# Semi-Supervised Learning for Semantic Video Retrieval

Ralph Ewerth
Dept. of Mathematics and Computer Science
University of Marburg
Hans-Meerwein-Str., D-35032 Marburg, Germany
ewerth@informatik.uni-marburg.de

Bernd Freisleben
Dept. of Mathematics and Computer Science
University of Marburg
Hans-Meerwein-Str., D-35032 Marburg, Germany
freisleb@informatik.uni-marburg.de

## ABSTRACT

The automatic understanding of audiovisual content for multimedia retrieval is a difficult task, since the meaning respectively the appearance of a certain event or concept is strongly determined by contextual information. For example, the appearance of a high-level concept, such as e.g. maps or news anchors, is determined by the used editing layout which usually is typical for a certain broadcasting station. In this paper, we show that it is possible to adaptively learn the appearance of certain objects or events for a particular test video utilizing unlabeled data in order to improve a subsequent retrieval process. First, an initial model is obtained via supervised learning using a set of appropriate training videos. Then, this initial model is used to rank shots for each test video v separately. This ranking is used to label the most relevant and most irrelevant shots in a video v for subsequent use as training data in a semi-supervised learning process. Based on these automatically labeled training data, relevant features are selected for the concept under consideration for video v. Then, two additional classifiers are trained on the automatically labeled data of this video. Adaboost and Support Vector Machines (SVM) are incorporated for feature selection and ensemble classification. Finally, the newly trained classifiers and the initial model form an ensemble. Experimental results on TRECVID 2005 video data demonstrate the feasibility of the proposed learning scheme for certain high-level concepts.

### Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods.*

### General Terms

Algorithms, Experimentation, Performance.

### Keywords

Semantic video retrieval, semi-supervised learning.

## 1. INTRODUCTION

The ultimate goal of research efforts in the domain of multimedia retrieval is the automatic understanding of audiovisual content. If it were feasible to automatically understand what is shown in a video shot or what is said in a news cast, then it would be much easier to answer user queries for multimedia databases. However, automatically recognizing objects and events in an image or in a video is a difficult problem, although there has been some success for particular object recognition tasks, such as face detection [20]. Naphade and Smith [14] state that first TRECVID [19] efforts showed that approaches like e.g. query by content using low-level features are inadequate to successfully search a large video collection. This is the reason why current research in the field of video retrieval focuses on the detection of so-called high-level features (concepts or topics) in order to index video shots to finally support different kinds of queries. Recent approaches to high-level concept detection are typically based on automatically extracted low-level features which are used by supervised machine learning methods to infer about semantic scene content. Such low-level features are, for example, color histograms, texture and shape descriptors [13], motion information [10] for video content, and the zero crossing rate ratio, the short-time energy ratio or the spectrum flux [12] for audio content. The incompatibility of low-level features that can be extracted automatically for an audiovisual scene and the high-level meaning associated by humans is considered as the "semantic gap" [5].

In an arbitrary video, the number of object classes, objects (or class instances), events and topics is very large, and it is not reasonable to use a particular detector for each possible concept, because it is not known in advance who or what will appear in a video. Furthermore, even for partially solved object detection or object recognition problems like face recognition, the system performance heavily depends on pose and illumination of a face [15]. In addition, in an arbitrary video, the context of events and person occurrences determines the complexity of the recognition task as well, e.g. a person's appearance depends on clothes and age etc. This does not only hold for person recognition but also for the overwhelming majority of semantically relevant objects and events. Often, the meaning respectively the appearance of a certain event or concept is strongly determined by contextual information. For example, the appearance of a high-level concept, such as e.g. maps or news anchors in a certain news program, is strongly determined by the editing layout which is specific for a certain broadcasting station.

In this paper, we propose a semi-supervised learning scheme to adaptively learn the appearance of certain objects or events for a particular video with the aim of enhancing the retrieval quality. In previous papers [6][7], we have shown that an initial object model obtained for a particular video via unsupervised learning can be improved adaptively for this video. This process has been


CIVR'07, July 9-11, 2007, Amsterdam, The Netherlands.

named self-supervised learning because only unlabeled data are used in this framework. We show that a similar semi-supervised learning scheme can also improve the retrieval performance for high-level concepts, although the performance of the available basis classifiers (based on supervised learning) is relatively low. To achieve this, we exploit the fact that there are concepts whose appearance is strongly related to a certain video source (e.g. maps in a news cast). The novel idea is to estimate the relevant features for a given concept in a particular test video v. It is assumed that an initial concept model is available which has been obtained via supervised learning using a set of appropriate training videos. Based on this initial model, shots are ranked separately for each test video v. Then, features are selected with respect to the concept's appearance in this video v and new classifiers are trained using only the best and worst ranked shots as positive and negative training samples. The feature set is split in order to train additional classifiers with different views to assemble a robust ensemble of classifiers which is finally used to re-classify respectively re-score the shots of this particular test video v. Experimental results for the MediaMill Challenge [18] part of the TRECVID 2005 video set demonstrate the feasibility of the proposed approach for certain concepts.

