AXES at TRECVid 2012: KIS, INS, and MED

INS+KIS: Robin Aly[1], Kevin McGuinness[2], Shu Chen[2], Noel E. O'Conner[2], Ken Chatfield[3], Omkar M. Parkhi[3], Relja Arandjelović[3], Andrew Zisserman[3], Basura Fernando[4], Tinne Tuytelaars[4],

MED: Dan Oneata[6], Matthijs Douze[6], Jérôme Revaud[6], Jochen Schwenninger[5], Danila Potapov[6], Heng Wang[6], Zaid Harchaoui[6], Jakob Verbeek[6], Cordelia Schmid[6]

[1]University of Twente, [2]Dublin City University, [3]Oxford University, [4]KU Leuven, [5]Fraunhofer Sankt Augustin, [6]INRIA Grenoble

***Abstract*— The AXES project participated in the interactive instance search task (INS), the known-item search task (KIS), and the multimedia event detection task (MED) for TRECVid 2012. As in our TRECVid 2011 system, we used nearly identical search systems and user interfaces for both INS and KIS. Our interactive INS and KIS systems focused this year on using classifiers trained at query time with positive examples collected from external search engines. Participants in our KIS experiments were media professionals from the BBC; our INS experiments were carried out by students and researchers at Dublin City University. We performed comparatively well in both experiments. Our best KIS run found 13 of the 25 topics, and our best INS runs outperformed all other submitted runs in terms of P@100. For MED, the system presented was based on a minimal number of low-level descriptors, which we chose to be as large as computationally feasible. These descriptors are aggregated to produce high-dimensional video-level signatures, which are used to train a set of linear classifiers. Our MED system achieved the second-best score of all submitted runs in the main track, and best score in the ad-hoc track, suggesting that a simple system based on state-of-the-art low-level descriptors can give relatively high performance. This paper describes in detail our KIS, INS, and MED systems and the results and findings of our experiments.**

I. Introduction

This paper describes the second participation of the EU Project AXES at TRECVid [28]. The AXES project aims to connect users and content from large multimedia archives by means of technology. The project partners involved in this year's participation (with references to earlier participations) were: 1) Dublin City University (CLARITY: Center for Sensor Web Technologies) [11], [32]; 2) University of Twente [1], [2]; 3) Oxford University [30]; 4) KU Leuven [17]; 5) INRIA Lear [13], [5]

Since AXES is about bringing users, technology, and content together, we conducted interactive user experiments in the instance search (INS) and the known-item search (KIS) tasks. Additionally, because users often search for events in multimedia content, we participated in the multimedia event detection task (MED). For INS and KIS, the collaboration with the BBC, as an industry partner in AXES, gave us a unique opportunity to conduct experiments in a realistic environment with professional users. We refer the reader to [19] for a description of these tasks.

For the Multimedia Event Detection task we developed an approach based on four low-level descriptors: SIFT features [**?**] for image content, trajectories+MBH features [31] for motion, MFCC for audio and optical character recognition with MSER regions [14] for text. The local low-level audio-visual descriptors were aggregated into high dimensional video-level signatures using Fisher vectors [23]. A combination of low-level specific linear classifiers allows to obtain event scores.

This paper is structured as follows: Section II describes the methods and the system we developed for this year's INS and KIS participation, including the system architecture and the user interface. Section III describes our participation in the MED task. Section IV describes the experiments and discusses the results and findings. Section V summarizes this paper.

II. INS and KIS Participation

In this section we describe the system we developed for this year's INS and KIS participation. We used a service oriented architecture for this year's TRECVid participation, see [15] for details. The central component of the system is our LIMAS service that merges search results from several retrieval services that each produce a confidence score for each shot whether it is relevant. The scores are then fused (combined) according to a single confidence score, producing a list of retrieval units (videos or shots). This list is then send back to the user interface. In the following, we first describe the individual retrieval services, the fusion scheme we used, and the employed user interface.

