scene classification, using a reasonably general scheme. In particular, combining the results of material detectors gives a belief vector for each image region. The location of each region is then encoded by imposing a regular grid structure on the image, similar to the case with low-level features [2][11]. Using the same classification scheme for both the semantic and low-level features should provide a fair assessment of the effectiveness of the features.

The primary contributions of this paper are two-fold. First, we provide a framework in which semantic features, i.e., the output of a bank of material detectors, can be used directly for *general* scene classification beyond the specific problems in this study. Second and more importantly, our scene classification results, based on a range of real and simulated material detectors, demonstrate that classification accuracy superior to that using low-level features can only be achieved given accurate enough material detectors.

2. LOW-LEVEL FEATURES

Low-level features are those that can be extracted directly from the image, such as color, texture, and edges. For distinguishing between certain types of outdoor scenes, color information is reasonably effective [1][12]. Furthermore, spatial information appears to be important as well: bright, warm colors at the top of an image may signal a sunset, while those at the bottom may come from desert rock. Therefore, we use spatial color moments in an Luv-like space as features, in the same fashion as [11][12]. This color space helps de-correlate chrominance from luminance and is nearly perceptually uniform: perceived color differences correspond closely to Euclidean distances in the color space [5]. Such preprocessing removes the burden from the classifier.

After conversion to Luv space, the image is divided into 49 blocks using a 7*x*7 grid. We compute the first and second moments (mean and variance) of each band, corresponding to a low-resolution image and to computationally inexpensive texture features, respectively. The end result is a 49*x*2*x*3 = 294-dimension feature vector per image.

3. SEMANTIC FEATURES

Object types occurring most frequently in consumer photos include sky, grass, and people, according to numerous studies [7]. Considering frequency and detectability by automatic algorithms, we chose a number of homogeneous material types for our detection system, as was done in [7]: *sky, cloud*, *grass, foliage, open water, snow, sand, rock*, and *pavement*. We also included a *manmade* object detector to detect skyscrapers, houses, and other manmade structures (such as boats in open water). While this list of detectors is by no means complete (e.g., we use no face or skin detector), these objects tend to occupy most of the background of outdoor scenes, have well-defined spatial relationships among each other, and usually define the scene. Any other region not detected by any of these detectors is unmodeled and thus ignored.

Our detectors for homogeneous materials, such as sky or grass, are described in detail in [7]. Each of the first nine detectors is similar. First, we train a neural network to discriminate between positive and negative examples of the material using colors (in Luv color space) and textures (6 multiresolution wavelet features). The 9-dimensional feature vector for each pixel is fed into the network, which produces a belief value in each pixel for a particular material type. The collection of pixel belief values forms a belief map of the image.

Second, we segment by thresholding the raw belief map to obtain spatially contiguous regions, then analyzing each region based on its characteristics; it is specific to each detector, such as the saturation gradient in blue sky [6]. Finally, each semantic region is given a uniform belief value, which is the average of the raw belief values for the pixels in that region.

Manmade structures, such as buildings, are also characterized by regular textures and to a lesser extent, colors. Our detector uses a block-based approach similar to [4].

While some of these detectors (e.g., for blue sky) have very good performance because of object-specific region analysis that removes false positives, other natural object detectors (e.g., for open water and snowfields) suffer substantially high misclassification rates. Used independently, the accuracies of our detectors range from 52% (open water) to 96% (blue sky).

4. ENCODING SPATIAL INFORMATION FROM SEMANTIC FEATURES

Spatial information can help scene classifiers using semantic features as well. While the presence or absence of certain objects predicts some scene types (such as sky, water, and sand in beaches), encoding the location of the objects can help overcome detector errors.

To isolate the expressiveness of the low-level features, we encoded the spatial location of the semantic regions in the same way as the color moments, i.e., using a 7*x*7 grid. Since this block-based scheme has been used with low-level features for a wide array of scene classification problems [2][11][12], we expect it to be equally applicable with semantic features and thus enable us to answer the main question we posed.

