The MediaMill TRECVID 2008 Semantic Video Search Engine

Draft notebook paper

C.G.M. Snoek, K.E.A. van de Sande, O. de Rooij, B. Huurnink, J.C. van Gemert[*], J.R.R. Uijlings, J. He, X. Li, I. Everts, V. Nedovic, M. van Liempt, R. van Balen, F. Yan[†], M.A. Tahir[†], K. Mikolajczyk[†], J. Kittler[†], M. de Rijke, J.M. Geusebroek, Th. Gevers, M. Worring, A.W.M. Smeulders, D.C. Koelma

ISLA, University of Amsterdam
Amsterdam, The Netherlands

[†]CVSSP, University of Surrey
Guildford, Surrey, UK

http://www.mediamill.nl

Abstract

In this paper we describe our TRECVID 2008 video retrieval experiments. The MediaMill team participated in three tasks: concept detection, automatic search, and interactive search. Rather than continuing to increase the number of concept detectors available for retrieval, our TRECVID 2008 experiments focus on increasing the robustness of a small set of detectors. To that end, our concept detection experiments emphasize in particular the role of sampling, the value of color invariant features, the influence of codebook construction, and the effectiveness of kernel-based learning parameters. For retrieval, a robust but limited set of concept detectors necessitates the need to rely on as many auxiliary information channels as possible. Therefore, our automatic search experiments focus on predicting which information channel to trust given a certain topic, leading to a novel framework for predictive video retrieval. To improve the video retrieval results further, our interactive search experiments investigate the roles of visualizing preview results for a certain browse-dimension and active learning mechanisms that learn to solve complex search topics by analysis from user browsing behavior. The 2008 edition of the TRECVID benchmark has been the most successful MediaMill participation to date, resulting in the top ranking for both concept detection and interactive search, and a runner-up ranking for automatic retrieval. Again a lot has been learned during this year's TRECVID campaign; we highlight the most important lessons at the end of this paper.

1 Introduction

Robust video retrieval is highly relevant in a world that is adapting swiftly to visual communication. Online services like YouTube and Truveo show that video is no longer the domain of broadcast television only. Video has become the medium of choice for many people communicating via Internet. Most commercial video search engines provide access to video based on text, as this is still the easiest way for a user to describe an information need. The indices of these search engines are based on the filename, surrounding text, social tagging, closed captions, or a speech transcript. This results in disappointing retrieval performance when the visual content is not mentioned, or properly reflected in the associated text. In addition, when the videos originate from non-English speaking countries, such as China, or the Netherlands, querying the content by text becomes even harder as robust automatic speech recognition results are difficult to achieve.

To cater for robust video retrieval, the promising solutions from literature are in majority concept-based [37], where detectors are related to objects, like a *telephone*, scenes, like a *kitchen*, and people, like *singing*. Any one of those brings an understanding of the current content. The elements in such a lexicon offer users a semantic entry to video by allowing them to query on presence or absence of visual content elements. Last year we presented the *MediaMill 2007* semantic video search engine [35] using a 572 concept lexicon, albeit with varying performance. Rather than continuing to increase the lexicon size, our TRECVID 2008 experiments focus on increasing the robustness of a small set of concept detectors by using a novel approach that builds upon recent findings in computer vision and pattern recognition. A robust but limited set of concept detectors necessitates the need to rely on as many information channels as possible for retrieval. To that end, we propose a novel framework for predictive video retrieval that automatically learns to trust one of three information channels that maximizes video search results for a given topic. To improve the retrieval results further, we extend our browsers by supplementing them with visualizations for swift inspection, and active learning mechanisms that learn to solve complex search topics by analysis from user browsing behavior. Taken together, the *MediaMill 2008* semantic video search engine provides users with robust semantic access to video archives.

The remainder of the paper is organized as follows. We first define our semantic concept detection scheme in Sec-

[*]Currently at: Willow, École Normale Supérieure Paris, France.

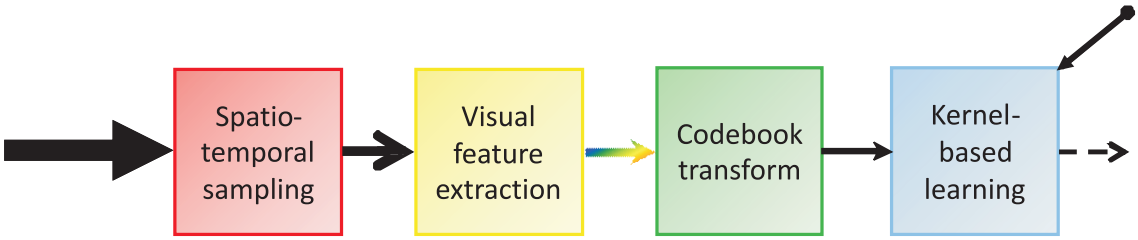

Figure 2: MediaMill TRECVID 2008 concept detection scheme, using the conventions of Figure 1. The scheme serves as the blueprint for the organization of Section 2.

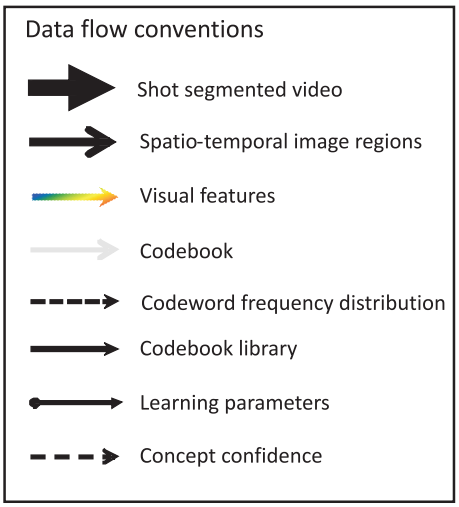

Figure 1: Data flow conventions as used in this Section. Different arrows indicate difference in data flows.

tion 2. Then we highlight our predictive video retrieval framework for automatic search in Section 3. We present the innovations of our semantic video search engine in Section 4. We wrap up in Section 5, where we highlight the most important lessons learned.

2 Detecting Concepts in Video

We perceive concept detection in video as a combined computer vision and machine learning problem. Given an n-dimensional visual feature vector x_i, part of a shot i [27], the aim is to obtain a measure, which indicates whether semantic concept ω_j is present in shot i. We may choose from various visual feature extraction methods to obtain x_i, and from a variety of supervised machine learning approaches to learn the relation between ω_j and x_i. The supervised machine learning process is composed of two phases: training and testing. In the first phase, the optimal configuration of features is learned from the training data. In the second phase, the classifier assigns a probability $p(\omega_j|x_i)$ to each input feature vector for each semantic concept.

Our TRECVID 2008 concept detection approach builds on previous editions of the MediaMill semantic video search engine [35,36,39]. In addition, we draw inspiration from the

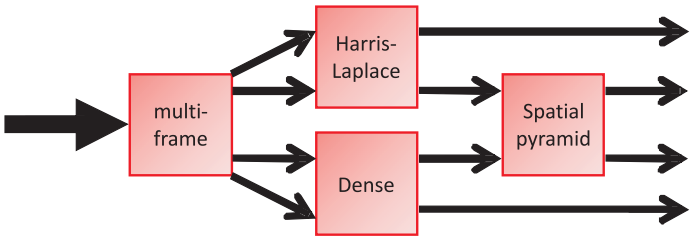

Figure 3: General scheme for spatio-temporal sampling of image regions, including temporal multi-frame selection, Harris-Laplace and dense point selection, and a spatial pyramid. Detail of Figure 2, using the conventions of Figure 1.

work of Schmid and her associates [24, 46], extending their work by putting special emphasis on video sampling strategies, keypoint-based color features [4, 33], codebook representations [8, 10], and kernel-based machine learning. We detail our generic concept detection scheme by presenting a component-wise decomposition. The components exploit a common architecture, with a standardized input-output model, to allow for semantic integration. The graphical conventions to describe the system architecture are indicated in Figure 1. Based on these conventions we follow the video data as they flow through the computational process, as summarized in the general scheme of our TRECVID 2008 concept detection approach in Figure 2, and detailed per component next.

