Multiple-Instance Learning for Natural Scene Classification

Oded Maron
Artificial Intelligence Lab
NE43-755, M.I.T.
Cambridge, MA 02139
oded@ai.mit.edu

Aparna Lakshmi Ratan
Artificial Intelligence Lab
NE43-739, M.I.T.
Cambridge, MA 02139
aparna@ai.mit.edu

Abstract

Multiple-Instance learning is a way of modeling ambiguity in supervised learning examples. Each example is a bag of instances, but only the bag is labeled - not the individual instances. A bag is labeled negative if all the instances are negative, and positive if at least one of the instances in positive. We apply the Multiple-Instance learning framework to the problem of learning how to classify natural images. Images are inherently ambiguous since they can represent many different things. A user labels an image as positive if the image somehow contains the concept. Each image is a bag, and the instances are various sub-regions in the image. From a small collection of positive and negative examples, we can learn the concept and then use it to retrieve images that contain the concept from a large database. We show that the Diverse Density algorithm performs well in this task, that simple hypothesis classes are sufficient to classify natural images, and that user interaction helps to improve performance.

1 INTRODUCTION

Scene classification is an open problem in machine vision and has applications in image and video database indexing. We investigate a method for learning visual concepts that encode the properties of a scene class from a small set of positive and negative examples. Extracted concepts are simple templates that capture some color and spatial properties of the class. Work by Lipson [Lipson et al., 1997] illustrates that simple, hand-crafted templates that describe the relative color and spatial properties in an image can be used successfully to classify natural scenes like fields, snowy mountains and waterfalls. In this paper we show that these templates can be learned. We describe a framework for learning scene-class concepts that can be used effectively for the task of content-based image retrieval from large databases. The learning framework we use in this paper is called Multiple-Instance learning [Dietterich et al., 1997],[Maron and Lozano-Pérez, 1998]. In this framework, examples are not labeled examples, but are labeled bags. Each bag is a collection of instances (Figure 1). A bag is labeled negative if all the instances in it are negative, and positive if at least one of the instances in it is positive. We use this framework to model the ambiguity in mapping an image to many possible templates which describe the image. Specifically, every image is a bag, and each possible template for describing the image is one instance in the bag. We discuss a method called Diverse Density [Maron and Lozano-Pérez, 1998] for learning concepts from Multiple-Instance examples.

We test our approach on images from the COREL photo library. We show that the system is succesful even when the hypothesis class involves very simple templates, and even when the images are sampled very coarsely. In addition, we show that user interaction (refining the hypothesis through the addition of more examples) is helpful in improving the performance of the learning system. In Section 2, we discuss previous and related work in image classification. We then describe the Multiple-Instance learning framework and the Diverse Density algorithm. In section 4 we detail our experimental setup and show results on various concept classes, hypothesis classes, and training regimes.

The third contribution of this paper (in addition to a novel application of Multiple-Instance learning and the discovery that surprisingly simple concepts do well on this task) is the development of a general architecture to combine ideas from the vision and machine learning communities. A key part of our system is the bag generator: a mechanism which takes an image and generates a set of instances, where each instance is a possible description of what the image is about. If an idealized object recognizer existed, then the bag generator would simply output a list of the objects in the image. The learning algorithm would be straightforward: find an intersection between the positive lists that didn't include elements from the negative lists. On the other extreme, if we had a learning algorithm that could handle billions of instances per bag, then we would not need an object recognizer. Instead, the bag generator would simply output every subcombination of pixels in the image. In this paper, we use a slightly more sophisticated bag generator (one that generates subregions), which limits the number of instances per bag and therefore allows us to use an algorithm such as Diverse Density. The key observation is that a better bag generator (progress in the vision community) leads to a simpler learning algorithm, while at the same time a better Multiple-Instance learning algorithm (progress in the machine learning community) allows us to use simpler segmentation algorithms. This is in contrast with the architecture of [Keeler et al., 1991], for example, where the learning mechanism is woven into the position-invariant representation of subimages.

2 IMAGE CLASSIFICATION SYSTEMS

In the past few years, the growing number of digital image and video libraries has led to the need for flexible, automated content-based image retrieval systems which can efficiently retrieve images from a database that are similar to a user's query. Because what a user wants can vary greatly, we also want to provide a way for the user to explore and refine the query by letting the system bring up examples.

One of the most popular global techniques for indexing is color-histogramming which measures the overall distribution of colors in the image. While histograms are useful because they are relatively insensitive to position and orientation changes, they do not capture the spatial relationships of color regions and thus have limited discriminating power. Many of the existing image-querying systems work on entire images or in user-specified regions by using distribution of color, texture and structural properties. The QBIC system [Flickner et al., 1995] is an example of such a system. Some recent systems that try to incorporate some spatial information into their color feature sets include [Smith and Chang, 1996, Huang et al., 1997, Belongie et al., 1998]. Promising work by Rubner [Rubner et al., 1998] on the earth mover's distance provides a metric that overcomes the binning problems of existing definitions of distribution distances for indexing. Most of these techniques require the user to specify the salient regions in the query image. One of the goals of our system is to learn the relevant color and spatial properties that best describe a particular class of natural scenes.