In general, our object detectors find irregularly-shaped image regions. However, encoding them as a fixed-size feature vector would require a fixed number of regions. We used a regular grid to solve this problem, weighting each region's beliefs by the fraction of the area it occupies in a block.

Let r be the number of semantic regions in the image. Each semantic region i, $i<=r$, has an associated belief vector $\mathbf{v_i}=<v_{i,1}, v_{i,2}, \ldots v_{i,m}>$ corresponding to the likelihood of each of m potential materials of interest being the true material, given the combined output of the detectors. In our study, m=10. A region with known class k (as we use in training) will have a 1 in the kth dimension and zeros elsewhere, the background is encoded using the zero vector, and in general, when derived from the material detector output, the values lie in the range [0,1]. Denote the belief vector for block j as $\mathbf{w_j}$, and the fraction of the area occupied by region i in block j as a_{ji}. Thus $\mathbf{w_j}$ is the weighted sum of the belief vectors of the regions occurring in it,

$$w_j = \sum_{i=1}^{r} a_{ji} v_i \,.$$

This weighting technique is similar in spirit to the voting technique used in [1]. Each image is thus encoded as a 49*x*10 = 490-dimension feature vector.

5. EXPERIMENTAL RESULTS

We conducted experiments using two sets of images, both taken from various film and digital sources. Each set contains images from six scene classes: beach, field, mountain, open-water (e.g., taken from a boat), urban street, and suburban. These classes have materials that our semantic detectors can find, but are not easily separable using semantic features without spatial information, because they contain many of the same materials (sky, grass, foliage, buildings). For example, both open-water and beach scenes often contain sky and water. Other classes, such as sunsets, can be found better using different approaches [2]. D1 contains a total of 937 images, while D2 contains 1153 images. Each image in D1 was segmented using an automatic algorithm [5] and each region of interest was manually labeled by a human, so it could be used as ground truth for training.

We used a Support Vector Machine (SVM) as a classifier, with block-based input feature vectors (either low-level or semantic). SVM classifiers have been shown to give better performance than other classifiers like Learning Vector Quantization (LVQ) on similar problems [11]. Our Gaussian kernel gives an RBF-style classifier. For multiclass classification, we used the "one-vs-all" approach: for each scene class, an SVM was trained to distinguish that class of images from the rest, test images are classified using each SVM and then labeled with the class corresponding to the SVM which gave the highest score.

We first trained a classifier using the color moment features and another using the semantic features. We obtained 67.4% accuracy using the color moments and 54.2% accuracy using the semantic features. Figure 1 compares the recall for both sets of features for each scene class. For half of the classes, the recall for color moment features is slightly higher. One exception is mountains, for which rock and snow were used as discriminatory features. In the other two exceptions, low-level features obtain *substantially* higher accuracy than semantic features. Many field scenes had grass that was difficult for our detector to find: the detector was trained to recognize lawn grass, while the fields often contained grains ready for harvest, such as wheat. In open-water scenes, the color features better handled images in which the water color changed slightly.

Thus, for semantic features to work for scene classification, we would need to improve the detectors substantially. Because the advantages of improved detectors are not guaranteed, we focused in our second experiment on the extreme case: full knowledge of the materials present. We computed accuracy using cross validation on image set D1, for which we know the true materials. When we used these *hand labels* (vs. detector output) as features, classification accuracy jumped to 93.1% (vs. 78.0% for color moments on this set), providing us with an upper bound on accuracy (Figure 2). We speculate that the remaining 7% in accuracy is due to configurations of regions that occur only once in the set, but it may also be partially due to subtle scene cues, materials for which we have no detector, or images with slightly ambiguous true classification.

We performed a final experiment to determine how good imperfect detectors would have to be in order to outperform low-level features in scene classification. We simulated a variety of imperfect detectors by randomly perturbing the manually labeled regions in D1 (details in [3]). We set the detection rates of individual detectors on each true material (both true and false positive rates) by counting performance of the corresponding actual detectors on a validation set. When they fire, they are assigned a belief that is distributed normally with mean μ. The parameter μ can be set differently for true and false positive detections; varying the ratio between the two is a convenient way to simulate detectors with different accuracies. Figure 3 shows how classification accuracy depends on detector accuracy.

