MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1669 Spetember, 1999

Boosting Image Database Retrieval

Kinh H. Tieu and Paul Viola
[1] **Artificial Intelligence Laboratory**
Massachusetts Institute of Technology
Cambridge, MA 02139
{*tieu,viola*}@*ai.mit.edu*

Abstract

We present an approach for image database retrieval using a very large number of highly-selective features and simple on-line learning. Our approach is predicated on the assumption that each image is generated by a sparse set of visual "causes" and that images which are visually similar share causes. We propose a mechanism for generating a large number of complex features which capture some aspects of this causal structure. Boosting is used to learn simple and efficient classifiers in this complex feature space. Finally we will describe a practical implementation of our retrieval system on a database of 3000 images.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research was provided in part by Nippon Telephone and Telegraph under grant number 9807-NTT03.

[1] http://www.ai.mit.edu/projects/lv

1 Introduction

Image database retrieval can be viewed as a particular case of information retrieval (IR) where the task is: "Given a few example images, learn to retrieve other examples of that class from a very large database." [9, 6, 13, 5, 7, 8, 11].

Retrieval differs from the more typical task of *classification* in that the number of potential image classes is extremely large and not known until query time. Thus traditional machine learning methods for classification are difficult to apply since they often require a small number of classes and a large set of labeled data $\{x^i, y^i\}$ (where x^i is an input image and y^i is the class label).

Several related issues conspire to make the learning for image retrieval surprisingly difficult. First, the number of training examples is small (perhaps 4 or 5 images); ii) the database of images is very large (perhaps 100,000 images); and iii) images belong to multiple classes. From this information one might conclude that the learning task for image database retrieval is essentially impossible.

An effective solution to this problem hinges on the discovery of a simplifying structure in the distribution of images. The distribution of natural images in constrained by the causal structure which generates them. The sparse causal structure of images is hard to ignore. A photograph chosen at random from the Web might be the "Eiffel Tower" or the "Taj Majal". While there are a very large number of possible objects, each image will contain just a few.

Classical image database retrieval approaches attempt to classify images based on a small number of common features, such as the number of vertical edges or the number of bright red pixels. Since both the Eiffel Tower and the Taj Majal have vertical edges, these features clearly cut across the boundaries of the causal structure. Learning the concept of "Eiffel tower" from example images using these features will require the learning algorithm to stake out a complex region in this feature space.

Our approach for image database retrieval depends on two related proposals: i) that images are best represented using a very large and selective set of features; and ii) that learning a query (image class) should quickly focus on just a few of these features.

In earlier work we proposed a scheme for computing a very large set of "complex features" [3]. In this paper we will expand upon and further justify this feature set. We will then describe an efficient mechanism for learning a query based on "Boosting" [4]. The trained classifier generalizes well, using between 20 and 50 complex features (less than one percent of the total number of complex features). These approaches are perfectly complementary: the complex feature set represents rich and subtle distinctions between images, while the boosted learning algorithm produces classifiers which generalize well and are very efficient.

2 Complex features

Most image database approaches use a very small set of "simple" features: for example the number of pixels of a given color, the number of vertical edges, or the number horizontal edges. Initial approaches focused exclusively on the color measures because they are pose insensitive [12]. Unfortunately these measures are also very non-selective. Many images have blue pixels or vertical edges. A query in such a simple feature space must stake out a very complex and irregular region in feature space. Finding such a region is a complex process that requires a lot of data. As a result these systems do not attempt to learn the query from the user. Instead users are asked to choose weights for the various features based on prior knowledge and intuition.

Our system differs from others because it detects not only simple first order features, such as oriented edges or color, but also measures how these first order features are related to one another. Thus by finding patterns between image regions with particular local properties, more complex – and therefore more discriminating – features can be extracted.

The process starts out by first extracting a feature map for each type of simple feature (there are 25 simple linear features including "oriented edges", "center surround" and bar filters). Each features map is rectified and downsampled by two. The 25 feature maps are then used as the input to another round of feature extraction (yielding 625 = 25 x 25 feature maps). The process is repeated again to yield 15,625 feature maps. Finally each feature map is summed to yield a single feature value.

