Putting Active Learning into Multimedia Applications: Dynamic Definition and Refinement of Concept Classifiers

Ming-yu Chen, Michael Christel, Alexander Hauptmann, and Howard Wactlar

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA 15213
1-412-268-{7003, 7799, 1448, 7458}

{mychen, christel, alex+, wactlar}@cs.cmu.edu

ABSTRACT

The authors developed an extensible system for video exploitation that puts the user in control to better accommodate novel situations and source material. Visually dense displays of thumbnail imagery in storyboard views are used for shot-based video exploration and retrieval. The user can identify a need for a class of audiovisual detection, adeptly and fluently supply training material for that class, and iteratively evaluate and improve the resulting automatic classification produced via multiple modality active learning and SVM. By iteratively reviewing the output of the classifier and updating the positive and negative training samples with less effort than typical for relevance feedback systems, the user can play an active role in directing the classification process while still needing to truth only a very small percentage of the multimedia data set. Examples are given illustrating the iterative creation of a classifier for a concept of interest to be included in subsequent investigations, and for a concept typically deemed irrelevant to be weeded out in follow-up queries. Filtering and browsing tools making use of existing and iteratively added concepts put the user further in control of the multimedia browsing and retrieval process.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *video.*

General Terms

Experimentation, Algorithms, Human Factors.

Keywords

Video retrieval, extensible concept classification, active learning.

1. INTRODUCTION

A 2004 report to the Council on Library and Information Resources opens as follows [19]:

> The rapid increase in the quantity of visual materials in digital libraries—supported by significant advances in digital imaging technologies—has not been supported by a corresponding advance in image retrieval technologies and techniques. Digital librarians sense that much could be done to improve access to visual collections and hope, perhaps vainly, that users' needs to identify relevant digital visual resources might be met more satisfactorily through search strategies based on visual characteristics rather than on textual metadata associated with the image, which are expensive to produce.

Similarly, a recent ACM strategic retreat examining the future of multimedia research identified three grand challenges, one of which is to "make capturing, storing, finding, and using digital media an everyday occurrence in our computing environment" [16]. The retreat report notes that with the widespread adoption of digital cameras and emergence of cell phones with built-in video cameras, coupled with increases in storage capacity and reductions in cost, we can now store massive amounts of image and video data, with the challenge being to make that data useful. The ACM report noted that better context and content descriptions could be used more thoroughly in multimedia interfaces.

The video analysis community has long struggled to bridge the gap from successful, low-level feature analysis (color histograms, texture, shape) to semantic content description of video. One plausible solution is to utilize a set of intermediate (textual) descriptors that can be reliably applied to visual scenes. Many researchers have been developing automatic concept classifiers like face, people, sky, grass, plane, outdoors, soccer goals, and buildings [14], showing that perhaps these classifiers will reach the level of maturity needed for their use as effective filters for video retrieval. It is an ongoing research issue as to how to best represent the high level semantics of a video shot, given current techniques for automatic lower-level feature extraction [10, 14], but we believe that extensibility will play a leading role in video retrieval systems of the future. It is too difficult to anticipate the set of concepts useful for a user addressing a particular need with a specific corpus. Instead, the user should be able to create and refine the set of classified concepts interactively and without much effort so that necessary concepts are available as filtering and browsing tools.

Shahraray notes that "well-designed human-machine interfaces that combine the intelligence of humans with the speed and power of computers will play a major role in creating a practical compromise between fully manual and completely automatic

MM'05, November 6–12, 2005, Singapore.

multimedia information retrieval systems" [4]. We describe a system in which the user plays a driving role in the creation and refinement of models for visual concepts applicable to video information access, rather than serving as only a consumer of pre-built automated concept classifiers.

Users have long been offered a more active role in information retrieval through relevance feedback techniques, where by interactively marking the correct (and, sometimes, the incorrect) items returned by a query a follow-up query can be made more precise. Limitations with relevance feedback techniques, however, include the user's unwillingness to invest time to label data and concern for introducing extra cognitive load to the user's primary tasks. The extensible video retrieval system described here simplifies the labeling task by folding it into the storyboard browsing activity and by carefully monitoring user activity to derive additional labeled data based on what the user passed over. It greatly reduces the need for labeled data by taking advantage of active learning, presented in Section 2. Section 3 presents the application focusing on its extensibility, with Sections 4 and 5 discussing multimodal learning and evaluation.

2. ACTIVE LEARNING

As outlined in [3], relevance feedback can be used as a query refinement scheme to derive or learn a user's query concept. To solicit feedback, the refinement scheme displays a few video shot instances and the user labels each shot as "relevant" or "not relevant." Based on the responses, another set of shots from the database is presented to the user for labeling. After a few such querying rounds, the refinement scheme returns a number of instances from the database that seem to fit the needs of the user. The construction of such a query refinement scheme can be regarded as a machine learning task. In particular, it can be seen as a case of pool-based active learning [12]. In pool-based active learning the query refinement scheme, i.e., the *learner,* has access to a pool of unlabeled data and can request the user's label for a certain number of instances in the pool. In the video retrieval domain with shots as the unit of information retrieval, the unlabeled pool would be the entire database of video. An instance would be a video shot, and the two possible labelings for each shot would be "relevant" or "not relevant". The goal for the active learner system is to learn the user's query concept.

Continuing the summary of [3], the main issue with active learning is finding a method for choosing informative shots within the pool to ask the user to label. The request for the labels of a set of shots can be termed a pool-query. Most machine learning algorithms are passive in the sense that they are generally applied using a randomly selected training set. The key idea with active learning is that it should choose its next pool-query based upon the past answers to previous pool-queries. In general, and for the video retrieval task in particular, such a learner must meet two critical design goals. First, the learner must learn target concepts accurately. Second, the learner must grasp a concept quickly, with only a small number of labeled instances, since most users are too impatient or preoccupied with more critical tasks to provide a great deal of feedback.

Active learning has demonstrated its effectiveness in reducing the cost of labeling data. Given an unlabeled pool *U*, an active learner *l* has three components (*f*, *q*, *x*). The first component is a classifier, *f*(x)→ (-1,1), trained on the current labeled data *x*. The second component *q*(x) is the querying function that, given a labeled set *x*, decides which instance in *U* to query next. The active learner can return a classifier *f* after each iteration or after some fixed number iterations. Figure 1 illustrates the framework of active learning. Given labeled data *x* (upper left pile), the classifier *f* trains a model based on *x*. The querying function *q* selects the informative data from unlabeled pool (the rectangle). Users annotate the selected data and feed them into the labeled data set.