The remainder of the paper is organized as follows. In section 2, some related work for video retrieval is discussed, in particular with respect to semi-supervised methods. The main processing steps of the semi-supervised retrieval system are presented in section 3. Experimental results using the TRECVID 2005 [19] video training set (and the MediaMill Challenge [18] system as the baseline system) are presented in section 4. Section 5 concludes the paper and outlines areas for future work.

## 2. RELATED WORK

First, we present state-of-the-art approaches to concept detection, such as the IBM Video Retrieval System [1] and the MediaMill system [17]. These approaches have in common that a mapping is learned between low-level and high-level features using machine learning techniques. Furthermore, several classifiers are trained on different feature subsets and on the whole feature set, respectively. Second, since we augment unlabeled data in our semi-supervised learning framework, we discuss related approaches that incorporate unlabeled data in the training process in order to reduce the labeling effort.

Amir et al. [1] present a set of different machine learning techniques to map low-level features to high-level semantic concepts. The machine learning techniques are: Support Vector Machine, Gaussian Mixture Models, Maximum Entropy Methods, a modified Nearest-Neighbor classifier, and Multiple Instance Learning. Furthermore, different fusion methods are investigated. The authors have conducted extensive experiments on TRECVID video data (the 2003 and 2005 development set) to select the best audiovisual and textual features for this task A semantic model vector (according to Smith et al. [16]) is built consisting of confidence scores for 39 LSCOM lite concepts. The idea of semantic model vectors is to train a binary classifier for each concept and combine the classifiers' outputs respectively their confidence scores in the so-called semantic model vector. Optionally, Principal Component Analysis can be applied to a model vector if dimensionality reduction is needed. Semantic model vectors represent the video shots and are finally used to compare different videos for retrieval purposes. In [1], several experiments are conducted where several strategies to fuse features or classifiers are investigated.

Snoek et al. [17] suggest the semantic pathfinder to extract semantic concepts from video shots. The idea is based on an authoring metaphor, i.e. videos are considered as edited entities which are produced with a certain intention and purpose. Videos are processed in the semantic pathfinder in several analysis steps: the "content analysis" step, the "style analysis" step and the "context analysis" step. In the content analysis step, the following modalities are considered: visual features and text which is obtained from transcribed speech. A multi-class SVM is trained for a number of pre-defined proto-concepts using only a few training samples. The feature vector representing the visual content consists of the percentage of pixels per proto-concept. For textual analysis, for each concept x a separate lexicon is created that contains the words (after stop word removal) that co-occur with x in the training set shots. The textual feature vector per shot consists of a histogram for the words related to concept x. The visual feature vector v and textual feature vector t are fused and serve as input to train a visual model for each concept using SVM which represents the first step: "content analysis". The second step, "style analysis", considers: 1.) layout (shot length, overlaid text, silence, voice-over), 2.) content (faces, face position, cars, object motion, frequent speaker; length of overlaid text, video text named entity; voice named entity), 3.) capture (camera distance, camera motion, camera motion type) and 4.) context, which serves to enhance or reduce the correlation between semantic concepts. The last step, "context analysis", uses the probabilities of style analysis that a concept is present in a shot: these probabilities are fused in the context vector c. Finally, a held-out validation set is used to find for each concept individually the best path through those three analysis steps for content, style, and context. This best concept path is then used to retrieve shots.

In a way, these approaches are representative for the concept detection part of the current generation of video retrieval systems. In recent years, first works have considered the incorporation of unlabeled data into the supervised learning process. Blum and Mitchell [2] suggest co-training in order to augment unlabeled data into the classification process. It is assumed that there are two different independent views on the classes respectively data. The feature set is divided according to these views for each sample and each classifier. Each classifier tries to classifiy unlabeled data and passes the data with the highest confidence score to the other classifier as training data. This process is repeated several times. Yan and Naphade [23] extend the co-training approach for semantic concept detection in video shots. They divide the feature set for a video shot into the textual and the visual representation and thus meet the prerequisite for co-training. Since classifiers for semantic video concepts are not sufficiently accurate and the use of incorrectly labeled data might degrade performance, they integrate a human annotator in the processing loop who reviews and eventually corrects the automatically labeled samples. They report experiments with several co-training iterations for four semantic concepts from the TRECVID 2004 high-level feature detection task: airplane, basketball, Bill Clinton and people. Experiments show that the performance degrades with the number of iterations when the original co-training approach has been used, presumably due to the low accuracy of the basis classifiers. In case when the

automatically labeled data were corrected by a human reviewer, the classification accuracy increased with the number of iterations, at least for most of the iterations. Anyway, compared to the basis performance, an improvement was achieved for any concept and any iteration number, which was not always the case for the standard co-training procedure.

Yan and Naphade [23] present another extension called "semi-supervised cross feature learning" for co-training. First, initial classifiers are trained on the labeled part of the training set. Then, in each iteration, samples that are classified with highest and lowest confidence by one classifier form a training set for the other classifier which is then trained only on the automatically labeled training set (without the samples from the original, manually labeled training set). Third, the performance of the newly trained classifiers is tested on a validation set and a corresponding weight is computed for each classifier. This weight is possibly 0 in case the classifier would degrade the performance of the initial classifier. The authors show theoretically that the minimal risk (to make an error) is never higher for the ensemble created in this way. Finally, they extend the learning approach for multiple views. Their experimental results for 11 concepts from the TRECVID 2003 data set show that the proposed semi-supervised approach slightly improves average precision of the baseline system from 0.216 up to 0.233 whereas co-training leads to a lower average precision of 0.177. In case the whole labeled training set was used, an average precision of 0.24 was achieved.