A. Speech and Metadata

We stored the available text for each shot in a text index. Both our KIS and INS search engines used ASR data; we used the provided ASR for KIS and extracted custom ASR for the INS task. We also used five metadata fields from the provided metadata XML files for the KIS task: title, description, keywords, subject, and uploader. At query time, the standard Lucene retrieval function was used to calculate a confidence retrieval score for each retrieval unit if the query contained any text terms. We used Lucene version 3.1.2 [29] in our experiments.

B. On the fly Object/Image Category Retrieval

The aim of the category recognition system is to quickly retrieve key-frames which contain queried general classes of objects (e.g. all *cars* in a dataset, or all examples of *gothic architecture*). The query is specified by entering a text term which is used to train a model for the query on-the-fly.

The system is based on the on the fly training of a discriminative classifier, and so in addition to the feature vectors for the dataset itself, features for negative and positive training data related to the target query are required. The negative training data is also sourced during the offline stage, and is fixed for all queries. Features are computed for $\sim 1,000$ images downloaded from Google Image search using the publicly available API and the search term 'things' and 'photos'.

The features for the positive training data are computed on the fly after the user has made a query, and again are sourced from Google Image search, which is used to translate the user's textual query into a set of images. We use the top-ranking ~ 200 images from a search for the query term entered by the user. Features are extracted from these images in the same way, and a linear SVM is trained against the pool of negative training features computed during the offline stage. The output of the classifier is a w vector of the same dimensions as the features, and the dot product between this and all features in the target dataset is then taken to provide an output score for each image. Finally, this score is used to rank the images in descending order of relevance to the entered query. Figure 1 shows some sample results. The system is described in detail in [6].

C. On the fly Face Retrieval

The aim of the face retrieval system is to retrieve key-frames based on the faces they contain. Given a query, a discriminative classifier is learnt using images containing faces downloaded from Google image search for that query.

To achieve real time performance, it is essential to perform as much of the processing in advance. In the offline processing faces are detected in every frame of every video and faces of same person are linked together within a shot to form face tracks. At the same time, nine facial features such as eyes, nose, mouth etc. are located within every face detection using pictorial structure based method [10], [9]. These features provide landmarks for computing facial descriptors (feature vectors). The whole process of representing faces in the videos by tracks results in substantial reduction in data to be processed. On the KIS dataset, tracking and filtering results in reduction in the granularity of the problem from 2.9 Million face detections to 17,390 face tracks.

Negative training images needed for training of the classifiers are taken from publicly available dataset [12]. These images are kept the same for all queries. The face detector, facial feature detector and appearance descriptor described above is applied to each of the negative images to produce feature vectors.

The online processing part consists of two steps collecting positive training images of faces from Google and training and ranking using a classifier. Once the features for positive training examples are computed, a linear SVM is trained, and used to assign scores to tracks in the corpus.

The resultant face search system can be used for searching both for specific people as well as those with specific (facial) attributes such as gender, facial hair, eyewear, etc. Figure 2 shows example results obtained for both these cases. For details of the method refer to [20].

D. On the fly Specific Object / Place Retrieval

The aim of the specific object / place recognition system is to quickly retrieve key-frames which contain queried specific objects or places based on their visual appearance. The query can be specified in two ways: (i) by uploading one or more images containing the object and optionally outlining regions of interest; (ii) by entering a textual query. Two varieties of method have been implemented: one that involves issuing multiple queries and combining the results (late fusion); the other is an early fusion method learns a more distinctive image representation on-the-fly by data mining the input query images. In both cases the final list is re-ranked using geometric verification.

The late fusion system architecture is identical to the one described in [3], which is based on the standard specific object retrieval approach by Philbin et al. [25] with some recent improvements which are discussed next. RootSIFT [4] descriptors are extracted from affine-Hessian interest points [22], [21] and quantized into 1M visual words using approximate k-means. Given a single query, the system ranks images based on the term frequency inverse document frequency (tf-idf) score [27]. The ranking is computed efficiently through the use of an inverted index. Spatial reranking is performed on the top 200 tf-idf results using an affine transformation [25].