More recently, work by Lipson and Sinha ([Lipson et al., 1997]) in scene classification illustrates that predefined flexible templates that describe the relative color and spatial properties in the image can be used effectively for this task. The flexible templates constructed by Lipson [Lipson et al., 1997] encode the scene classes as a set of image patches and qualitative relationships between those patches. Each image patch has properties in the color and luminance channels. These templates describe the color relationship (relative changes in the R,G,B channels), luminance relationship (relative changes in the luminance channel) and spatial relationship between two image patches. Lipson hand-crafted these flexible templates for a variety of scene classes and showed that they could be used to classify natural scenes of fields, waterfalls and snowy mountains efficiently and reliably. For example, the following concept might be learned for the snowy-mountain class: "if the image contains a blue blob which is above a white blob which is above a brown blob, then it is a mountain". In this paper, we would like to learn such concepts for natural images given a small set of positive and negative examples.

All of the systems described above require users to specify precisely what they want. Minka and Picard [Minka and Picard, 1996] introduced a learning component in their system by using positive and negative examples which let the system choose image groupings within and across images based on color and texture cues; however, their system requires the user to label various parts of the scene, where as our system only gets a label for the entire image and automatically extracts the relevant parts of the scene. In this paper, we focus on learning natural scene concepts by extracting color and spatial relations between image patches using a small set of positive and negative examples.

Our system uses a small set of user-selected positive and negative examples to learn a scene concept which is used to retrieve similar images from the database. The system also lets the user add more positive and negative examples after each iteration in order to refine the concept.

3 MULTIPLE-INSTANCE LEARNING

In traditional supervised learning, a learning algorithm receives a training set which consists of individually labeled examples. There are situations where this model fails, specifically, when the teacher cannot label individual instances, but only a collection of instances. For example, given a picture containing a waterfall, what is it about the image that causes it to be labeled as a waterfall? Is it the butterfly hovering in the corner, the blooming flowers, or the white stream of water? It is impossible to tell by looking at only one image. The best we can say is that at least one of the objects in the image is a waterfall. Given a number of images (each labeled as waterfall or non-waterfall), we can attempt to find commonalities within the waterfall images that do not appear in the non-waterfall images. Multiple-Instance learning is a way of formalizing this problem, and Diverse Density is a method for finding the commonality.

In Multiple-Instance learning, we receive a set of *bags*, each of which is labeled positive or negative. Each bag contains many *instances*, where each instance is a point in feature space. A bag is labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the learner tries to induce a concept that will label unseen bags correctly. This problem is harder than even noisy supervised learning because the ratio of negative to positive instances in a positively-labeled bag (the noise ratio) can be arbitrarily high.

The multiple-instance learning model was only recently formalized by [Dietterich et al., 1997], where they develop algorithms for the drug activity prediction problem. This work was followed by [Long and Tan, 1996, Auer et al., 1996, Blum and Kalai, 1998], who showed that it is difficult to PAC-learn in the Multiple-Instance model unless very restrictive independence assumptions are made about the way in which examples are generated. [Auer, 1997] shows that despite these assumptions, the MULTINST algorithm performs competitively on the drug activity prediction problem. [Maron and Lozano-Pérez, 1998] develop an algorithm called *Diverse Density*, and show that it performs well on a variety of problems such as drug activity prediction, stock selection, and learning a description of a person from a series of images that contain that person.

3.1 MULTIPLE-INSTANCE LEARNING FOR SCENE CLASSIFICATION

In this paper, each training image is a bag. The instances in a particular bag are various subimages. If the bag is labeled as a waterfall (for example), we know that at least one of the subimages (instances) is a waterfall. If the bag is labeled as a non-waterfall, we know that none of the subimages contains a waterfall. Each of the instances, or subimages, is described as a point in some feature space. As discussed in section 4, we experimented with several ways of describing an instance. We will discuss one of them (`single blob with neighbors`) in detail: a subimage is a 2x2 set of pixels (referred to as a *blob*) and its four neighboring blobs (up, down, left, and right). The subimage is described as a vector $[x_1, x_2, \ldots, x_{15}]$, where x_1, x_2, x_3 are the mean RGB values of the central blob, x_4, x_5, x_6 are the differences in mean RGB values between the central blob and the blob above it, etc. One bag is therefore a collection of instances, each of which is a point in a 15-dimensional feature space. We assume that at least one of these instances is the template that contains the waterfall.