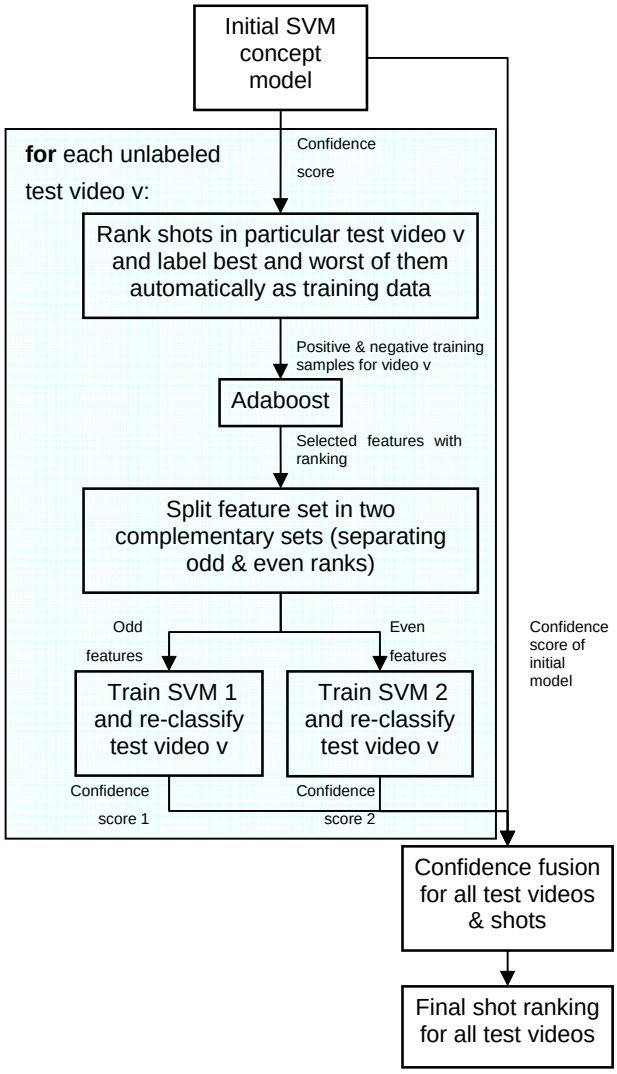
Wu et al. [21] state that the appearance of concepts changes over time and address the problem of concept drifting in videos. They use Gaussian Mixture Models (GMM) to model a concept and propose an incremental online learning framework to cope with concept drifting. For this purpose, videos are processed in a batch mode. The first batch of pre-labeled data is used to learn a global GMM for each concept. The next batch of data (five videos are considered as one batch in their experiments) is employed to learn a set of locally optimized GMMs for each concept from the first unlabeled portion of the new data, aiming at an optimal classification performance on the current test batch. At first, the local models are used to classify the test data. Then, they are used to update the global GMMs. This online process is repeated for each newly upcoming test batch. Their experiments show that locally optimized GMMs outperform the updated global GMMs.

Yan and Hauptmann [22] argue that textual information is the most useful information source for video retrieval but claim that additional audiovisual features can improve a retrieval result. Based on this argumentation, they use textual information to achieve a first ranking and employ other features (audio, visual, motion) to re-rank and refine a retrieval result. However, learning based on a retrieval result must deal with noisy labels because the retrieval result is not perfect. The authors argue that the first k returned documents represent sufficiently positive training samples and consider the remaining set of documents as negative training samples. This training set is used to perform a boosting process that is augmented with audiovisual features, called co-retrieval. The authors report that the boosted re-ranking approach improves the precision compared with the baseline system which used only textual features.

# 3. SEMI-SUPERVISED LEARNING FOR SEMANTIC VIDEO RETRIEVAL

As discussed above, a common approach to detect semantic concepts in videos is based on a mapping between low-level features and the high-level features. It can be observed for some concepts that their visual appearance is strongly related to a particular video source. For example, weather news are typically related to the following concepts. A map usually presents the area of interest, and some symbols indicate rain, clouds, wind and sun. Furthermore, displayed text gives information about locations, temperatures, all together explained and moderated by a human expert. For a particular instance of weather news, those general elements take a concrete shape, e.g. the map color, the font type and size of text and the style of the symbols are identical for a certain TV program. In addition, the spatial composition of the shot will be specific for this TV cast, e.g. the moderator's position and the camera distance will be specific as well.

**Figure 1: A concrete realization of the framework for semi-supervised learning for video retrieval.**



Our proposed approach is aimed at learning the specific appearance of a certain concept in a particular unlabeled test video v based on an initial model. To improve the retrieval performance for such concepts, we suggest the following concept detection and retrieval framework to exploit the specific appearance of a concept. Its main processing steps are as follows (see also Figure 1). First, a basis system using supervised learning (the SVM data of the MediaMill Challenge system [18] are used in our prototype) is used to map low-level features to high-level concepts. Second, the learned model is used to achieve a first separate ranking for the shots of each test video v according to the SVM confidence scores. Third, the best features are selected for this test video v and optionally split into two disjoint feature sets.