In the on-the-fly system, given a text query of an object or place, example images are retrieved by textual Google image search using the publicly available API. A visual query set is constructed from the top 8 retrieved Google images. To retrieve from the corpus, a visual query is issued for each image in the query set independently and retrieved ranked lists are combined by scoring each image by the maximum of the individual scores obtained from each query. This is the MQ-Max method from [3], where further details are given. Figure 3 sketches the on-the-fly process and gives examples of retrieved key-frames.

For the version with early fusion, we first build a query specific model of the object or place. To this end, we mine local-bag-of words around the keypoints detected in the query images, resulting in a more powerful mid-level representation tuned towards the object or place we want to retrieve. Using this query specific model we construct a new histogram representation on the fly for each database image and retrieve images using a tf-idf based retrieval approach, using an inverted file system. This is again followed by spatial verification.

E. Score Fusion

Because our main focus this year lies in incorporating different retrieval services, we chose a relatively simple algorithm to fuse the scores from the above retrieval services. We first normalized the scores of each component to the interval $[0, 1]$ by dividing them through the maximum score and then fused them using a linear combination as follows (see also [26]):

$$score = \sum_{i=1}^{n} score_i \tag{1}$$

(a) Top 10 Google Image results for the textual query "forest"

(b) Top 10 INS dataset results retrieved for the query "forest"

Fig. 1: **On-The-Fly Object/Image Category Retrieval.** Images downloaded from Google using a textual query (a) are used to train a classifier and then retrieve images from the INS dataset (b).

(a) Top 20 INS retrieved results for specific person query "George Bush"

(b) Top 20 INS retrieved results for attribute query "Eyeglasses"

Fig. 2: **On-The-Fly Person Retrieval.** Retrived results from the INS dataset.

where $score$ is the final score, and $score_i$ is the confidence score of the ith retrieval service. Note that a retrieval function of weighted confidences scores is likely to perform better [33]. However, because our main focus this year was on the interactive use of the on-the-detectors we plan non-uniform weighting schemes for future work.

F. User Interface

The user interface used for both the KIS and INS tasks was based on a version of the AXES Professional search system interface that was developed by the AXES consortium based on professional user requirements and feedback from TRECVid 2011. Figure 4 shows a screenshot of the AXES professional user interface. As with our 2011 TRECVid interface, the AXES professional interface is a browser-based user interface targeted at traditional desktop-based interaction. The client-side interface uses HTML5, CSS3, and Javascript, and AJAX to communicate asynchronously with the server side.

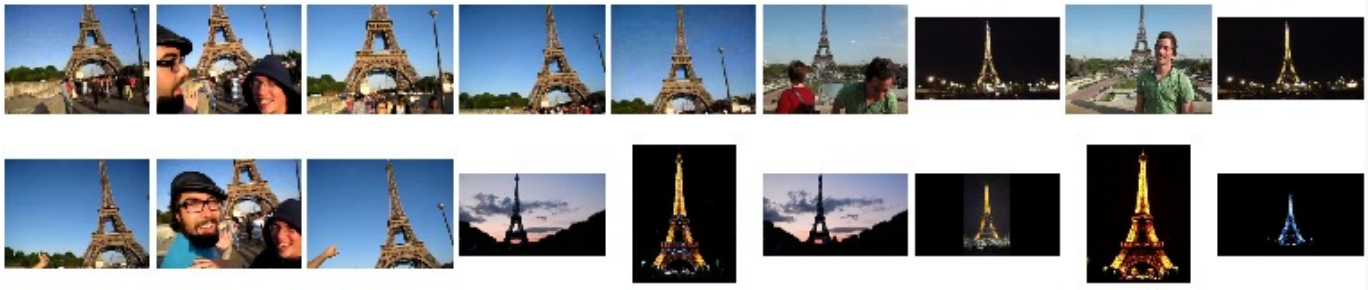
The interface is composed of two panels: the *search archive* panel and the *retrieved results* panel. The search archive panel allows user to formulate text-based, concept-based, or image-based queries. It supports predefined visual concept selection, visual similarity search, and video saving and download. The retrieved results panel shows the results of a query in various ways. There are four areas contained in search archive panel: Search, Concept Classifiers, Similarity Search, and Saved Videos. To ensure that there is always enough space on the users display, even for different resolutions and window sizes, each of these panels is collapsible.