We would now like to find a description which will correctly classify new images as waterfalls or non-waterfalls. This can be done by finding what is in common between the waterfall images given during training and the differences between those and the non-waterfall images. The main idea behind the *Diverse Density* (DD) algorithm is to find areas in feature space that are close to at least one instance from every positive bag and far from every negative instance. The algorithm searches the feature space for points with high Diverse Density. Once the point (or points) with maximum DD is found, a new image is classified positive if one of its subimages is close to the maximum DD point. As seen in Section 4, the entire database can be sorted by the distance to the learned concept. Figure 1 is a schematic of how the system works.

In the following subsection, we will describe a derivation of Diverse Density and how we find the maximum in a large feature space. We will also show that the appropriate scaling of the feature space can be found by maximizing DD not just with respect to location in

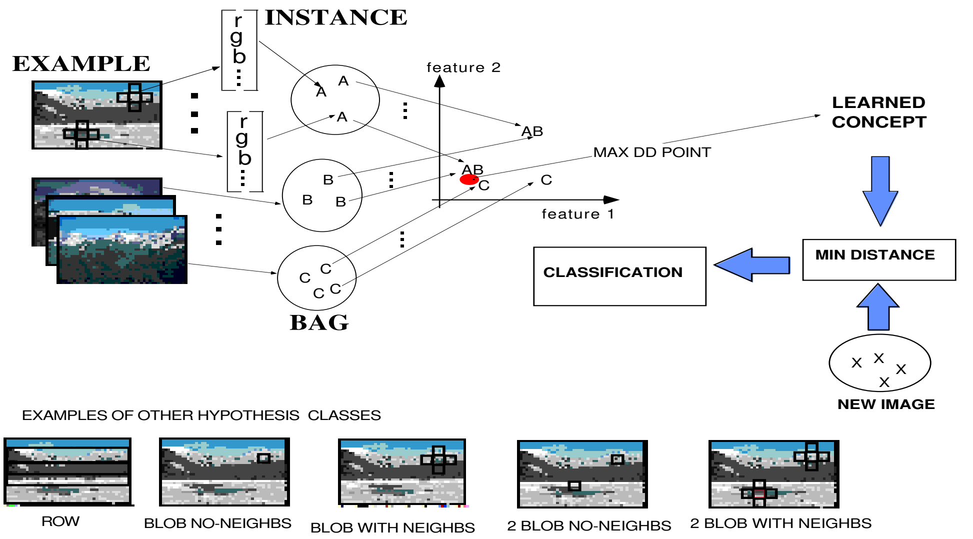

Figure 1: System Diagram

feature space, but also with respect to a weighting of each of the features.

3.2 DIVERSE DENSITY

In this section, we derive a probabilistic measure of Diverse Density. More details are given in [Maron, 1998]. We denote positive bags as B_i^+, and the j^{th} instance in that bag as B_{ij}^+. Likewise, B_{ij}^- represents an instance from a negative bag. For simplicity, let us assume that the true concept is a single point t in feature space. We can find t by maximizing $\Pr(t \mid B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^-)$ over all points in feature space. Using Bayes' rule and a uniform prior over the concept location, we see that this is equivalent to maximizing the likelihood:

$$\arg\max_t \Pr(B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^- \mid t). \qquad (1)$$

By making the additional assumption that the bags are conditionally independent given the target concept t, this decomposes into

$$\arg\max_t \prod_i \Pr(B_i^+ \mid t) \prod_i \Pr(B_i^- \mid t) \qquad (2)$$

which is equivalent (by similar arguments as above) to maximizing

$$\arg\max_t \prod_i \Pr(t \mid B_i^+) \prod_i \Pr(t \mid B_i^-) \qquad (3)$$

This is a general definition of Diverse Density, but we need to define the terms in the products to instantiate it. In this paper, we use the noisy-or model as follows:

$$\Pr(t \mid B_i^+) = 1 - \prod_j (1 - \Pr(t \mid B_{ij}^+)). \qquad (4)$$

The noisy-or model makes two assumptions: one is that for t to be the target concept it is caused by (hence close to) one of the instances in the bag. It also assumes that the probability of instance j not being the target is independent of any other instance not being the target.

Finally, we estimate the distribution $\Pr(t \mid B_{ij}^+)$ with a Gaussian-like distribution of $\exp(- \parallel B_{ij}^+ - t \parallel^2)$. A negative bag's contribution is likewise computed as $\Pr(t \mid B_i^-) = \prod_j (1 - \Pr(t \mid B_{ij}^-))$. A supervised learning algorithm such as nearest-neighbor or kernel regression would average the contribution of each bag, computing a density of instances. This algorithm computes a product of the contribution of each bag, hence the name Diverse Density. Note that Diverse Density at an intersection of n bags is exponentially higher than it is at an intersection of $n-1$ bags, yet all it takes is one well placed negative instance to drive the Diverse Density down.

