RC 22231 (W0111-015) November 2, 2001
Computer Science

IBM Research Report

Learning to Annotate Video Databases

Milind R. Naphade, John R. Smith, Sankar Basu, Belle L. Tseng, Ching-Yung Lin
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Learning to Annotate Video Databases

Milind R. Naphade, Ching-Yung Lin, John R. Smith, Belle Tseng, Sankar Basu

Pervasive Media Management Group
IBM T J Watson Research Center
Hawthorne, NY 10532,USA

ABSTRACT

Model-based approach to video retrieval requires ground-truth data for training the models. This leads to the development of video annotation tools that allow users to annotate each shot in the video sequence as well as to identify and label scenes, events, and objects by applying the labels at the shot-level. The annotation tool considered here also allows the user to associate the object-labels with an individual region in a key-frame image. However, the abundance of video data and diversity of labels make annotation a difficult and overly expensive task. To combat this problem, we formulate the task of annotation in the framework of supervised training with partially labeled data by viewing it as an exercise in active learning. In this scenario, one first trains a classifier with a small set of labeled data, and subsequently updates the classifier by selecting the most informative, or most uncertain subset of the available data-set. Consequently, propagation of labels to yet unlabeled data is automatically achieved as well.

The purpose of this paper is primarily twofold. The first is to describe a video annotation tool that has been developed for the purpose of annotating generic video sequences in the context of a recent video-TREC benchmarking exercise. The tool is semi-automatic in that it automatically propagates labels to "similar" shots, which requires the user to confirm or reject the propagated labels. The second purpose is to show how active learning strategy can be potentially implemented in this context to further improve the performance of the annotation tool. While many versions of active learning could be thought of, we specifically report results on experiments with support vector machine classifiers with polynomial kernels.

Keywords: Video annotation, Active learning, Model based retrieval, Support vector machines, Relevance feedback.

1. INTRODUCTION

Accessing content at a semantic level is essential for efficient utilization of multimedia databases. Recent work in the semantic indexing of video includes.[7] Studies reveal that most queries to content-based retrieval systems are phrased in terms of keywords. To support exhaustive indexing of content using such semantic labels, it is necessary to annotate the multimedia databases. While manual annotation is being used currently, automation of this process to some extent can greatly reduce the burden of annotating large databases. We investigate a system that relies on supervised and semi-supervised learning to aid the annotation of multimedia content. Using simple active learning techniques, the system builds multimodal representations of semantic classes. These representations are then used to aid the annotation through smart propagation of labels to content similar in terms of the representation. We want to compare the utility of this annotation tool using objective measures of human intervention required with and without the use of the active learning algorithm. A by-product of this work can be the creation of a labeled corpus with crude models of semantics that can be further refined off-line to build efficient and accurate models of semantic concepts using supervised training methods.

As the user starts annotating the video database, the learning component of the system attempts to propagate user-provided labels to regions with similar spatio-temporal characteristics. Proximity in feature space and consistency with pre-annotated user-provided labels are used to suggest labels for the videos that the user is annotating. The active learning component of the system prompts the user to label those segments of video, which according to the algorithm form the minimal subset that lead to maximum disambiguation. The user then needs to label the examples *actively* selected by the system. The aim is to propagate the labels with confidence levels desired by the user with minimal user-interaction in terms of the number of annotations that

the user needs to provide. We show that the number of examples that need to be annotated can be significantly reduced without sacrificing propagation performance by the incorporation of a support vector machine based active learner.

2. ACTIVE LEARNING, SUPPORT VECTOR MACHINES AND VIDEO ANNOTATION

In supervised learning, the task is to design a classifier when the sample data-set is completely labeled. In situations where there is an abundance of data but labeling is too expensive in terms of money and time the strategy of active learning can be adopted. In this approach, one trains a classifier based only on a selected subset of the labeled data-set. Based on the current state of the classifier, one selects some of the "most informative" subset of the unlabeled data so that knowing labels of the selected data is likely to greatly enhance the design of the classifier. The selected data is to be labeled and added to the training set. This procedure can be repeated, and our goal is to have the human label as little data as possible to achieve a certain performance. This approach to boost classification performance without labeling a large data set has a large literature, some recent examples of which are cited in the bibliographoical references at the end of the present paper.[3,5,8,14] It may be remarked in this context that the larger problem of using unlabelled data to enhance classifier performance, of which active learning can be viewed as a specific solution, can also be alternatively approached via other passive learning techniques such as those elaborated in[8,11] etc.

We shall briefly review and discuss few previous approaches to active learning in general. Unless otherwise noted most of the discussion in the following section relates to learning theory in general, and is not necessarily in the domain of video retrieval.