2.1 Spatio-Temporal Sampling

The visual appearance of a semantic concept in video has a strong dependency on the spatio-temporal viewpoint under which it is recorded. Salient point methods [43] introduce robustness against viewpoint changes by selecting points, which can be recovered under different perspectives. Another solution is to simply use many points, which is achieved by dense sampling. Appearance variations caused by temporal effects are addressed by going beyond the key frame level. By taking more frames into account during analysis, it becomes possible to recognize concepts that are visible in the shot, but not necessarily in a single key frame. We summarize our spatio-temporal sampling approach in Figure 3.

Temporal multi-frame selection We demonstrated in [38] that a concept detection method that considers more visual content obtains higher performance over key frame-based methods. We attribute this to the fact that the content of a shot changes due to object and camera motion, and imperfect shot segmentation results. Therefore, we employ a multi-frame sampling strategy. To be precise, we sample a maximum of 4 additional frames distributed around the (middle) key frame of each shot.

Harris-Laplace point detector In order to determine salient points, Harris-Laplace relies on a Harris corner detector. By applying it on multiple scales, it is possible to select the characteristic scale of a local corner using the Laplacian operator [43]. Hence, for each corner the Harris-Laplace detector selects a scale-invariant point if the local image structure under a Laplacian operator has a stable maximum.

Dense point detector For concepts with many homogenous areas, like scenes, corners are often rare. Hence, for these concepts relying on a Harris-Laplace detector can be suboptimal. To counter the shortcoming of Harris-Laplace, random and dense sampling strategies have been proposed [6,19]. We employ dense sampling, which samples an image grid in a uniform fashion using a fixed pixel interval between regions. In our experiments we use an interval distance of 6 pixels and sample at multiple scales.

Spatial pyramid weighting Both Harris-Laplace and dense sampling give an equal weight to all keypoints, irrespective of their spatial location in the image frame. In order to overcome this limitation, Lazebnik *et al.* [20] suggest to repeatedly sample fixed subregions of an image, *e.g.* 1x1, 2x2, 4x4, *etc.*, and to aggregate the different resolutions into a so called spatial pyramid, which allows for region-specific weighting. Since every region is an image in itself, the spatial pyramid can be used in combination with both the Harris-Laplace point detector and dense point sampling. Reported results using concept detection experiments are not yet conclusive in the ideal spatial pyramid configuration, some claim 2x2 is sufficient [20], others suggest to include 1x3 also [24]. We use a spatial pyramid of 1x1, 2x2, and 1x3 regions in our experiments.

2.2 Visual Feature Extraction

In the previous section, we addressed the dependency of the visual appearance of semantic concepts in a video on the spatio-temporal viewpoint under which they are recorded. However, the lighting conditions during filming also play an important role. Burghouts and Geusebroek [4] analyzed the properties of color features under classes of illumination and viewing changes, such as viewpoint changes, light intensity changes, light direction changes, and light color changes. Van de Sande *et al.* [33] analyzed the properties of color features under classes of illumination changes within the diagonal model of illumination change, and specifically for data sets as considered within TRECVID. Another comparison of our invariant visual features, emphasizing discriminatory power, and efficiency of the feature representation is presented by Van Gemert *et al.* [10]. Here we summarize their main findings. We present an overview of the visual features used in Figure 4.

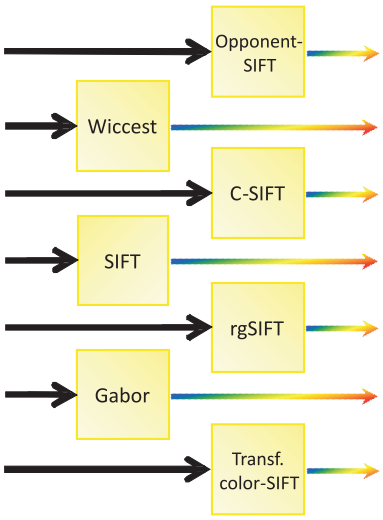

Figure 4: General scheme of the visual feature extraction methods used in our TRECVID 2008 experiments.

Wiccest Wiccest features [11] utilize natural image statistics to effectively model texture information. Texture is described by the distribution of edges in a certain image. Hence, a histogram of a Gaussian derivative filter is used to represent the edge statistics. It was shown in [13] that the complete range of image statistics in natural textures can be well modeled with an integrated Weibull distribution. In effect, reducing a histogram to just two Weibull parameters, see [10]. The Wiccest features for an image region consist of the Weibull parameters for the color invariant edges in the region. Thus, the two Weibull parameters for the x-edges and y-edges of the three color channels yield a 12-dimensional feature.

Gabor Gabor filters may be used to measure perceptual surface texture in an image [3]. Specifically, Gabor filters respond to regular patterns in a given orientation on a given scale and frequency, see [10]. In order to obtain an image region feature with Gabor filters we follow these three steps: 1) parameterize the Gabor filters 2) incorporate color invariance and 3) construct a histogram. First, the parameters of a Gabor filter consist of orientation, scale and frequency. We use four orientations, $0°, 45°, 90°, 135°$, and two (scale, frequency) pairs: (2.828, 0.720), (1.414, 2.094). Second, color responses are measured by filtering each color channel with a Gabor filter. The W color invariant is obtained by normalizing each Gabor filtered color channel by the inten-

sity. Finally, a histogram of 101 bins is constructed for each Gabor filtered color channel.

SIFT The SIFT feature proposed by Lowe [23] describes the local shape of a region using edge orientation histograms. The gradient of an image is shift-invariant: taking the derivative cancels out offsets [33]. Under light intensity changes, *i.e.* a scaling of the intensity channel, the gradient direction and the relative gradient magnitude remain the same. Because the SIFT feature is normalized, the gradient magnitude changes have no effect on the final feature. To compute SIFT features, we use the version described by Lowe [23].

OpponentSIFT OpponentSIFT describes all the channels in the opponent color space using SIFT features. The information in the O_3 channel is equal to the intensity information, while the other channels describe the color information in the image. The feature normalization, as effective in SIFT, cancels out any local changes in light intensity.

C-SIFT In the opponent color space, the O_1 and O_2 channels still contain some intensity information. To add invariance to shadow and shading effects, we have proposed the C-invariant [12] which eliminates the remaining intensity information from these channels. The C-SIFT feature uses the C invariant, which can be intuitively seen as the gradient (or derivative) for the normalized opponent color space $O1/I$ and $O2/I$. The I intensity channel remains unchanged. C-SIFT is known to be scale-invariant with respect to light intensity. Due to the local comparison of colors, as effective due to the gradient, the color component of the feature is robust to light color changes. See [4,33] for detailed evaluation.

rgSIFT For rgSIFT, features are added for the r and g chromaticity components of the normalized RGB color model, which is already scale-invariant [33]. In addition to the r and g channel, this feature also includes intensity. Because the SIFT feature uses derivatives of the input channels, the rgSIFT feature becomes shift-invariant as well. However, the color part of the feature is not invariant to changes in illumination color.

Transformed color SIFT For the transformed color SIFT, we normalize each RGB channel independently [33]. For every normalized channel, the SIFT feature is computed. The feature is scale-invariant, shift-invariant and invariant to light color changes and shift.

We compute the Wiccest and Gabor features on densely sampled image regions [10], the SIFT [23] and ColorSIFT [33] features are computed around salient points obtained from the Harris-Laplace detector and dense sampling. For all visual features we take several spatial pyramid configurations into account.