More formally the characteristic signature of an image is given by:

$$S_{i,j,k,c}(I) = \sum_{pixels} E_{i,j,k}(I_c) \tag{1}$$

where I is the image, i, j and k are indices over the different types of linear filters, and the I_c are the different color channels of the image. The definition of E is:

$$E_i(I) = 2 \downarrow [abs(F_i \otimes I)] \tag{2}$$

$$E_{i,j}(I) = 2 \downarrow [abs(F_j \otimes E_i(I))] \tag{3}$$

$$E_{i,j,k}(I) = 2 \downarrow [abs(F_k \otimes E_{i,j}(I))] \; . \tag{4}$$

where F_i is the ith filter and $2 \downarrow$ is the downsampling operation.

We conjecture that these features do in fact reflect some of the sparse causal structure of the image formation process. One piece of evidence which supports this conclusion is the statistical distribution of the complex feature values. Evaluated across an image database containing 3000 images, these features are distinctly non-gaussian. The average kurtosis is approximately 8 and some of the features have a kurtosis as high as 120 (the gaussian has a kurtosis of 3). Observing this type of distribution in a filter is extremely unusual and hence highly meaningful. Since the distribution of the pixels is sub-gaussian, a random combination of pixel values would yield approximately gaussian distribution. [1] The discovery of features with these types of properties is highly unlikely. Recall that the final step in the feature computation is summing the pixels in each feature map. Given that the sum of independent variables tends toward a gaussian very quickly, the high kurtosis of the feature values is even more surprising. The kurtosis of the feature map pixels is much greater than the kurtosis of the features themselves. In our experiments, the top ten features had average kurtoses as high as 304. Experiments using only features with the lowest kurtoses resulted in poorer performance.

3 The Query Learning Process

At first it might seem that the introduction of tens of thousands of features could only make the query learning process infeasible. How can a problem which is difficult given ten to twenty features become tractable with 10,000. Two recent results in machine learning argue that this is not necessarily

[1] It is not unusual to observe kurtosis in the distribution of a non-linear feature. For example one could easily square a variable with gaussian distribution in order to yield a higher kurtosis. The complex features do *not* contain these sorts of non-linearities. At each level the absolute value of the feature map is computed.

a terrible mistake: "support vector machines (SVM)" and "boosting" [2, 4]. Both approaches have been shown to generalize well in very high dimensional spaces because they maximize the margin between positive and negative examples. Boosting provides the closer fit to our problem because it greedily selects a small number of features from a very large number of potential features.

In its original form the AdaBoost learning algorithm is used to combine a collection of weak classifiers to form a stronger classifier. The task of a weak learner is to search over a very large set of simple classification functions to find one with low error. The learner is called weak because we only expect that the returned classifier will correctly classify slightly more than one half of the examples. In order for the weak learner to be boosted, it is called upon to solve a sequence of learning problems. In each subsequent problem examples are reweighted in order to emphasize those which are incorrectly classified. The final strong classifier is a weighted combination of weak classifiers.

The weak learner used in the image query domain attempts to select the single complex feature along which the positive examples are most distinct from the negative examples. For each feature, the weak learner computes a gaussian model for the positives and negatives, and returns the feature for which the two class gaussian model is most effective. In practice no single feature can perform the classification task with 100% accuracy. Subsequent weak learners are forced to focus on the remaining errors through example re-weighting. The AdaBoost algorithm re-weights the incorrectly classified examples in the following way. Define the classification error rate for the k^{th} weak learner as η_k, the initial example weights as w_i, and define $\beta_k = \frac{\eta_k}{1-\eta_k}$. The new weights are $\hat{w}_i = w_i \beta_k^{1-e_i}$, where e_i is 1 if the i^{th} example is in error and 0 otherwise.

3.1 Learning from User Input

The user defines a new query in an interactive fashion. The first step is to select two or three positive examples. Users found that it was somewhat tedious to hand pick negative examples. Instead we randomly choose 100 images to form a set of generic negative examples. This policy for selecting negatives is somewhat risky because it is possible that the set may contain true positives. We run AdaBoost for 30 iterations which is usually sufficient to achieve zero error on the training set. Each image in the database is then ranked by the margin of the strong classifier. The first goal of an image retrieval program is to present the user with useful images which are related to the query. Since the learning algorithm is most certain about images with a large positive margin, a set of these images are presented. Without further refinement this set of images often contains many false positives.