2.1. Active learning

Active learning strategies can be broadly classified into three different categories. One approach to active learning is "uncertainty sampling" in which instances in the data that need to be labeled are iteratively identified based on some measure that suggests that the predicted labels for these instances are uncertain. A variety of methods for measuring uncertainty can be used. For example, a single classifier can be used that produces an estimate of the degree of uncertainty in its prediction and an iterative process can then select some fixed number of instances with maximum estimated uncertainty for labeling. The newly labeled instances are then to be added to the training set, and a classifier trained using this larger training set. This iterative process is continued until the training set reaches a specified size or a certain performance is achieved on a validation set. This method can be further generalized by using more than one classifier. For example, one classifier can determine the degree of uncertainty and another classifier can perform classification. An alternative, but related approach - sometimes referred to as *Query by Committee*. Here, two different classifiers consistent with the already labeled training data are randomly chosen. Instances of the data for which the two chosen classifiers disagree are then candidates for labeling. As an example of a *Adaptive resampling* methods are being increasingly used to solve the classification problem in various domains with high accuracies. A third strategy to active learning is to exploit such techniques. A boosting-like technique that adaptively resample data biased towards the misclassified points in the training set and then combine the predictions of several classifiers is also used.[5] For comprehensive listing of prior work in this are of research the reader is also referred to the work of Iyengar, Apte and Zhang.[5]

Even among the uncertainty sampling methods mentioned above a variety of classifiers and measures of degree of uncertainty of classification have been proposed. Two specific classifiers that have been widely used for this purpose are the Support Vector Machine (SVM) classifier and gaussian Mixture Model based classification scheme. For SVM classifier the distance of an unlabeled data-point from the separating hyperplane in the high dimensional feature space could be taken as a measure of uncertainty (alternatively, a measure of confidence in classification) of the data-point. For the gaussian mixture model based classifier the likelihood of the new data-point given the current parameters of the gaussian mixture model can be used as measure of this confidence. This strategy has been used[14] in the context of relevance feedback for video retrieval[15] for text classification, whereas discussions on gaussian mixture model based active learning have also appeared in other contexts.[3]

Our approach described in this paper will be more akin to the SVM approach[14] but the difference is in the fact that we are interested in applying it for *persistent semantic annotation*.

There exists still other examples of the use of active learning to address issues arising from a large volume of unlabeled data. The problem of parameter estimation in dynamic bayesian networks, where the examples to be labeled are again chosen depending on the information content of the data presented, has been addressed.[16] In addition, the more ambitious problem of learning the causal structure of the graph associated with the bayesian network has been considered[17] in an active learning framework, where the learner suggests experiments that are more informative towards revealing the causal structure of the problem.

2.2. Support Vector Machines

Recently, Support Vector Machines (SVM) have enjoyed large popularity in the machine learning community and excellent treatments on SVMs are available.[4,18] Essentially, a support vector machine is a linear classifier that attempts to find a separating hyperplane that maximally separates the two classes under consideration. A distinguishing feature of the support vector machines is that although it makes use of a linear hyperplane separator between the two classes, the hyperplane lives in a higher dimensional induced space obtained by nonlinearly transforming the feature space in which the original problem is posed. This "blowing up" of the dimension is achieved by a transformation of the feature space by proper choice of a Kernel function that allows inner products in the high dimensional induced space to be conveniently computed in the lower dimensional feature space in which the classification problem is originally posed. Commonly used examples of such (necessarily nonlinear) kernel functions are polynomial kernels, Radial basis function etc. Design of the kernel function occupies much of recent interest in SVM research. The virtue of nonlinearly mapping the feature space to a higher dimensional space is that it can be shown analytically that the generalization capability of the classifier is, thus, largely enhanced. This fact, strangely at odds with the "curse-of-dimensionality" lies at the heart of the success of SVM classifiers with relatively small data-sets. The key idea here is that the true complexity of the problem is not necessarily in the classical dimension of the feature space, but in the the so called Vapnik Chervonenkis dimension,[18] which does not increase in transforming the space via properly chosen kernel function.

Another interesting and somewhat more intuitive fact is that the feature points near the decision boundary have a rather large influence on determining the position of the boundary. These so called *support vectors* turn out to be few in number and facilitate computation to a large degree. In our context of use of SVMs for active learning, these play an even more important role. Some of the unseen data samples that lie near the decision boundary are potential candidates for new support vectors and also the most *informative*. Indeed, in our application we learn an SVM on the existing labeled data and select the next data point to be worthy of labeling only if it comes "close" to the separating hyperplane in the induced higher dimensional space. Of course, it is possible to think of several ways of measuring this closeness to the separating hyperplane. We will describe this is more detail in the section 4 to follow.