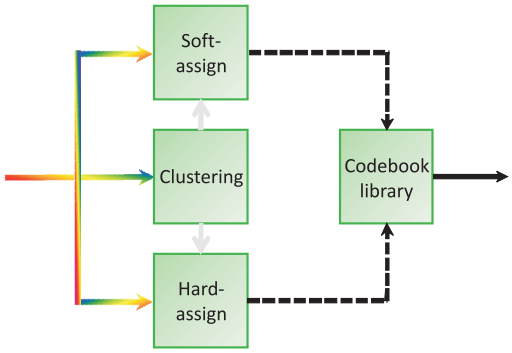

Figure 5: General scheme for transforming visual features into a codebook, where we distinguish between codebook construction using clustering and codeword assignment using soft and hard variants. We combine various codeword frequency distributions into a codebook library.

2.3 Codebook Transform

To avoid using all visual features in an image, while incorporating translation invariance and a robustness to noise, we follow the well known codebook approach, see *e.g.* [8, 10, 19, 21, 33, 34]. First, we assign visual features to discrete codewords predefined in a codebook. Then, we use the frequency distribution of the codewords as a compact feature vector representing an image frame. Two important variables in the codebook representation are *codebook construction* and *codeword assignment*. An extensive comparison of codebook representation variables is presented by Van Gemert *et al.* in [8,10]. Here we detail codebook construction using clustering and codeword assignment using hard and soft variants, following the scheme in Figure 5.

Clustering We employ two clustering methods: k-means and radius-based clustering. K-means partitions the visual feature space by minimizing the variance between a predefined number of k clusters. The advantage of the k-means algorithm is its simplicity. A disadvantage of k-means is its emphasis on clusters of dense areas in feature space. Hence, k-means does not spread clusters evenly throughout feature space. In effect biasing frequently occurring features. To overcome the limitation of k-means clustering, while maintaining efficiency, Jurie and Triggs [19] proposed radius-based clustering. The algorithm assigns visual features to the first cluster lying within a fixed radius of similarity r. Hence, the radius determines whether two visual features describe the same codeword. As an implementation of radius-based clustering we use Astrahans algorithm, see [10]. For both k-means and radius-based clustering we fix the visual codebook to a maximum of 4000 codewords.

Hard-assignment Given a codebook of codewords, obtained from clustering, the traditional codebook approach describes each feature by the single best representative codeword in the codebook, *i.e.* hard-assignment. Basically, an

image is represented by a histogram of codeword frequencies describing the probability density over codewords.

Soft-assignment In a recent paper [8], we show that the traditional codebook approach may be improved by using soft-assignment through kernel codebooks. A kernel codebook uses a kernel function to smooth the hard-assignment of image features to codewords. Out of the various forms of kernel-codebooks, we selected *codeword uncertainty* based on its empirical performance [8].

Codebook library Each of the possible sampling methods from Section 2.1 coupled with each visual feature extraction method from Section 2.2, a clustering method, and an assignment approach results in a separate visual codebook. An example is a codebook based on dense sampling of rgSIFT features in combination with k-means clustering and hard-assignment. We collect all possible codebook combinations in a visual codebook library. Naturally, the codebooks can be combined using various configurations. For simplicity, we employ equal weights in our experiments when combining codebooks to form a library.

2.4 Kernel-based Learning

Learning robust concept detectors from large-scale visual codebooks is typically achieved by kernel-based learning methods. From all kernel-based learning approaches on offer, the support vector machine is commonly regarded as a solid choice. We investigate the role of its parameters and how to select the optimal configuration for a concept, as detailed in Figure 6.

Support vector machine Similar to previous years, we use the support vector machine framework [44] for supervised learning of semantic concepts. Here we use the LIBSVM implementation [5] with probabilistic output [22, 28]. It is well known that the parameters of the support vector machine algorithm have a significant influence on concept detection performance [1, 25, 39, 45]. The only parameters of the support vector machine we optimize are C and the kernel function $K(\cdot)$. In order to handle imbalance in the number of positive versus negative training examples, we fix the weights of the positive and negative class by estimation from the class priors on training data. While the radial basis kernel function usually perform better than other kernels, it was recently shown by Zhang *et al.* [46] that in a codebook-approach to concept detection the earth movers distance [32] and χ^2 kernel are to be preferred. In general, we obtain good parameter settings for a support vector machine, by using an iterative search on both C and $K(\cdot)$.

Episode-constrained cross-validation From all parameters q we select the combination that yields the best average precision performance, yielding q^*. We measure performance of all parameter combinations and select the combination that yields the best performance. We use a 3-fold cross validation to prevent over-fitting of parameters. Rather than using regular cross-validation for support vector machine parameter optimization, we employ an *episode-constrained* cross-validation method, as this method is known to yield a less biased estimate of classifier performance [9].

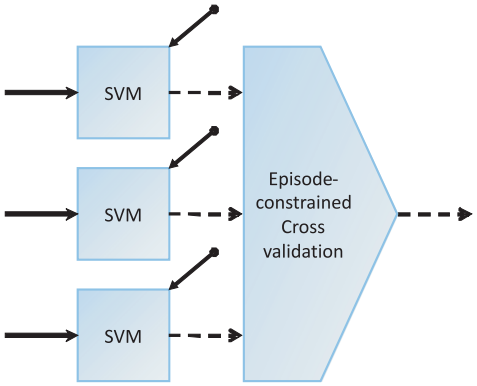

Figure 6: General scheme for kernel-based learning using support vector machines and episode-constrained cross-validation for parameters selection.

The result of the parameter search over q is the improved model $p(\omega_j|x_i, q^*)$, contracted to $p^*(\omega_j|x_i)$, which we use to fuse and to rank concept detection results.

2.5 Submitted Concept Detection Results

We investigated the contribution of each component discussed in Sections 2.1–2.4, emphasizing in particular the role of sampling, the value of color invariance, the influence of codebook construction, and the effectiveness of kernel-based learning parameters. In our experimental setup we used the TRECVID 2007 development set as a training set, and the TRECVID 2007 test set as a validation set. The ground truth used for learning and evaluation are a combination of the common annotation effort [2] and the ground truth provided by ICT-CAS [42]. The positive examples from both efforts were combined using an OR operation and subsequently verified manually. Based on our extensive experiments (data not shown) we arrived at the conclusion that a codebook library employing dense sampling and Harris-Laplace salient points in combination with a spatial pyramid, one of the three following (color) SIFT features: SIFT, OpponentSIFT, and transformed color SIFT, and a codebook representation based on k-means clustering and soft-assignment, is a powerful baseline for concept detection in video. This codebook library, consisting of 6 books in total, is our baseline. The baseline was not submitted for evaluation in the high-level feature extraction task, but post-TRECVID experiments indicates it would have obtained a mean infAP of 0.152. It was, however, the basis of all our TRECVID 2008 submissions. An overview of our submitted concept detection runs is depicted in Figure 7,