**Semi-Supervised-Learning-for-Retrieval**($svm_C$, V)
**Input**: Initial SVM model $svm_C$ for concept C;
Set of test videos V in the database;
**Output**: List of ranked shots of all test videos;

**Algorithm:**

**for** each test video v ∈ V
Rank shots(v) according to $svm_C$ confidence score in descending order;
// generate training samples for this video
*nrPotentialPositiveSamples* := |{shot s ∈ shots(v) | $svm\_confidence_C$(s) >= *min_Confidence*}|;

// if not enough training data exist
**if** *nrPotentialPositiveSamples* < *minNrPosShots* **then**
Normalize $svm_C$ scores for all shots ∈ v to ensemble score and store them;
**continue** loop with next video;

P:= Select *max_percentage_Pos* shots s with $svm\_confidence_C$(s) >= *min_Confidence* as positive training samples;
N:= Select *max_percentage_Neg* shots s with $svm\_confidence_C$(s) < *min_Confidence* as negative training samples;

// Get list of ranked features for concept C for this
// particular video v using a modified Adaboost;
F:=AdaboostFeatureSelection(P, N);

Split F in two sets $F_{odd}$ and $F_{even}$;

Train $newClassifier_1$ with features $F_{odd}$ and training data P and N;
Train $newClassifier_2$ with features $F_{even}$ and training data P and N;
**for** each shot s ∈ shots(v)
Obtain two confidence scores $confidence1_C(s)$ and $confidence2_C(s)$ using $newClassifier_1$ and $newClassifier_2$;
$total\_score(s) := a_1 * svm\_confidence_C(s) + a_2 * confidence1_C(s) + a_3 * confidence2_C(s)$;
**end for;**
**end for;**

**return** ranking of all shots of all test videos according to *total_score(s)* in descending order;

**Figure 2: Pseudo code for the semi-supervised learning retrieval scheme.**

These feature sets are used to train new SVM classifiers using only the automatically labeled data from the current test video v under consideration. Then, the additional classifier(s) and the original classifier form an ensemble which is used to re-compute a total confidence score for each shot in the test video v.

Finally, the confidence scores for all shots of all test videos v are used to rank them, and shots are returned to the user according to this ranking. The algorithmic steps are now explained in more detail (see also the pseudo code in Figure 2). To give a complete picture of the system, we also discuss in section 3.1 the employed low-level features which have been donated by Snoek et al. [18].

## 3.1 Learning the Initial Concept Model

In general, any classifier that produces a confidence score regarding a concept's presence in a video shot can be used as the basis classifier in our system. For ease of implementation, the visual features, SVM models and confidence scores of the MediaMill Challenge system [18] are used in our system, which have been kindly donated by Snoek et al. These features are aimed at describing a complex scene as a composition of 15 so-called proto-concepts, as e.g. building, car, desert, maps, mountain, road, sky, snow, water etc. Based on a small training set per proto-concept (between 20 and 320 samples), the texture characteristics are captured for each proto-concept. Therefore, the color channels are decorrelated and then an invariant edge detector (equivalent to a Gaussian derivative filter) is applied to each of the newly obtained color channels. The distribution of edges in an image region is then represented by a Weibull distribution which is described by three parameters: μ describes the origin of distribution, β describes the width of the distribution, and γ stands for the peakness of the distribution of edge responses [9]. So far, only models of the proto-concepts are obtained. An image is now divided into a number of regions and each region is described by a set of features that reflect the similarity to each of the pre-defined proto-concepts. A similarity measure $C^2$ has been suggested in [9] to compare two Weibull distributions F and G reflecting the squared error between them:

$$C^2(F,G) = \sqrt{\frac{\min(\gamma_F,\gamma_G)}{\max(\gamma_F,\gamma_G)}\frac{\min(\beta_F,\beta_G)}{\max(\beta_F,\beta_G)}} \qquad (1)$$

Finally, two different histograms are created for each proto-concept to describe an image. The first one, $H_{accu}$ adds the similarity values for all pairs of image regions and annotated training samples, whereas $H_{best}$ is the maximum of this set of similarity values. Hence, $H_{accu}$ captures global information about the image and $H_{best}$ captures local image information. Since both features are extracted for two different Gaussian smoothing filters and two different regions sizes, eight features are obtained for each of the proto-concepts, yielding a total number of 120 features. For further details, the reader is referred to [9].

As mentioned above, these features, SVM models and related confidence scores for 101 concepts were provided by Snoek et al. [18] for the TRECVID 2005 training video set. SVM models are learned based on 70% of this training set, while the remaining 30% are used as the test set. Confidence scores are computed for the test set partition for each concept using the SVM models.

## 3.2 Adapting a Concept Model to a Video

In this section, we propose a semi-supervised learning scheme to learn a specific concept model for its appearance in a certain test video v and utilize it for the retrieval process.

