M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 445
Appeared: IEEE Intl Workshop on Content-based Access of Image and Video Databases, Jan 1998

Indoor-Outdoor Image Classification

Martin Szummer and Rosalind W. Picard

MIT Media Lab Rm E15-384; 20 Ames St; Cambridge MA 02139; USA
szummer@media.mit.edu, picard@media.mit.edu
http://www-white.media.mit.edu/~szummer/

Abstract

We show how high-level scene properties can be inferred from classification of low-level image features, specifically for the indoor-outdoor scene retrieval problem. We systematically studied the features: (1) histograms in the Ohta color space (2) multiresolution, simultaneous autoregressive model parameters (3) coefficients of a shift-invariant DCT. We demonstrate that performance is improved by computing features on subblocks, classifying these subblocks, and then combining these results in a way reminiscent of "stacking." State of the art single-feature methods are shown to result in about 75–86% performance, while the new method results in 90.3% correct classification, when evaluated on a diverse database of over 1300 consumer images provided by Kodak.

1 Introduction

The scene classification problem is one of the holy grail challenges of computer vision. Given an arbitrary photograph, we would like to describe what type of *semantic* scene it depicts. Currently, very little work has been done in this area, probably because the problem is very difficult and also because there is no agreed-upon scene description language. Most computer vision research involves low-level image analysis that rarely tries to bridge the gap to semantic scene description.

Our purpose is to show how one particular semantic scene description problem can be approached. The task is to determine whether a consumer photograph depicts an indoor or an outdoor scene. This problem is relatively unambiguous, and is motivated by several practical applications.

There certainly exist photographs for which the indoor-outdoor distinction is unclear. Examples include shots made through a window (with visible window edges), photographs of paintings, and extreme close-ups of faces. Fortunately, such compositions are rare in consumer photographs. Our database of 1343 images was classified by two independent people, and only nineteen out of these were labeled as unclear and omitted from this study.

The applications of this problem are interesting. Knowledge about the scene enables more intelligent image processing. For example, when film is developed and prints are made from the negatives, the exposure and color is automatically adjusted. Unfortunately, the automatic correction does not take into account the content of the photograph. If the machine could distinguish indoor from outdoor images, it could adjust these classes differently, rather than adjusting everything towards one "ideal" exposure and color. This observation can also be applied to image scanners, photocopiers, fax machines, image processing software, etc.

Another important application is image retrieval. Let's say we would like to find a beach scene. A helpful step would be to limit the search to outdoor scenes. Unfortunately, this is not possible even in state-of-the-art image retrieval systems such as QBIC [1], Virage [2] and VisualSEEk [3]. These systems are based mainly on color histograms and primitive texture measures. The user builds a query by selecting colors from a palette, a texture from a chart, and then indicates how to weight the color versus the texture. Unfortunately, it is difficult for a user to know how to weight the different features to get a beach scene. The systems' level of abstraction is much too low.

Query by image example enables the user to select one image and find other similar images, making it easier to specify the relevant color and texture query. Most systems still require the user to select weights for the different features. An exception to this is FourEyes [4], which can learn the relevant feature combination based on several positive and negative examples. In an initial quick attempt to teach FourEyes to solve the indoor-outdoor classification problem using whole images with no specific subblock guidance, we did not meet with significant success. Although FourEyes can learn any classification, it was not very efficient on this one, probably due to the noisiness of the two classes being considered.

This work was supported in part by Kodak, NEC and Hewlett Packard Labs.

In this work we propose a different classification approach that exploits the same idea in FourEyes of non-linearly combining features from multiple models, but does so in a different way. This new way is successful for accurately distinguishing indoor from outdoor scenes.

2 Background

Several attempts at recognizing high-level scene properties using low-level features have been made. Gorkani and Picard [5] discriminate between photos of city scenes and photos of landscape scenes. They use a multiscale steerable pyramid to find dominant orientations in 4×4 subblocks of the image. The image is classified as a city scene if enough subblocks have strong dominant vertical orientation, or alternatively medium-strong vertical orientation and also horizontal orientation.