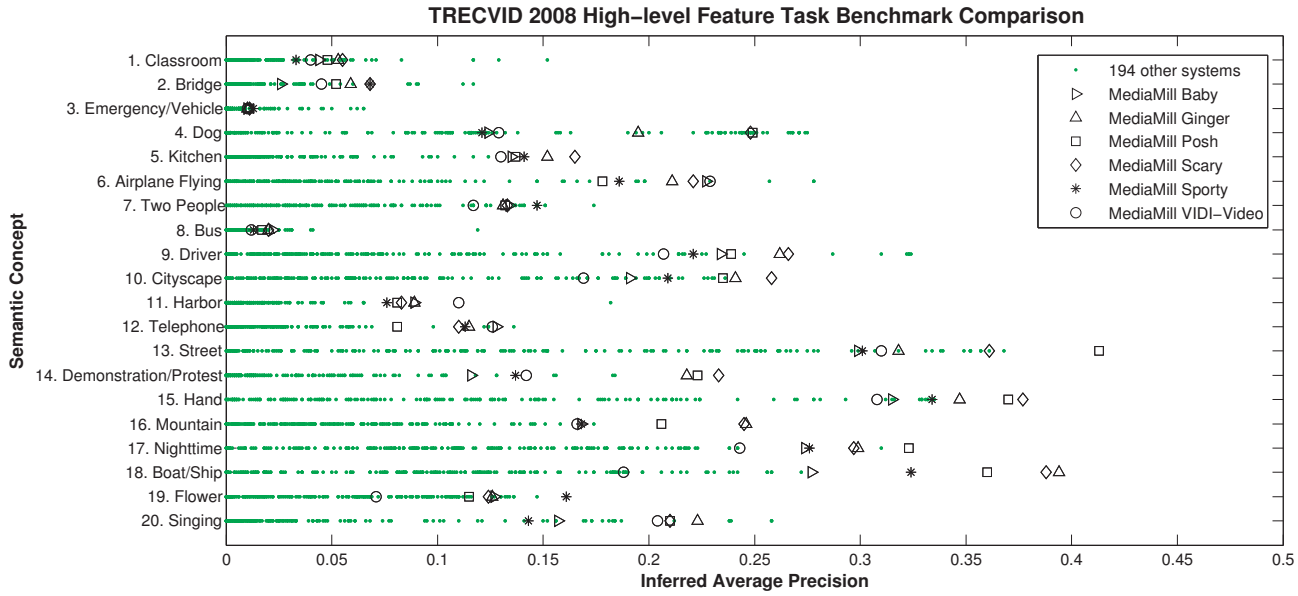

Figure 7: Comparison of MediaMill video concept detection experiments with present-day concept detection approaches in the TRECVID 2008 High-level Feature Task benchmark.

and detailed next.

Baby run The Baby run extends upon the baseline run by also including codebooks for the *rg*SIFT and C-SIFT features. This results in a codebook library of 10 books. This run achieved a mean infAP of 0.155. Indeed, only a small improvement over our baseline.

Sporty run The codebook library used in the Sporty run extends upon the Baby run by also including the Wiccest and Gabor features, and their early fusion. We apply the standard sequential forward selection feature selection method [18] on this large codebook library. This run achieves the overall highest infAP for the concept *Flower*, and has a mean infAP of 0.159.

VIDI-Video run This run is a cooperation between the University of Amsterdam and the University of Surrey. It uses multiple kernel learning [41] on the codebook library of the Baby run together with another codebook library based on SIFT only. The weights of the kernels, *i.e.*, the relative importance of the 2 codebook libraries, are learnt from the training data. It achieved a mean infAP of 0.148.

Ginger run The Ginger run extends the codebook library of 6 books from the baseline run to the temporal domain. For every shot, up to 5 frames are processed, and the results are averaged. This run achieves the overall highest infAP for the concepts *Mountain* and *Boat/Ship*, and has a mean infAP of 0.185.

Posh run The Posh run is based on a codebook library in a temporal setting. The codebooks to use per concept were chosen on the basis of hold-out performance on the validation set. There were 3 sets of codebooks to chose from, together with the method for temporal aggregation, which could be either the average or the maximum concept likelihood. This run achieves the overall highest infAP for the concepts *Street* and *Nighttime*, and has a mean infAP of 0.184.

Scary run The Scary run applies the standard sequential forward selection feature selection method on several codebook libraries, all of which have been applied spatio-temporally to up to 5 frames per shot. This run achieved the overall highest mean infAP in the TRECVID2008 benchmark (0.194), with the overall highest infAP for 4 concepts: *Kitchen*, *Cityscape*, *Demonstration or protest*, and *Hand*.

2.6 57 Robust Concept Detectors

In order to have as many concept detectors as possible available for video retrieval, we have employed a graceful degradation approach in previous TRECVID editions [35, 36]. This has resulted in lexicons containing close to 600 concept detectors, albeit with mixed performance. In contrast to previous TRECVID editions, we aim for a small but robust lexicon of concept detectors this year. To that end we have employed our baseline codebook library on the concept sets of TRECVID 2008 (20 concepts), TRECVID2007 (36 concepts) and an additional black/white detector. Comparative experiments with our baseline codebook library and last years approach indicates a performance increase of 100%. Hence, the 2008 MediaMill semantic video search engine includes 57 robust concept detectors and a powerful codebook library for retrieval.

3 Automatic Video Retrieval

The TRECVID automatic search task has, over the previous years, shown that topic type directly relates to the best type of information for querying. Specifically, named entity queries can best be answered using speech search. Furthermore, if a robust concept detector is available for a query, detector-based search should provide reliable results. These principles drive the query-dependent, predictive video retrieval strategy of the MediaMill 2008 automatic video search system.

3.1 Predictive Video Retrieval

Inspired by work in query-difficulty prediction, we predict which of the three retrieval channels (speech, detector, or example-based search) should be trusted to provide the best results for a given topic. The top search results from the trusted retrieval channel are used as the basis set of search results, and are then reranked with information from the remaining two retrieval channels. By trusting one, and only one, information channel, we reduce the need for parameter estimation associated with fusion approaches that consider all results from all channels. In addition, the prediction framework allows for a query-class independent approach that we expect will generalize well to include other channels of information.

3.1.1 Prediction Features

We found, after evaluating topics from previous TRECVID benchmarks, that two topic features were especially good indicators of the best retrieval channel: 1) named entity occurrence, and 2) exact matches to 'informative' detectors.

Named entities were extracted from the topics using the Stanford Named Entity tagger [7]. Binary occurrence of named entities was used as a feature, so either a topic contained at least one named entity, or it did not.

To find exact detector matches, both the detectors and the topics were linked to WordNet (noun) synsets. If a topic synset directly matched a detector synset, this was considered a direct match. To determine detector informativeness, the information content was calculated using Resnik's measure of information content [29]. If a matched detector had an information content higher than 5, it was considered informative. Resnik's measure is corpus-based, we use a very large Google-based corpus kindly provided to us by Haubold and Natsev [14]. This allowed us to gain relatively accurate, 'real world' frequency counts than as compared to more traditional (and smaller) news corpora. As a result, for a topic such as *find shots of people with a body of water* a general detector such as *people* would not be considered informative, as opposed to a more specific detector such as *waterscape*.

3.1.2 Multimodal Fusion and Reranking

We experimented with combining the evidence from multiple retrieval channels using a trust-based framework. This is a three-step procedure, as follows. First, given the three channels, namely, speech, detector, and example-based search, we select the trusted channel on which the re-ranking of the result list will be based. If a named entity is detected, the speech channel is trusted. If an informative detector directly matches the query, then we trust the detector channel. Otherwise we trust the example-based search results. Second, we truncate the trusted result list to the top 1000 results. The secondary result lists, any shots that do not occur in the trusted result list are removed, considering the top 1000 results in the list. Third, we combine the result lists using rank-based fusion. Any results that are contained in more than one list will be boosted.

3.1.3 Retrieval Channel Implementation

Our predictive retrieval framework is built on search results from three retrieval channels: speech, detector, and example-based search. These are implemented as follows:

Speech-based search Our speech based search approach is similar to that of last year, incorporating both the original Dutch automatic speech transcripts donated by the University of Twente [15], and the automatic machine translation provided by Queen Mary, University of London. At retrieval time, each topic statement was automatically translated into Dutch using the online translation tool `http://translate.google.com`, allowing a search on the machine-translated transcripts with the original (English) topic text, and a search on transcripts from automatic speech recognition using the translated Dutch topic text. The two resulting ranked lists were then combined to form a single list of transcript-based search results. To compensate for the temporal mismatch between the audio and the visual channels, we used our temporal redundancy approach [16]. To summarize this approach, the transcript of each shot is expanded with the transcripts from temporally adjacent shots, where the words of the transcripts are weighted according to their distance from the central shot.

