A Framework for Learning Query Concepts in Image Classification *

A. L. Ratan O. Maron W. E. L. Grimson T. Lozano-Pérez

Artificial Intelligence Lab,
Massachusetts Institute of Technology.
(aparna,oded,welg,tlp) @ai.mit.edu

Abstract

In this paper, we adapt the Multiple Instance Learning paradigm using the Diverse Density algorithm as a way of modeling the ambiguity in images in order to learn "visual concepts" that can be used to classify new images. In this framework, a user labels an image as positive if the image contains the concept. Each example image is a bag of instances (sub-images) where only the bag is labeled — not the individual instances (sub-images). From a small collection of positive and negative examples, the system learns the concept and uses it to retrieve images that contain the concept from a large database. The learned "concepts" are simple templates that capture the color, texture and spatial properties of the class of images.

We introduced this method earlier in the domain of natural scene classification using simple, low resolution sub-images as instances. In this paper, we extend the bag generator (the mechanism which takes an image and generates a set of instances) to generate more complex instances using multiple cues on segmented high resolution images. We show that this method can be used to learn certain object class concepts (e.g. cars) in addition to natural scenes.

1 Introduction

In the past few years, the growing number of digital image and video libraries has led to the need for automated content-based image retrieval systems. Because what a user wants can vary greatly, we also want to provide a way for the user to explore and refine a query by letting the system bring up examples. In this paper, we develop a general architecture to learn visual query concepts similar to the ones that Lipson pre-defined in [11] from a small number of examples. The extracted concepts are simple, flexible templates that capture some color, texture and spatial properties of a class of images.

The learning framework we use in this paper is called Multiple-Instance learning. In this framework, examples are not labeled examples, but take the form of labeled bags (e.g. the whole image is labeled but not individual regions). Each bag is a collection of instances (e.g. subimages). 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 what it represents. We discuss a method called Diverse Density [13] for learning concepts from Multiple-Instance examples. We introduced this framework in [14] and showed that it can be used effectively for the task of natural scene classification.

In the domain of natural scenes, the predominant color and spatial relations explained most parts of the image and were preserved even in very low resolution images. For many other types of queries, we might want to retrieve new images based on smaller subregions of the query image. For these queries, we need to preserve some detailed high resolution properties of image sub-regions for classification. In this paper, we address these issues by introducing a few extensions to the previous system. We use the segmented regions in high resolution images to generate complex instances (conjunctions and disjunctions of segmented components) using color, texture and simple geometric properties. We show that the new instances can be used to retrieve images of object classes like cars in addition to images of natural scenes like fields and waterfalls. Our experiments on "car" classification show that a combination of cues (color, texture and simple shape), some feature selection, and more complicated concepts (conjunctions) play a significant role in improving classifier performance.

2 Image Classification Systems

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. Some recent systems try to incorporate some spatial information into their color feature sets [22, 5, 8, 2] among others. More recently, work by Lipson [11] illustrates that pre-defined flexible templates that capture the relative color and spatial properties in the im-

*Research supported in part by Parent Grant#:F49620-93-1-0263, and ARPA under ONR contract N00014-95-1-0600

0-7695-0149-4/99 $10.00 © 1999 IEEE

age can be used effectively to classify natural scenes like mountains and waterfalls. In this paper, we would like to learn such concepts/templates for natural scenes and other object classes from a small set of positive and negative examples.

All of the systems described above require users to specify precisely the desired query. Minka and Picard [15] 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. Forsyth et al.[5] learned representations for horses by training the system using the appropriate color, texture and edge configuration. Cox et al. [4] modeled relevance feedback in a Bayesian framework that uses an explicit model of the user's selection process. More recently, Nastar et al. [16] introduced a relevance feedback system for query refinement over time using a set of positive and negative images. They estimate the distribution of relevant images and minimize the probablility of retrieving non-relevant images. Our system uses the Multiple Instance learning framework to learn a query concept and feature weights automatically from a small set of positive and negative examples. The system finds the simplest concept (template) that can be used to explain the query set and retrieves similar images from the database by finding the location of the concept in the image.

3 Multiple-Instance Learning

In traditional supervised learning, a learning algorithm receives a training set which consists of individually labeled examples. There are situations, however, 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? 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 (labeled as waterfalls and non-waterfalls), 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 and instances 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 [7], where they develop algorithms for dealing with the drug activity prediction problem. This work was followed by [1], 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. [13] 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 in image database indexing

In this paper, each training image is a bag. The instances in a particular bag are various subimages. Each of the instances, or subimages, is described as a point in some high dimensional feature space. We experimented with combinations of cues to describe an instance. Details on the generation of these instances will be discussed in section 6.