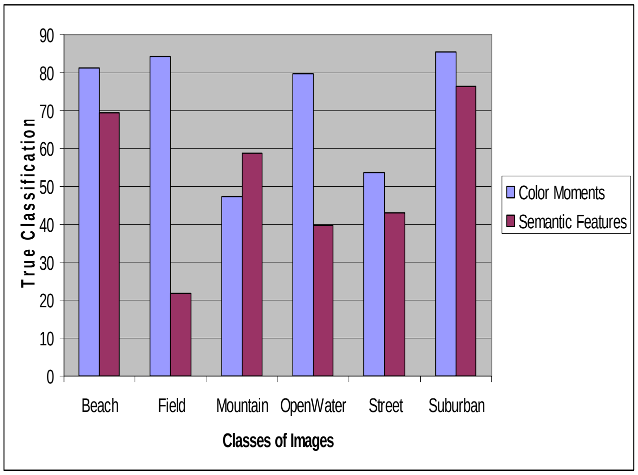


Figure 1. Per-class recall for each scene type using detected and low-level feature sets.

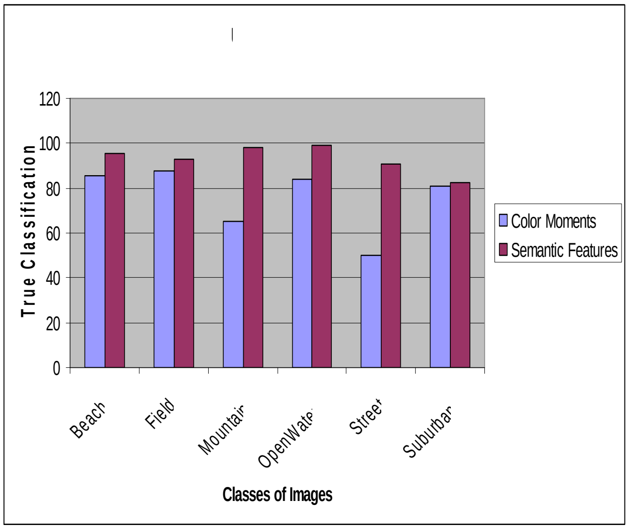


Figure 2. Per-class recall for each scene type using true semantic and low-level feature sets.

For each operating point in Figure 3, the testing images are perturbed as mentioned above. Furthermore, perturbed versions of the training images (perturbed in the same fashion) are added to the training set. This has two effects. For small perturbations, adding extra data to the training set has the effect of better populating the feature space with detection results similar to what the detectors actually produce, leading to higher classification accuracy. For large perturbations, it adds a large amount of noise to the system, lowering the classification accuracy. Determining the optimal amount of perturbation is left to future work. Without perturbation in the training images, i.e., the detectors are always perfect, the SVM would learn only the spatial layout of materials and not the expected confidence in the detection.

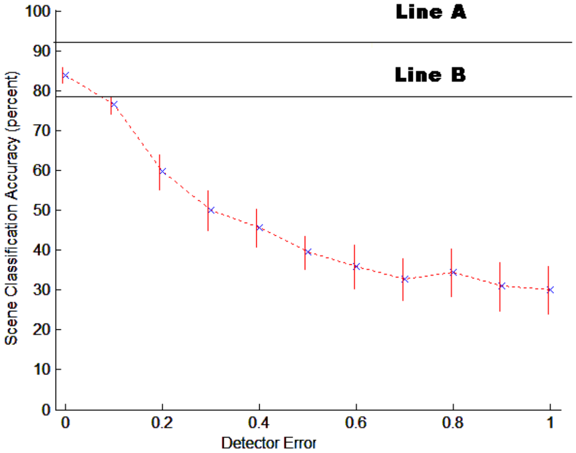


Figure 3. Scene classification accuracy as a function of detector errors (μ_{TP}/μ_{FP}). The control points to the right correspond to extremely faulty detectors, while those to the left are better detectors. Accuracy using the best simulated detector is higher than using the low-level features (78%; line B), but is below the upper bound using this feature set (93%; line A)

Figure 3 shows that accuracy higher than that with low-level features can be obtained using the semantic features if the material detectors can be trained to yield confidence in true positive detections substantially greater than in false positive detections.