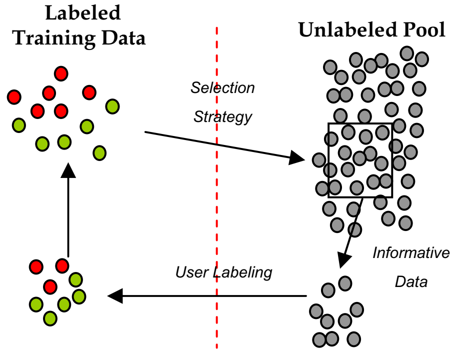


Figure 1. Illustration of active learning.

The main difference between an active learner and a regular passive learner is the querying component *q*. This brings us to the issue of how to choose the next unlabeled instance in the pool to query, and what is informative data. This issue also relates to which classifier you will use. In our framework, we employ Support Vector Machine (SVM) [2] as our classifier algorithm.

2.1 Support Vector Machine (SVM)

The basic idea of SVM is to separate samples with a hyperplane that has a maximal margin between two classes. To formulate the problem of classifying synthesized feature vectors, the training data are represented as $\{x_i, y_i\}$, i = 1,2, … , n, y_i is either -1 (negative examples) or 1 (positive examples), n is the number of training samples. Suppose all training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1 \quad \text{when } y_i = 1 \qquad (1)$$
$$x_i \cdot w + b \leq -1 \quad \text{when } y_i = -1$$

The distance between the hyperplane "$x_i \cdot w + b \geq +1$" and the hyperplane "$x_i \cdot w + b \leq -1$" is $2/||w||$, where $||w||$ is the Euclidean norm of *w*. Therefore, by minimizing $||w||^2$ we get the two hyperplanes with maximal margins. Quadratic programming provides well-studied optimizations to maximize the quadratic functions subject to the linear constraints in equation 1, which guarantees finding the global maximum.

More generally, SVM can project the original training data in space ***X*** to a higher dimensional feature space ***F*** via a Mercer kernel operator ***K***.

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) \qquad (2)$$

When ***K*** satisfies Mercer's condition [2] we can write: ***K***(u,v) = Φ(u)·Φ(u) where $\Phi : \boldsymbol{X} \rightarrow \boldsymbol{F}$ and "·" denotes an inner product. We can then rewrite ***f*** as:

$$f(x) = w \cdot \Phi(x), \text{ where } w = \sum_{i=1}^{n} \alpha_i \Phi(x_i) \qquad (3)$$

With the ***K*** function, we are implicitly projecting the training examples into a different feature space ***F*** and employ the same optimization problem as Equation (1) to maximize the margin of hyperplane in ***F***. By choosing different kernel functions we can project the training data to different spaces to make more complex decision boundaries than in the original space. A commonly used kernel is the radial basis function (RBF) kernel ***K***(u,v) = (e-r(u-v)*(u-v)) which induces boundaries by placing weighted Gaussians [2]. Our base classifier algorithm is this RBF SVM.

2.2 SVM Active Learning Algorithm

In active learning, we want to choose the most informative data to annotate. Following the procedure of [18], we learn a SVM on the existing labeled data and choose as the next examples those which come closest to the hyperplane in ***F***. This scheme for choosing new examples will reduce the corresponding version space of the SVM, i.e., the "most informative" data for the next round of annotation are those examples closest to the hyperplane in ***F***.

We can also explain this scheme more explicitly. We choose the examples between or close to hyperplanes which will change SVM hyperplanes, and are more complicated for the current model to explain. Therefore, those examples will change the current existing hyperplanes and force the model to deal with those difficult examples.

To summarize, the SVM active learning algorithm performs the following steps for each round of user-directed feedback:

1. Randomly select examples from unlabeled pool to annotate as initial set, or let user decide on an initial set to label.
2. Train a RBF SVM based on the labeled set.

3a. Select the examples which are closest to the hyperplane to be annotated. Based on [18], if the user's goal will be to generate the most accurate model, these examples are the set to annotate next.

3b. Alternatively, return the best-ranked data to the user, in cases when the user is motivated to achieve improved precision at the top N documents. By annotating these assumed "best" items, the precision at N can be more quickly improved.

4. Add annotated examples into labeled set.
5. Repeat step 2 to step 4.

3. ENVIE: EXTENSIBLE NEWS VIDEO INFORMATION EXTRACTION

Regardless of which concept classifiers are provided as part of a baseline video retrieval system, the user is likely to have information requirements that are not addressed, a data set to which classifiers need to be tailored and trained for acceptable accuracy, and/or security concerns that prohibit the user from broadly communicating a given need. We believe the best way for an application to support video exploration and retrieval is by making extensibility a priority, which led to the development of ENVIE targeting the broadcast news genre, where ENVIE is an acronym for Extensible News Video Information Extraction. The user can extend ENVIE by updating existing classifiers through positive and negative examples, by developing new classifiers that take advantage of those concepts already classified within the system, and by creating summary and video skim templates appropriate to his or her needs. Our focus here is on the dynamic definition and refinement of concept classifiers through active learning, with the architecture for this process shown in Figure 2.

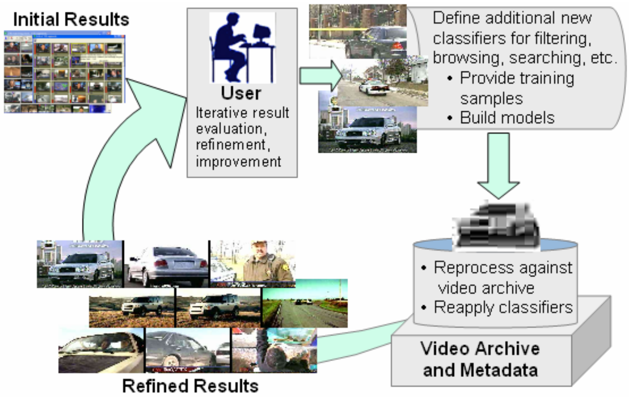

Initial Results
User
Iterative result evaluation, refinement, improvement
Define additional new classifiers for filtering, browsing, searching, etc.
• Provide training samples
• Build models
• Reprocess against video archive
• Reapply classifiers
Video Archive and Metadata
Refined Results

Figure 2. ENVIE architecture: User iteratively builds and refines concept classifiers for filtering and browsing video.

Since a user may frequently encounter new training examples to update or improve a particular concept classifier, an approach is needed that provides for quick incorporation of new data. Some stochastic learning algorithms allow the classifier to focus on learning new examples, instead of building a complete classification model from scratch each time more examples are added in. Most studies of stochastic learning algorithms have focused on "on-line learning" [7]. In each iteration, the algorithm is fed with one more training example and the model is updated accordingly. However, the on-line learning algorithm cannot revisit previous training examples. Compared to standard machine learning algorithms, on-line learning algorithms cannot take full advantage of all the existing data because training data cannot be revisited. Standard machine learning algorithms likewise also fail to support dynamic extensibility, requiring too much training data or performing expensive re-evaluations of all the training data each time the training data is modified. Active learning offers the advantages of achieving high accuracy while significantly reducing the need for labeled training instances [3, 8, 13, 15, 18, 20].