Given some labeled bags, we would like to find a description which will correctly classify new images as waterfalls and non-waterfalls. 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. In Section 6, new images are sorted by their distance from the maximum DD point.

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 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. 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. In other words, the intersection of all positive bags minus the union of all negative bags is a single point. 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 $\exp(- \parallel B_{ij}^+ - t \parallel^2)$ [1]. 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

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

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 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:

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

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 in a bag, we are very likely to find the maximum DD point. When we need to find both the location and the scaling of the

[1] Any distribution which is monotonically decreasing as distance from the mean increases would be suitable here.

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.

4 Segmentation

In the domain of natural scenes, the predominant color and spatial relations in the target concept explained most parts of the image and were preserved even in very low resolution images. For many other types of queries, we might not be interested in the whole image and might want to retrieve new images based on specific parts of the query image. For example, we might want to retrieve images that contain a certain object class like cars or tigers. For these queries, we need to preserve some detailed high resolution properties of image sub-regions in order to be able to (a) extract the query object from the positive examples as the target concept and (b) retrieve new images containing the object.

To do this, the system needs to ignore the background and clutter and select out the object of interest from the positive examples. The architecture we have described in the previous sections can be applied here as well. The instances here differ from the low-resolution sub-images used in [14] in that they are segmented components of a high resolution image with a rich set of descriptive properties. There have been several methods proposed in the literature for segmenting images into multiple regions with coherent properties [18, 19]. In this paper we use the method by Felzenszwalb et al. [19] to roughly decompose the image into the dominant set of regions using its color properties. The segmentation helps generate instances that correspond to salient regions in the image and reduces both the number of instances and the running time for more complicated concepts (e.g. conjunctions).

5 Bag Generator using Multiple Cues

In this section, we will demonstrate how the Diverse Density algorithm can be used to learn a representation for the class of cars by describing a new bag generator which uses a combination of cues and rough segmentation to generate the instances. For example, a flexible car template which encodes the geometric properties of the two wheels and the color and texture relations between the wheel regions and its neighbours could be used to detect cars while accomodating within-class variations in color, texture and pose. We would like to show that we can use the Multiple Instance Learning framework to learn the car concept from a few examples by using a set of primitives which includes color, texture and simple shape properties (a wheel/circle detector in this case). We also demonstrate the feature selection capability of the system which determines the relevant features that are needed to detect cars vs other natural scene (e.g. fields and waterfalls).

In our experiments, all images were pre-segmented using the algorithm described in [19]. We tried to learn the concept using the hypothesis classes given below:

1. cc1: each instance is a 13 dimensional vector which describes the color, texture and basic shape properties of a segmented region in the image. $< x_1, x_2, ...x_{13} >$. x_1, x_2 gives the position of the centroid of the component, x_3, x_4, x_5 gives the representative color of the component.

We use the Hausdorff matching technique [9, 17] to detect instances of a simple shape primitive (circle) in the segmented component. The circles are detected in the edge detected image. The Hausdorff matcher searches for the circle over all 2 dimensional translations and scale thus detecting ellipses as well as circles. x_6, x_7, x_8, x_9 gives the location and scaling of the circle in the image. x_6 gives the distance to the detected circle, x_7 gives the Hausdorff fraction that says how well it matched the model circle, x_8, x_9 gives the scaling of the model circle in the image.
We use the output of the Steerable Pyramid [20] to get a multi-scale, multi-orientation image decomposition with a 4-orientation-band filter set in a neighborhood around the centroid of the connected component. This is included to give some measure of texture where the segmented regions are non-uniform (e.g. snow capped mountains, wheel regions of a car) [2]. $x_{10}, x_{11}, x_{12}, x_{13}$ give the steerable filter responses for 4 orientations across 3 scales.

2. cc2 : an instance is the conjunction of two connected components (cc1 vectors)

6 Experiments

In the following sections, we show three different types of results from running the system: one is that Multiple-Instance learning is applicable to this domain. A second result is that while one does not need very complicated hypothesis classes to learn concepts from the natural image domain [14], more complex instances which describe a combination of properties using multiple cues help extend the system to learn a

[2] These filter responses were obtained using the Steerable Filter Software Library designed by Simoncelli et al.[20, 21]

more diverse set of queries (e.g. object queries). In the case of object queries, the target concept is only a small part of the image. The third result compares the performance of the system on the "car" class (1) using simple (single component) vs. more complicated (conjunction) concepts (2) with and without feature selection. This result shows that complicated concepts and feature selection help improve classifier performance.