Figure 4 shows example results for four images per class. We consider the first column. In row 2, rocks covering a substantial portion of the image suggested a mountain. In row 3, atypical sand color caused the spatial color moment-based classification to fail. In row 4, the viewpoint of the beach, with water both in front of and behind the sand, caused both the semantic features and the low-level features to classify it as open-water.

6. CONCLUSIONS AND FUTURE WORK

Using a generally applicable scene classification scheme, we have shown that semantic features can be used to classify outdoor scenes with various levels of accuracy. Our currently available semantic features did not outperform the low-level features, primarily due to the poor performance of the current material detectors that generate these features. When the materials are poorly detected, it does not pay to use them; when they become sufficiently accurate (>90%), the benefit starts to show.

One potential bias that exists against the semantic features is the higher dimensionality of the feature vector (490 vs. 294). It is more difficult to obtain enough training data to cover all spatial configurations of objects, the "curse of dimensionality".

The obvious direction for future work is to improve upon our material detectors, perhaps by using a more powerful classifier such as an SVM. Another direction is to apply our technique to indoor scene classification. Because our system is modular, we can do so easily by replacing our detectors with appropriate detectors for typical indoor objects.

7. REFERENCES

[1] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Recognition of images in large databases using a learning framework", U. Cal. Berkeley Technical Report 97-939, 1997.

[2] M. Boutell, J. Luo, and R. T. Gray, "Sunset scene classification using simulated image recomposition," in *Proc. IEEE Intl. Conf. on Multimedia & Expo*, 2003.

[3] M. Boutell, J. Luo, and C. Brown, "Scene parsing using region-based generative models," *IEEE Trans. on Multimedia*, to appear.

[4] B. Bradshaw, J. Platt, and B. Scholkopf, "Kernel methods for extracting local image semantics," *Microsoft Research Technical Report MSR-TR-2001-99*, 2001.

[5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(5): 603-619, 2002.

[6] J. Luo and S. P. Etz, "A physical model-based approach to detecting sky in photographic images," *IEEE Trans. on Image Processing*, vol. 11, pp. 201–212, 2002.

[7] J. Luo, A, Singhal, and W. Zhu, "Natural object detection in outdoor scenes based on probabilistic spatial context models," in *Proc. IEEE Intl. Conf. on Multimedia & Expo*, 2003.

[8] J.-H. Lim and J.-S. Jin, "Unifying local and global content-based similarities for home photo retrieval," in *Proc. Int'l Conf on Img. Proc.*, 2004.