The initial feature space is probably not the most suitable one for finding commonalities among images. Some features might be irrelevant or redundant, while small differences along other features might be crucial for discriminating between positive and negative examples. The Diverse Density framework allows us to find the best weighting on the initial feature set in the same way that it allows us to find an appropriate location in feature space. If a feature is irrelevant, then removing it can only increase the DD since it will bring positive instances closer together. On the other hand, if a relevant feature is removed then negative instances will come closer to the best DD location and lower it. Therefore, a feature's weight should be changed in order to increase DD. Formally, the distance between two points in feature space (B_{ij} and t) is

$$\| B_{ij}^{+} - t \|^2 = \sum_k w_k (B_{ijk} - t_k)^2 \tag{5}$$

where B_{ijk} is the value of the k^{th} feature in the j^{th} point in the i^{th} bag, and w_k is a non-negative scaling factor. If w_k is zero, then the k^{th} feature is irrelevant. If w_k is large, then the k^{th} feature is very important. We would like to find both t and w such that Diverse Density is maximized. We have doubled the number of dimensions in our search space, but we now have a powerful method of changing our representation to accomodate the task.

We can use also use this technique to learn more complicated concepts than a single point. To learn a 2-disjunct concept $t \vee s$, we maximize Diverse Density as follows:

$$\arg\max_{t,s} \prod_i (1 - \prod_j (1 - \Pr(t \vee s \mid B_{ij}^{+}))) \prod_i \prod_j \Pr(t \vee s \mid B_{ij}^{-}) \tag{6}$$

where $\Pr(t \vee s \mid B_{ij}^{+})$ is estimated as $\max\{\Pr(t \mid B_{ij}^{+}), \Pr(s \mid B_{ij}^{+})\}$. Other approximations (such as noisy-or) are also possible.

Finding the maximum Diverse Density in a high-dimensional space is a difficult problem. In general, we are searching an arbitrary landscape and the number of local maxima and size of the search space could prohibit any efficient exploration. In this paper, we use gradient ascent (since DD is a differentiable function) with multiple starting points. This has worked successfully because we know what starting points to use. The maximum DD point is made of contributions from some set of positive points. If we start an ascent from every positive point, one of them is likely to be closest to the maximum, contribute the most to it and have a climb directly to it. Therefore, if we start an ascent from every positive instance, we are very likely to find the maximum DD point. When we need to find both the location and the scaling of the concept, we perform gradient ascent for both sets of parameters at the same time (starting with all scale weightings at 1). The number of dimensions in our search space has doubled, though. When we need to find a 2-disjunct concept, we can again perform gradient ascent for all parameters at once. This carries a high computational burden because the number of dimensions has doubled, and we perform a gradient ascent starting at every pair of positive instances.

Our goal in the next section is to show that: (1) Multiple-Instance learning by maximizing diverse density can be used in the domain of natural scene classification, (2) simple concepts in low resolution images are sufficient to learn some of these concepts (3) adding false positives and false negatives over mutiple iterations (user interaction) can be used to improve the classifier performance.

4 EXPERIMENTS

In this section, we show four different types of results from running the system: one is that Multiple-Instance learning is applicable to this domain. A second result is that one does not need very complicated hypothesis classes to learn concepts from the natural image domain. We also compare the performance of various hypotheses, including the global histogram method. Finally, we show how user interaction would work to improve the classifier.

4.1 EXPERIMENTAL SETUP

We tried to learn three different concepts: waterfall, mountain, and field. For training and testing we used natural images from the COREL library, and the labels given by COREL. These included 100 images from each of the following classes: waterfalls, fields, mountains, sunsets and lakes. We also used a larger test set of 2600 natural images from various classes.

We created a *potential training set* that consisted of 20 randomly chosen images from each of the five classes mentioned above. This left us with a *small test set* consisting of the remaining 80 images from each of the five classes. We seperated the potential training set from the testing set to insure that results of using various training schemes and hypothesis classes can be compared fairly. Finally the *large test set* contained 2600 natural images from a large variety of classes.

For a given concept, we create an *initial training set* by picking five positive examples of the concept and five negative examples, all from the potential training set. After the concept is learned from these examples (by finding the point in and scaling of feature space with maximum DD), the unused 90 images in the potential training set are sorted by distance from the learned concept[1]. This sorted list can be used to simulate what a user would select as further refining examples. Specifically, the most egregious false positives (the non-concept images at the beginning of the sorted list) and the most egregious false negatives (the concept images at the end of the sorted list) would likely be picked by the user as additional negative and positive examples.