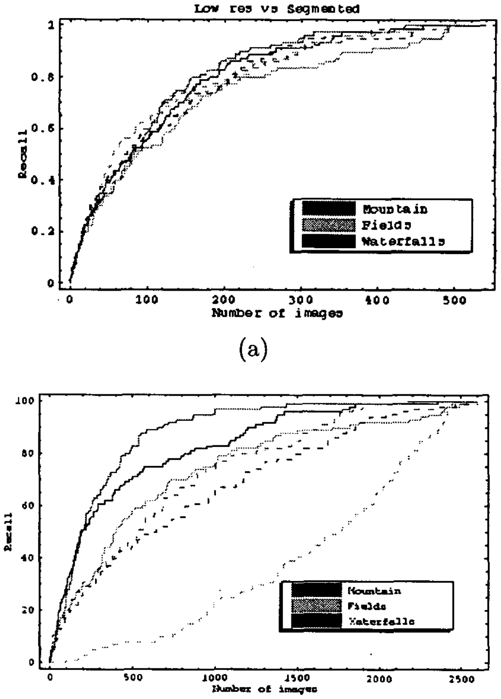
7 Experimental setup

We tried to learn each of four concepts ("fields", "mountains", "waterfalls", "cars") using the new bag generator. 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, lakes, cars, race-cars. We had a training set of 140 images with 20 images from each of the classes, a small test set of 538 images which was disjoint from the training set, and a larger test set of 2600 images (which included the training images). The training scheme used five positives and five negative examples. We attempted different training schemes: `initial` is simply using the initial five positives and five negative examples. `+5fp` adds the five most egregious false positives after a round of testing on a held-out set of images. `+10fp` repeats the `+5fp` scheme twice. This simulates the behavior of a user interacting with the system.

8 Natural Scene Classification

We compared the best curves for three natural scene classes ("fields", "mountains", "waterfalls") with the previous version of the system from [14] that uses low resolution sub-images as instances. Figure 1 shows that the performance is comparable to our previous results in this domain and is significantly better than global histogramming. This performance continues for the larger test set of 2600 images as well. Figure 3 shows a snapshot of the new system working on the larger dataset of 2600 images for the waterfall concept. In addition, the current system has the advantage of better running time and a more general framework which can be extended to other classes of objects.

Even when using extremely low resolution images (8x8), learning a concept using the previous system took anywhere from a few seconds for the simple hypotheses to a few days for the more complicated hypotheses (conjunctions and disjunctions). 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). However, the current system uses a better method of generating instances (a rough segmentation using connected components) and this reduces both the number of instances and the running time by orders of magnitude. There are roughly 15–20 components per image and the system takes a few seconds to learn the simple `cc1` concept and tens of minutes to learn the more complicated ones



(a)

(b)

Figure 1: (a) Comparison of classification performance with segmentation (solid curves) and without segmentation (dashed curves) for 3 natural scene concepts on 538 images from the small test set. (b) This figure shows the poor performance of global histogramming on the larger dataset of 2600 images.

9 Classification of Cars

Figure 2 (a) compares the performance of the two hypothesis classes `cc1` (denoted by the solid lines) and `cc2` (denoted by the dashed lines) on the "cars" class. We see that both the recall and precision-recall[3] are better for the conjunction concept `cc2`. This indicates

[3] 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

that a single circle detector alone is not sufficient to classify cars.

Figure 2 (b and c)shows the role of scaling (feature selection) in the presence of multiple cues. The solid lines represent the case where there is no feature selection and the dashed lines represent the case where there is feature selection. Both the recall and precision-recall is significantly better when there is feature selection. This indicates that for this class certain dimensions are redundant or irrelevant and selecting the salient dimensions helps improve classifier performance.

Figure 4 shows a snapshot of the system working on the cars concept on the test set of 538 images which are disjoint from the training set. A new image has the rating of the minimum distance of one of its instances to the learned car concept, where the distance metric uses the learned scaling to account for the importance of the relevant features. As we see, the system is able to extract the concept without having the user specify salient regions within the example images.