### *3.2.1 Generating Training Data for a Video*

To adapt a model to its appearance in a particular test video v, the basis SVM model is used to rank the shots in this video according to their probability containing the concept. Then, the p shots with the highest probability and the n shots with the lowest probability serve as positive and negative samples for the subsequent adaptive learning process. The positive samples must exceed a minimum SVM confidence score *min_Confidence* to be considered for the subsequent training process. In case that the number of positive samples is below another threshold *minNrPosShots* defining a minimum number of positive samples, the semi-supervised learning process is not applied for this test video v. In this case, the initial scores are normalized with respect to the final number of ensemble classifiers. In addition, two further thresholds define the maximum percentage of shots in a video which are used as positive and negative training samples, respectively. The reader is referred to the pseudo code for these algorithmic steps in Figure 2.

### *3.2.2 Selecting Features*

Once the positive and negative training samples have been obtained automatically for a test video v, feature selection is conducted using a slightly modified version of the Adaboost procedure (see pseudo code in Figure 3). Adaboost is a meta-classifier that provides an ensemble decision rule that is a weighted sum of the n different classifiers. It was shown by Freund and Schapire [8] that Adaboost minimizes the error on the training data as the number of training rounds (and hence the number of classification models) increases. A very nice property of Adaboost is that this is guaranteed as long as each selected classification model achieves an error rate below 0.5, i.e. Adaboost is able to build a classification model (a "strong classifier") with an arbitrarily low error rate on the training data based on the estimated combination of possibly "weak classifiers". We employ the modified version of Adaboost shown in Figure 3 in order to perform feature selection.

As stated above, Adaboost combines a number of n (possibly weak) classifiers to build a strong classifier within n rounds of training. Therefore, a classification error is estimated for each (weak) classifier by estimating the best threshold that separates positive and negative samples using the one-dimensional data of a single feature. The classification error is calculated based on the current sample weights of misclassified samples. The training samples are weighted equally in the beginning. Training samples which are misclassified by the current classification model are re-weighted such that they have more impact in the next training round for the next "weak classifier". Thus, a newly selected feature has a higher probability to correctly classify those training samples that have been misclassified in preceding rounds.

### *3.2.3 Building an Ensemble of Classifiers*

After the selected features have been obtained, they can be used to train any other classifier for the test video v under consideration: in our system, a SVM is used [4]. Another possibility is to build an ensemble of classifiers using majority voting. In this case, the classifiers forming an ensemble should have a certain level of independence. There is evidence [11] that a reasonable degree of independence of ensemble classifiers improves accuracy and guarantees at least the accuracy of the weakest classifier in the ensemble if the classifiers' accuracies exceed a certain value. There is no generally accepted measure for the diversity of classifier ensembles, and it has not been shown theoretically how to successfully build a classifier ensemble, as remarked by Brown et al. [3]. Nevertheless, as stated before, independence between the ensemble members is beneficial. In [6], we have successfully exploited the boosting procedure of Adaboost to assign features alternating between two different classifiers according to their odd respectively even rank in the feature selection process. This approach is motivated by the fact that during the Adaboost process the weights of those samples are increased which have been misclassified by the preceding weak classifier. Thus, the next selected classifier should be partially independent of its direct predecessor. This property is our motivation to split the feature set depending on the rank of features for subsequent training in order to increase the independence of the obtained feature sets. In case of n classifiers, the feature with rank k is assigned to classifier k modulo n. In our proposed system, the feature set is split into two disjoint feature sets where each of them is used in a separate subsequent SVM training process. In this SVM training, concept models are learned based only on the automatically labeled training examples taken from the current test video v, yielding two new SVM models.

**AdaboostFeatureSelection**(P, N)
**Input**: Training shot set S consisting of positive sample set P and negative sample set N;
// each sample shot s ∈ S is described via a set of
// features F
**Output**: List of ranked features;

**Algorithm:**

**for** each training instance instX ∈ S:
    weight[instX] = 1/|S|;
F'= ∅;
**for** i=1 to N boosting rounds
    **for** each feature f
        // the next if-condition is normally *not* part
        // of an Adaboost procedure but inserted
        // in order to avoid selecting a feature two
        // times (with possibly different thresholds)
        **if** f ∈ F' **then**
            **continue** loop with next feature;
        Find the threshold t that separates the classes A and B using feature f with the lowest error e, this error e is the sum of weights of the misclassified samples;
    Choose f with minimum error $e_{min}$ as current feature f';
    **if** ($e_{min}$ == 0 || $e_{min}$ >= 0.5) **then** terminate;
    **for** each instance instX
        **if** instX is classified correctly **then**
            weight[instX] *= $e_{min}$ / (1- $e_{min}$);
    Normalize all weights[i] (sum is 1);
    F' = F' ∪ {f'};
    Feature f' has rank i in the feature selection ranking;

**return** list of ranked features;

**Figure 3: Pseudo code for our modification of Adaboost to employ a feature selection process.**

Finally, three SVM models are available for each video, the initial (global) model and two (local) models that are learned with training data that have been labeled automatically for this particular test video v using the initial model.