Text-based queries can be entered via the search panel in the top-left of the interface. If the user checks the metadata or spoken words options, the relevance score will be calculated based

(a) Top 8 Google Image results for the textual query “Saint Peter’s Basilica”

(b) Top retrieved INS results for the query “Saint Peter’s Basilica”

(c) Top 8 Google Image results for the textual query “Eiffel tower”



(d) Top retrieved INS results for the query “Eiffel tower”

Fig. 3: **On-The-Fly Specific Object/Place Retrieval.** Images downloaded from Google using a textual query (a,c) are used to retrieve images from the INS dataset (b,d). Note the diversity of retrieved key-frames: (c) contains key-frames from the interior and exterior of Saint Peter’s Basilica, while (d) shows key-frames of the Eiffel tower at day and night.

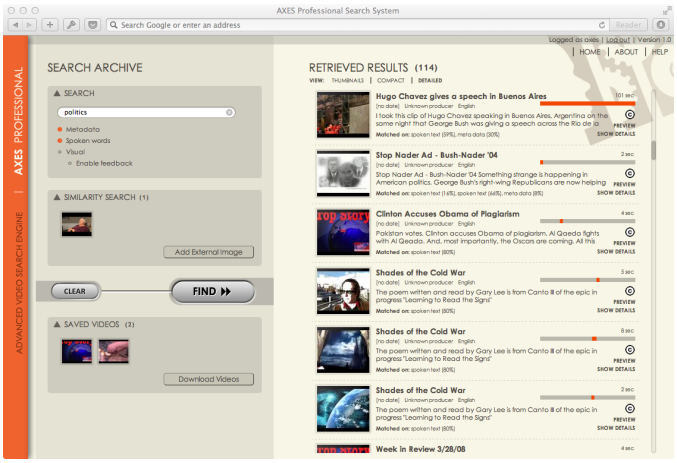


Fig. 4: The AXES professional user interface showing a detailed view of the search results.

on textual metadata (author, title, short description) or audio transcripts generated using automatic speech recognition. The visual search check box enables the on the fly visual concept classification, which uses images from an external source to build a visual model of the specified text. When enable feedback is selected, the intermediate results from the external search engine are presented to the user in a popup overlay. This overlay allows the user to exclude specific images from the model before training the on the fly classifier. The search options in this panel can be combined, in which case the results are compiled by fusing the output of the selected search components.

The concept classifier panel allows the user to specify a number of pre-defined high-level concepts be used in the search. The available concept list is dynamic and retrieved from the link management and structured search system on initialization. Each concept is a tri-state toggle that can be marked as positive, negative, or off. Clicking on the individual concepts in the list cycles the selection between these options. Marking a concept as positive will boost results containing that

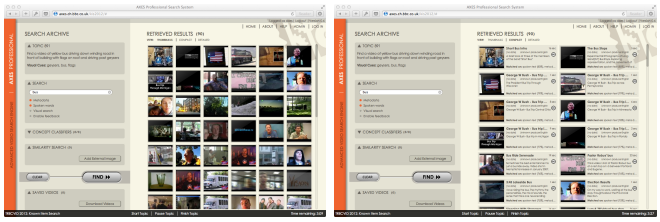


Fig. 5: The AXES user interface adapted for KIS showing two different result views.

concept so that they appear higher in the result list. Marking a concept as negative has the opposite effect: it indicates that the user wishes to see fewer results containing the concept and so demotes videos that are likely to contain the concept.