Yiu [6] uses the same dominant orientation features and also color information to classify indoor and outdoor scenes. She uses nearest neighbor and support vector machine classifiers. The former classifier is better at color, the latter at dominant orientation. Yiu reports accuracies similar to those in our work, but they are believed to have a high variance because they were not thoroughly evaluated with a leave-one-out method, and a much smaller database of only 500 images was used. Furthermore, the texture features used here give significantly better results than her dominant orientation detector. The work here also takes advantage of a spatial tessellation of the image, which we found provides a significant gain in performance.

Instead of building a specific scene class detector, Lipson [7] describes a general scene query approach. Scenes are described by graphs representing relations between image regions. The relationships include relative color, spatial location, and highpass frequency content. Unfortunately, the templates have to be constructed manually for each scene layout. These templates are also quite specific, which makes them fine for limited special cases such as "sky over mountain over lake" but difficult for the case considered here of capturing a broad concept like an outdoor scene.

Yu [8] learns a statistical template from examples. She computes vector quantized color histograms for subblocks of the image. Then she trains a one-dimensional hidden Markov model along vertical or horizontal segments of specific scene layouts, such as sky-mountain-river scenes. Unfortunately, the one-dimensional model cannot describe spatial relationships well, and a two-dimensional generalization such as Markov random fields is desirable, but raises many new kinds of problems.

3 Features

3.1 Image Database

The image database in these experiments consists of 1343 consumer photographs collected and labeled by Kodak. They depict typical family and vacation scenes, and are taken by many different individuals, at all different times of they year. The database is quite diverse, and includes snow, bright sun, sea, sunset, night and, silhouette scenes. Image types not in our database can be easily added to the training set without changing any algorithms.

The images were hand-labeled by two independent people (not the authors), resulting in 694 (52%) labeled as outdoor, and 630 (48%) labeled as indoor. We have excluded 19 (1.4%) images which were labeled as ambiguous. All the images in the set have landscape orientation and are right-side up. The full resolution of the images was 768×512, but for most experiments we used half- or quarter resolution, as will be described later. The images originally came in 36-bit color, but were quantized down to 24-bit color. At the same time we performed basic color balancing according to steps provided by Kodak, which simply clipped the top and bottom 5% of the intensity-levels, shifted the histograms to the middle, and stretched them to occupy the 24-bit range.

3.2 A baseline experiment

The problem is quite challenging. A naive approach, such as a traditional color histogram, will not give good classification performance. To illustrate this, we computed 32-bin histograms, uniformly spaced, for each RGB channel, concatenated them into a feature vector, and applied a nearest neighbor classifier. The distance between feature vectors was measured using the Euclidean norm. The resulting leave-one-out classification performance was 69.5%. This number is only somewhat better than just guessing that each image is outdoor, which would be 52% correct. Nevertheless, a very similar color histogram is used at the heart of most image retrieval systems.

Below, we describe and evaluate several methods which perform much better. These included more sophisticated features and classifiers, which tessellate the image into subregions, and combine the results from different features and different spatial regions to result in significantly improved performance.

3.3 The features

We have used three types of features: one each for color, texture and frequency information. These features were computed both for the whole image and for each subblock of a 4×4 image tessellation.

The color feature is a color histogram, and has 32 bins per channel like our baseline. However, the three channels come from the Ohta color space [9]. The color axes of this space are the 3 largest eigenvectors of the RGB space, found through principal components analysis of a large selection of natural images. This yields:

$$I1 = R + G + B \tag{1}$$

$$I2 = R - B \tag{2}$$

$$I3 = R - 2G + B \tag{3}$$

The advantage of the Ohta color space is that the color channels are approximately decorrelated, which makes it a good choice for computing per-channel histograms. The change of color spaces, from RGB to Ohta, raises the performance of color histogram based recognition to 73.2%.

Moreover, instead of using the Euclidean norm for measuring distances between histograms, we use the histogram intersection norm [10]. It measures the amount of overlap between corresponding buckets in the two histograms h^1 and h^2, and is defined as:

$$\text{dist}(h^1, h^2) = \sum_{i=1}^{N} (h_i^1 - min(h_i^1, h_i^2))$$

When both the Ohta color space and histogram intersection is used, the classification rises to 74.2% correct. The intersection norm is better than the Euclidean norm, possibly because it penalizes linear error as opposed to squared error, reducing sensitivity to outliers. In the rest of the paper we exclusively apply the Ohta color space with histogram intersection.