Detector-based search The detector-based search, using our lexicon of 57 robust concept detectors, consisted of two main steps: 1) concept selection and 2) detector combination. We evaluated a number of concept selection approaches using a benchmark set of query-to-concept mappings, adapted from [17] to the new lexicon. We found an example-based concept selection strategy to deliver the best concept selection results. The final concept selection method used for automatic search was to average the score for a concept detector on the provided topic video examples, and select concepts that scored over a threshold (a threshold of 0.5 was used, as this gave the best results). As for the combination of multiple selected concepts for a topic,

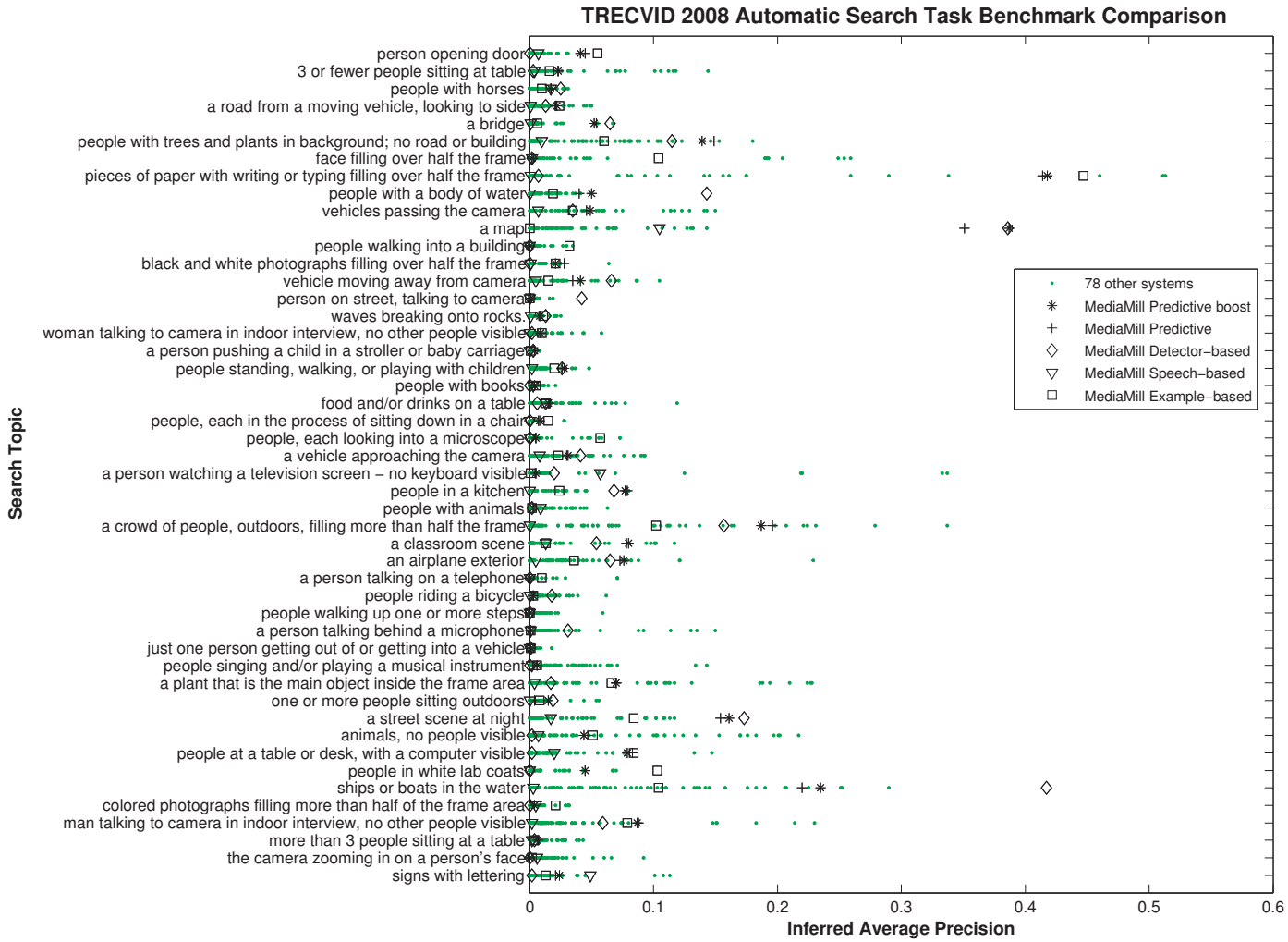

Figure 8: Comparison of MediaMill automatic video search experiments with present-day automatic search approaches in the TRECVID 2008 benchmark.

this was done by simply taking the product of the raw selected detector scores for each shot as its retrieval score. No extra normalization or parametrization was done, nor were concepts weighted according to their computed score for the examples. Rather, we used the triangulation of concept detector scores to provide information as to the relevance of a shot to a query.

Example-based search This is the first year that we included example-based search. Similarly to [26], we treat example-based search as an on-the-fly concept learning problem, with the provided topic video examples as positive examples, and randomly selected shots from the test collection as pseudo-negative examples. Spatio-temporal sampling of interest regions, visual feature extraction, codebook transform, and kernel-based learning were done as described in Section 2.5. The resulting model was applied to the shots in the test collection, shots were ranked according to the probabilistic output score of the support vector machine.

3.1.4 Automatic Search Results

We submitted four official runs to the TRECVID automatic search task. These runs include two of the three retrieval channel searches (the required text and visual baselines), and two combination runs with slightly different reranking schemes. For the sake of completeness, we also include the remaining retrieval channel, *i.e.* example-based, as a supplementary run (not submitted to TRECVID) in our results and analysis.

Due to a lack of named entity queries amongst the TRECVID topics, the speech baseline (**UvA-MM-6**) had the lowest overall mean infAP of the 4 runs. The visual baseline run (**UvA-MM-5**) was done using detector-only search, and performed especially well for topics where one or more strongly related detectors were available. The two combination runs, (**UvA-MM-4** and **UvA-MM-3**), created using our novel predictive video retrieval framework, performed consistently well over a range of topics. In terms of mean infAP our predictive video retrieval approach was

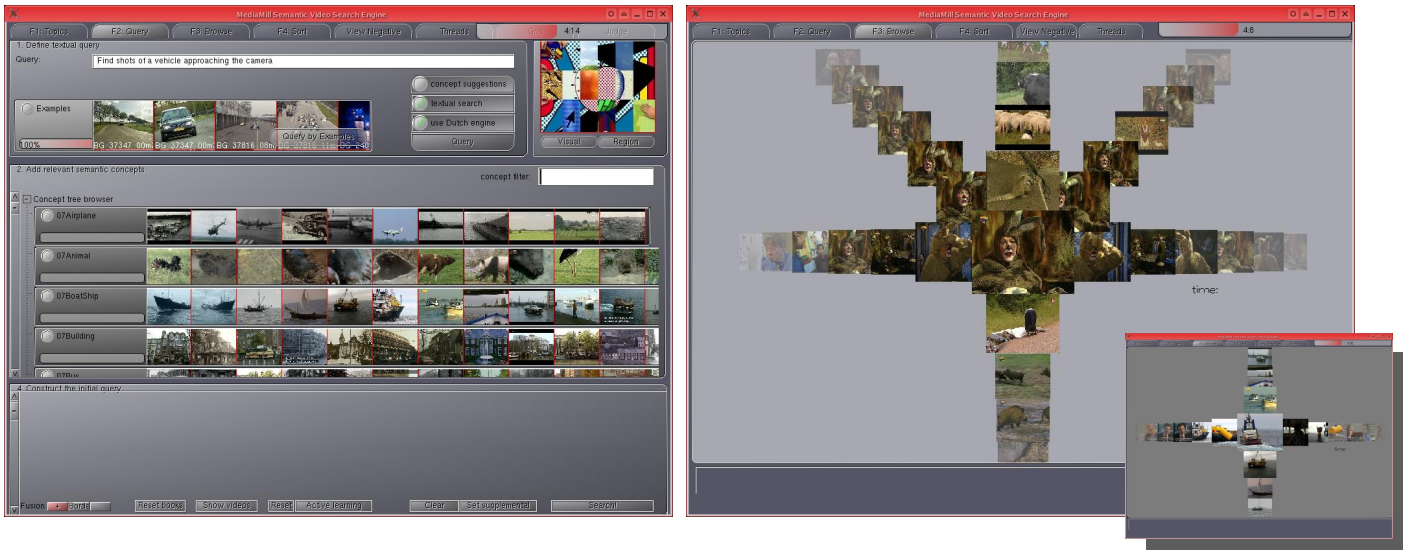

Figure 9: Screenshots of the MediaMill semantic video search engine with its query interface (left), its ForkBrowser [31] (right), and its CrossBrowser [40] (inset).

one of the top performers in this year's automatic search task. When the correct channel was accurately predicted, the final combined results either approached those of the best channel or exceeded them, showing that the secondary channels can provide valuable (re)ranking information.