Retrieval results can be improved greatly if the user is given the opportunity to select new training examples. Toward this end three sets of images are presented: i) test set images with large positive margin; ii) generic negative images which are close to the decision boundary; and iii) test set images which are close to the boundary. The first set is intended to allow the user to select new negative training examples which are currently labelled as strongly positive. The second set allows the user to discard generic negatives which are not true negatives. The third set allows the user to refine the decision boundary by labelling examples which determine the margin.

In every case the final query is produced by running AdaBoost for 30 iterations. This yields a strong classifier which is a simple function of 30 complex features. Since image databases are very large, the computational complexity of the final classifier is a critical aspect of image database retrieval performance.

4 Experiments

Experimental verification of image database retrieval programs is a very difficult task. There are few if any standard datasets, and there are no widely agreed upon evaluation metrics.

To test the classification performance of the system we constructed five classes of natural images (sunsets, lakes, waterfalls, fields, and mountains) using the Corel Stock Photo [2] image sets 1, 26, 27, 28, and 114 respectively [1, 10]. Each class contains 100 images. We also used sets 1 through 30 for a 3000 image data set to test retrieval performance.

Figures 1 and 2 show results of queries for race cars, flowers, and waterfalls in the 3000 image data set.

Figure 3 shows the average recall and precision for the five classes of natural images, each over a set of 100 random queries. Recall is defined as the ratio of the number of relevant images returned to the total number of relevant images. Precision is the ratio of the number of relevant images returned to the total number of images returned.

5 Conclusion

We have presented a framework for image database retrieval based on representing images with a very large set of highly-selective, complex features and interactively learning queries with a simple Boosting algorithm. The selectivity of the features allow effective queries to be formulated using just a small set of features. This supports our observation of the "sparse" causal structure of images. It also makes training the classifier simple, and retrieval on a large database fast.

Acknowledgments

This work is supported in part by Nippon Telegraph and Telephone.

References

[1] Corel Corporation. Corel stock photo images. http://www.corel.com.

[2] C. Cortes and V. Vapnik. Support vector networks. *Mach. Learn.*, 20:1–25, 1995.

[3] J. S. DeBonet and P. Viola. Structure driven image database retrieval. In *Advs. Neural Information Processing Systems*, volume 10, 1998.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Comp. and Sys. Sci.*, 55(1):119–139, 1997.

[5] M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the candid approach. *SPIE Vol. 2420 Storage and Retrieval for Image and Video Databases III*, pages 238–248, 1995.

[6] V. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: querying images by content using color, texture, and shape. *IS&T/SPIE 1993 Intern. Symp. on Electronic Imaging: Science & Technology*, 1908:173–187, 1993.

[7] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. Technical Report 3, MIT Media Lab, 1995.

[2] This publication includes images from the Corel Stock Photo images which are protected by the copyright laws of the U.S., Canada and elsewhere. Used under license.

Figure 1: Race cars: The top portion shows the positive examples followed by the top twenty retrieved images. The first row of the bottom portion lists the negative images in the training set which are close to the decision boundary and the second row lists images in the test set which are near the boundary.

