Indoor/outdoor scene classification project

Paul Fitzpatrick
Pattern Recognition and Analysis (MAS 622J)
MIT ID 984985527
paulfitz@ai.mit.edu

1 The data set

The data set for this project consisted of a collection of 1343 consumer photographs. The images had been hand-labeled as either indoor or outdoor. The goal of the project was to automate this classification task.

The photographs had also been placed into categories not directly relevant to the project, but interesting to glance at for what their distribution reveals about the nature of the problem. These categories were based on the presence or absence of green, people, or sky. Figure 1 shows the breakdown of the data into the different categories. The breakdown is in fact quite encouraging, because it shows that the indoor/outdoor category – which seems a fairly high-level semantic concept – is strongly correlated with categories we could imagine treating statistically. If we could replicate the human labeling of green and sky, the data suggests we could achieve success rates of around 86%. Detecting people would seem a little more difficult, but this dimension has in fact little power for discrimination anyway, since people tend to appear in any photograph regardless of category (naturally enough).

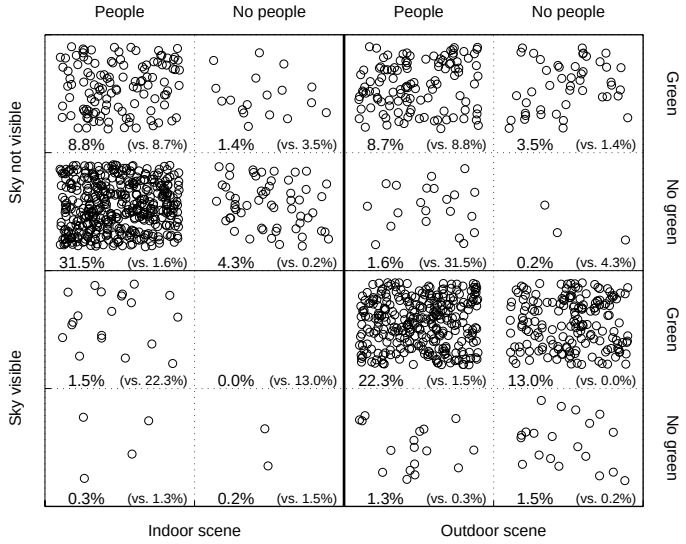

Figure 1: The project data, broken down into the categories for which labels were available.

2 Features provided

The following pre-calculated features were made available for this project:

- ▷ Color histogram (96-bin)
- ▷ Autoregressive texture feature
  (15 dimensional mean, 15×15 dimensional covariance)
- ▷ DCT–based texture feature
  (10 dimensional mean, 10×10 dimensional covariance)

These features were computed for each of the 1343 images. The features were also calculated for 16 sub-regions of the images. This makes for a very large amount of data, which suggests using either:

- ▷ A k nearest neighbor classifier, or
- ▷ Dimensionality reduction to make other classifiers applicable.

I chose to try both a k nearest neighbor classifier, and a linear classifier on a reduced subset of the features. However since the nature of the data as summarized in Figure 1 suggested that perhaps 80% of scenes can be classified quite easily, I wanted to start with a very simple classifier that captured these easy cases and could give me a baseline for comparison. The features listed above are not ideal for this without dimensionality reduction, so I chose to exercise an option given in the project of deriving features from the raw image data.

3 Features extracted

The details of the features I chose are not particularly important; I tried several and almost any arbitrary collection of simple features gave about the same discriminatory power. This suggests that they are in fact capturing the "easy" indoor/outdoor classification cases.

One group of features is shown in Figure 2. The first feature tries to capture the amount of green in the scene, since Figure 1 showed this was a particularly salient feature. The second pair of features are designed to detect sky by measuring the difference in brightness between the upper part of the image and the lower part, and (far less reliable) the amount of blue in the scene. The other features are easily measurable quantities that happened to show some separation in their distributions for indoor and outdoor scenes.