The texture features are computed using the multiresolution, simultaneous autoregressive model (MSAR) [11]. These are among the best texture features bench-marked on the Brodatz album [12]. The model constructs the best linear predictor of a pixel based on a noncausal neighborhood. The features are the weights of the predictor. Three different neighborhoods at scales 2, 3, and 4 are used, and the weights are concatenated to yield a 15-dimensional vector, as in [12]. The Mahalanobis norm is used to measure feature vector distance (covariances are estimated from several subwindow estimates). We extracted these features from gray scale images at two resolutions (half and quarter), using a suitable antialiasing filter.

The MSAR classification is 82.2% and 77.7% correct at half and quarter resolutions respectively, which is significantly better than the best color classification. This is surprising since the MSAR feature presupposes a single texture (characterized by a second-order stationary autoregressive process), whereas a typical image consists of many textures and is definitely nonstationary. As we shall see, we can do even better by dividing the image into 4×4 subblocks and computing the features separately over each block.

The frequency features are obtained by first calculating the 2D DFT magnitude, and then taking the 2D DCT. The first step is shift-invariant, and for periodic textures it shows a regular pattern of peaks (fundamental and harmonic frequencies). The second step replaces all related frequencies by one coefficient. All computations are done over 8×8 pixel blocks, and the results are averaged over the image region, also producing covariances used for the Mahalanobis distance metric.

3.4 Classification

The performance numbers so far refer to nearest neighbor classification of features computed on the whole image. Unfortunately, this method cannot exploit local properties of the image, e.g. blue sky at the top. Nevertheless, most image retrieval systems use features computed for the whole image, thus the numbers here are useful for comparisons. The results are summarized in Table 1 for a K-nearest neighbor classifier. We have also tried a 3-layer neural network with sigmoid nonlinearities, but training was slow and the results were worse than for the nearest neighbor algorithm when compared on the color histogram features.

To allow local and spatial properties to improve the classification, we divided the image into 4×4 subblocks and computed the features separately within them. Now the question arises how to classify these blocks. One possibility would be to concatenate feature vector from all subblocks of an image, and apply a classifier to this vector. The problem is that such a feature is very high-dimensional (e.g. $4 \times 4 \times 32 = 512$). It is difficult to estimate covariances for such a large vector, and we encounter general curse-of-dimensionality problems.

Instead, we chose to pursue a multi-stage classification approach, classifying the subblocks independently and then performing another classification on these answers (Figure 1). This is reminiscent of stacking [13] except that the subblock classifiers here were trained on their own data. Not surprisingly, the individual subblock classifiers are less accurate than a whole image classifier. Ideally, we would keep a confidence or probabilistic value associated with each subblock classification, as opposed to the binary decision "in" or "out" shown in Figure 1. In theory, the mixture of experts method applied below takes care of this case; this will be described later. For now, Table 2 shows the results

Table 1: Whole image classification results, using k-nearest neighbor. The best result in each row is marked with (*).

Feature	k=1	k=3	k=5	k=9	k=13
RGB histogram euclidean	69.5	72.0	72.4	73.9	74.0 (*)
RGB histogram intersection	69.1	71.3	72.4	73.5 (*)	72.9
Ohta histogram euclidean	73.2	75.0	75.3	75.2	75.4 (*)
Ohta histogram intersection	74.2	74.0	75.1	75.3	75.6 (*)
MSAR quarter resolution	77.7	80.2	81.9	82.3	83.0 (*)
MSAR half resolution	82.2	85.2	84.9	86.2 (*)	86.1
DCT half resolution	80.4	81.0	81.3	81.3	81.9 (*)

of the use of the k-nearest neighbor classifier on the color features, where each subblock is compared to all subblocks in the database regardless of spatial location, excluding subblocks from the same image.

When the results of the subblocks are combined, the classification can be greatly improved. Three ways to combine the features were systematically tried: (1) a simple majority classifier that assigns the label for the image to the most common class label among the subblocks, (2) a one-layer neural net, and (3) a Mixture of Experts classifier. The first method was evaluated using the leave-one-out method, and was found to give good results (Table 3). The other two methods, because of their long training time, were only evaluated with a few runs of "leave 100 out," training on 4/5 of the data and testing on the other 1/5. In these limited tests, which are subject to higher variance than the leave-one-out test method, we got slightly better results than the majority classifier, but not significantly better.