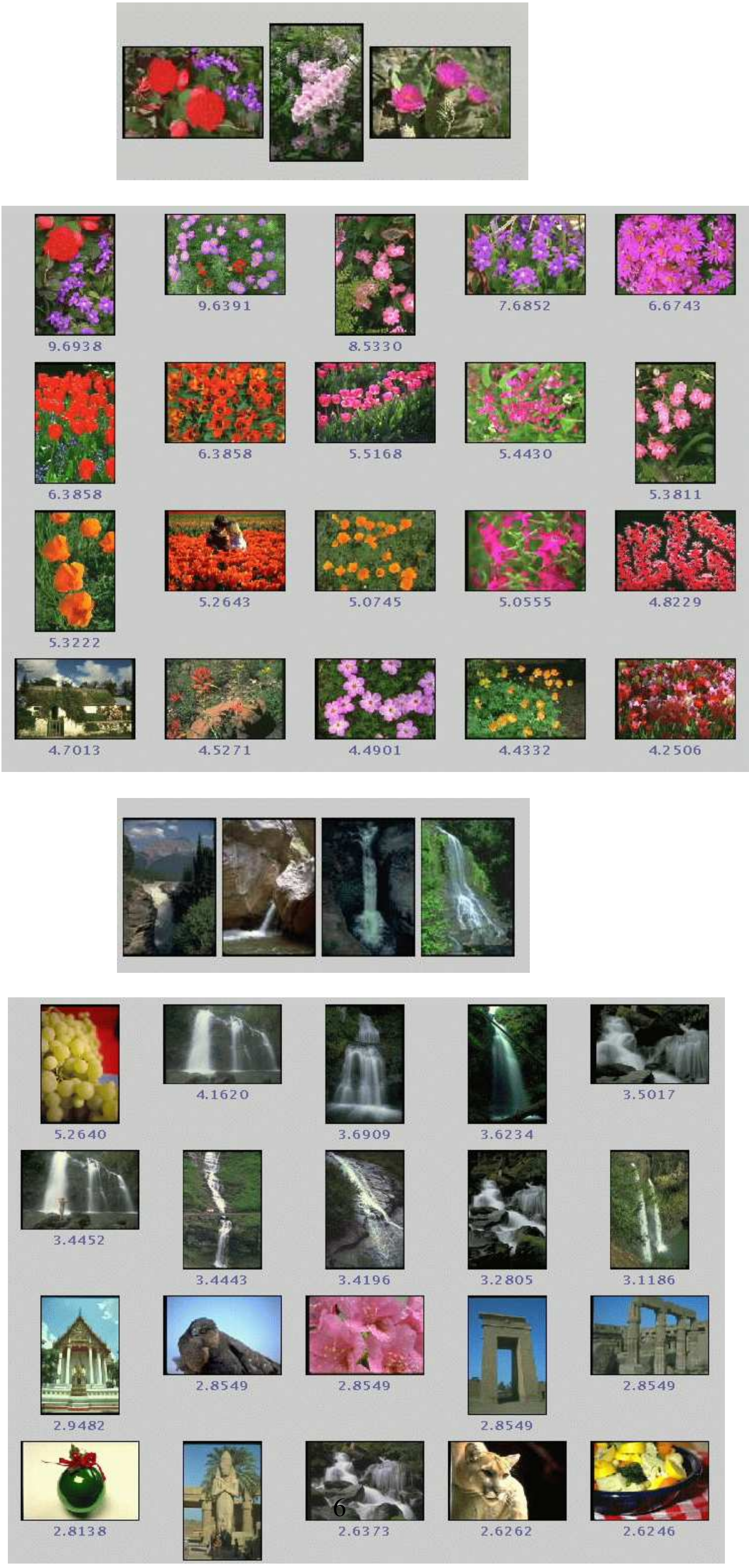

Figure 2: Flowers and waterfalls: The positive examples are shown first followed by the top twenty retrieved images.

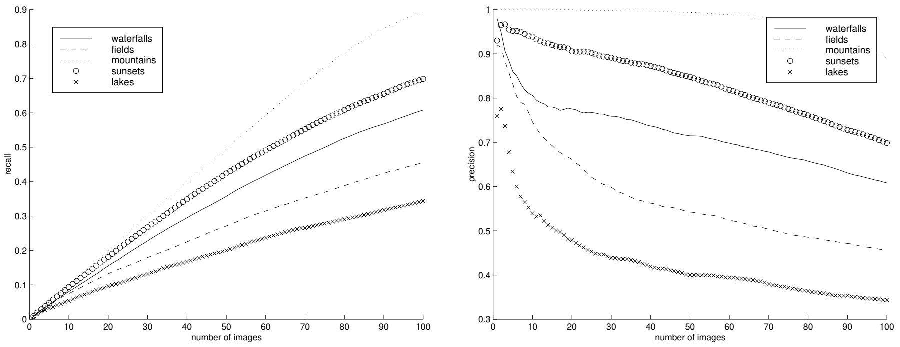

Figure 3: Average recall and precision for the five classes of natural images.

[8] R. W. Picard and T. Kabir. Finding similar patterns in large image databases. *Int. Conf. Acous., Speech, Sig. Proc.*, V:161–164, 1993.

[9] QBIC. The ibm qbic project. Web: http://www.qbic.almaden.ibm.com.

[10] A. L. Ratan and O. Maron. Multiple instance learning for natural scene classification. In *Int. Conf. Mach. Learn.*, pages 341–349, 1998.

[11] S. Santini and R. Jain. Gabor space and the development of preattentive similarity. In *Int. Conf. Patt. Recog.*, 1996.

[12] M. J. Swain and D. H. Ballard. Color indexing. *Int. J. Comp. Vis.*, 7(1):11–32, 1991.

[13] Virage. The virage project. Web: http://www.virage.com.