Figure 8 provides a topic-level summary of the performance of the MediaMill automatic search runs. We see that speech generally gave very low performance in general, due to a lack of named entity topics. It performed fairly well for a few topics, namely *shots with a map*, *television screen without keyboard*, and *signs with lettering*. This indicates that speech can be of help for some queries, even when they do not contain named entities. Example-based search gave higher performance. As expected it did well when no directly related detectors were present, *e.g. person opening a door* and *people in white lab coats*. Detector-based search performed very well for a number of topics, providing the highest overall infAP scores multiple times. This search performed very well when one highly related detector was used for search - for example *shots of a map*, which triggered only the *graphical map* detector for search. In addition, it also performed especially well when multiple related detectors were selected for search — for example *people where a body of water can be seen* which triggered the *outdoor*, *sky*, *person*, and *waterscape* detectors. Keeping in mind that our detector combination is based on a simple probabilistic fusion strategy, we attribute the success of the detector-based retrieval approach to a number of factors: 1) high-quality concept selection, 2) robust concept detectors, and 3) accurate estimation of detector reliability and collection frequency by the detector itself. The two predictive search runs performed best overall, with no significant difference between the two. Compared to the individual retrieval channels, they do not always give the best retrieval results for a particular topic. However, they give consistently good results over a number of topics, resulting in a better mean infAP than any of the retrieval channel searches.

The predictive search strategy was influenced by prediction accuracy: the best performing channel was not always correctly selected. In fact, the best performing channel was selected correctly for exactly half the topics. However, many of the incorrect predictions occurred for topics where infAP scores were very low, so it can be argued that for these topics none of the channels could be trusted. When considering the 20 topics where at least one of the channels had an infAP higher than 0.05, the correct channel was predicted correctly 75% of the time. When the correct channel was predicted, performance either increased or decreased slightly. When the incorrect channel was trusted, the infAP of the prediction runs was almost invariably higher than the infAP of the (incorrectly) trusted channel.

Preliminary experiments had indicated that the detector channel generally gave better results than the other two retrieval channels, and should therefore be given an extra boost during retrieval. The results show a slight improvement with this approach, but not significantly so. We plan to investigate more refined weighting for reranking in future work.

4 Interactive Video Retrieval

From the past five years of TRECVID experience we have learned that the ideal interactive video retrieval approach depends on many factors, such as the type of query, the query method, the browsing interface, the interaction scheme, and the level of expertise of the user. Moreover, when search topics are diverse, it is hard to predict which combination of factors yields optimal performance. Therefore, the MediaMill video search engine has traditionally offered multiple query methods in an integrated browse en-

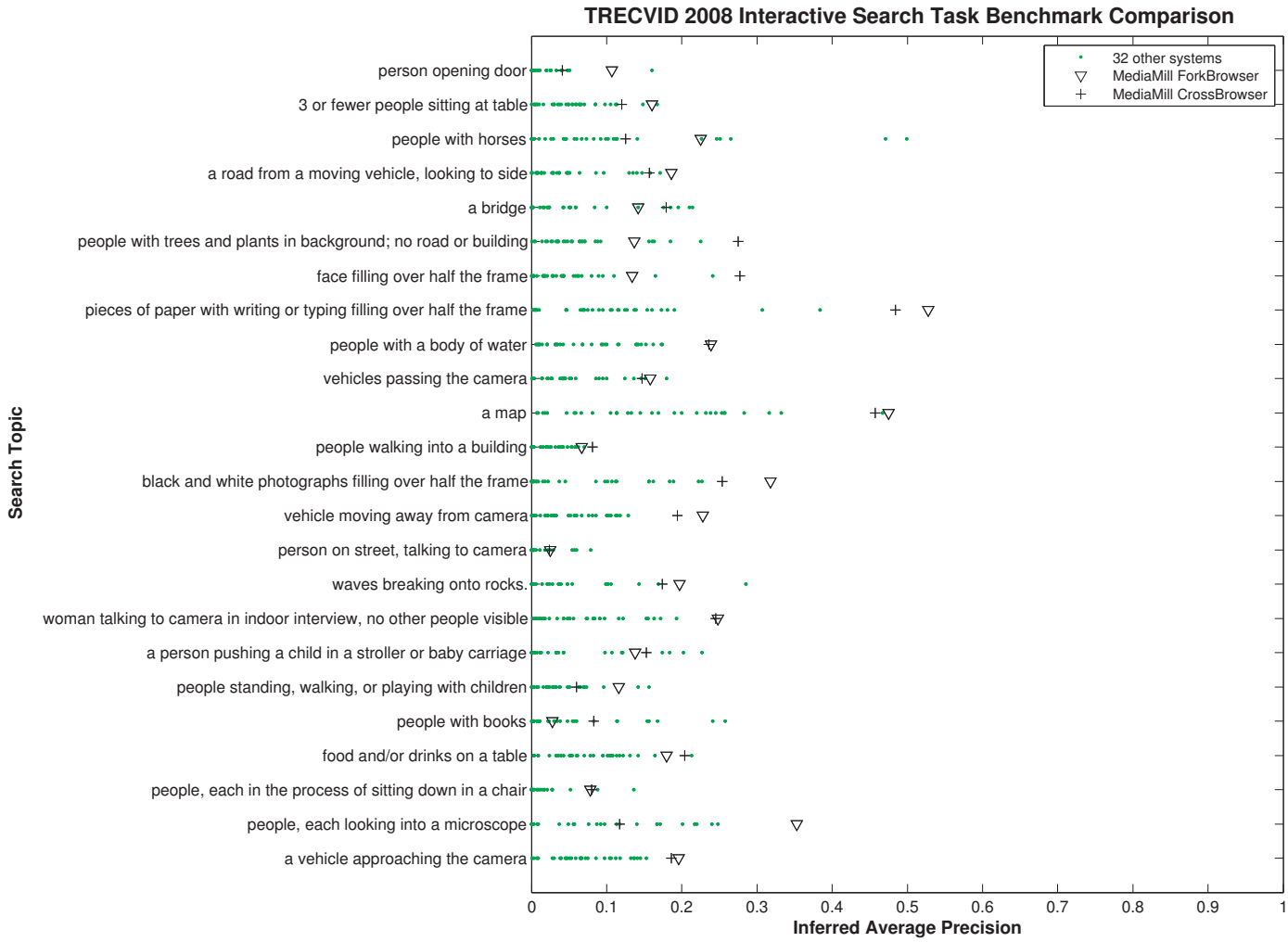

Figure 10: Comparison of MediaMill interactive video search experiments with present-day interactive video search engines in the TRECVID 2008 benchmark.

vironment. While this gives the user complete control over which strategy to use for which topic, it often causes the user to select a sub-optimal strategy. In order to alleviate this problem, our TRECVID 2008 experiments focus on supporting the user in determining the utility of the retrieval strategy and to guide her on the path to a correct set of results. Our contribution is twofold; first we introduce a novel PreviewBrowser, which helps the user to either quickly determine that results are not valid at all, or to help find a starting point within the selected results. Second, we introduce a novel active learning strategy, based on passive sampling of user browsing behavior, for those topics that have no valid starting points in the video collection.