Indoor	Outdoor
	
↓ Amount of "green"	↑ Amount of "green"
↓ Amount of "blue"	↑ Amount of "blue"
↓ Degree of vertical change in brightness	↑ Degree of vertical change in brightness
↑ Degree of vertical orientation	↓ Degree of vertical orientation
↑ Degree of local homogeneity	↔ Degree of local homogeneity
↑ Degree of horizontal symmetry	↑ Degree of horizontal symmetry

Figure 2: Features extracted from raw images, and an example of how they might differ between "stereotypic" indoor and outdoor scenes.

A simple linear classifier (using Fisher's Linear Discriminant) based on these features gives a successful classification rate of 84.7% measured using leave-one-out cross-validation. For comparison, blind classification would be expected to have a success rate of 52.4% (the proportion of the data that was outdoors). This success rate confirms what Figure 1 suggests: a large proportion of images can be classified correctly quite easily. It is interesting to compare this figure with the best result quoted in the Szummer and Picard paper of 90.3%, also measured using leave-one-out cross-validation. This suggests that after the "easy" cases have been classified, things get very hard very fast.

To get an initial hint as to how advantageous the use of a k nearest neighbor classifier would be for this problem, I applied one to my feature set. The results are shown in Figure 3, with performance again measured by leave-one-out

cross-validation. Distance was measured using Euclidean distance over the features, after the features were scaled to have equal variances. I also tried the Mahalanobis distance metric, but this gave relatively poor results (around 80%). The performance level is quite flat once k is large. For $k = 13$ it is 85.2%, which appears just a little better than the results for the linear discriminant.

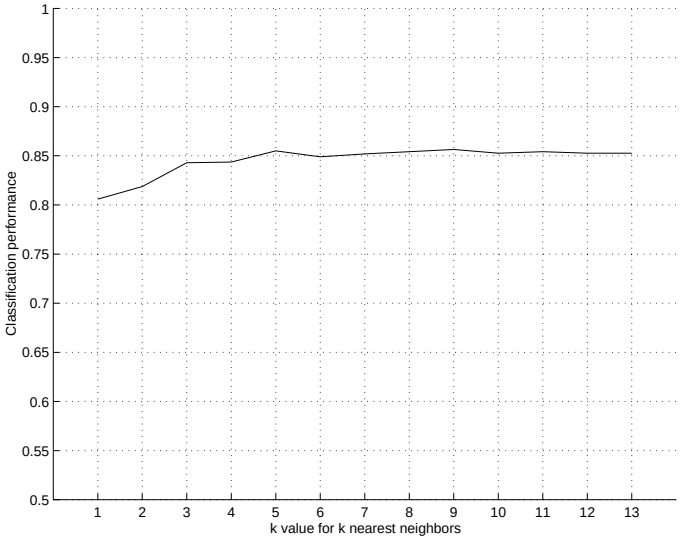

Figure 3: k nearest neighbor performance.

Given that performance levels of about 85% are easy to obtain, while a level of about 90% takes a great deal of effort, it is clear that increments in performance need to be measured quite finely. It therefore was important for me to understand how reliable my measurements of performance were. The advantage of using leave-one-out cross-validation is that it gives a performance measure with lower variance than alternatives. But there is one potential problem with it when used in conjunction with nearest neighbor classification in this problem domain. Examining the image database, there are a number of pairs of images such as those shown in Figure 10, and occasional sequences such as those in Figure 11 (towards the end of the report). When one of these images is left out, and is classified, the classifier still has an almost exact labeled copy of the image available to it. This seems dubious, in that it is possible to imagine scenarios where this leads to overly optimistic performance estimates that fail to generalize to new data. Although for large k it is difficult to imagine this having a significant effect, it seemed worth checking.

To determine whether or not image repeats do in fact have any impact on performance measures, I first tried to get an estimate of how often repeats occur. Figure 4 shows the distance in feature space between successive pairs of images, compared with the distance between each image and a random image. A difference in distribution is entirely to be expected, since successive images are by no means independent (for example, 82% of the images belong to the same indoor/outdoor category as the image preceding them). However note the comparatively high number of successive images with very low distance between them. Examining these directly shows up many image pairs.