2.3. Multimedia Annotation

Early work on video annotation has been reported in the work of Minka and Picard,[9] which considers a system that combines low level textures with high level descriptions to assist users in annotation. The system dynamically selects multiple texture models based on the behavior of the user in selecting a region for labeling. This work presents the use of pre-computed trees of clusters as internal representations. This allows for flexibility to allow combinations of clusters from different models. If no single model was the best then it could produce a new hypothesis by by pruning and merging relevant pieces from the model tree. The technique does not make use of similarity metric during annotation: the metrics are used only to cluster the patches into hierarchy of trees allowing fast tree search permitting online comparison among multiple models. Thus, the system maintains persistent knowledge to some extent by storing what it has learned in terms of positive and negative examples to improve its labeling ability. There is no attempt to modify the tree structure during annotation.

A more recent application of the active learning paradigm to retrieval of 3D models can be found in the work of work Zhang and Chen.[19] We have already mentioned application of active learning to image retrieval

in a relevance feedback scenario by Simon and Chang.[14] By contrast, the work presented here describes an annotation tool developed for the task of video retrieval. Unlike a relevance feedback setting, we are interested in persistent propagation of semantically meaningful labels to enable semantic retrieval. Our approach also generates a set of initial models for semantic classification as a by-product.

3. DESCRIPTION OF THE VIDEO ANNOTATION TASK

We report our experiments with active annotation using the TREC Video Corpus that resulted from a recent video retrieval benchmarking exercise more details of which can be found in.[12] The TREC video corpus[1] is divided into the training set and the testing set. The training set consists of 19 videos, and the testing set consists of 28 videos. The total of 47 sequences results in about 11 hours of MPEG video. These videos include documentaries from space explorations, US government agencies, river dams, wildlife conservation, and instructional videos. From the given content, we define a lexicon for our video description and used for labeling the training set.

For each video sequence, first shot detection is executed to divide the video into multiple shots by using the CueVideo algorithm.[2] CueVideo segments an input video sequence into smaller units, where scene cuts, dissolves, and fades are detected. The 47 videos result in a total of 5882 detected shots. Because each shot can be described and retrieved independently of each other, the next step is to define our lexicon for shot descriptions.

Our video content description methodology is motivated largely by the previous work of Naphade and Huang.[7] A video shot can fundamentally be described by three attributes. The first is the background surrounding of where the shot was captured by the camera, which is referred to as the *static scene*. The second attribute is the collection of significant subjects involved in the shot sequence, which is referred to as the (key) *object*. Lastly, the third attribute is the corresponding actions taken by some of the key objects, which is referred to as the *event*. These three types of lexicon define the vocabulary for our video content.

Our vocabulary for the static scenes included *indoors*, *outdoors*, and *outer space*. Furthermore, each category is hierarchically sub-classified to comprise more specific scene descriptions. Our simplified vocabulary for the key objects includes the following categories: *animals, human, man-made structures, man-made objects, nature objects, graphics and text, transportation*, and *astronomy*. In addition, each key object category is subdivided into more specific object descriptions, i.e., *rockets*, *fire*, *flower* and *robots*. For our events vocabulary, only eight events are of specific interest to the given retrieval project, and some of them include *water skiing*, *boat sailing*, *person speaking*, *landing*, *take-off* or *launch*, and *explosion*.

Using the defined vocabulary for static scenes, key objects, and events, the lexicon is imported into our IBM VideoAnn Annotation Tool for describing and labeling each video shot. The shots are labeled for its content to serve two main purposes. The first is to train our models. The second is to verify our retrieval results. The labeled data provide the ground truth we need. We describe the VideoAnn annotation tool next.

3.1. The Annotation Tool

The operation of our VideoAnn annotation tool hereafter referred to as VideoAnn is described as follows. Each shot in the video sequence can be annotated with static scene descriptions, key object descriptions, event descriptions, and other keywords. These descriptions are provided for each shot and are stored as MPEG-7 descriptions in the output XML file. VideoAnn can also open MPEG-7 files in order to display the annotations for the corresponding video sequence.

The required inputs to VideoAnn are a video sequence and its corresponding shot file. The shot file can be generated by the CueVideo shot detection algorithm. CueVideo segments an input video sequence into smaller units called video shots, where scene cuts, dissolves, and fades are detected.