Utilizing support vector machines and active learning, users can develop and refine their own concept classifiers, based on model-building details discussed in Section 4. The users need not know anything of these details, instead reviewing results following the generation of a new model. Based on the actions then taken by the user, the model itself can be tuned to better meet his or her information requirements.

Consider a user who needs high precision for a concept. For example, the user may need to find examples of "aircraft" to serve as launching points for further inquiry without regard to whether all of the shots satisfying the "aircraft" concept are retrieved. In this case, the user can provide feedback on the top-ranked shots in order to revise the classifier for the concept so that it delivers higher precision at the top-N ranked items.

Instead, the user may want a better model of the concept to apply in multiple settings, e.g., to filter out the unwanted shots having that concept following queries for varying topics, or to isolate shots with that concept in other query result sets. In that case, the user can provide feedback on shots close to the SVM decision boundary, shown to be the most beneficial in iteratively improving the model through active learning [18].

3.1 ENVIE Concept Building Procedure

Video is decomposed into shots with shots each represented by a keyframe image, as is typical in news video retrieval systems today. The user can browse thumbnail representations of keyframes for shots in many different arrangements, including map and timeline layouts, named entity graphs, and storyboards, with "storyboards" the focus here as they provide an ordered set of shots to the user. Storyboards are also a commonly encountered interface widget for video retrieval systems [11]. Interactive search experiments conducted with this same ENVIE set of storyboards for TRECVID documentary and news retrieval confirm that both novices and experts can utilize the storyboards efficiently and effectively for video browsing and selection [5].

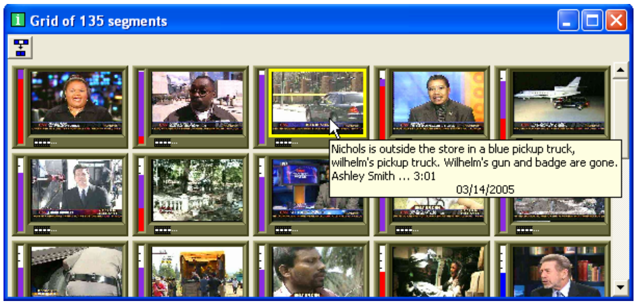
A key idea of ENVIE is to reduce the amount of labeling necessary by the user in building a concept classifier, and so when the user marks shots in the storyboard for use as a positive sample set, the shots that they skipped over, up to the last shot considered, are automatically collected into an "implied" negative sample set. Likewise, if the user marks shots in storyboards for use as a negative sample set, the skipped over shots are automatically collected into an "implied" positive sample set. The user can review and clean up either the positive or negative sample set if they so wish. More typically, based on early trials, the user launches the concept classification process and evaluates a set returned by that process to iteratively improve the model's overall accuracy or top-ranked precision.

The user initializes a classifier by identifying shots that are positive and negative examples for a new concept to be tagged. While we anticipate employing different learning strategies to improve the classification, with the user positioned to evaluate outputs and determine which classifier, if any, should be preserved within ENVIE, for this paper we focus on the use of RBF SVM and active learning. Concepts that are approved by the user for broader applicability and preservation in the corpus could be employed as input features for building follow-up concept classifiers. The goal is that with an increasing number of concepts, higher order semantics can be derived with a confidence measure based on the confidence of the contributing classes. For example, if ENVIE is already armed with detectors for people, people sitting, and indoors, then a "meeting" classifier might be developed where "meeting" might be inferred most strongly by more than 2 people sitting indoors.

This work for now deals with global visual frame classification only, i.e.., identifying that a concept is represented somewhere in the keyframe characterizing a video shot, rather than identifying the precise time and region occupied by a concept in that shot. Several approaches have been proposed to detect specific objects; a broad review of this research is given in [6]. However, the number and type of objects that can be detected by template or model based methods is limited. In order to work on a large set of classes, combinations of several approaches are needed. Some scenes can be identified by using features extracted from the entire image (e.g., outdoor scenes have certain color and texture distributions, but no specific shapes or objects). Some objects can be detected by region-based methods (e.g., an airplane can be defined as a gray region in the middle of a blue region that corresponds to sky), whereas faces can be classified with a model-based approach based on specific feature points [17]. Temporal features such as camera and object motion direction and rate of motion can also be specified for inclusion into classifiers. ENVIE supports a user-driven interactive process for classifier creation, allows the user more control over which features are utilized and their relative contributions by providing a fluid, effortless means for defining positive and negative example sets for active learning. The two example cases presented in the next sections illustrate this process, working with a three month test corpus of American, Arabic, and Chinese news broadcasts.

3.2 Defining a Vehicle Classifier, Revised for Browsing

Consider a user interested in identifying shots with vehicles. The user issues a text search "car truck automobile" that for the news test corpus returns 135 segments, with 270 shots at or near the aural mention of "car", "truck", "automobile", or derivatives, or at or near the showing of such words in overlaid text on the broadcast. A thumbnail-based view of the data termed a *segment grid*, presents the thumbnail for the highest rated shot by the text query service, one thumbnail per news story segment, as shown in part in Figure 3.