The one-layer neural net can give us information about what subblocks are important for the classification task. The net has a sigmoid nonlinearity at the output, and optimizes the cross-entropy cost function between the network output and the true class. By observing the weights learned by the network, we found that it favors especially the top row but also the lower-center of the image for classification (figure 2) (the subblocks were classified using color histograms). In some images, sky occurs in this region, and is perhaps easier to classify correctly and hence is heavily weighted.

The mixture of experts classifer [14] is similar to but more flexible than the above neural network. It learns "experts" for specific subproblems. The experts are selected depending on the input data, and each expert can weight the data differently. For example, if the top of the image is classified as outdoor, an expert can weight it more heavily than if the top of the image is classified as indoor. The technique is similar to softly clustering the data and assigning a set of weights for each

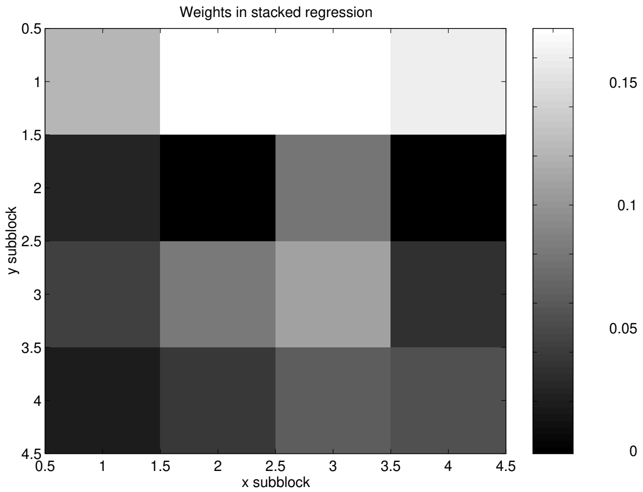

Figure 2: Weights of subblocks in image classification. The larger the weights, the brighter the square.

cluster (however, the clustering and weight assignment occur simultaneously). Somewhat disappointingly, the classification results are approximately the same as for the neural network. Moreover, we are required to set additional parameters (the number of experts) and be careful to avoid overfitting. Thus, we judged that the additional effort was not justified.

3.5 Multiple feature combination

So far, we have combined multiple subblock classifications, but these were all based on a single image feature. To gain robustness, we can use multiple image features simultaneously. One common way to do this is to concatenate different feature vectors into a longer vector. Unfortunately, this step increases the dimensionality of the problem, and requires a metric which is simultaneously good for e.g. color histograms and MSAR. The relationship between two features is almost certainly not linear, so such a metric is difficult to construct. There-