The similarity search panel is drag-and-drop based: upon retrieving a set of results, the user can drag any thumbnails to the similarity search panel to use them as query images. By default, the entire image is used as the query, but the user may also specify a region of interest by double clicking on the thumbnail and dragging a rectangle around the relevant region. Images from external websites and search engines, as well as images from the users local machine can also be added to the similarity search panel by clicking on the add external image button on the bottom right of the similarity search area. Clicking on this button displayed a selection overlay that allows the user to upload local images or specify external images by pasting in the URL for the image. Images can be removed from the similarity search panel simply by dragging them back into the results area.

The saved videos panel, located at the lower-left of interface, allows users to save video shots for subsequent use. As with the similarity search area, videos can be saved using drag-and-drop. Users can review saved videos by double clicking their thumbnails to play back corresponding video.

The retrieved results area on the right of the interface displays all videos retrieved that match the user's query. The results area allows the user to view the result list using three different views: thumbnails, compact, and detailed. In the thumbnail view (Figure 5, left), each retrieved video is represented as a single thumbnail. User can double-click any thumbnail to quickly preview the entire content of corresponding video in a popup overlay. If the retrieved result is a segment from the video, then the preview overlay will automatically jump to the relevant location in the video. Users can drag thumbnails to and from this view into the saved videos panel or similarity search panel. The advantage of this panel is that it provides a global overview of a large number of retrieved videos on a single screen; the disadvantage is the lack of detailed information on the videos.

The compact view (Figure 5, right) shows each retrieved result as a thumbnail along with some very brief accompanying metadata and match information. The displayed metadata includes the video title, publication date, producer, language, description, license, and clip duration. If any of these fields are too long to fit in the available screen space, the fields are truncated. Information is also shown to help the user to understand the reason that the system retrieved this particular

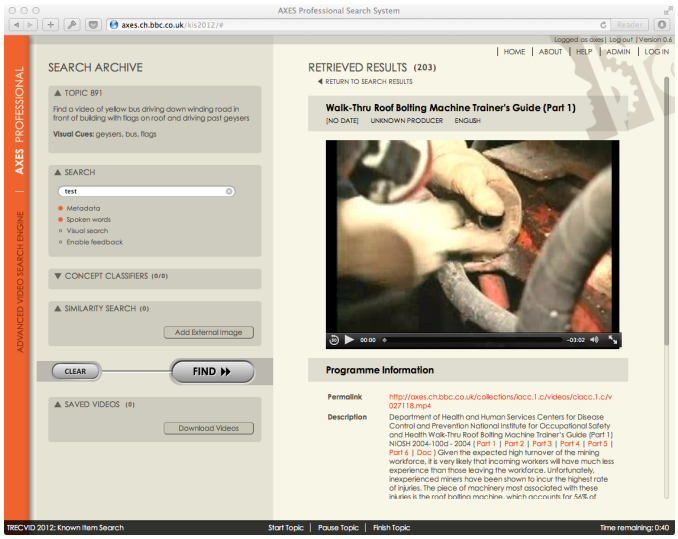


Fig. 6: The asset view showing detailed information about a single video.

result. This information is shown in a *matched on* field. This is particularly helpful when a query is specified using multiple modalities. For example, consider a text-based query in which the user has selected both spoken words and visual (on the fly concept) search. In this case, a particular match may not look visually similar to the concept specified by the query text, but may be retrieved because a person in the video spoke the words specified in the query text. The matched on field shows the user that the video was matched based on spoken words, thus helping the user to understand the system's behaviour and increasing confidence in the results. As with the thumbnail view, results can be freely dragged to and from the compact view and the similarity search and saved videos panels.