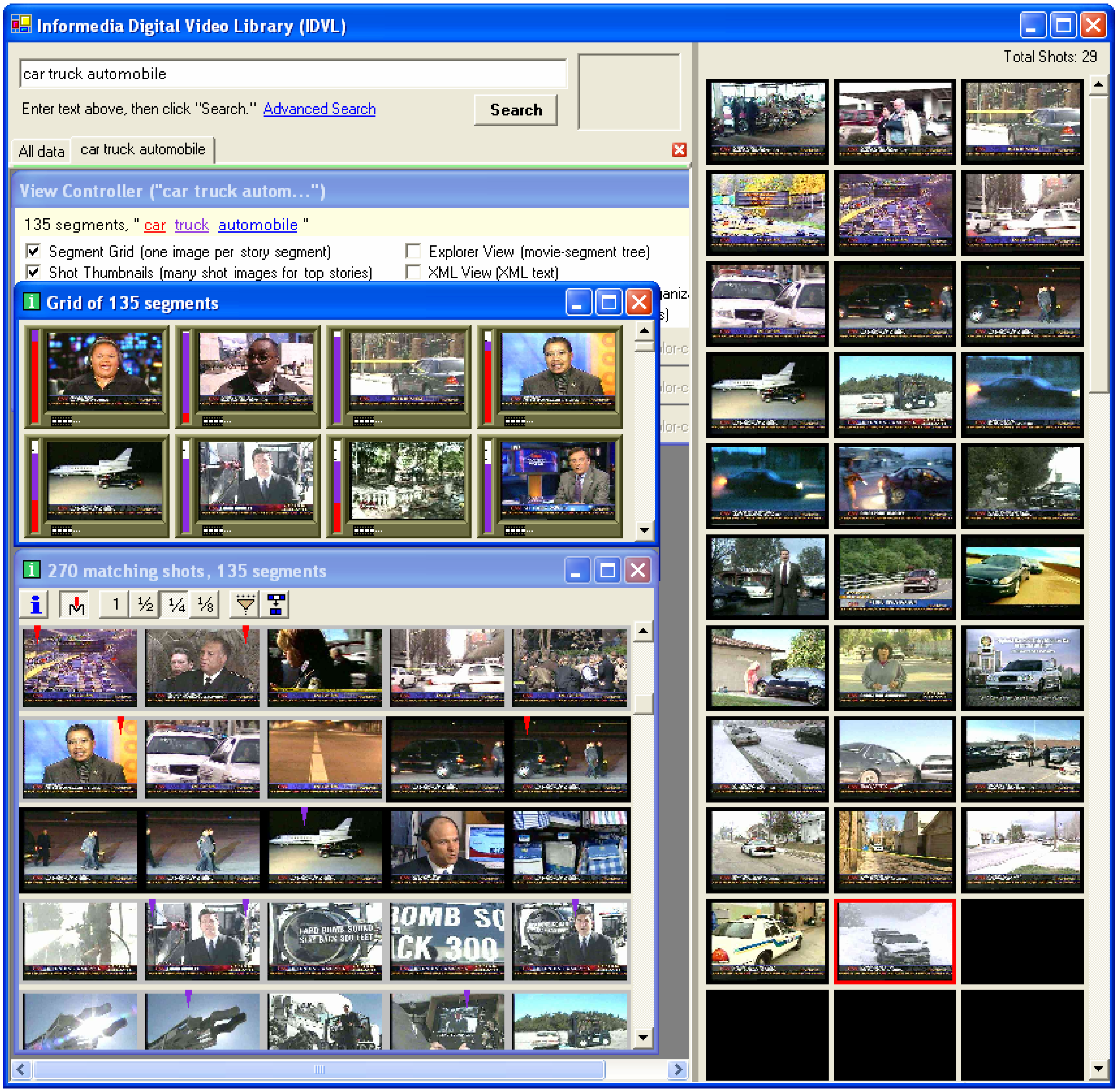
Nichols is outside the store in a blue pickup truck,
wilhelm's pickup truck. Wilhelm's gun and badge are gone.
Ashley Smith ... 3:01
03/14/2005

Figure 3. Segment grid with each news story segment represented by one thumbnail image, ordered by segment relevance to the query "car truck automobile."

Another traditional storyboard view shows one thumbnail image for each of the 270 match shots, as shown in part in the lower left of Figure 4. The user, interested in building a vehicle detector, moves the mouse over the thumbnails in the *storyboard* and selects those that are positive examples of "vehicle" such as the white car images in the top 2 storyboard rows shown in Figure 4. Selection is accomplished via a keyboard shortcut while the mouse hovers over the thumbnail being judged (fastest operation), by right-clicking the mouse and selecting from a context-sensitive menu, or by dragging the thumbnail into the shot collector area. The user can also review the *segment grid* and mark items that are "vehicle" such as the middle image on the top row. In this manner, the user can make use of multiple views (segment grid for query, storyboard for query, perhaps storyboard for a particular daily news broadcast, segment grids and storyboards for other queries, etc.), as source material for assembling a positive sample set for the concept "vehicle."

The shots that the user skips over, e.g., the first and second images in the segment grid, become the "implied" negative training set for the concept "vehicle." Considering the 25 thumbnails shown in the storyboard view of Figure 4 (lower left) as shots 1, 2, …, 25, the user selects shots 1, 4, 7, 9, 10, 13, and 25 as positive examples, which causes shots 2, 3, 5, 6, 8, 11, 12, and 14-24 to be labeled as implicit negative examples. While it is true that the user might make mistakes and skip over something that actually is the concept, or that the skipped over shot should be considered more of a "can't tell" ambiguous shot than a shot which is part of the negative training set, the advantages in speed for quickly defining positive and negative training sets without unduly burdening the user with detail have outweighed these disadvantages. Furthermore, by employing active learning the user is encouraged to correct for any such error, not by revisiting and correcting the positive and negative sample sets from "round 1" of the concept build, but by evaluating and responding to the round 1 concept classifier output in order to generate an improved round 2 (and follow-up) classifier.

The user collects positive examples in this manner, where ENVIE informs the user in the status bar if a shot being judged as positive is already a member of the positive sample set. Figure 5 shows a snapshot of the process when 29 examples were identified, shown in the shot collector area docked to the right of the application window, and also showing views of the segment grid (Figure 3) and storyboard. When finished after a few minutes of reviewing thumbnails, the positive example set holds 42 shots, and the implied negative example set holds 228 shots, with some shots at the tail of the segment grid and storyboard not judged explicitly as relevant nor judged implicitly as irrelevant.


Informedia Digital Video Library (IDVL)
Search
All data
View Controller ("car truck autom...")
Grid of 135 segments
270 matching shots, 135 segments
1
½
¼
⅛
Total Shots: 29

car truck automobile

Enter text above, then click "Search." Advanced Search

135 segments, " car truck automobile "

☑ Segment Grid (one image per story segment)
☐ Explorer View (movie-segment tree)
☑ Shot Thumbnails (many shot images for top stories)
☐ XML View (XML text)

Figure 4. ENVIE screen shot during collection of shots defining "vehicle" (right pane).

The user launches a dialog to build a new concept classification for a concept she names "vehicle." The vehicle classifier is built asynchronously, with the goal of quick performance supporting interactive review and iteration. Within a minute the classifier returns the availability of the new concept for review, and the user thinks about how she wishes to employ this classifier. She wants to make use of it to browse vehicle shots in the corpus at large, and to perhaps very restrictively filter down queries to a few shots with high likelihood of being vehicles. The user is hence interested in high precision, and so reviews the top-ranked 200 vehicle shots in a storyboard view, shown in Figure 5. Of this set, 47 are actually vehicle shots.