10 Conclusions

In this paper, we have demonstrated a general system for query learning and for image classification using a small number of examples. We describe an architecture that allows the user to train the system, by selecting positive and negative examples, letting the system create and use an initial template based on those examples and finally refine the template by adding incorrect matches (false positives). Our approach to training indexing systems treats query learning as a Multiple Instance Learning problem and builds on the method of Diverse Density. The system has been tested for a few classes on a database of 2600 images from the COREL photo library.

We show that rough segmentation of high resolution images into salient connected components greatly reduces the number of instances and the running time of the algorithm especially when low resolution pixel-instances are not sufficient (e.g. when we move from the domain of natural scenes to objects). We also have experiments to show that by using a more complex representation and features within the same framework, we can learn certain object classes (e.g. cars). Our experiments on "car" classification show that segmentation, combination of cues (color, texture and simple shape), some feature selection and more complicated concepts (conjunctions) help improve classifier performance.

In this paper, we have used a small combination of primitives which were suitable for natural scenes and vehicles to demonstrate the learning, feature selection and retrieval capabilities of our system. However, the issue of which primitives/invariants we need to use to generate the instances is still open. In the future, we would like to explore this issue and extend the bag generator to use a base set of low-level primitives that captures a larger set of class concepts.

This paper will be available on-line (with color images) at the URL http://www.ai.mit.edu/people/aparna/

Acknowledgements

We would like to thank Kinh Tieu for providing data for some of the comparison experiments used to evaluate the system.

References

[1] Peter Auer, Phil M. Long, and A. Srinivasan. Approximating hyper-rectangles: learning and pseudorandom sets. In *COLT*, 1996.

[2] 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.

[3] A. Blum and A. Kalai. A note on learning from multiple-instance examples. *To appear in Machine Learning*, 1998.

[4] I. Cox, M. Miller, T. Minka, P. Yianilos An Optimized Interaction Strategy for Bayesian Relevance Feedback *CVPR*, 1998.

[5] D. Forsyth, M. Fleck Body Plans *CVPR*, 1997.

[6] J. S.DeBonet and P. Viola Structure driven image database retrieval *NIPS*, 10, 1997.

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence Journal*, 89, 1997.

[8] J. Huang, S. Ravikumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR*,1997.

[9] D.P. Huttenlocher and G.A. Klanderman and W.J. Rucklidge. Comparing images using the Hausdorff distance. In *IEEE Trans. Patt. Anal. Mach. Intell.*,15:850–863, 1993.

[10] James D. Keeler, David E. Rumelhart, and Wee-Kheng Leow. Integrated segmentation and recognition of hand-printed numerals. In *NIPS*,3, Morgan Kauffman, 1991.

[11] P. Lipson, E. Grimson, and P. Sinha. Context and configuration based scene classification. In *CVPR*, 1997.

[12] P. M. Long and L. Tan. Pac-learning axis alligned rectangles with respect to product distributions from multiple-instance examples. In *COLT*, 1996.

[13] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*,10, MIT Press, 1998.

[14] O. Maron and A. Ratan. Multiple Instance Learning for Natural Scene Classification In *ICML*, 1998.

[15] T. Minka and R. Picard. Interactive learning using a society of models. In *CVPR*, 1996.

[16] C.Nastar, M.Mitschke, C.Meilhac Efficient Query Refinement fpr Image Retrieval In *CVPR*, 1998.

[17] W. J. Rucklidge Locating Objects Using the Hausdorff distance. In *ICCV*, 457–464, 1994.

[18] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. In *Proc. CVPR*, 1997.

[19] P. Felzenszwalb and D. Huttenlocher. Image Segmentation Using Local Variation. In *Proc. CVPR*, 1998.

[20] W. Freeman and E.H. Adelson The Design and Use of Steerable Filters. In *PAMI*, 13(9), 1991.

[21] E. Simoncelli, R. Buccigrassi and H. Farid. Shiftable Pyramid Software Library. Developed at *Computer Inf. Science Dept, University of Pennsylvania*, 1995.

[22] J. Smith and S. Chang. Visualseek: a fully automated content-based image query system. In *Proceedings of the ACM Int. Conf. on Multimedia*, 1996.