We attempted four different training schemes: `initial` is simply using the initial five positives and five negative examples. `+5fp` adds the five most egregious false positives. `+10fp` repeats the `+5fp` scheme twice. `+3fp+2fn` adds 3 false positives and 2 false negatives.

All images were smoothed using a gaussian filter and subsampled to 8×8. We used the RGB color space in these experiments. For every class and for every training scheme, we tried to learn the concept using one of seven hypothesis classes (Figure 1 shows some examples):

1. `row`: an instance is the row's mean color and the color difference in the rows above and below it.
2. `single blob with neighbors`: an instance is the mean color of a 2×2 blob and the color difference with its 4 neighboring blobs.
3. `single blob with no neighbors`: an instance is the color of each of the pixels in a 2×2 blob.
4. `disjunctive blob with neighbors`: an instance is the same as the single blob with neighbors but the concept learned is a disjunction of two single blob concepts.
5. `disjunctive blob with no neighbors`: an instance is the same as the single blob with no neighbors but the concept learned is a disjunction of two single blob concepts.
6. `two blob with neighbors`: an instance is the mean color of two descriptions of two `single blob with neighbors` and their relative spatial relationship (whether the second blob is above or below, and whether it is to the left or right, of the first blob).
7. `two blob with no neighbors`: an instance is the mean color of two descriptions of two `single blob with no neighbors` and their relative spatial relationship.

Learning a concept took anywhere from a few seconds for the simple hypotheses to a few days for the 2-blob and disjunctive hypotheses. The more complicated hypotheses take longer to learn because of the higher number of features and because the number of instances per bag is large (and to find the maximum DD point, we perform a gradient ascent from every positive instance). Because this is a prototype, we have not tried to optimize the running time; however, a more intelligent method of generating instances (for example, a rough segmentation using connected components) will reduce both the number of instances and the running time by orders of magnitude.

[1] An image/bag's distance from the concept is the minimum distance of any of the image's subregions/instances from the concept.

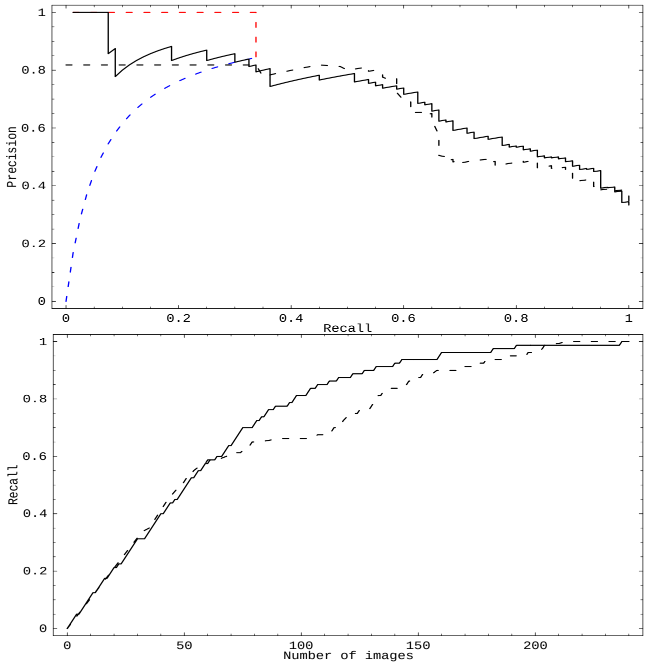

Figure 2: Comparison of learned concept (solid curves) with hand-crafted templates (dashed curves) for the mountain concept on 240 images from the small test set. The top and bottom dashed precision-recall curves indicate the best-case and worst-case curves for the first 32 images retrieved by the hand-crafted template which all have the same score.

4.2 RESULTS

In this section we show results of testing the various hypothesis classes, training schemes, and concept classes against the small test set and the larger one. The small test set does not intersect the potential training set, and therefore more accurately represents the generalization of the learned concepts. The large test set is meant to show how the system scales to larger image databases.

The graphs shown are precision-recall and recall curves. Precision is the ratio of the number of correct images to the number of images seen so far. Recall is the ratio of the number of correct images to the total number of correct images in the test set. For example, in Figure 3, the waterfall precision-recall curve has recall 0.5 with precision of about 0.7, which means in order to retrieve 40 of the 80 waterfalls, 30% of the images retrieved are not waterfalls. We show both curves for because (1) the beginning of the precision-recall is of interest to applications where only the top few objects are of importance, and (2) the middle of the recall curve is of interest to applications where correct classification of a large percentage of the database is important.