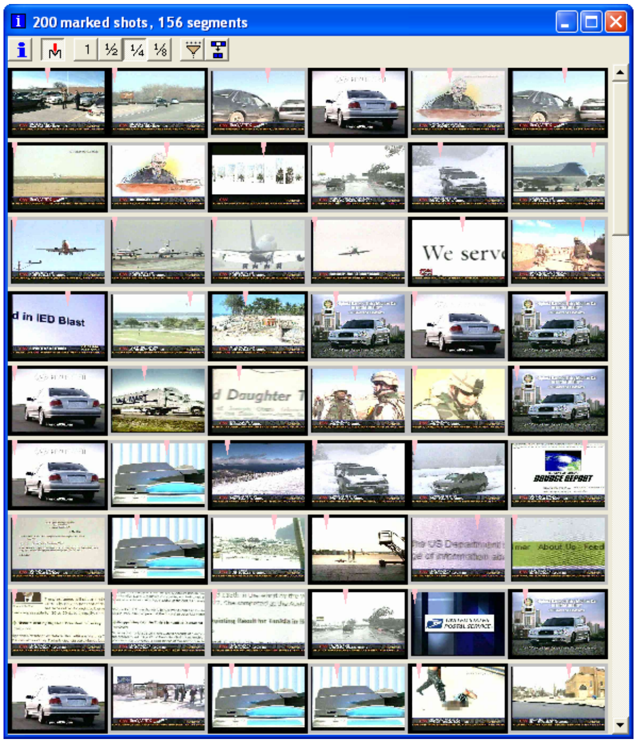


Figure 5. Best "vehicle" shots, version 1 of vehicle classifier.

The user wants better precision, and after clearing out the shot collector with a simple "Clear…" menu operation, decides to start collecting negative examples by marking shots from the "best vehicles" storyboard of Figure 5 that are in fact not vehicle shots. She stops after adding 100 shots to the explicit negative example set, with an implicit positive example set being generated based on what the user skipped over in this iteration. Now when the user selects to rebuild the vehicle classifier, the previous positive and negative examples from prior rounds are combined with the new example sets from the latest round as follows:

- The newest round's explicitly marked set is taken as highest confidence truth and overrides all prior choices.
- The newest round's implicitly marked set adds to prior choices, but if any conflict arises, the prior judgment is kept, as the new round is only "implicit" and hence lacks the authority to challenge and change prior judgments.

So, for the case of the round 2 vehicle classification, the 100 negative example shots are definitely part of the new negative example set for round 2. The implicit positive example shots, if formerly judged as negative in the prior set of 228 negative shots, would be kept as negative; otherwise they are added to the positive example set. The resulting positive example set for round 2 classification holds 89 shots, with the negative example set holding 328 shots.

When the round 2 classification is done in a minute or so, the user checks the "Best Vehicle Version 2" shot set and decides that performance is good enough for use elsewhere. An inspection of the best 200 shots, partially shown in Figure 6, finds 116 of 200 correct, nearly three times better than the initial version.

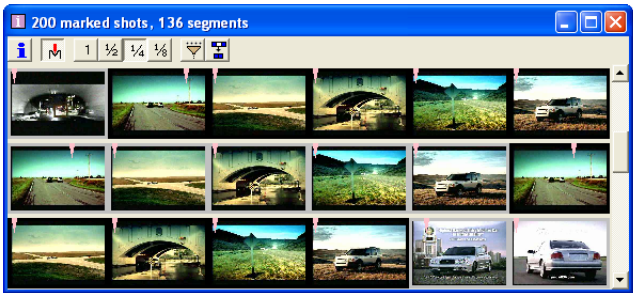


Figure 6. Best "vehicle" shots, version 2 of vehicle classifier.

The user in this session, or perhaps in a later session, decides to investigate "Baghdad Iraq" which returns over 500 shots. Wanting to zero down to just the vehicle shots in this set, the user opens up a filter tool that provides dynamic query-based sliders [1] for use in restricting the storyboard to only show shots meeting the given filter. Version 2 of the vehicle detector is available for use, and by restricting the display to just the shots considered as vehicles the display of Figure 7 is produced, showing 17 shots filtered from the set of 524, of which 9 actually contain a vehicle. The precision with the quickly built classifier is high enough to enable investigations to be launched with some target shots satisfying the need, i.e., vehicles from "Iraq" query, even if recall is not optimized because of the manner in which the concept model was built. For the second example in the next section, recall is optimized by iterating on shots located near the decision boundary, rather than evaluating the top-ranked set.

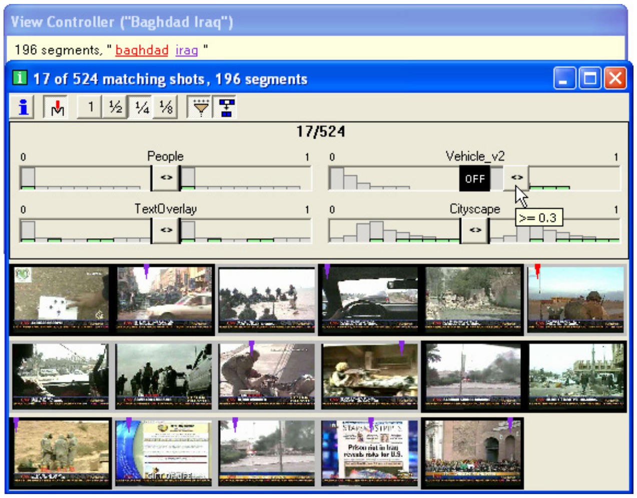

View Controller ("Baghdad Iraq")
196 segments, " baghdad iraq "
17 of 524 matching shots, 196 segments
1 ½ ¼ ⅛
17/524
0
People
1
0
Vehicle_v2
1
OFF
0
TextOverlay
1
0
Cityscape
1
>= 0.3

Figure 7. Filtering capability using newly built classifier to limit 524 shots down to set of 17 with the vehicle concept.

3.3 Defining a Taiwanese News Anchor Classifier, Revised for Filtering

Consider a user working through a multilingual news corpus and discovering that the provided anchorperson detector is not classifying anchorperson shots for a particular Taiwanese broadcaster well. That broadcaster makes use of numerous digital effects and keys in field footage as a backdrop for the anchor shots, rather than keeping the backdrop a consistent image as is typical for other broadcasters. The user decides he wants to filter out as many Taiwanese anchor shots as possible from future queries, and so begins the process of defining a Taiwanese anchorperson detector (T-Anchor) for subsequent use.

As with Section 3.2, the user starts by browsing a storyboard, in this case the storyboard for the full Taiwanese broadcast of January 7 as shown in part in Figure 8. Counting these shots as 1, 2, …, 30, the user marks shots 5-11 and 29-30 as T-Anchor shots, causing the remainder (1-4, 12-28) to be implicitly labeled as the negative example set. The user quickly repeats the process for the first shots of a January 11 broadcast, producing a positive example set of 18 shots and implicit negative example set of 114 shots.