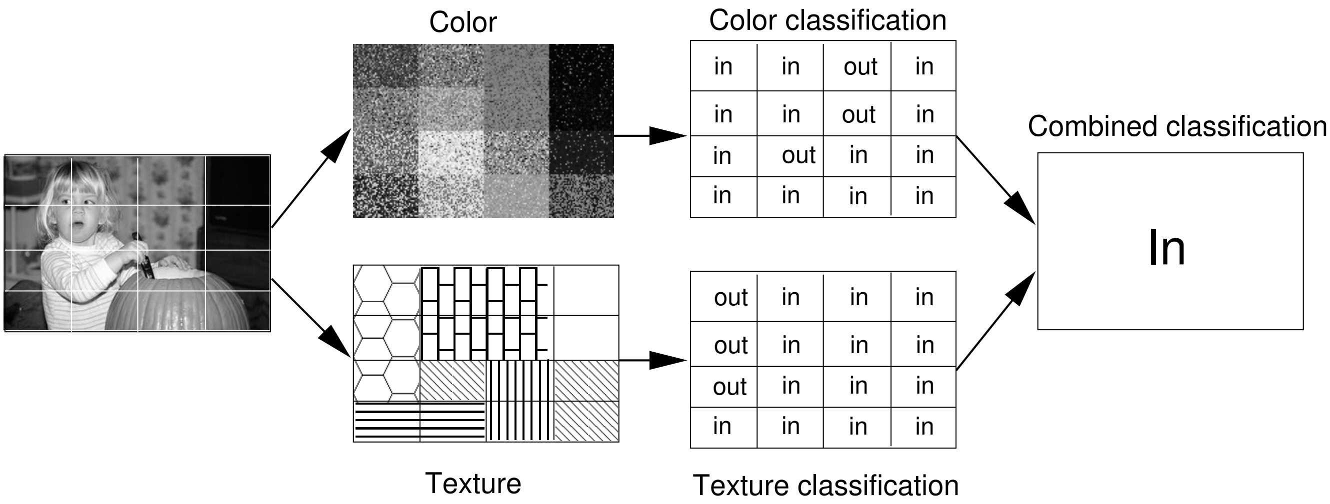

Figure 1: Two-stage classification combining color and texture.

Table 2: K-nearest neighbor classifications on subblocks. The best result in each row is marked with (*).

Feature	k=1	k=3	k=5	k=9	k=13
Ohta histogram intersection	66.0	68.1	69.2	70.1	70.3 (*)
MSAR half resol	69.2	72.0	73.1	74.2	74.7 (*)
MSAR quarter resol	64.5	67.6	68.9	70.0	70.6 (*)
DCT half resol	66.4	69.7	71.0	72.0	72.3 (*)

Table 3: Majority classifier based on k-nearest neighbor. The best result in each row is marked with (*).

Feature	k=1	k=3	k=5	k=9	k=13
Ohta histogram intersection	78.2	80.2	81.0 (*)	81.0 (*)	80.5
MSAR half resol	82.0	84.4	85.0 (*)	84.5	84.0
MSAR quarter resol	80.0	83.0	81.9	82.6	84.0 (*)
DCT half resol	82.0	84.4	85.0 (*)	84.5	84.0

Table 4: Combined feature classifier, k=13. MSAR features were measured at half resolution.

Feature	Performance
Color, MSAR	90.3
Color, DCT	89.0
MSAR, DCT	86.5
Color, MSAR, DCT	89.9

Table 5: Confusion matrix for color MSAR combination.

True Class	Classified as	
	indoor	outdoor
indoor	561	69
outdoor	60	634

fore, we think this approach is a mistake, even though it is commonly used by researchers working in content-based retrieval.

There is a way to combine feature vectors just by concatenation, by first translating all the features into a common language. This was done in the FourEyes system by using the common language of clusters. Our common language is different: the subblock classes assigned by the k-nearest neighbor classifer. In other words, we simply concatenate the subblock classifications based on different image features, and then do a second-stage classification. In the second stage, we get significantly improved results by applying the majority classifier to the combined vectors (Table 4).

4 Discussion

The best classification results were generally obtained by combining color features with texture features. Both the MSAR and DCT-based features capture shift-invariant intensity variations over a range of scales, so combining them does not provide as much gain as combining color with one of them.

The color-MSAR combination gives us the best result, 90.3% correct, measured using leave-one-out cross validation. The confusion matrix for this result (Table 5) shows that it is approximately equally likely to mistakenly label indoor images as being outdoor or vice-versa. The proportion of indoor to outdoor images in the database is 48% vs. 52%, which is fairly balanced.

Figure 3 shows several correctly classified images by the combined color and MSAR algorithm (see `http://www.media.mit.edu/~szummer/caivd98/` for color versions of the images). These images were incorrectly labeled when using only color information. The color algorithm easily mistakes photos containing green or navy blue as outdoor images. Conversely, it often mistakes photos containing white areas (e.g. snow scenes) and brown colors as indoor images. The texture feature disregards color and the combination gives the right answer.

Figures 4 and 5 show samples of images that were misclassified by the combined color and MSAR algorithm. Some of the missclassified indoor images contain green plants, Christmas trees or green walls, since green is a typical color of outdoor images. Another difficult indoor image is a picture of the top of a shelf and the ceiling, which looks blue under flash light, like sky. The missclassified outdoor images are often night-time flash photographs. White outdoor walls and hazy white sky are also sometimes mistaken to be indoor, probably because they are very common in indoor scenes. Close-ups are always challenging, because they are dominated by one object and provide little background.

It is tempting to believe that outdoor images can be easily classified by building a blue sky detector. A quick look at the database disspells this myth, at least for amateur photographs: only about one in five outdoor images have clear blue sky; in most outdoor images, the sky is not visible, or is cloudy white or gray. These cloudy colors can unfortunately be produced by flash light as well, making them difficult to use for discrimination.