Figure 5 gives a more direct feel for how significant (or otherwise) this issue is. The "d=0" line is a repeat of Figure 3, and shows the unencumbered performance of the k nearest neighbor classifier. The "d=1" line shows the performance of the classifier if the images immediately prior to and after a given image in the database are made inaccessible to the classifier. The performance drops for low k, but is essentially unchanged for large k. This shows that this issue is likely to be irrelevant for large k, as anticipated. If images within a distance of 10 in the database are withheld, performance drops by about 1%. For larger thresholds, it is likely that performance drops are caused more by starving the classifier of relevant samples than anything else.

4 Combining classifiers

To extend the classifier to use the features provided, rather than the toy features I derived, it is necessary to combine information from the different feature types and from different segments within the image. For both k nearest neighbors and linear discriminants, I did this by having individual classifiers for particular feature types in particular image segments, and then combining their results.

When combining output from classifiers based on linear discriminants, the approach I found to work best was a

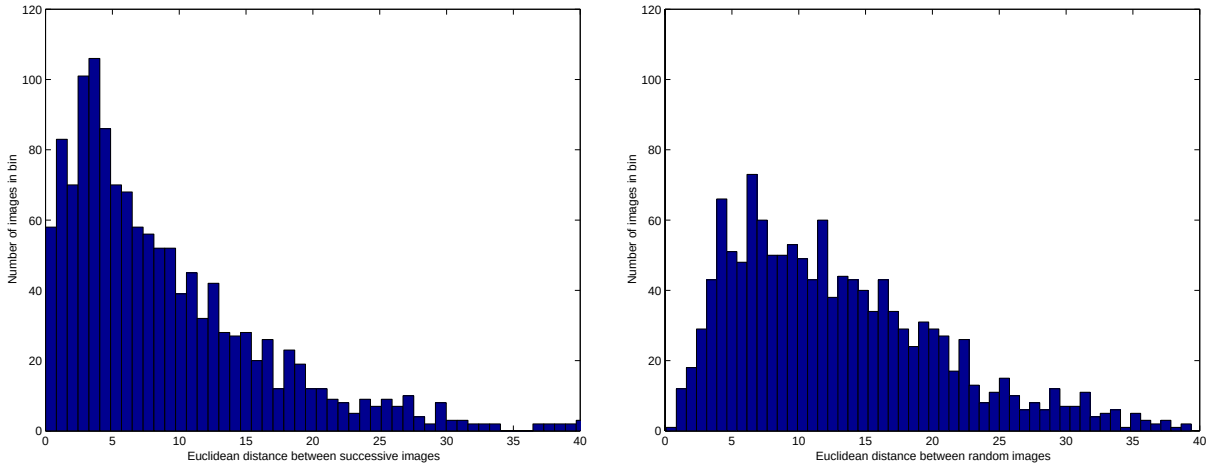

Figure 4: Distance in feature space between successive images (left) and random pairs of images (right).

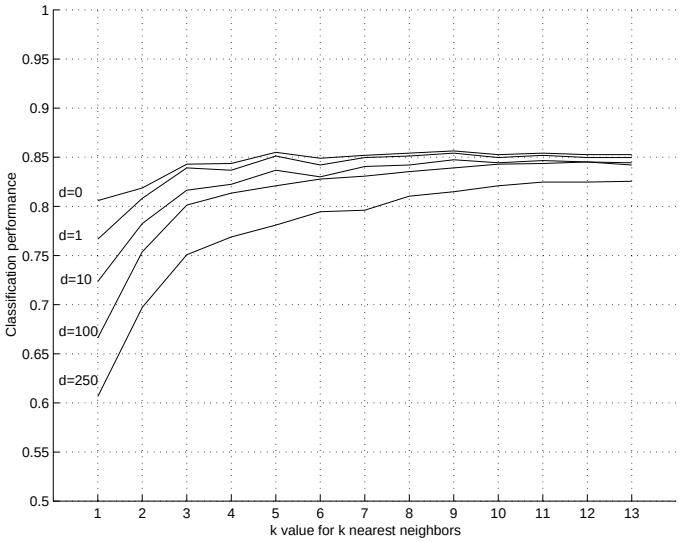

Figure 5: Constrained k nearest neighbor performance.