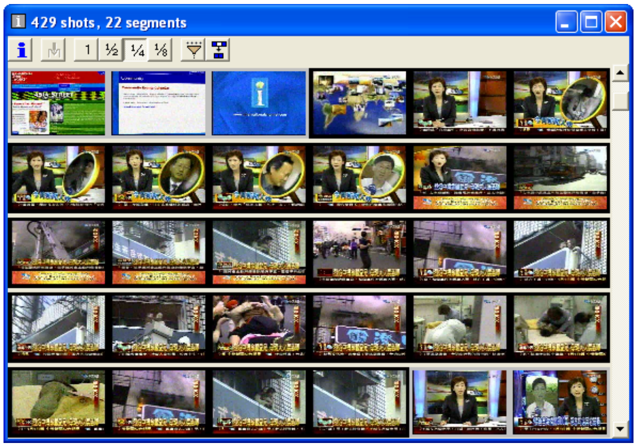


Figure 8. Taiwanese news broadcast storyboard.

The user through a dialog box initiates the building of a model from the collected shots, naming it "T-Anchor." Based on experience with TRECVID concept classification, we note that some concepts apply exclusively or primarily to one broadcaster, and so we instrumented the ENVIE classifier building dialog to let the user limit the applicability of the classifier being generated. Such is the case here: the T-Anchor classifier should only consider the Taiwanese news broadcasts, not the CNN news broadcasts or other broadcasters in the test corpus. The user indicates so, and the result is an asynchronously built model via a spawned process against the Taiwanese news. The user is notified when the classification has completed a minute or so later.

The user inspects the results by opening up a Taiwan news broadcast from February 1 (different from the test set) and filtering the storyboard for that day's half-hour show (393 shots) into just the T-Anchor shots. He sees that the filter shows 26 anchor shots in a set of 34, but notes that more should have been found. In actuality, there are 40 anchor shots in this set of 393, so the round 1 performance tested on this one broadcast is precision 0.76, recall 0.65. He issues a geographic query for the Hong Kong area, returning 288 matching shots from the multilingual corpus, and sees that filtering out T-Anchor shots drops out 17 shots. Again, he expected a bit better (in actuality, there are 33 T-Anchor shots in this set of 288, so round 1 performance is precision 1, recall 0.52).

Wanting to build a better model for T-Anchor, the user takes ENVIE's active learning suggestion and inspects a set of 500 shots located close to the decision boundary, i.e., the set of shots corresponding to step (3a) from the algorithm in Section 2.2. He browses the storyboard within a few minutes and marks 38 shots to be in the positive example set. He initiates a dialog to create a version 2 of the T-Anchor model, which causes a merge of the prior positive and negative example sets with the latest ones as discusses in Section 3.2. In this case, the new positive example set contains 56 shots (18 before plus 38 new ones now), and the new negative example set contains 570 shots (114 before, plus 456 new ones now). Note that the user did not mark 456 shots explicitly; instead, the system recognized that these shots were passed over in the storyboard when gathering the positive example set and so they were implicitly marked as negative training examples.

The updated T-Anchor model earns the approval of the user. Inspecting the February 1 storyboard, he sees that the T-Anchor filter returns 39 of 54 at a relaxed setting (precision 0.72, recall 0.98), 37 of 43 shots when restricting to higher confidence for T-Anchor (precision 0.86, recall 0.93). The model generated confidences for each shot in the range [0, 1], with 1 indicating complete confidence that a shot possesses a concept. Similarly, for the Hong Kong set of 288 images, filtering to just T-Anchor shots produces the set of 32 shown in Figure 9 (precision 1, recall 0.97). The user intends to use the concept to filter out T-Anchor shots, e.g., direct inspection to the Hong Kong matching shots other than T-Anchor ones shown in Figure 9, which is trivial to accomplish by reversing the interactive filter.

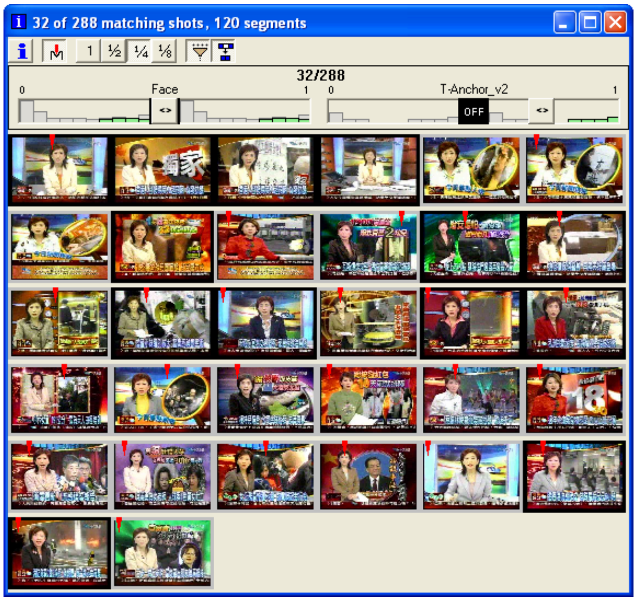


Figure 9. Taiwanese anchorperson shots after one iteration of active learning, demonstrating significant performance boost.

These examples of the T-Anchor and Vehicle concept classifiers serve to illustrate the interactive, extensible concept building environment available with ENVIE, and the use of concepts for browsing and filtering. While anecdotal, the evidence is convincing that iterative shot labeling improves learner performance, in agreement with prior literature on the topic. The benefits of ENVIE include streamlining the labeling process for positive and negative examples, quick model building and notification back to the user when the model is ready for use, and dynamic query sliders allowing fine user control over concepts for filtering and browsing. The remaining sections of the paper discuss the underlying model building and evaluation more thoroughly.

4. MULTIMODAL ACTIVE LEARNING

For any multimedia source, there are many different variants of features (various texture computations, alternate color spaces, different audio feature types, etc.) to represent its content. Assume we have r different feature sets, our training data x_i is composed of $\{x_{ij}\}, j=1, 2, \ldots, r$. Most of the time, the easiest way to deal with this kind of data, is to concatenate it as a larger feature vector x_i and employ a machine learning algorithm, such as SVM. This creates two main problems, first and foremost, the curse of dimensionality [9]. One ends up needing much more labeled data for the learning algorithm due to the increase in dimensionality of the feature vector. Second, it becomes more difficult for a human to understand and analyze the relative importance and the performance corresponding to a particular feature set. Furthermore, we effectively eliminate the variations of individual feature sets and only maintain one, undifferentiated global model to explain all the data. From our TRECVID experiments, concatenating feature vectors always perform worse in evaluation than intelligently selected feature sets.