### *3.2.4 Re-Ranking all Video Shots*

At this stage, the question is how to utilize the SVM models, in particular with respect to those models which were trained adaptively for a test video v, to achieve a retrieval result for all (test) videos in the database. After the video-specific learning of concept models, three different classifiers are available for each shot and the concept under consideration: the initial model learned on the whole *training* set and two SVM classifiers which have been learned specifically for a *test* video v. The two newly trained SVM models are now used to obtain additional confidence scores which indicate the probability that a shot exhibits a certain concept. Then, a final confidence score is computed for each shot using the three models (weighted sum of these three scores) which is finally used to rank the shots of all test videos.

## 4. EXPERIMENTAL RESULTS

In our experiments, the usefulness of the proposed learning framework has been investigated for several video concepts which were expected to have a specific appearance in a certain video or TV program, respectively. The TRECVID 2005 training set consisting of 137 videos was used for this purpose. As mentioned in the previous section, the MediaMill challenge [18] features, SVM models and related confidence scores are used as our baseline system. In the challenge scenario, the features are extracted from the TRECVID 2005 training video set and SVM models have been learned using 70% of this training set, whereas the remainder is used as the test set. Hence, the proposed system is tested on the remaining 30% as well. The minimum probability that must be assigned to a shot to serve as a positive training sample was set to 0.01, and at least one shot in a test video v must have fulfilled this condition. Otherwise, semi-supervised learning is not conducted for this test video v. The libSVM [4] is used in our implementation to learn SVM models and to obtain confidence scores. Average precision is used to measure the retrieval performance and is defined as:

$$avg\_precision = \frac{1}{|R|}\sum_{k=1}^{|A|}\frac{\left|R \cap L^k\right|}{k}\psi(l_k) \qquad (2)$$

where A is the result set of returned documents, $L_k = \{l_1, \ldots, l_k\}$ is the subset of the k responses which are the most similar responses in A with respect to a confidence score, R is the set of relevant documents and $\psi(l_k)$ is a function which evaluates to 1 for $l_k \in R$ and to 0 for $l_k \notin R$.

We expected the following high-level concepts to have an appearance related to a specific video: anchor, charts, maps, and overlaid text. Several experiments were conducted for these high-level features. First, the performance of the proposed semi-supervised framework is compared with the MediaMill baseline system. In this scenario, 90 features are selected and this feature set is split to train two additional SVMs for each video.

**Table 1: Experimental results for the high-level feature „maps": Average precision values for the MediaMill challenge baseline system and the proposed semi-supervised approach are presented.**

| *Average Precision [%]* | *Basis system* | *Semi-Supervised Learning* | *Improvement* |
|---|---|---|---|
| Top-100 | 76.0 | 89.6 | +13.6 |
| Top-1000 | 58.2 | 69.0 | +10.8 |
| Top-2000 | 52.7 | 56.6 | +3.9 |
| All shots | 47.6 | 51.8 | +4.2 |

**Table 2: Experimental results for the high-level feature „anchor": Average precision values for the MediaMill challenge baseline system and the proposed semi-supervised approach are presented.**

| *Average Precision [%]* | *Basis system* | *Semi-Supervised Learning* | *Improvement* |
|---|---|---|---|
| Top-100 | 97.1 | 98.0 | +0.9 |
| Top-1000 | 83.1 | 84.8 | +1.7 |
| Top-2000 | 75.3 | 78.2 | +2.9 |
| All shots | 63.1 | 69.4 | +6.3 |

**Table 3: Experimental results for the high-level feature „charts": Average precision values for the MediaMill challenge baseline system and the proposed semi-supervised approach are presented.**

| *Average Precision [%]* | *Basis system* | *Semi-Supervised Learning* | *Improvement* |
|---|---|---|---|
| Top-100 | 60.6 | 73.4 | +12.8 |
| Top-1000 | 44.4 | 51.3 | +6.9 |
| Top-2000 | 40.5 | 42.6 | +2.1 |
| All shots | 32.7 | 35.7 | +3.0 |

**Table 4: Experimental results for the high-level feature „overlaid text": Average precision values for the MediaMill challenge baseline system and the proposed semi-supervised approach are presented.**

| *Average Precision [%]* | *Basis system* | *Semi-Supervised Learning* | *Improvement* |
|---|---|---|---|
| Top-100 | 97.5 | 96.9 | -0.6 |
| Top-1000 | 87.9 | 85.1 | -2.8 |
| Top-2000 | 83.9 | 80.4 | -3.5 |
| All shots | 66.9 | 62.1 | -4.8 |

For this experiment, the results are presented for each concept separately in terms of average precision in Table 1 - Table 4 for several sizes of retrieved shots (documents). The semi-supervised learning scheme improves the average precision for three of those four concepts, except for the concept "overlaid text". The average precision with respect to 100 retrieved documents is more than 10% higher for the concepts "maps" and "charts" compared with the baseline system. In case of the concept "anchor", the top-100 and top-1000 average precision of the baseline is already rather high, so there is only a little improvement. However, average precision increased by about 3% with respect to the top-2000 result. Let us consider the results for the other concepts at the level of top-2000 average precision which is used in the TRECVID evaluation series: for the concept "maps", average precision increased by about 4% and for the concept "anchor" by about 3%, respectively 4% and 6% in case all shots are considered as retrieved. For the concepts "maps" and "charts" the performance improvement is rather high for the top-100 and top-1000 average precision. This is a useful system property for a user who first wants to browse through these returned shots to access relevant shots more quickly.