Threads as basis for navigation The basic building block behind the browsers of the MediaMill semantic video search engine is the thread; a linked sequence of shots in a specified order, based upon an aspect of their content [30]. These threads span the video archive in several ways. For example, time threads span the temporal similarity between shots, visual threads span the visual similarity between shots, a query thread spans the similarity between a shot and a user-imposed query, and history threads span the navigation path the user follows.

Thread visualization The MediaMill video search engine allows the user to choose between two modes for thread visualization. Each mode starts with a query thread as the basic entry point for the visualization. The first visualization, the CrossBrowser then shows the query thread and the time thread in a cross formation. This visualization is most efficient for topics where a single concept query is sufficient for solving a topic [36, 40]. The second visualization, the ForkBrowser, provides the user with two extra diagonal threads, and a history thread. The ForkBrowser is more efficient in handling complex queries where no direct mapping between available concept detectors is possible [31].

Guiding the user to results In order to guide the user in finding results, we introduce the PreviewBrowser. This

browser helps the user to quickly determine the validity of a chosen set of results, by visualizing a large set of results from a single thread at once. To keep the user experience as seamless as possible this is done without changing to another browser. The user is then able to either continue browsing the thread, or change the set of results by changing the query.

When multiple searches yield limited effect, a different strategy is needed to find results. For this scenario, the system continuously monitors user behavior and uses this information on-demand to generate a new set of results. It does so by using a real-time support vector machine. The active learning is performed on the entire collection of positive and negative examples based on what the user selected *and* what the user viewed. This results in a new thread which is available to the user for visualization.

Both methods yield a new thread with possible results for the same topic. A possible downside of such threads is that they will contain the same shots as in a previously visited thread. To further guide the user to correct results we extended the CrossBrowser and ForkBrowser to automatically hide previously seen results after a while. This decision is based on the user monitoring strategy as also employed in the active learning algorithm. This ensures that the users only see new results, and do not see results they have already seen over and over again.

4.1 Interactive Search Results

We submitted two runs for interactive search. Both interactive runs were performed by expert users, one used the ForkBrowser (**UvA-MM1** run), and another one used the CrossBrowser (**UvA-MM2** run) for retrieval. The two users had access to the real-time active learning approach as well as the PreviewBrowser. We present an overview of achieved results per topic in Figure 10.

Analysis of logging data indicates that the users employed a variety of strategies to retrieve results. In particular, we observe the following topic-dependent patterns:

- **Topics maps to an available concept detector:** When relevant concept detectors are available for a topic, these are taken as the entry point for search by both users. For example, the users selected the *Face* detector for the topic *a person's face filling more than half the frame.*

- **Topic asks for explicit motion:** When an explicit form of motion is requested by the search topic, the best strategy to validate the presence of the motion within individual shots is to display animated key frames. For example, the expert users select the *Car* detector as an entry point in the following topics: *road taken from a moving vehicle, looking to the side*, *a vehicle moving away from the camera* and *a vehicle approaching the camera.* This yields the same set of results for all topics. By watching the individual shots, and looking for the requested motion pattern, valid results are retrieved and selected.

- **Topic examples have small variability in appearance:** For topic examples that have a small variability in their visual appearance, such as a *map* or *piece(s) of paper with writing on it* the users employed the real-time active learning approach on the provided video examples. Here, the system automatically selects the center frame from each video example as a positive example, and automatically selects 50 pseudo-negative key frame examples from the video collection. The optimal strategy then seems to be to quickly validate the resulting 'topic detector' with the PreviewBrowser, which is able to select large batches of results in few keystrokes.

- **Topic asks for complex information need:** Often topics express a complex information need, which combines all of the above. For example, the best result for topic *one or more people looking into a microscope* was obtained by 1) using visual similarity to the examples to gather a few initial results. The best performing expert user then alternated between 2) using the ForkBrowser to significantly expand this set, 3) using animated key frames to verify that the action actually occurred, and then 4) using active learning to find even more results based on the current selection and automatically selected negatives.

- **Topic with a limited number of initial results:** For some topics it happened that the users were able to find some relevant shots using any kind of entry point, but were not able to retrieve more relevant results. In this case the users generated new retrieval results based on real-time active learning on the previously selected shots. In most cases this provided the users with, up to then, unseen but correct retrieval results.

Overall our approach is on-par with the state of the art in interactive video retrieval, yielding the highest infAP scores for 12 out of 24 topics. This indicates that our multi-strategy approach combined with robust concept detectors and active learning yields good search results.

5 Lessons Learned

TRECVID continues to be a rewarding experience in gaining insight in the difficult problem of concept-based video retrieval. The 2008 edition has been our most successful participation to date resulting in top ranking for both concept detection and interactive search and a runner-up ranking for automatic retrieval, see Figure 11 for an overview. To conclude this paper we highlight our most important lessons learned:

- *Spatial-temporal processing improves classification accuracy [38];*

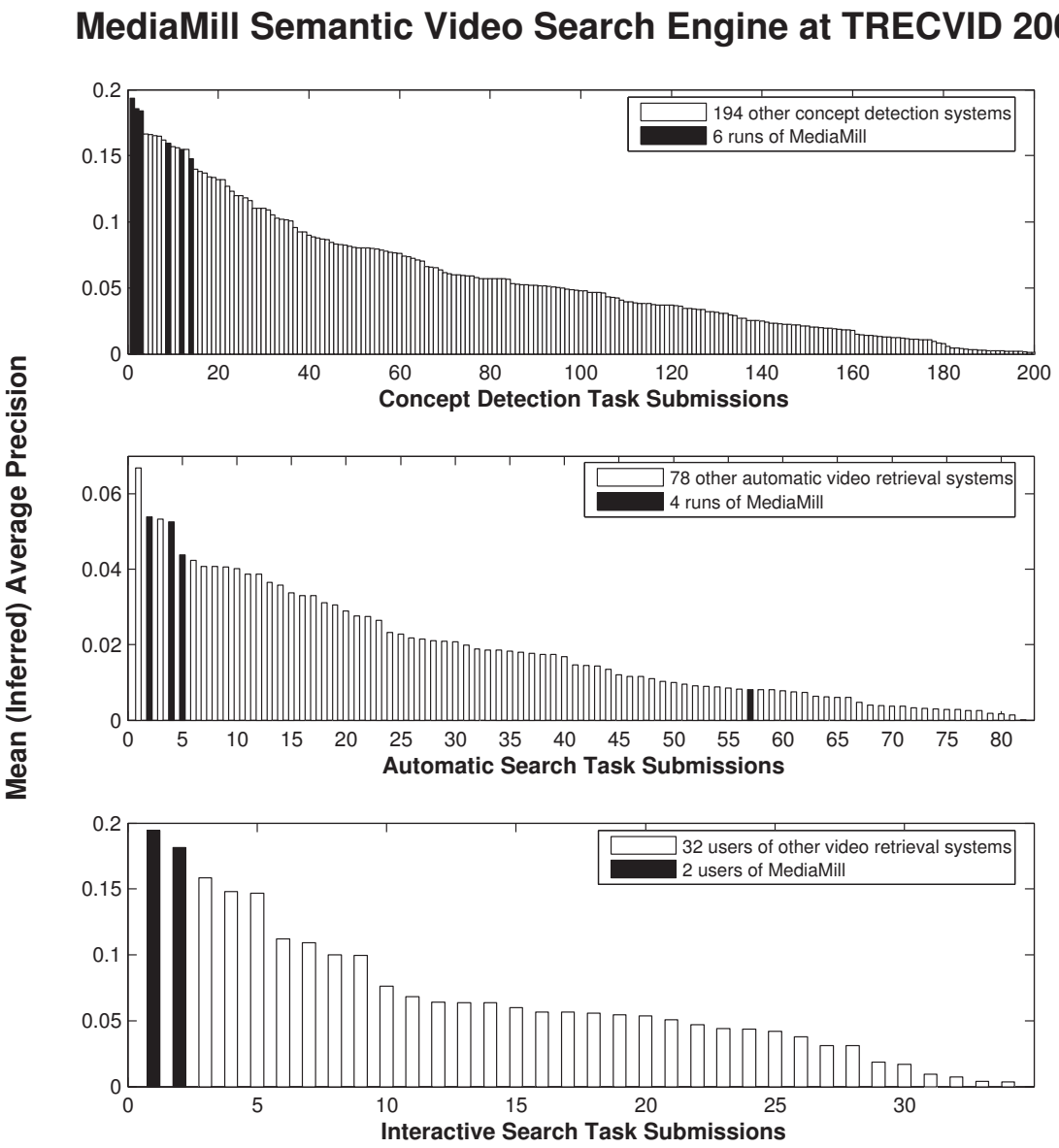

Figure 11: Overview of all 2008 TRECVID benchmark tasks in which MediaMill participated. From top to bottom: concept detection, automatic search, and interactive search, runs ranked according to mean inferred average precision.