Therefore, multi-modality fusion can lead us to a better approach than the concatenation method. Assume we have r different feature sets; we can construct r individual sub-models for each feature set. Each model represents its own information according to the feature space.

We fuse the sub-models by linear combination via a held-out set to obtain a global model for the multimodality data. This approach is motivated by an attempt to keep the locality of different feature spaces but still have a global model to represent the classification concept. The α in Equation (4) is the weight parameter for each sub-model.

$$f(x) = \sum_{j=1}^{r} \alpha_j g_j(x_j) \qquad (4)$$

The fusion approach requires a held-out set to learn the combinational parameter. Usually, a split of training data is required and this reduces the number of examples we can use in training the classifier. However, with the active learning algorithm, we will choose some informative data from the unlabeled data pool iteratively, and this data has not already been used in training process. This provides us with the held-out set we need for multimodality fusion.

Multimodality active learning works as follows:

1. Randomly select examples from unlabeled data pool. This is the initial training set for active learning.
2. Build r individual sub-models for the training set according to the different feature sets and apply their learned models to the unlabeled data.
3. For each sub-model, choose k examples which are closest to its hyperplane. In total, $k*r$ unlabeled examples will be chosen for annotation in each iteration.
4. Annotate these examples. The multi-modality fusion weights are then trained using these new annotated examples. A global model can then be constructed and evaluated.
5. Add the newly annotated examples into training set.
6. Repeat step 2 to step 4.

The multimodal active learning algorithm can be formulated as follows:

```
Unlabeled data D = $\{x_i\}$ $i = 1, 2, \ldots, n$
$D_0 = \{x_i\}$ which randomly chooses from $D$
$D_{0m} = D_0$
for $j = 1$ to $t$
{
    for $m = 1$ to $r$
    {
        $g_{jm}$ is the model constructed from $D_{j-1m}$
        $d_{jm} = \{x_{jm}\}$ the set of examples closest to hyperplane of $g_{jm}$
        $D_{jm} = D_{j-1m} \cup d_{jm}$

    }
    $F_j(x) = \sum_{m=1}^{r} \alpha_m g_{jm}$ combination parameters trained by $d_{jm}$
}
```

Some interesting issues are raised by this approach. The main idea we want to achieve is to train and select each feature set individually. Therefore, we split the training data for each feature set; let's call it D_{jm}, which is the training data of feature m in j iteration. After each iteration, we select new examples for each feature to annotate and obtain d_{jm} of them. This means, that for m different features, we select m sets of data according to each feature and build sub-models. The reason we keep every feature set separately is to maintain the specificity of that feature. We want to train locally for each feature set instead of a global model.

Through experiments, we found the problem of active learning is that it makes a strong assumption about the correctness of the previous model and the selected data is to improve the boundary. However, this assumption leads the whole model to a more and more restricted area in the feature space with each iteration. Our hope is that with separate sub-models for each feature set, we can expand the selected data from different feature spaces and avoid this problem.

5. EXPERIMENTAL EVALUATION

In this section, we describe experiments on semantic concept extraction using the development set of the TRECVID 2004 feature extraction task to demonstrate the performance of our multi-modality active learning approach. We selected 20 concepts in TRECVID 2003 and 2004 semantic feature extraction tasks. The development set is the collection of news video from ABC and CNN. It contains 52943 shots and is totally around 60 hours. The 20 concepts are as follows:

- Outdoors
- News subject face
- Building
- Road (2003)
- Road (2004)
- Vegetation
- Female speech
- Car/truck/bus
- News subject monologue
- Non-studio setting
- Sporting event
- Weather news
- Boat/Ship
- Bill Clinton
- Basket scored
- People walking/running
- People
- Aircraft
- Animal
- Beach

Low-level features including color, edge, texture, and face are generated to learn the semantic features. After dividing an image into 5 by 5 grids, the color feature in each grid is computed as the mean and variance of color histogram from HSV color space. A canny edge detector is applied to extract edges from the images. The edge histogram for 5 by 5 grids is quantized at 45 degree intervals. Six oriented Gabor filters are applied to extract texture features. Schneiderman's face detection algorithm [17] is used to extract frontal and profile faces. The size and location of faces represent the face detection result.

Figure 10 compares the performance between the multi-modality active learning approach and single-modality active learning. We start the initial data with 1000 examples and during each iteration we choose 250 new examples from the 4 individual feature sets (for a total of 1000 new examples). The curve labeled multi-active depicts the results of the new approach and the curve labeled "single-active" is the approach which concatenates the 4 feature sets into one larger feature vector. The baseline uses the complete training data set without any active learning. Our evaluation experiments are performed on the TRECVID 2003 and 2004 ground truth provided by NIST in a separate test set. Our measurement is the macro-average mean average precision (MAP) of those 20 topics. From Figure 10, we note that active learning is very effective. Even the single-modality active learning approach can reach the same performance as using the whole training data set with only 7% of the labeled data (4000 over 52943). Furthermore, the new approach works much more effectively than the single-modality approach. Its performance was comparable to the baseline with only 3% of the training data.

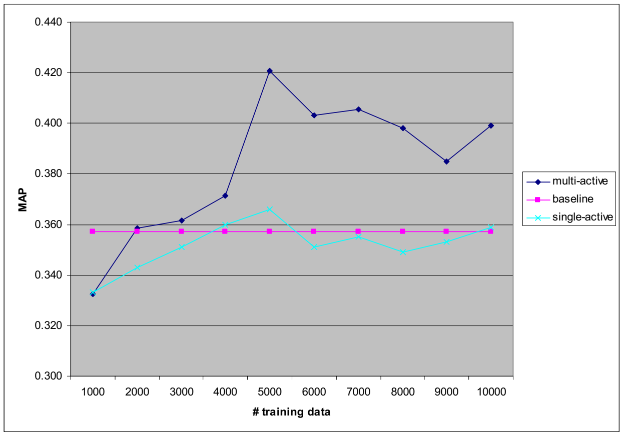


Figure 10. Classification performance for multimodality active learning and basic active learning.

Figure 11 compares the performance between the multimodality active learning approach and optimal fusion approach. By *optimal fusion* we mean that we did all possible combinations of different feature sets and choose the best performing combination as the result. It means if we have *r* different feature sets, we need to run our classification processes up to 2*r* times. We use *best-active* when, for each active learning iteration, we only choose the best fusion result. The *best-baseline* means we use the whole training set but fuse the multimodality by optimal fusion. The result shows our multimodality approach can reach as good as optimal fusion although needing more iterations. However, the optimal fusion is very computationally expensive. In our experiments, we have 4 different feature sets, so that for optimal fusion we need to consider 2^4 = 16 different combinations. The computation is 16 times as expensive.