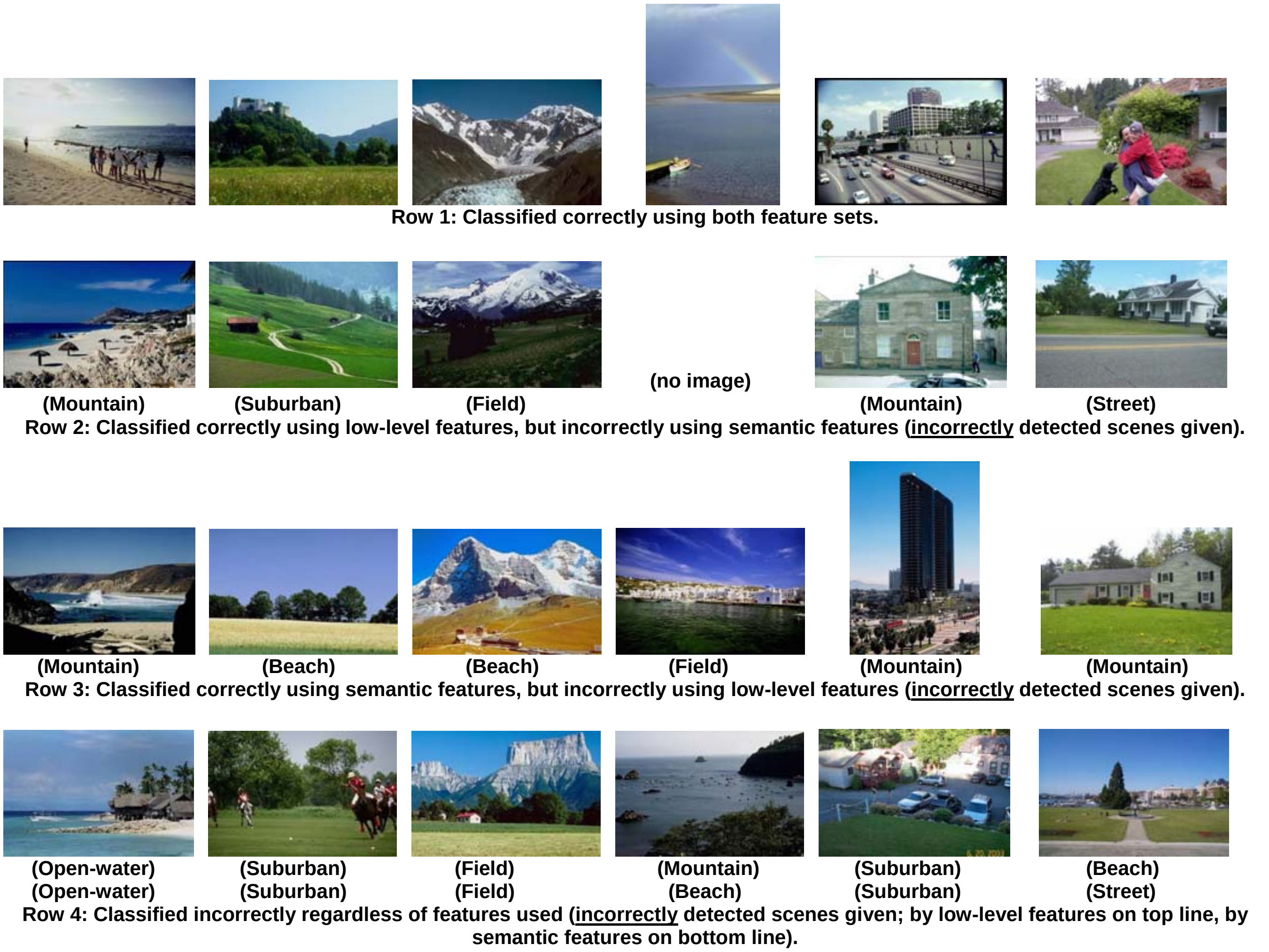
[9] J. Smith, C. Lin, M. Naphade, and B. Tseng, "Multimedia semantic indexing using model vectors," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, 2003.

[10] Y. Song and A. Zhang, "Analyzing scenery images by monotonic tree," *ACM Multimedia Systems J.*, 10(3), 2002.

[11] Y. Wang and H.-J. Zhang. "Detecting image orientation based on low-level visual content," *Computer Vision and Image Understanding*, 93(3), 328-346, 2004.

[12] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang, "Content-based hierarchical classification of vacation images," in *Proc. IEEE Multimedia Systems*, 1999.

Beach	Field	Mountain	Open water	Street	Suburban


Row 1: Classified correctly using both feature sets.

Row 2: Classified correctly using low-level features, but incorrectly using semantic features (incorrectly detected scenes given).

Row 3: Classified correctly using semantic features, but incorrectly using low-level features (incorrectly detected scenes given).

Row 4: Classified incorrectly regardless of features used (incorrectly detected scenes given; by low-level features on top line, by semantic features on bottom line).

Fig. 4. Examples of images in the six classes (in six columns): Beach, Field, Mountain, Open-water, Street, and Suburban. Note that correctly detected scene labels are not shown but can be found at the top of the figure.