**Table 5: Experimental results for the high-level feature „maps" for different learning strategies. The number in brackets indicates the number of selected features by Adaboost.**

| *Average Precision [%]* | *Only new SVM model (30)* | *Initial model & 1 new SVM model (all)* | *Initial model & 1 new SVM model (30)* | *Initial model & 1 new SVM model (90)* | *Initial model & 2 new SVM models (90)* |
|---|---|---|---|---|---|
| Top-100 | 35.4 | 85.3 | 86.6 | 87.8 | **89.6** |
| Top-1000 | 18.7 | 64.0 | 67.3 | 68.5 | **69.0** |
| Top-2000 | 15.0 | 58.9 | 58.4 | **59.7** | 56.6 |
| All shots | 13.4 | 52.5 | 52.0 | **52.5** | 51.8 |

**Table 6: Experimental results for the high-level feature „anchor" for different learning strategies. The number in brackets indicates the number of selected features by Adaboost.**

| *Average Precision [%]* | *Only new SVM model (30)* | *Initial model & 1 new SVM model (all)* | *Initial model & 1 new SVM model (30)* | *Initial model & 1 new SVM model (90)* | *Initial model & 2 new SVM models (90)* |
|---|---|---|---|---|---|
| Top-100 | 46.9 | **98.9** | 97.8 | 97.1 | 98.0 |
| Top-1000 | 50.5 | 83.8 | 83.7 | 84.5 | **84.8** |
| Top-2000 | 47.4 | 76.6 | 76.8 | 75.5 | **78.2** |
| All shots | 42.8 | 68.3 | 68.1 | 66.8 | **69.4** |

Further experiments have been conducted for those concepts for which the retrieval performance could be improved. Five additional learning strategies were investigated: 1.) Only one newly trained model and the related score is used (without the initial model) to finally rank all test video shots in the database. 2.) The initial baseline SVM model and one additional newly learned SVM model is employed, all features are used; 3.) The initial baseline SVM model and one additional newly learned SVM model is employed, 30 features are selected for each video; 4.) As 3.), but 90 features are used; 5.) The proposed semi-supervised learning with two additional SVM models and 90 features. The experimental results are presented for each concept separately in terms of average precision in Table 5 - Table 7. At first, it is observable that using only the newly trained model deteriorates average precision significantly. However, the results also show that the proposed semi-supervised learning scheme using two additional SVM models is superior in nearly all cases in terms of the top-100 and top-1000 average precision, while the use of only one additional model achieves sometimes a slightly better result. Since the improvement is clearer for the former results, the semi-supervised framework that uses three classifiers in total seems to be superior for most scenarios. Overall, we have shown that the proposed semi-supervised learning scheme can learn the specific appearance of certain concepts in a video and improve the retrieval performance.

**Table 7: Experimental results for the high-level feature „charts" for different learning strategies. The number in brackets indicates the number of selected features by Adaboost.**

| *Average Precision [%]* | *Only new SVM model (30)* | *Initial model & 1 new SVM model (all)* | *Initial model & 1 new SVM model (30)* | *Initial model & 1 new SVM model (90)* | *Initial model & 2 new SVM models (90)* |
|---|---|---|---|---|---|
| Top-100 | 11.5 | 63.1 | 70.1 | 68.8 | **73.4** |
| Top-1000 | 8.3 | 48.1 | 50.5 | 49.7 | **51.3** |
| Top-2000 | 7.6 | 41.7 | **45.1** | 43.9 | 42.6 |
| All shots | 6.4 | 34.9 | 36.5 | **36.7** | 35.7 |

# 5. CONCLUSIONS

In this paper, we have proposed a semi-supervised learning framework for video retrieval with respect to high-level feature (concept) detection. For this purpose, the fact was exploited that there are concepts whose appearance or layout is strongly related to a certain video source or TV program (e.g. maps in a news cast). Relevant features are selected for a given concept based on an initial classification model specifically for each particular test video. Based on an initial model, shots are ranked separately for each test video v. Then, features are selected with respect to the concept's appearance in this video and new classifiers are trained. The feature set is split in order to train additional classifiers with different views to assemble a robust ensemble of classifiers for a particular test video. This ensemble is used to re-classify the shots of this video, and finally the total confidence score is employed to obtain a ranking for all video shots in a database. Experimental results on the TRECVID 2005 training set have demonstrated the

feasibility of the proposed approach for certain concepts which are strongly related to a video source or TV program. Furthermore, the experiments have demonstrated that the exclusive use of the newly trained models deteriorates retrieval performance, whereas the proposed feature split in conjunction with an ensemble of classifiers achieves the best results in most cases.

There are several issues for future work. First, an iterative learning procedure should be tested in the semi-supervised learning framework. Second, other feature types should be investigated since there might be features that are more clearly related to concept appearance or layout than those which have been used in our current system.

## 6. ACKNOWLEDGMENT

This work is financially supported by the Deutsche Forschungsgemeinschaft (SFB/FK 615, Teilprojekt MT).