Figure 2 shows that the performance of the learned mountain concept is competitive with a hand-crafted mountain template (from [Lipson et al., 1997][2]). The test set consists of 80 mountains, 80 fields, and 80 waterfalls. It is disjoint from the training set. The hand-crafted model's precision-recall curve is flat at 84% because the first 32 images all receive the same score, and 27 of them are mountains. We also show the curves if we were to retrieve the 27 mountains first (best-case) or after the first five false positives (worst-case).

In Figure 3, we show the performance of the best hypothesis and training method on each concept class. The dashed lines show the poor performance of the global histogram method. The solid lines in the precision-recall graph show the performance of `single blob with neighbors` with `+10fp` for waterfalls, `row` with `+10fp` for fields, and `disjunctive blob with no neighbors` with `+10fp` for mountains. The solid lines in the recall curve show the performance of the `single blob with neighbors` with `+10fp` for waterfalls, `single blob with neighbors` with `+3fp+2fn` for fields, and `row` with `+3fp+2fn` for mountains. This behavior continues for the larger test set.

In Figure 4, we show the precision-recall curves for each of the four training schemes. We average over all concepts and all hypothesis classes. We see that performance improves with user interaction. This behavior continues for the larger test set as well.

In Figure 5, we show the precision-recall and recall curves for each of the seven hypotheses averaged over all concepts and all training schemes. Note that these curves are for the larger 2600 image database. We see that the single blob with neighbors hypothesis has good precision. We also see that the more complicated hypothesis classes (i.e. the disjunctive concepts and the two-blob concepts) tend to have better recall curves.

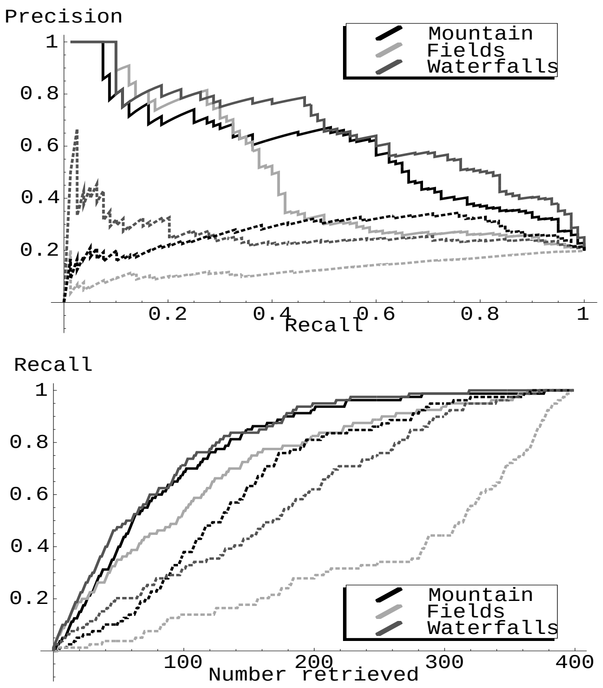

Figure 3: The best curves for each concept using a small test set. Dashed curves are the global histogram's performance.

In Figure 6, we show a snapshot of the system in action. The system is trained using training scheme `+10fp` for the waterfall concept. It has learned a waterfall concept using the `single blob with neighbors` hypothesis. The learned waterfall concept is that somewhere in the image there is a blob whose left neighbor is less blue, whose own blue value is 0.5 (where RGB values are in the $[0, 1]$ cube), whose neighbor below has the same blue value, whose neighbor above has the same red value, whose green value is 0.55, whose neighbor above has the same blue value and whose red value is 0.47. These properties are weighted in the order given, and any other features were found to be irrelevant. A new image has the rating of the minimum distance of one of its instances to the learned concept, where the distance metric uses the learned scaling to account for the importance of the relevant features. As we can see in the figure, this simple learned concept is able to retrieve a wide variety

[2] Lipson's classifier was modified to give a ranking of each image, rather than its class.

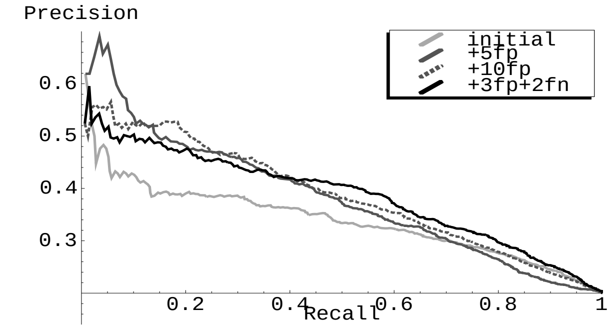

Figure 4: Different training schemes, averaged over concept and hypothesis class, using a small test set.

of waterfall scenes.

The top 20 images in the figure are the training set. The first 10 images are the initial positive and negative examples used in training. The next 10 images are the false positives added. The last 30 images are the top 30 returned from the large dataset.