3.2. Overview of Graphical User Interface

VideoAnn is divided into four graphical sections as illustrated in Figure 1. On the upper right-hand corner of the tool is the Video Playback window with shot information. On the upper left-hand corner of the tool is the Shot Annotation with a key frame image display. On the bottom portion of the tool consists of two different View Panels of the annotation preview. A fourth component, not shown in Figure 1, is the Region Annotation pop-up window for specifying annotated regions. These four sections provide interactivity to the use of the annotation tool.

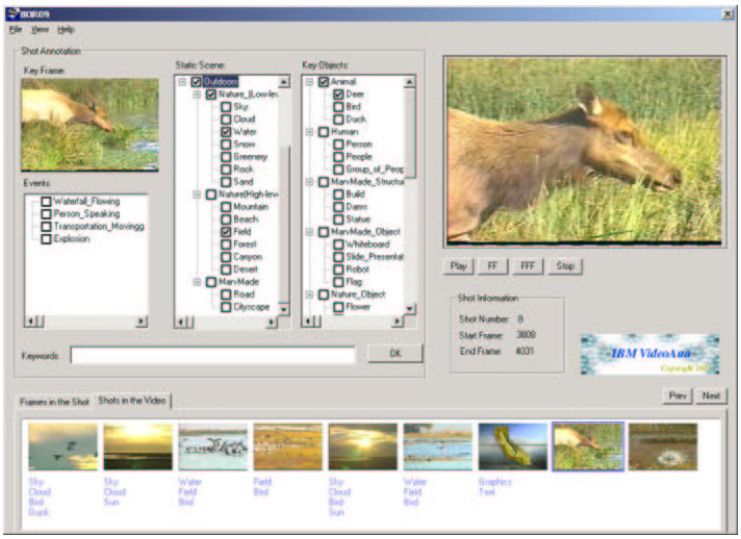

Figure 1. IBM VideoAnn Annotation Tool divided into three regions: (1) Video Playback, (2) Shot Annotation, and (3) Views Panel.

The Video Playback window on the upper right-hand corner displays the opened MPEG video sequence as show in Figure 2. The four playback buttons directly below the video display window include:

- **Play** - Play the video in normal real-time mode.
- **FF** - Play the video in fast forward mode [display I- and P-frames].
- **FFF** - Play the video in super fast forward [display only I-frames].
- **Stop** - Pause the video in the current frame.

As the video is played back in the display window, the current shot information is given as well. These shot information include the current shot number, the shot start frame, and the shot end frame. Note that the first shot starts at number 0.

The Shot Annotation module on the upper left-hand corner displays the defined annotation descriptions and the key frame window as depicted in Figure 3. As the video is displayed on the Video Playback, a key frame image of the current shot is displayed on the Key Frame window. The key frame is a representative image of the video shot segment, and thus offer an instantaneous recap of the whole video shot. Consequently, the key frame may provide the author with immediate assistance in annotating the shot descriptions. In the shot annotation module, the annotation lexicon is also displayed. There are three types of lexicon as follows:

- Events - List the action events that can be used to annotate the shots.

Figure 2: Video palyback of the IBM VideoAnn annotation tool

- Static Scene - List the background static scenes that can be used to annotate the shots.
- Key Objects - List the significant objects that are present in the shots.

In each of the three lexicons, the descriptions are organized in a hierarchical tree structure. These annotation descriptions have corresponding check boxes for the author to select. Furthermore, there is a Keywords box for customized annotations. Once the check boxes have been selected and the keywords typed, the author hits the **OK** button to advance to the next shot.

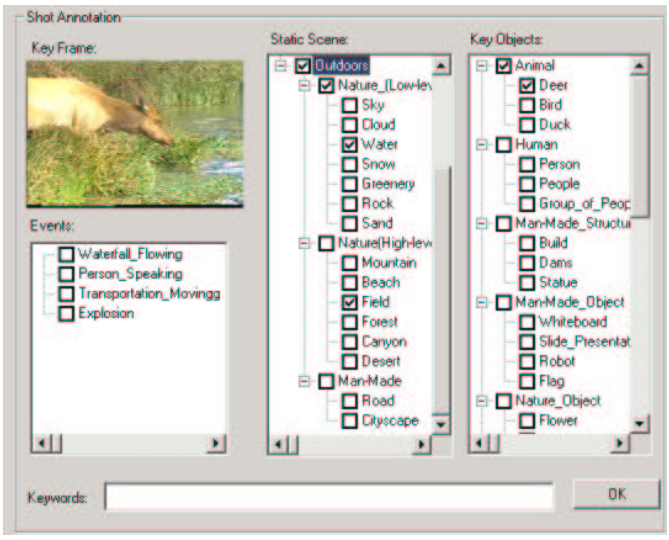

Figure 3: Shot Annotation of the IBM VideoAnn Annotation Tool.