simple caricature of the "Bayesian" approach presented in class. Each feature type for each segment of the scene first went to an individual classifier. The results of these classifiers were then combined by assigning weights to each classifier according to the probabilities it assigned to the true classes of each training sample. I tried arranging the classifiers in a hierarchy according to feature type or according to image segment, but had best results with a flat arrangement.

A problem I had with the "Bayesian" approach for combining classifiers was that for a large data set, it tended to pick a single classifier. To sketch why, suppose the outputs from two classifiers λ_1 and λ_2 are being combined, and the data set consists of the same sample x_0 from class ω_0 repeated N times. The ratio between the weight chosen for λ_2 versus that chosen for λ_1 will be:

$$\frac{p(\omega_0|\lambda_2, x_0)^N}{p(\omega_0|\lambda_1, x_0)^N}$$

For $N = 1000$, $p(\omega_0|\lambda_1, x_0) = 0.61$, and $p(\omega_0|\lambda_2, x_0) = 0.6$, this becomes $6.6 \times 10^{-8} \approx 0$. In other words, the classifier with slightly better performance would be given full weighting, with the other classifier ignored. This is a very simple degenerate example, but this was the performance I met with for my actual data. In one sense it is reasonable behavior, but is quite sub-optimal for this particular problem. Better performance resulted from taking the N^{th} roots of the probability products, effectively weighting performances according to a geometric mean of the

probability they assign to the true class of each sample. The arithmetic mean worked equally well. These are clearly just hacks I introduced when the "Bayesian" approach gave poor results – I have no principled basis for this. Another equally unprincipled hack that had some value was to generate two separate weights for how much each classifier was trusted for each class, indoor or outdoor.

To use this method for combining classifiers, I needed the classifiers to produce probabilities as their output. My approach was simply to estimate the class-conditional distributions of the outputs as Gaussians and use these to generate the necessary probabilities. This worked well, although the actual distribution of the output of some classifiers was decidedly non-Gaussian. The better distributions looked like the example shown in Figure 6. Perhaps representing the distributions directly as histograms might have been preferable.

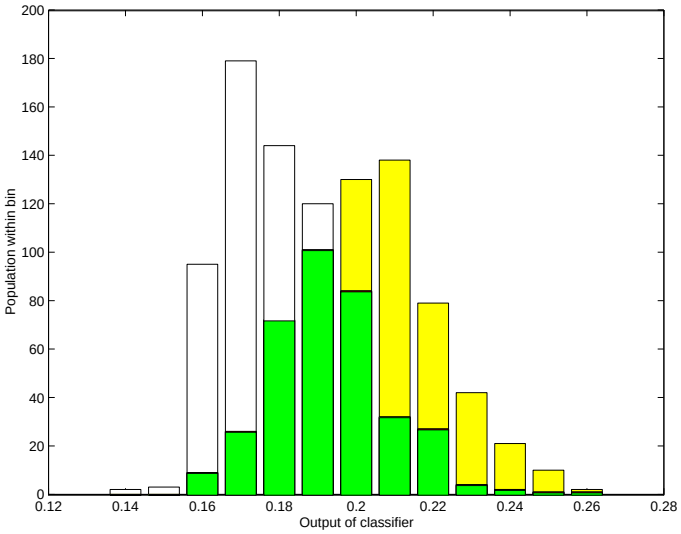

Figure 6: Output of an example classifier.

Another alternative I tried was using Fisher's Linear Discriminant over the output of individual classifiers. This gave reasonable results only when the number of classifiers was small. Many other possibilities existed for combining the classifiers, but it did not seem reasonable to fight so hard for a percentage point or two of questionable variance.