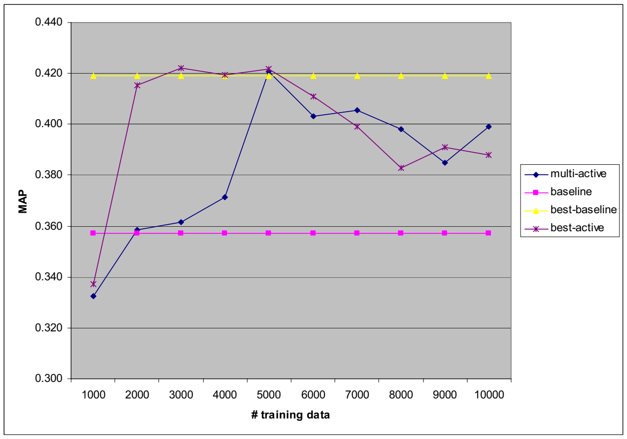


Figure 11. Classification performance for multi-modality active learning and optimal fusion approaches.

6. CONCLUSIONS

Automated concept classification provides the user with tools to filter and browse large collections of video for shots of interest. We present ENVIE, an application allowing the user to dynamically define and revise additional concepts in a timely manner through a simple, well understood interface. The concepts are built using multimodality active learning with RBF SVMs as the discriminative classifier, without these underlying details presenting additional cognitive load for the user or introducing new interface complexity. Rather, through active learning the user can efficiently improve the accuracy of the classifier through reduced numbers of training examples compared to passive machine learning algorithms.

Other researchers have contributed new work toward determining what imagery should next be labeled in the iterative step of active learning [3, 15] for better model performance. These recommendations will be folded into ENVIE so that the user is asked to annotate even fewer shots, and more informative shots, between iterations. One of ENVIE's goals is to provide new visual search capability for broad multilingual news corpora, where text metadata is either missing or fails to bridge the different source languages, but where visual concepts like indoor, outdoor, face, vehicle, etc., can provide the search strategies based on visual characteristics alluded to by Trant regarding growing multimedia collections [19]. Through performance evaluations using open testing procedures, metrics, and data, we plan to assess the benefits of ENVIE and its active learning component for interactive video information retrieval, with ENVIE's development driven by the goal of providing efficient, effective access to relevant shots from video collections.

7. ACKNOWLEDGMENTS

This material is based on work supported by the Advanced Research and Development Activity (ARDA) under contract numbers H98230-04-C-0406 and NBCHC040037. We thank the members of the Informedia research team who make this work possible (http://www.informedia.cs.cmu.edu), especially Bob Baron for his database administration and management, and James Lanagan for his implementation of the slider+histogram interface.

8. REFERENCES

[1] Ahlberg, C. and Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proc. CHI '94*, ACM Press, 1994, 313-317.

[2] Burges, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery, 2,* 2 (1998), 121-167.

[3] Chang, E.Y., Tong, S., and Goh, K.-S. Support Vector Machine Concept-Dependent Active Learning for Image Retrieval. *IEEE Transactions on Multimedia* (anticipated 2005), http://mmdb2.ece.ucsb.edu/~echang/mm000540.pdf.

[4] Chang, S.-F., moderator. Multimedia Access and Retrieval: The State of the Art and Future Directions. In *Proc. ACM Multimedia '99* (Orlando FL, Nov. 1999), ACM Press, 443-445.

[5] Christel, M. and Conescu, R. Addressing the Challenge of Visual Information Access from Digital Image and Video Libraries. In *Proc JCDL '05*, ACM Press, 2005, 69-78.

[6] Forsyth, D., and Ponce, J. *Computer Vision: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 2002.

[7] Freund, Y., and Schapire, R.E. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences, 55,* 1, 1997, 119-139.

[8] Gosselin, P.H., and Cord, M. RETIN AL: An active learning strategy for image category retrieval. In *Proc. IEEE Conf. Image Processing* (Singapore, October 2004), 2219-2222.

[9] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[10] Hauptmann, A.G., and Christel, M.G. Successful Approaches in the TREC Video Retrieval Evaluations. *Proc. ACM Multimedia '04*, ACM Press (2004), 668-675.

[11] Lee, H. and Smeaton, A.F. Designing the User Interface for the Físchlár Digital Video Library, *J. Digital Info.* 2(4), http://jodi.ecs.soton.ac.uk/Articles/v02/i04/Lee/, May 2002.

[12] McCallum, A., and Nigam, K. Employing EM in pool-based active learning for text classification. In *Proc. Int'l Conf. on Machine Learning*. Morgan Kaufmann, 1998, 350-358.

[13] Naphade, M., and Smith, J.R. Active Learning for Simultaneous Annotation of Multiple Binary Concepts. In *Proc. IEEE Intl. Conf. on Multimedia and Expo (ICME)* (Taipei, Taiwan, June, 2004), 77-80.

[14] Naphade, M.R., and Smith, J.R. On the Detection of Semantic Concepts at TRECVID. *Proc. ACM Multimedia '04*, ACM Press (2004), 660-667.

[15] Nguyen, H.T., and Smeulders, A. Active Learning Using Pre-clustering. In *Proc. Int'l Conf. on Machine Learning* (Banff, Canada, July 2004). ACM Press, 2004.

[16] Rowe, L.A. and Jain, R., *ACM SIGMM Retreat Report on Future Directions in Multimedia Research*, http://www.sigmm.org/Events/reports/retreat03/sigmm-retreat03-final.pdf, March, 2004.

[17] Schneiderman, H., and Kanade, T. Probabilistic Modeling of Local Appearance and Spatial Relationships of Object Recognition. In *Conf. Computer Vision and Pattern Recognition (CVPR '98)* (Santa Barbara, CA, June, 1998). IEEE Computer Society, 1998, 45-51.

[18] Tong, S., and Chang, E. Support Vector Machine Active Learning for Image Retrieval. In *Proc. ACM Multimedia 2001* (Ottawa, Canada, October, 2001). ACM Press, 2001, 107-118.

[19] Trant, J. *Image Retrieval Benchmark Database Service: A Needs Assessment and Preliminary Develoment Plan.* Council on Library and Information Resources and the Coalition for Networked Information, Archives & Museum Informatics, http://www.clir.org/pubs/reports/trant04/tranttext.pdf, January 2004.

[20] Wang, L., Chan, K.L., and Zhang, Z. Bootstrapping SVM Active Learning by Incorporating Unlabelled Images for Image Retrieval. In *Conf. Computer Vision and Pattern Recognition (CVPR '03)* (Madison, WI, June, 1998). IEEE Computer Society, 2003, 629-634.