The Views Panel on the bottom displays two different previews of representative images of the video. They are:

- Frames in the Shot - Display representative images of the current video shot.
- Shots in the Video - Display representative images of the entire video sequence.

The Frames in the Shot view shows all the I-frames as representative images of the current shot as shown in Figure 4. A maximum of 18 images can be displayed in this view. This allows the author to obtain an instantaneous temporal insight into the video shot without having to playback the video shot over time. The **Prev** and **Next** buttons refresh the view panel to reflect the previous and next shot frames in the video sequence. Also, one can double-click on any of the representative images in the panel. This action designates that selected image to be the new key frame for this shot, and is respectively displayed on the Key Frame window. In this preview mode, if the author clicks the **OK** button on the Shot Annotation Window then the video will stop playback of the current shot and advance to play the next shot.

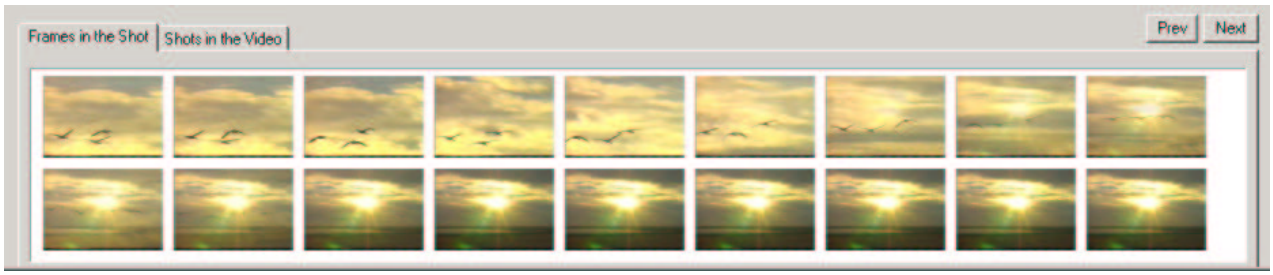

Figure 4: Frames in the Shot of the Views Panel in the IBM VideoAnn Annotation Tool.

The Shots in the Video view shows all the key frames of each shot as representative images over the entire video as illustrated in Figure 5. Below each shot's key frame is the annotated descriptions, if indeed they have already been provided. The author can peruse the entire video sequence in this view and examine the annotated and non annotated shots. The **Prev** and **Next** buttons scroll the view panel horizontally to reflect the temporal video shot ordering. Also, one can double-click on any of the representative images in the panel. This action instantiates the selection of the corresponding shot, resulting in (1) the appropriate shot being displayed on the Video Playback window, (2) the simultaneous key frame being displayed on the Key Frame window, and (3) the corresponding checked descriptions on the Shot Annotation panels. In this preview mode, if the author clicks the **OK** button on the Shot Annotation Window then the video will FFF playback of the current shot and advance to play the next shot in normal playback mode.

Figure 5: Shots in the Video of the Views Panel in the IBM VideoAnn Annotation Tool.

The Region Annotation pop-up window shown in Figure 6 allows the author to associate a rectangular region with a labeled text annotation. After the text annotations are identified on the Shot Annotation window, each description can be associated with a corresponding region on the selected key frame of that shot. When the author finishes check marking the text annotations and clicks the **OK** button, then the Region Annotation

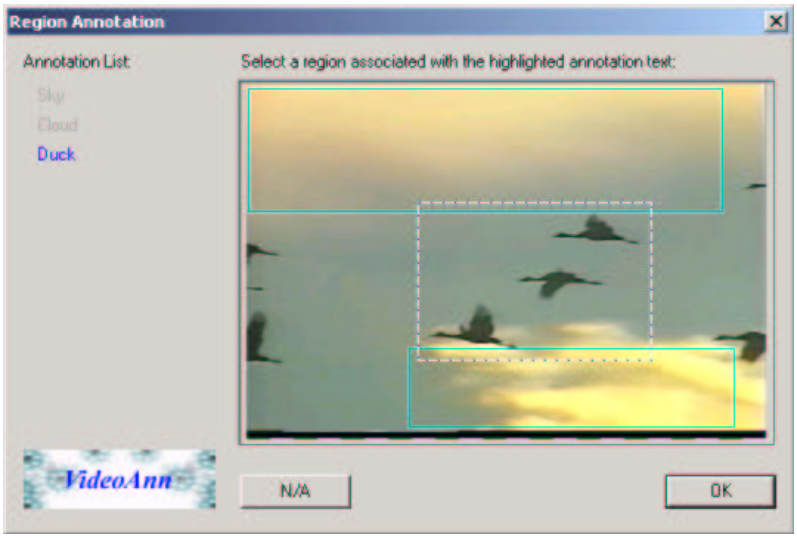

Figure 6: Region Annotation of the IBM VideoAnn Annotation Tool.

window appears. On the left side of the Region Annotation window is a column of descriptions listed under **Annotation List**. On the right side is the display of the selected key frame for this shot along with some rectangular regions. For each description on the **Annotation List**, there may be one or no corresponding region on the key frame.