When combining k nearest neighbor classifiers, I tried a number of schemes but the best seemed to be a simple majority voting arrangement. A slight improvement resulted from weighting votes by how "unanimous" the decision of each individual classifier was, as given by the fraction of the image's neighbors that were in fact from the class chosen.

5 Results

The results shown here list the performance of combined classifiers based on Fisher's Linear Discriminant. Performance is averaged over 50 trials, with training on a randomly picked set of 800 samples and testing on a random set of 100 samples. The training and testing samples were chosen such that the two images before and two images after each test image in the database would not appear in the training set. Success rates varied across trials with a large standard deviation of about 3.2%. The average success rates shown have a standard deviation of about 0.2%.

Subdivided image features

The results show performance for various combinations of features. The variances of the DCT and MSAR features are useful for calculating distances using the Mahalanobis distance metric in k nearest neighbor classification, but I did not use them when working with linear classifiers (although the MSAR variance does actually show some power of discrimination, 66%). I performed dimensionality reduction on the Ohta color histograms by taking the first 10 principle components. The results are lower than those in the Szummer and Picard paper, but show some similar trends. In particular, best results are achieved when color and texture features are combined. The hand-crafted features from the simple classifier developed earlier do not give a significant performance increase. Other combinations of features not shown here result in poorer performance than combinations of which they are a subset.

Handcrafted	Ohta	DCT mean	MSAR mean	Performance
×	√	√	√	87.0%
×	×	√	√	84.9%
×	√	×	√	86.9%
×	√	√	×	85.5%
√	√	√	√	87.3%

Whole image features

It is interesting to briefly look at performance using features calculated for the whole image alone. The best result is less than that for subdivided image features, but only marginally so. This may suggest that my arrangement of classifiers is failing to capitalize on the extra detail subdivided image features offer.

Handcrafted	Ohta	DCT mean	MSAR mean	Performance
√	√	√	√	86.7%
×	√	√	√	84.0%

The best results for k nearest neighbor classification was 86.2%. This was using whole image features. For subdivided image features, I couldn't use the full set of 21488 segments and still get results in a reasonable length of time, and this hurt performance.

6 Analysis

Figure 7 characterizes the performance of the best classifier when the threshold used to separate (suspected) indoor scenes from outdoor scenes is varied. The plot shows the success rate on indoor scenes versus the failure rate on outdoor scenes. It confirms that there are indoor scenes that are being classified quite confidently as outdoor scenes, and vice versa. Some outdoor scenes are classified as indoor scenes with particularly high probability. While it is difficult to be sure what features are dominating the classification, these are typically pictures taken against the front of a house, or otherwise dominated by artificial structures. Figure 12 shows some examples. With lower confidence, close-ups and pictures containing large regions of white may be misclassified. Indoor scenes that get misclassified typically have brown or green colors, or indoor plants. Misclassifications are similar to those given in the Szummer and Picard paper.

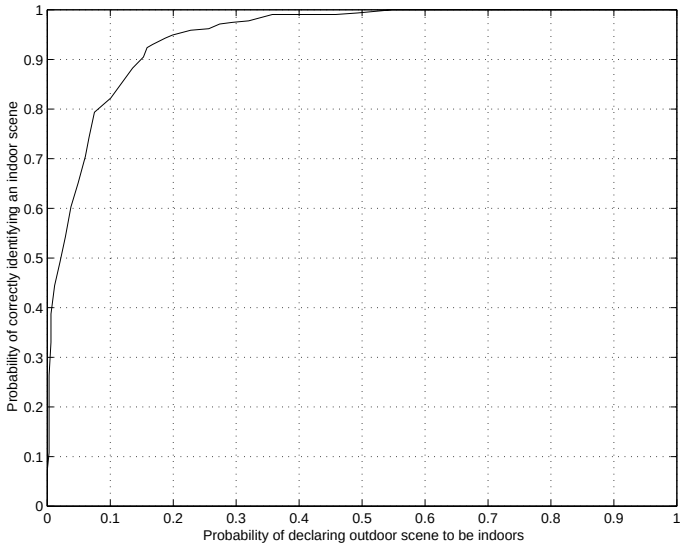

Figure 7: "ROC" curve for the classifier (taking indoor scenes to be "hits").