5 CONCLUSIONS

In this paper, we have shown that Multiple-Instance learning by maximizing diverse density can be used to classify images of natural scenes. Our results are competitive with hand-crafted models, and much better than a global histogram approach. We have also demonstrated that simple learned concepts that capture color relations in low resolution images can be used effectively in the domain of natural scene classification. Our experiments indicate that complicated concepts (e.g. disjunctive concepts) tend to have better recall curves and that user interaction (adding false positives and false negatives) over multiple iterations can improve the performance of the classifier. Our architecture, by seperating the bag generator from the learning mechanism, allows progress in the field of computer vision to benefit the field of machine learning and vice versa.

Acknowledgements

We thank Tomás Lozano-Pérez, Eric Grimson, and Pam Lipson for their advice and AFOSR ASSERT program Parent Grant#:F49620-93-1-0263, and ARPA under ONR contract N00014-95-1-0600 for their support of this research.

References

[Auer *et al.*, 1996] Peter Auer, Phil M. Long, and A. Srinivasan. Approximating hyper-rectangles: learning and pseudorandom sets. In *Proceedings of the 1996 Conference on Computational Learning Theory*, 1996.

[Auer, 1997] Peter Auer. On Learning from multi-instance examples: Empirical evalutaion of a theoretical approach. In *Proceedings of the 14th Internation Conference on Machine Learning*, 1997.

[Belongie *et al.*, 1998] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and Texture based image segmentation using EM and its application to content-based image retrieval. In *International Conference on Computer Vision*, 1998.

[Blum and Kalai, 1998] A. Blum and A. Kalai. A Note on Learning from Multiple-Instance Examples. *To appear in Machine Learning*, 1998.

[Dietterich *et al.*, 1997] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the Multiple-Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence Journal*, 89, 1997.

[Flickner *et al.*, 1995] M. Flickner, , and et al. Query by image and video content: The QBIC System. *IEEE Computer*, 28:23–32, 1995.

[Huang *et al.*, 1997] J. Huang, S. Ravikumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition*, 1997.

[Keeler *et al.*, 1991] James D. Keeler, David E. Rumelhart, and Wee-Kheng Leow. Integrated Segmentation and Recognition of Hand-Printed Numerals. In *Advances in Neural Information Processing Systems 3*. Morgan Kauffman, 1991.

[Lipson *et al.*, 1997] P. Lipson, E. Grimson, and P. Sinha. Context and Configuration Based Scene Classification. In *Computer Vision and Pattern Recognition*, 1997.

[Long and Tan, 1996] P. M. Long and L. Tan. PAC-learning axis alligned rectangles with respect to product distributions from multiple-instance examples. In *Proceedings of the 1996 Conference on Computational Learning Theory*, 1996.

[Maron and Lozano-Pérez, 1998] O. Maron and T. Lozano-Pérez. A framework for Multiple-Instance learning. In *Advances in Neural Information Processing Systems 10*. MIT Press, 1998.

[Maron, 1998] O. Maron. Learning from Ambiguity. Doctoral Thesis, Dept. of Electrical Engeineering and Computer Science, M.I.T., June 1998.

[Minka and Picard, 1996] T. Minka and R. Picard. Interactive Learning using a society of models. In *Computer Vision and Pattern Recognition*, 1996.

[Rubner *et al.*, 1998] Y. Rubner, C. Tomasi, and L. Guibas. A Metric for Distributions with Applications to Image Databases. In *Proceedings of IEEE Int. Conf. on Computer Vision*, 1998.

[Smith and Chang, 1996] J. Smith and S. Chang. VisualSEEK: a fully automated content-based image query system. In *Proc. ACM Interational Conference on Multimedia*. Morgan Kaufmann, 1996.