The descriptions under the **Annotation List** may be presented in one of four colors:

1. Black - the corresponding description has not been region annotated.
2. Blue - the corresponding description is currently selected.
3. Gray - the corresponding description has been labeled with a rectangular region.
4. Red - the corresponding description has no applicable region. (i.e., when you click **N/A**)

The regions on the Key Frame image may be presented in one of two colors:

1. Blue - the region is associated with one of the not-current descriptions (i.e., the description in Gray color).
2. White -the region is associated with the currently selected description (i.e., the description in Blue color).

When the Region Annotation window pops up, the first description on the **Annotation List** is selected and highlighted in Blue, while the other descriptions are colored Black. The system then waits for the author to provide a region on the image where the description appears by click-and-drag a rectangular bounding box around the area of interest. Right after the region is designated for one description, the system advances to the next description on the list. If there is no applicable region on the key frame image, click the **N/A** button, and the corresponding description will appear in Red. At any time, the author can click any description on the **Annotation List** to make that selection current. Thus the description text will appear in Blue and the corresponding region, if any, will appear in White. Furthermore, this action allows the author to modify the current region of any description at any time.

4. EXPERIMENTS WITH SVM BASED ACTIVE LEARNING

In this section we report preliminary simulation experiments to demonstrate the effectiveness of our support vector machine based active learning algorithm on the video-TREC database previously mentioned. We use polynomial kernel machines * Of the many labeled examples that are available via the use of VideoAnn tool on our video-TREC database we will report results on indoor-outdoor classification only. There are $9,045$ examples in the database to be annotated. To begin with, we choose approximately 1% of the data and accept their labels as provided by human annotators †. Subsequently, the initial support vector classifier is then built on the basis of this annotated data-set and new unseen examples are presented to the classifier in steps. Each unseen example is classified by the SVM classifier and the confidence in classification is taken to be inversely proportional to the distance of the new feature from the separating hyperplane in the induced higher dimensional feature space. If this distance is less than a specified threshold then we select the new sample to be included in the training set. We adopt three different selection strategies:

1. In the first strategy, we measure the distance from the hyperplane in the higher-dimensional space. We refer to these as experiments of type-I.

2. In the second strategy, we also consider absolute distance but select points to be included in the training set only if the point is classified negatively - the rationale for this being that we wish to balance the lack of positively labeled data in the training set. We refer to these as experiments of type-II.

3. In the third strategy, we re-scale the ratio of distance of points classified negatively to points classified positively by a factor $2:1$ before making a decision whether to select a point or not. The rationale for this ratio again comes from the fact that there are approximately twice as many negatively labeled examples compared to the positively labeled examples. We refer to these as experiments of type-III.

The SVM classifier is retrained after every decision to include a new example in the training set. Note that if the example is not selected then the uncertainty associated with its classification is low and its label can be automatically propagated. Iterative updates of the classifier can proceed in this manner until a desirable performance level is reached.

The precision recall curves for retrieval performance achieved by the classifiers so trained are shown in Figure 7. The lowermost dotted curve and uppermost continuous curve show the performance of the classifier when only 10% and 90% of the labeled training data are respectively chosen (without any "active" role of the classifier) for passive supervised training. These two curves serve the purpose of comparing the effectiveness of active (semi-supervised) learning as against passive (supervised) learning. The remaining three curves refer to precision recall behavior of the classifiers trained with 10% data by adopting active learning strategies of types I, II and III. It is remarkable that with all three training strategies, active learning with only 10% data shows performance almost as good as passive training with 90% data and much better than passive training with 10% data.

The ROC curves in Figure 8 show the detection to false alarm ratio as another composite measure of retrieval performance with progress of iterations. The results are in conformity with those in Figure 7. We again observe remarkably improved detection to false alarm ratio for all three types of active learning compared to passive learning. In fact, the simplest type I training seems to perform the best - especially after the iterations have progressed substantially, indicating that there is probabaly little to be gained from complicated weighting schemes in measuring distance from the decision boundary.

5. CONCLUSION AND FUTURE DIRECTIONS

In this paper we present an active annotation paradigm based on the principle of active learning for sample selection for maximum disambiguation. The proposed system can propagate persistent semantic labels with

*The SVM-light software[6] is used for simulations.