In the detailed view (Figure 4), each row contains one retrieved video with more detailed information than what is presented in compact panel. As with the compact view, each video is displayed as a thumbnail with associated metadata, and the thumbnail may be double-clicked to start a preview playback. The metadata and matching information is located besides the thumbnails. The information presented is much the same as with the compact view, the difference being that there is more screen space dedicated to this information, so that less information needs to be truncated. A coloured segment location bar is also shown in this view. It describes the temporal location of the retrieved video segment with respect to the overall video. The length of the grey bar indicates video duration, while the length of orange bar describes the duration of video segment and the position where it is located. The duration of segment is displayed textually over the segment location bar. Below it, there are two buttons: preview and show details. Clicking on preview button shows the standard preview overlay. The show details button slides out the retrieved results and displays a detailed *asset view* (Figure 6) for the selected video.

Two forks of the AXES Professional user interface were created to facilitate AXES participation in the 2012 TRECVid benchmarking activity: one for the instance search task and

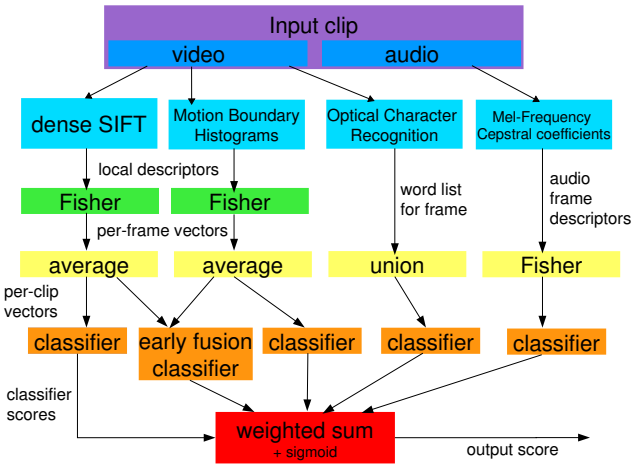


Fig. 7: Overview of the AXES event detection system.

one for the known-item search task. These interfaces added the extra features required for the experiments. Both of these interfaces contained extra functionality for topic assignment, topic specification, and task timing. The instance search interface also delivered the necessary topic sample images to the user. The known item search interface (Figure 4) added functionality to validate saved videos with the KIS oracle and inform the user if they had found the correct video.

III. Multimedia Event Detection

In this section we describe the AXES submission to the Multimedia Event Detection (MED) track. Figure 7 gives an overview of the system which we describe in more detail below. In Section III-A we describe the three low-level features we used, and in Section III-B how we encode them to obtain a video-level signature. Section III-C gives more detail on finding a balance between the computational cost and discriminative power of the MBH motion feature. Section III-D describes the fourth feature that is based on text detected by OCR. Details on classifier training and feature fusion are provided in III-E.

A. Low-level audio-visual features

For the audio channel we down-sample the original audio track to 16 kHz with 16 bit resolution and then compute Mel-frequency cepstral coefficients (MFCC) with a window size of 25 ms and a step-size of 10 ms, keeping the first 12 coefficients of the final cosine transformation plus the energy of the signal. We enhance the MFCCs with their first and second order derivatives.

The visual content is described by static appearance and motion features. For static visual appearances we use SIFT features [**?**] extracted for one frame out of 60 frames. We compute SIFT descriptors at multiple scales at points on a spatially dense sampling grid (21×21 patches at 4 pixel steps). Motion information is captured using the recently trajectory + Motion Boundary Histogram (MBH) features [31], which have shown to obtain state-of-the-art results for human action recognition.

The MBH feature is similar to SIFT, but based on motion information derived from optical flow fields. Where SIFT computes gradient orientation histograms over pixel gray value intensities, MBH computes these over both the vertical and horizontal spatial derivatives of the optic flow field. A second difference is that SIFT descriptors are computed around small patches in the image plane, where MBH descriptors are computed along feature tracks. This ensures that each descriptor is computed from the spatio-temporal volume which follows the motion. Just like in SIFT, gradient orientation histograms are computed in several regular cells along each trajectory, and then concatenated. The procedure is illustrated in Figure 8. The top row of the figure shows (left) a video frame, together with (center) its flow field (direction indicated by hue, and magnitude coded by saturation), and (right) the gradients computed in the horizontal component of the flow field (using the same color coding). The bottom left panel shows a set tracked features along each of which an MBH descriptor is computed. The bottom right panel shows spatio-temporal cells aligned with the feature track, for each of which gradient orientation histograms are computed and then concatenated. Since the MBH feature is relatively expensive to compute, we consider in Section III-C the trade-off between the accuracy of the descriptor vs. the computational efficiency by down-sampling videos over space and time.