- *The addition of ColorSIFT, with different levels of invariance to changes in illumination conditions, on top of intensity SIFT improves concept detection accuracy [4, 33];*
- *Kernel codebooks suffer less from the curse of dimensionality and give better performance in larger data sets. [8];*
- *A kernel codebook outperforms the traditional codebook model over several feature dimensions, codebook sizes, and data sets [8];*
- *The codebook library proves to be a valuable addition over a single codebook;*
- *Good retrieval ingredients matter;*
- *The more sources of information, the better the retrieval performance;*
- *Topic examples are valuable for automatic retrieval (but can we expect users to give them?);*
- *Simple fusion techniques suffice when concept detectors are robust and well selected;*
- *Predictive combination of retrieval channels pays off;*
- *Multi-thread browsing with the ForkBrowser, combined with quick result visualization of single threads, yields a fast browsing experience which is suited for a broad range of topics;*

- *Monitoring retrieval behavior of users combined with real time active learning help the user find new results effectively and efficiently;*

Acknowledgments

We thank Vivek Edatan Puthiya Veettil and Fangbin Liu for annotation help. This research is sponsored by the European VIDI-Video, IST-CHORUS, and MultiMatch projects, the BSIK MultimediaN project, and the NWO MuNCH and QASSIR projects. The authors are grateful to NIST and the TRECVID coordinators for the benchmark organization effort.

References

[1] A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. R. Naphade, A. P. Natsev, C. Neti, H. J. Nock, J. R. Smith, B. L. Tseng, Y. Wu, and D. Zhang. IBM research TRECVID-2003 video retrieval system. In *Proc. TRECVID Workshop*, Gaithersburg, USA, 2003.

[2] S. Ayache and G. Quénot. Video corpus annotation using active learning. In *Proc. ECIR*, pages 187–198, Glasgow, Scotland, 2008.

[3] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. PAMI*, 12(1):55–73, 1990.

[4] G. J. Burghouts and J. M. Geusebroek. Performance evaluation of local color invariants. *CVIU*, 2009. In press.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scence categories. In *Proc. IEEE CVPR*, pages 524–531, 2005.

[7] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. ACL*, pages 363–370, Ann Arbor, USA, 2005.

[8] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, Marseille, France, 2008.

[9] J. C. van Gemert, C. G. M. Snoek, C. Veenman, and A. W. M. Smeulders. The influence of cross-validation on video classification performance. In *Proc. ACM Multimedia*, pages 695–698, Santa Barbara, USA, 2006.

[10] J. C. van Gemert, C. G. M. Snoek, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek. Comparing compact codebooks for visual categorization. *CVIU*, 2009. Submitted.

[11] J.-M. Geusebroek. Compact object descriptors from local colour invariant histograms. In *BMVC*, Edinburgh, UK, 2006.

[12] J. M. Geusebroek, R. Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Trans. PAMI*, 23(12):1338–1350, 2001.

[13] J. M. Geusebroek and A. W. M. Smeulders. A six-stimulus theory for stochastic texture. *IJCV*, 62(1/2):7–16, 2005.

[14] A. Haubold and A. P. Natsev. Web-based information content and its application to concept-based video retrieval. In *Proc. ACM CIVR*, pages 437–446, Niagara Falls, Canada, 2008.

[15] M. Huijbregts, R. Ordelman, and F. M. G. de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proc. SAMT*, volume 4816 of *LNCS*, pages 78–90, Berlin, 2007. Springer-Verlag.

[16] B. Huurnink and M. de Rijke. Exploiting redundancy in cross-channel video retrieval. In *Proc. ACM SIGMM MIR Workshop*, pages 177–186, Augsburg, Germany, 2007.

[17] B. Huurnink, K. Hofmann, and M. de Rijke. Assessing concept selection for video retrieval. In *Proc. ACM MIR*, pages 459–466, Vancouver, Canada, 2008.

[18] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. PAMI*, 22(1):4–37, 2000.

[19] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. IEEE ICCV*, pages 604–610, 2005.

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE CVPR*, volume 2, pages 2169–2178, New York, USA, 2006.

[21] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

[22] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[24] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition, 2007. Visual Recognition Challange workshop, in conjunction with ICCV.

[25] M. R. Naphade. On supervision and statistical learning for semantic multimedia analysis. *Journal of Visual Communication and Image Representation*, 15(3):348–369, 2004.

[26] A. P. Natsev, M. R. Naphade, and J. Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *Proc. ACM Multimedia*, pages 598–607, Singapore, 2005.

[27] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. In *Proc. TRECVID Workshop*, Gaithersburg, USA, 2004.

[28] J. C. Platt. Probabilities for SV machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[29] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conference on Artificial Intelligence*, pages 448–453, Montréal, Canada, 1995.

[30] O. de Rooij, C. G. M. Snoek, and M. Worring. Query on demand video browsing. In *Proc. ACM Multimedia*, pages 811–814, Augsburg, Germany, 2007.

[31] O. de Rooij, C. G. M. Snoek, and M. Worring. Balancing thread based navigation for targeted video search. In *Proc. ACM CIVR*, pages 485–494, Niagara Falls, Canada, 2008.

[32] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 40(2):99–121, 2000.

[33] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *Proc. IEEE CVPR*, Anchorage, Alaska, 2008.

[34] J. Sivic and A. Zisserman. Efficient visual search for objects in videos. *Proc. IEEE*, 96(4):548–566, 2008.

[35] C. G. M. Snoek, I. Everts, J. C. van Gemert, J.-M. Geusebroek, B. Huurnink, D. C. Koelma, M. van Liempt, O. de Rooij, K. E. A. van de Sande, A. W. M. Smeulders, J. R. R. Uijlings, and M. Worring. The MediaMill TRECVID 2007 semantic video search engine, 2007.

[36] C. G. M. Snoek, J. C. van Gemert, T. Gevers, B. Huurnink, D. C. Koelma, M. van Liempt, O. de Rooij, K. E. A. van de Sande, F. J. Seinstra, A. W. M. Smeulders, A. H. C. Thean, C. J. Veenman, and M. Worring. The MediaMill TRECVID 2006 semantic video search engine, 2006.

[37] C. G. M. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2009. Submitted.

[38] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, and F. J. Seinstra. On the surplus value of semantic video analysis beyond the key frame. In *Proc. IEEE ICME*, Amsterdam, The Netherlands, 2005.

[39] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. M. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Trans. PAMI*, 28(10):1678–1689, 2006.

[40] C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. *IEEE Trans. MM*, 9(2):280–292, 2007.

[41] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[42] S. Tang et al. TRECVID 2008 high-level feature extraction by MCG-ICT-CAS. In *Proc. TRECVID Workshop*, Gaithersburg, USA, 2008.

[43] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

[44] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 2nd edition, 2000.

[45] D. Wang, X. Liu, L. Luo, J. Li, and B. Zhang. Video Diver: generic video indexing with diverse features. In *Proc. ACM SIGMM MIR Workshop*, pages 61–70, Augsburg, Germany, 2007.

[46] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.