†The warm-up set can be selected randomly. It can also be selected by unsupervised clustering[8]

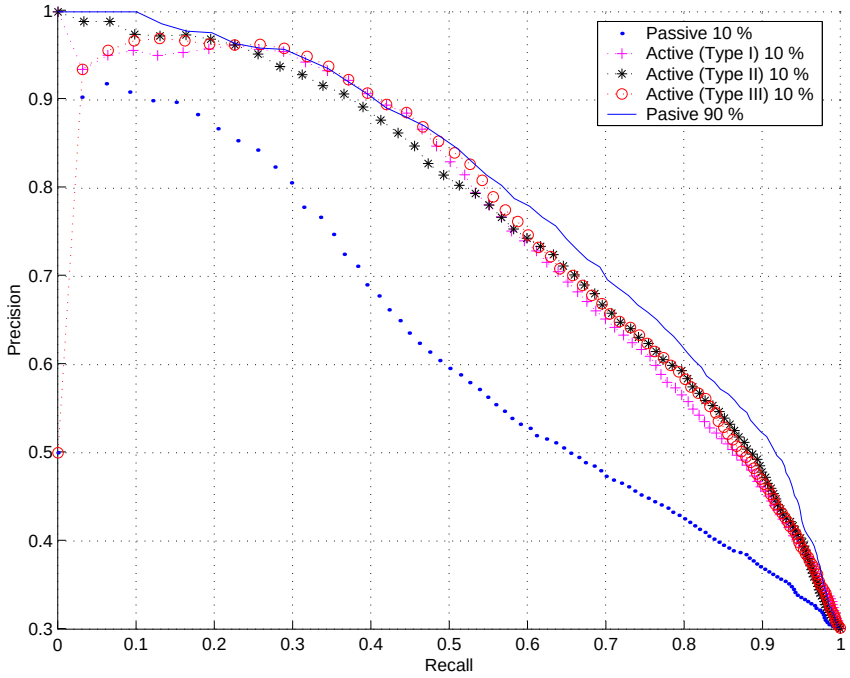

Figure 7. Comparison of precision-recall curves showing classification performance for different active learning strategies with that using passive learning when only 10% and 90% of the training data were used.

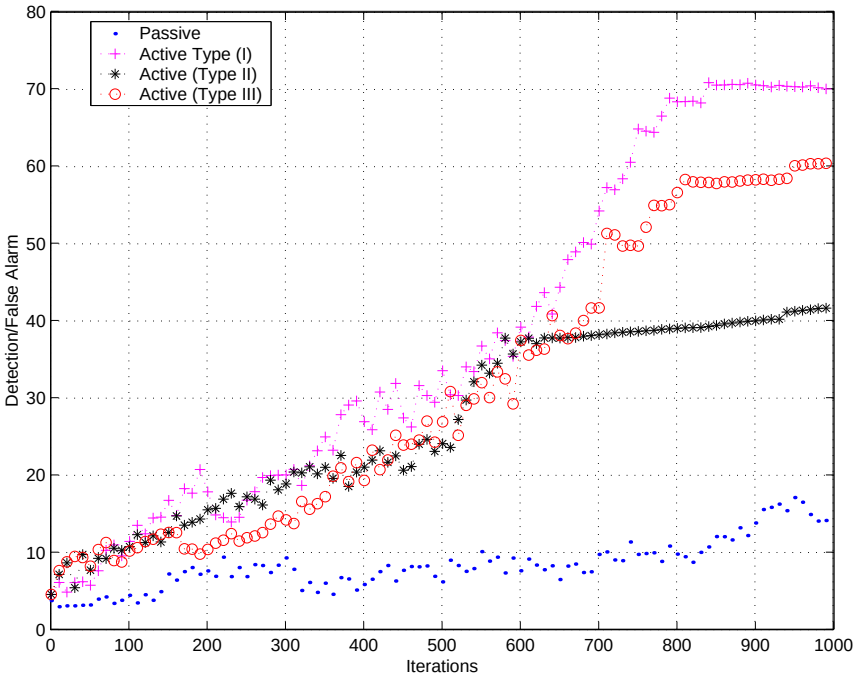

Figure 8. Comparison of detection to false alarm ratio for all three active learning strategies and passive learning with progress of iterations.

minimal user input by selecting the most "difficult" or "ambiguous" examples for the user to annotate. We use an incrementally trained support vector machine to build a model that assists the user in annotation. A desirable characteristic of such a system is extremely high precision, which our system achieves successfully. We show that the user needs to annotate very few "selected" examples and in fact achieve much better classification as well as propagation performance with the models thus trained.