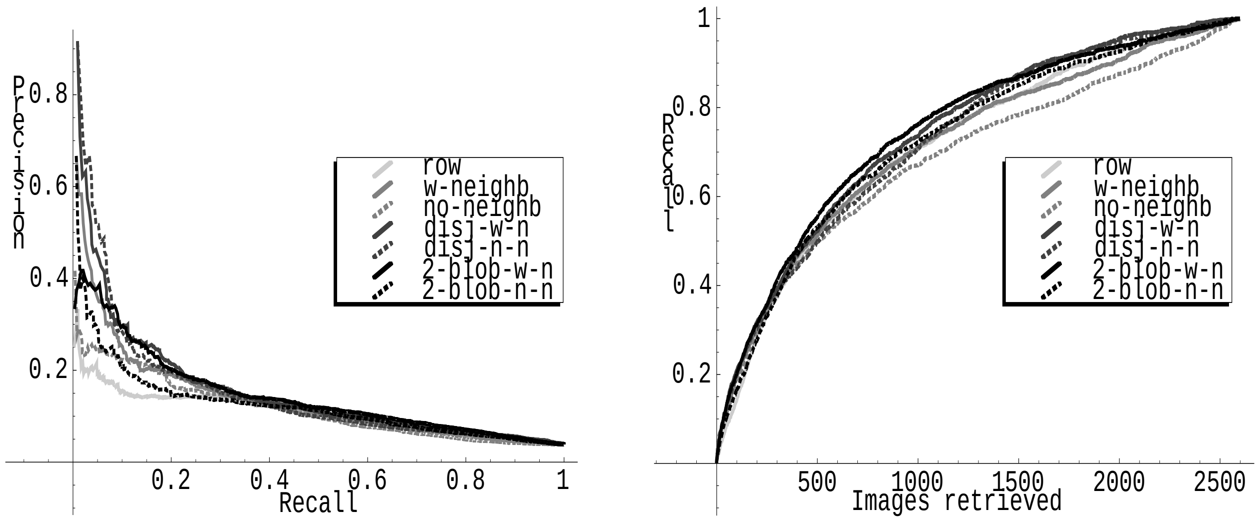

Figure 5: Different hypothesis classes averaged over concept and training scheme, using a large test set with 2600 images.

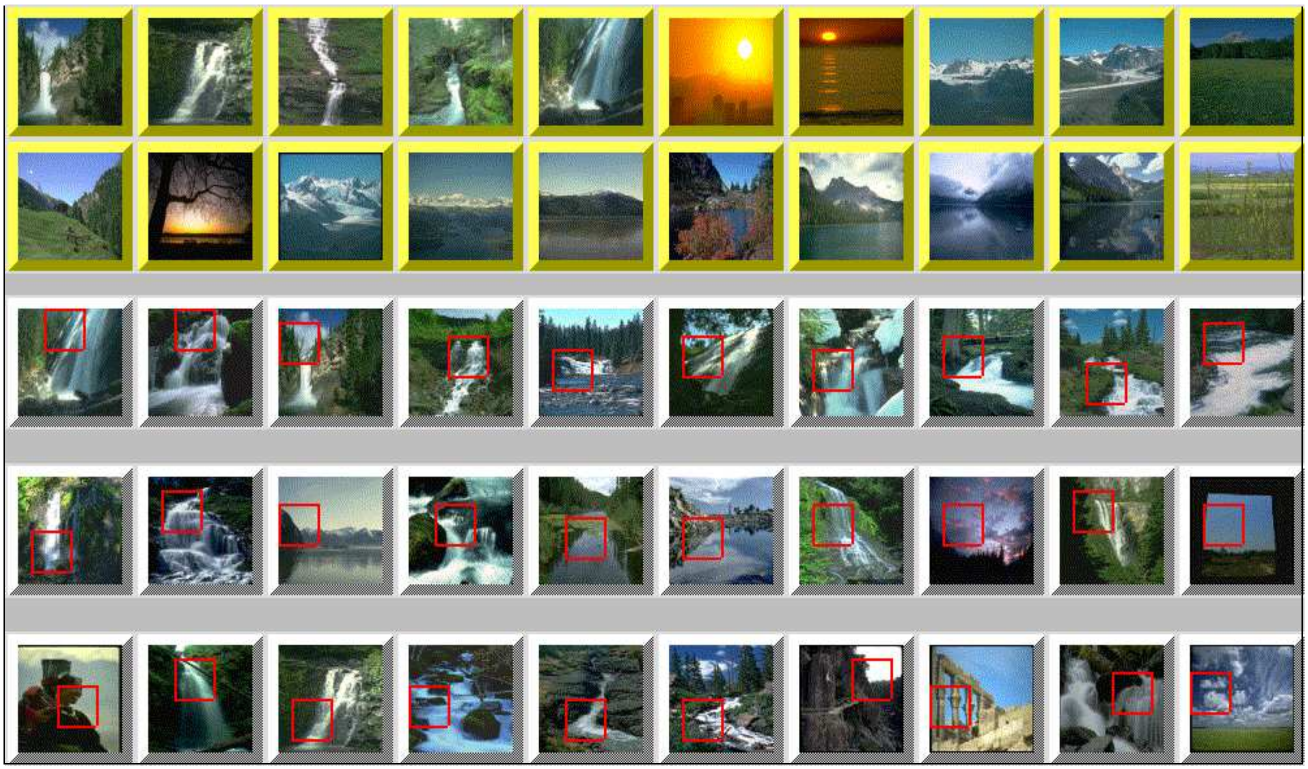

Figure 6: Results for the waterfall concept using the `single blob with neighbors` concept with `+10fp`. Top row: Initial training set–5 positive and 5 negative examples. Second Row: Additional false positives. Last three rows: Top 30 matches retrieved from the large test set. The red squares indicate where the closest instance to the learned concept is located.