B. Low-level feature encoding

Once the three local low-level features are extracted, we use them to construct a signature to characterize the video. For this feature encoding step we proceed in the same manner for all three low-level features by using a Fisher Vector (FV) representation [23]. This is an extension of the bag-of-visual-words (BoV) representation, which is widely used for image classification and retrieval since its introduction in [7], [27]. The BoV approach is based on a quantization of the local descriptor space (typically obtained off-line using a k-means clustering on a large collection of local descriptors). A video is then represented by a histogram that counts how many local descriptors of that video are assigned to each quantization cell. The size of histogram equals the number of quantization cells.

Fisher vector (FV) records, for each quantization cell, not only the number of assigned descriptors, but also their mean and variance along each dimension. Therefore, a smaller number of quantization cells can be used than for BOV. This leads to a signature with a dimension of $K(2D+1)$ for K quantization cells and D dimensional descriptors. Since the assignment of local descriptors to quantization cells is the main computational cost, the FV signature is faster to compute. Instead of using a k-means clustering, a Mixture of Gaussian clustering is used in the FV representation. Local descriptors are then assigned not only to a single quantization cell, but in a weighted manner to multiple clusters using the posterior component probability given the descriptor. In addition, we apply power and L2 normalization, as introduced in [24].

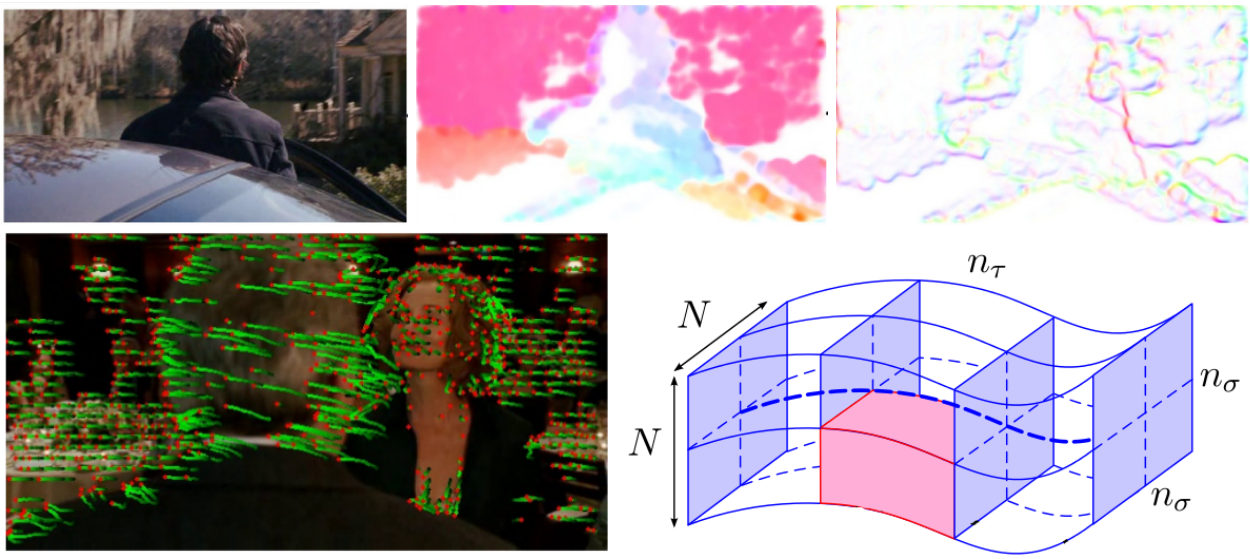


Fig. 8: Illustration of the MBH motion feature computation, see text for more details.