There are several directions for future research. In the work reported here, we considered annotation of video sequences based on keyframes only. This is due to the reason that the retrieval was to be performed on a static basis i.e., the static image associated with the keyframe was assumed to fully represent the video shot. Alternatively, retrieval based on dynamic or time varying nature of the video signal can be sought. An example of the type of tools useful for such study are dynamic models such as the Hidden Markov model (HMM), which require labeled sequence from an event. Active learning based annotation can potentially be used here as well in order to further enhance the design of annotation tool presented here. A second direction is that of multiple instance learning.[10] Here keyframes are considered bags of objects of interest, and the frames are labeled positive and negative depending on the occurrence of specific objects in the frame. While multiple instance learning itself could potentially facilitate annotation by not requiring the human annotator to label the specific objects in the frame, an active version of multiple instance learning could add another layer to this strategy. A less ambitious or more short term goal could be extending the two-class active learning method adopted here to multiclass problems. Yet another such objective is to propagate labels in a video frame on basis of temporal proximity of some of the frames in the past as well as semantic proximity when multiple labels are being assigned to each keyframe simultaneously.

REFERENCES

1. The Video-TREC benchmarking exercise, National Institute of Science and Technology, 2001.
2. A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan, and G. Cohen, Using audio Time scale modification for video browsing, Hawaii Int. Conf. on System Sciences, HICSS-33, Maui, January 2000.
3. David A. Cohn, Zhoubin Ghahramani and Micahel I. Jordan, Active learning with statistical models, Journal of artificial intelligence research (4),1996, 129-145.
4. Nello Cristianini, John Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-Based Learning Methods, Cambridge University Press, 2000.
5. Vijay Iyengar, Chid Apte, and Tong Zhang, Active learning using adaptive aesampling, pp. 91-98, ACM SIGKDD 2000.
6. T. Joachims, Making large-scale SVM learning practical, Advances in Kernel Methods - Support Vector Learning, B. Schlkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.
7. M. R. Naphade and T. S. Huang, A probabilistic framework for semantic video indexing, filtering and retrieval, *IEEE Transactions on Multimedia, special issue on Multimedia over IP*, vol. 3, pp. 141-151, March 2001.
8. M. R. Naphade, X. Zhou, and T. S. Huang, Image classification using a set of labeled and unlabeled images, in Proceedings of SPIE Photonics East, Internet Multimedia Management Systems, pp. 13-24, 2000.
9. R. W. Picard and T. P. Minka, Vision texture for annotation, ACM/Springer Verlag Journal of Multimmedia Systems, vol 3, pp. 3-14, 1995.
10. A. Ratan, O.Maron, W. Grimson, and T. LozanoPerez, A framework for learning query concepts in image classification, Proceedings of Computer Vision and Pattern Recognition, vol. 1, pp. 423–429, 1999.
11. B. Shahshahani, D. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, IEEE Transactions on Geoscience and Remote Sensing, 32, 1087-1095, 1994.
12. John R. Smith, Savitha Srinivasan, Arnon Amir, Sankar Basu, Giri Iyengar, Ching-Yung Lin, Milind Naphade, Dulce Ponceleon and Belle Tseng, Integrating Features, Models, and Semantics for Content-based Retrieval, NIST video-TEC notebook, Novemver 2001.
13. Zhong Su, Stan Li and Hong-Jiang Zhang, Extraction of feature subspaces for content based retrieval using relevance feedback, ACM Multimedia, pp.98-101, 2001.

14. Simon Tong, Edward Chang, Support vector machine active learning for image retrieval, ACM Multimedia, pp. 107-118, 2001.
15. S. Tong and D. Koller, Support vector machine active learning with applications to text classification. Proceedings of the 17th International Conference on Machine Learning, pages 401-412, June 2000.
16. Simon Tong, Daphne Koller, Active learning for structure in bayesian Networks, Seventeenth International Joint Conference on Artificial Intelligence, pp. 863-869, 2001.
17. Simon Tong, Daphne Koller, Active learning for parameter estimation in bayesian networks, Neural Information Processing Systems, pp. 647-653, 2000.
18. V. Vapnik, Statistical Learning Theory, Wiley John & Sons, 1998.
19. Cha Zhang and Tsuhan Chen, Active learning for information retrieval:Using 3D models as an Example, Carnegie Mellon University, Advanced Multimedia Processing Laboratory, Tech. Rep. AMP 01-04, 2001.
20. Tong Zhang and Frank Oles, A probabilistic analysis of the value of unlabelled data for classification problems, Proceedings of Int. Conf. Machine Learning, pp. 1191-1198, 2000.