5 Conclusions

We have shown how high-level scene properties can be inferred from low-level image features. The indoor-outdoor classification problem is only one example of a high-level scene property, and we believe that many other properties can be inferred in a similar way. Since people often reason in terms of *semantic* image properties, it is important for vision systems to extract them.

We found that it is quite difficult to predict the performance of a feature or feature combination; often combining two weaker features with a k-nearest neighbor classifier consistently produced better results than a single good feature. Moreover, relatively simple classifiers (k-nearest neighbors) performed better than the more sophisticated neural networks and mixture of expert classifiers. These empirical results suggest that a theoretical investigation should be undertaken in an effort to better understand the relative merits of these methods.

Nevertheless, we believe that performance will scale well to larger databases of consumer photography. After a thorough examination, we settled for simple but robust classifiers that require few parameter settings. Of course, it is always possible to devise scenes that will fool any system. However, our system can always be provided with more ground truth for new image types, which is likely to increase the performance on such images. In the domain of consumer photography, we have used a large enough sample to show that accurate classification is possible.

6 Acknowledgement

The authors wish to thank Thomas Minka for help with the FourEyes software and Bob Gray at Kodak for suggestions. Portions of the research in this paper use the Kodak Image Research Database. This work was supported in part by Kodak, NEC and Hewlett Packard Labs.

7 References

[1] Myron Flickner, Harpreet Sawhney, et al. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, Sept 1995.

[2] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5), 1997. `http://www.virage.com/research.htm/vir_cacm.pdf`.

[3] J. R. Smith and S.F. Chang. Visualseek: a fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, Nov 1996.

[4] T. P. Minka and R. W. Picard. Interactive learning using a 'society of models'. In *Proceedings of CVPR*, pages 447–452, San Francisco, CA, June 1996. IEEE Computer Society.

[5] Monika Gorkani and Rosalind W. Picard. Texture orientation for sorting photos at a glance. In *Proc. Int. Conf. Pat. Rec.*, volume I, pages 459–464, Jerusalem, Israel, Oct. 1994.

[6] Elaine C. Yiu. Image classification using color cues and texture orientation. Master's thesis, MIT, dept EECS, 1996.

[7] Pamela R. Lipson. *Context and Configuration Based Scene Classification*. PhD thesis, MIT, EECS dept, 1996.

[8] Hong-Heather Yu and Wayne Wolf. Scenic classification methods for image and video databases. In *Proc. SPIE, Digital Image Storage and Archiving systems*, pages 363–371, 1995. `http://www.ee.princeton.edu/~heathery/`.

[9] Y-I Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Comp. Graph. and Img. Proc.*, 13:222–241, 1980.

[10] Michael Swain and Dana Ballard. Color indexing. *Int. J. of Comp. Vis.*, (1):11–32, 1991.

[11] Jianchang Mao and Anil K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[12] Rosalind W. Picard, Tanweer Kabir, and Fang Liu. Real-time recognition with the entire Brodatz texture database. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 638–639, New York, June 1993. *MIT Media Lab Perceptual Computing TR* 215.

[13] Leo Breiman. Stacked regression. `ftp://ftp.stat.berkeley.edu/pub/users/breiman/stacked.abstract`, 1994.

[14] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

Figure 3: Images correctly classified by the MSAR and color combination, but incorrectly classified when using only color. The top 6 are outdoor images, the bottom 6 are indoor. The color classifier often assumes that white areas are part of an indoor image (presumably a wall), thereby missclassifying snow scenes. The MSAR corrects these scenes. See `http://www.media.mit.edu/~szummer/caivd98/` for color versions of the images.

Figure 4: 12 out of 60 missclassified outdoor images (combined color and MSAR classifier). Outdoor flash photos in dusk or at night are especially difficult, as are scenes with white regions (walls, hazy sky). Close-ups are also challenging, since they are dominated by one object and do not provide much context. The other 634 outdoor images in the database were correctly labelled.

Figure 5: 12 out of 69 missclassified indoor images (combined color and MSAR classifier). Plants, Christmas trees, green walls and brown floors are sometimes mistakenly thought to belong to outdoor scenes. The other 561 indoor images were correctly classified.