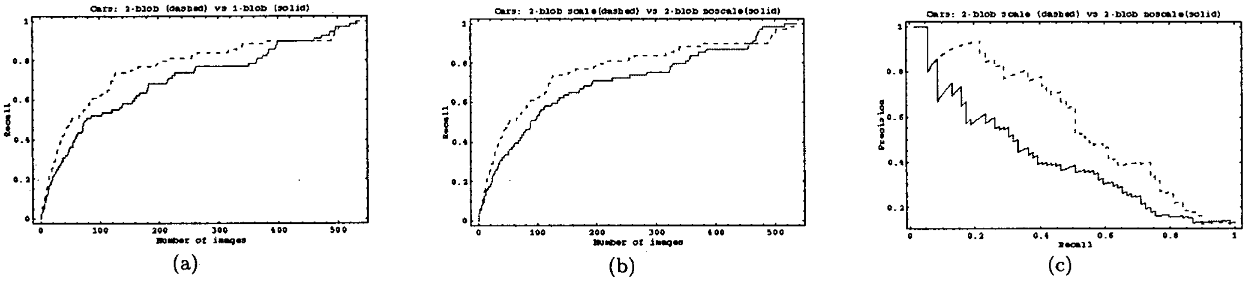


Figure 2: (a) Recall curve comparing classification performance for "cars" using a single component concept (solid curve) vs a conjunction of components concept (dashed curve). (b) and (c) Recall and precision curves comparing classification performance for "cars" with feature selection (dashed curves) and without feature selection (solid curves).

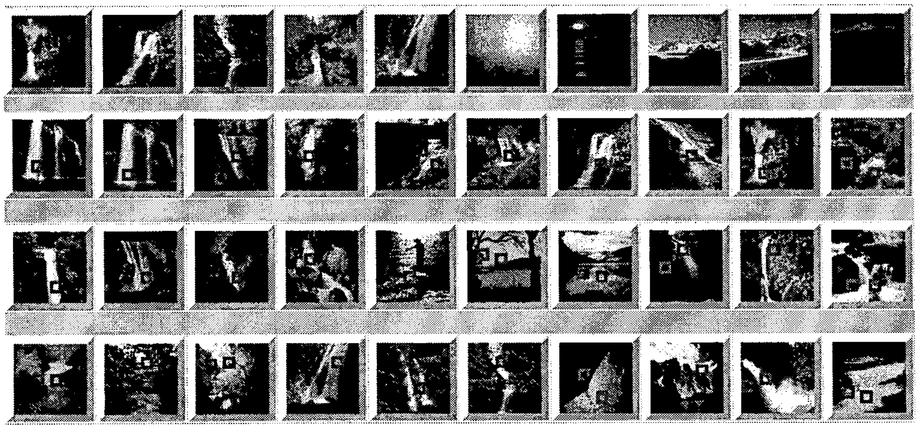


Figure 3: The figure shows the results for the waterfall concept using the cc2 concept. Top row: Initial training set–5 positive followed by 5 negative examples. Last three rows: Top 30 matches retrieved from the test set. The red squares indicate where the closest instance to the main component of the learned concept is located.

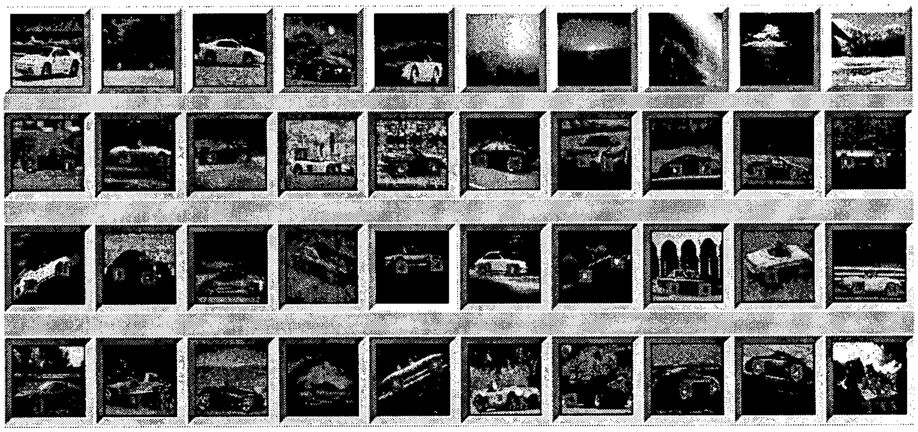


Figure 4: The figure shows the results for the cars concept using the cc2 concept. Top row: Initial training set–5 positive followed by 5 negative examples. Last three rows: Top 30 matches retrieved from the small test set of 538 images. The red squares indicate where the closest instance to the learned concept is located.

Session 4-A

Sensors