## 7. REFERENCES

[1] Amir, A.., Argillander, J., Campbell, M., Haubold, A., Iyengar, G., Ebadollahi, S., Kang, F., Naphade, M. R., Natsev, A., Smith, J.R., Tesic, J., and Volkmer, T. IBM Research TRECVID-2005 Video Retrieval System, in TREC Video Retrieval Online Proceedings, (2005), http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[2] Blum, A. and Mitchell, T. Combining Labeled and Unlabeled Data with Co-Training. In Proceedings of the 11th Conference on Computational Learning Theory, Madison, Wisconsin, USA, 1998, 92-100.

[3] Brown, G. Wyatt, J., Harris, R., and Yao, X. Diversity Creation Methods: A Survey and Categorisation. In Information Fusion 6 (2005), Elsevier, 2005, 5-20.

[4] Chang, C.-C. and Lin, C.-J. LIBSVM: A Library for Support Vector Machines, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] Dorai, C. and Venkatesh, S. Media Computing – Computational Media Aesthetics. Kluwer Academic Publishers, Boston, 2002.

[6] Ewerth, R. and Freisleben, B. Self-Supervised Learning for Robust Video Indexing. In Proc. of the IEEE Conference on Multimedia & Expo 2006, Toronto, 2006, 1749-1752.

[7] Ewerth, R., Mühling, M., and Freisleben, B. Self-Supervised Learning of Face Appearances in TV Casts and Movies. In Proceedings of the IEEE Symposium on Multimedia, San Diego, CA, USA, 2006, 78-85.

[8] Freund, Y. and Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In Journal of Computer and System Sciences, 55(1), 1997, 119-139.

[9] Gemert, J., Geusebroek, J., Veenman, C., Snoek, C., and Smeulders, A. Robust Scene Categorization by Learning Image Statistics in Context. In Proceedings of Int'l Workshop on Semantic Learning Applications in Multimedia, in conjunction with CVPR'06, New York, USA, 2006.

[10] Jeannin, S. and Mory, B. Video Motion Representation for Improved Content Access. In IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, 2000, 645-655.

[11] Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., and Duin, R. P. W. Limits on the Majority Vote Accuracy in Classifier Fusion. In Pattern Analysis and Applications, 6, 2003, Springer-Verlag, 22-31.

[12] Lu, L., Jiang, H., and Zhang, H.-J. A Robust Audio Classification and Segmentation Method. In Proc. of the ACM Conf. on Multimedia 2001, Ottawa, Canada, 203-211.

[13] Manjunath, B. S., Ohm, J.-R., Vasudevan, V., and Yamada, A. Color and Texture Descriptors. In IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, 2001, 703-715.

[14] Naphade, M. R. and Smith, J. R. On the Detection of Semantic Concepts at TRECVID. In Proceedings of the ACM Conference on Multimedia, 2004, New York, 660-667.

[15] Phillips, P.J., Grother, P., Micheals, R. J., Blackburn, D. M., Tabassi, E., and Bone, M. Face Recognition Vendor Test 2002. Evaluation Report IR 6965, National Institute of Standards and Technology, www.itl.nist.gov/iad/894.03/face/face.html, March 2003.

[16] Smith, J. R., Naphade, M. R., and Natsev, A. Multimedia Semantic Indexing Using Model Vectors. In Proceedings of the IEEE International Conference on Multimedia & Expo 2003, Volume 2, Baltimore, Maryland, USA, 2003, 445-448.

[17] Snoek, C. G. M., Worring, M., Geusebroek, J.-M., Koelma, D. C., Seinstra, F. J., and Smeulders, A. W. M. The Semantic Pathfinder: Using an Authoring Metaphor for Generic Multimedia Indexing. In IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 28(10), 2006, 1678-1689.

[18] Snoek, C. G. M., Worring, M., van Gemert, J. C., Geusebroek, J.-M., and Smeulders, A. W.M. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In Proceedings of ACM the Conference on Multimedia, Santa Barbara, USA, 2006, 421-430.

[19] TRECVID: TREC Video Retrieval Evaluation Series. http://www-nlpir.nist.gov/projects/trecvid/

[20] Viola, P. and Jones, M. Robust Real-Time Face Detection. In International Journal of Computer Vision, Volume 57 (2), (2004), Kluwer Academic Publishers, 2004, 137–154.

[21] Wu, J., Ding, D., Hua, X.-S., and Zhang, B. Tracking Concept Drifting with an Online-Optimized Incremental Learning Framework. In Proc. of the 7th ACM Int'l Workshop on Multimedia Information Retrieval, 2005, Singapore, 33 – 40.

[22] Yan, R. and Hauptmann, A. G. Co-Retrieval: A Boosted Reranking Approach for Video Retrieval. In Proc. of the Int'l Conf. on Image and Video Retrieval, Dublin, Ireland, 2004, 60-69.

[23] Yan, R. and Naphade, M. Co-Training Non-Robust Classifiers for Video Semantic Concept Detection. In Proceedings of the IEEE International Conference on Image Processing 2005, Vol. 1, Singapore, 1205-1208.

[24] Yan, R. and Naphade, M. Semi-Supervised Cross Feature Learning for Semantic Concept Detection in Videos. In Proc. of the IEEE Int'l Conf. on Computer Vision and Pattern Recognition, 2005, Vol. 1, San Diego, CA, USA, 657-663.