7 Allowing abstentions

One advantage of using a classifier based on linear discriminants is that it is fast and has much lower memory requirements than a k nearest neighbor classifier. It is worthwhile therefore considering whether a linear classifier could be used to catch many of the "easy" scenes to classify, while leaving difficult cases to a k nearest neighbor classifier. Figure 8 presents some evidence that this may be possible. Suppose the classifier in the Szummer and Picard paper is available, with its success rate of 90.3%. Then the linear classifier can be configured to classify 80% of what it receives, while passing the remaining difficult cases on to the nearest neighbor classifier. This may leave the overall success rate undiminished — assuming that the mistakes the linear classifier makes even when allowed to abstain would also have been made by the nearest neighbor classifier, which may or may not be the case.

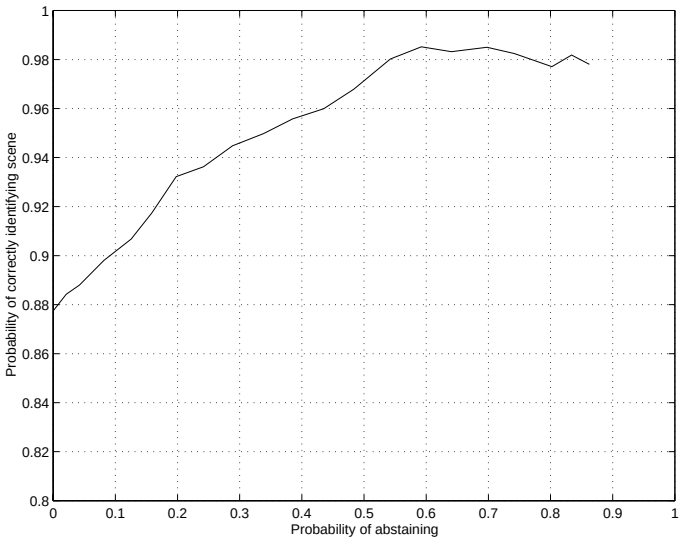

Figure 8: Effect of allowing abstentions on performance.

8 Generalization

Finally, it is worth considering how well the classifier will generalize to data beyond the database supplied. Figure 9) shows that the classifier gives good results with relatively few training samples. This doesn't prove anything, but does bode well for generalization.

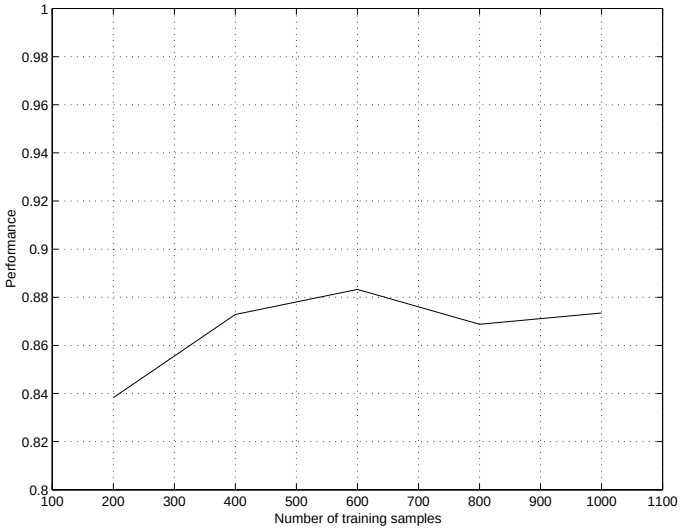

Figure 9: Effect of quantity of training data on performance.

Figure 10: Some image pairs in the database.

Figure 11: Some image sequences in the database.

Figure 12: Difficult outdoor scenes that are classified as indoor scenes with high confidence.