C. Video rescaling for efficient MBH computation

The videos in the TRECVid MED dataset vary strongly in size: the duration ranges from few seconds to one hour, while the resolution goes from low quality (e.g., 128×88) to full HD (1920×1080). We normalize the videos to make the MBH feature extraction feasible and to ensure that the features across different videos are comparable.

We experimented with rescaling the videos such that its width is at most 160 pixels (`small`), 320 pixels (`medium`), 640 pixels (`large`), or 1280 pixels (`maximum`); the video's height is set to preserve the original aspect ratio.

For the temporal rescaling, we tried two approaches: computing features only on parts of the video (`chunk`) and dropping some frames uniformly, with a similar effect to increasing the frame rate (`skip`). For both cases, we selected the parameters such that we only process a quarter ($T/4$) or half ($T/2$) of the video frames: for the `chunk` case, we used each 120 frames (around 4 s) discarding the next 360 ($T/4$) or the next 120 ($T/2$); and, for the `skip` case, we skip either three frames out of each four ($T/4$) or one out of two ($T/2$). From the timings in Table I we can see that the cost of MBH feature extraction scales roughly linearly with the number of pixels in the spatial-temporal video volume.

For computational efficiency we decided to rescale the videos to either `small` or `medium`. To finally determine the spatial and temporal resolution, we conducted experiments using training set of Trecvid 2011 MED. This data set of about 7500 videos was divided itself in a train and test set, each containing about 100 videos for 15 event categories, plus additional videos from the null class. We trained per-event classifiers on samples from that event and the null class. We evaluated in a similar scenario, using mean average precision as metric. The results in Table II, show that the `skip` strategy yields higher quality descriptors, and that spatial sub-sampling yields a smaller degradation of performance than temporal sub-sampling.

Based on these results, we decided to use the `small` spatial resolution, and skipped every second frame. These design choices allowed us to compute the complete motion feature pipeline —video re-scaling, MBH extraction, and FV encoding— in 2.42 times the real-time duration of the video on a single core.

		Temporal subsampling		
	Resolution	Original	$\frac{1}{2}$	$\frac{1}{4}$
`chunk`	`maximum` (1280 × 720)	2:36:11	1:15:25	37:08
	`large` (640 × 360)	37:17	18:08	8:46
	`medium` (320 × 180)	9:08	4:29	2:10
	`small` (160 × 90)	2:07	1:01	0:30
`skip`	`large` (640 × 360)		20:53	10:58
	`medium` (320 × 180)		5:01	2:40
	`small` (160 × 90)		1:08	0:36

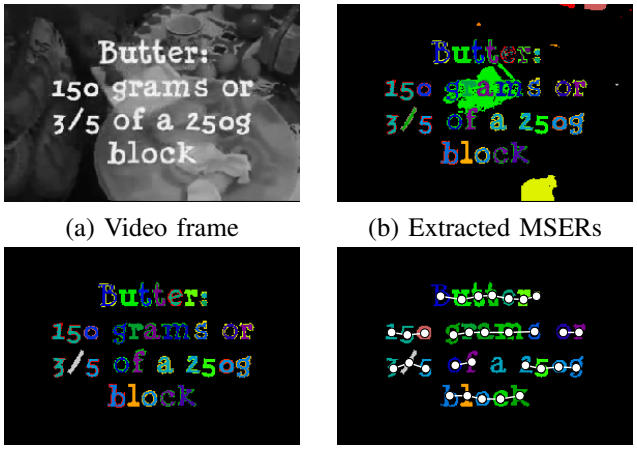
TABLE I: CPU time for MBH feature extraction for various rescalings of a 1280×720 video with length 2m15s.

		Temporal subsampling		
	Resolution	Original	$\frac{1}{2}$	$\frac{1}{4}$
`chunk`	`medium`	52.630	49.069	44.045
	`small`	49.313	46.443	41.843
`split`	`medium`		52.099	50.603
	`small`		53.174	49.827

TABLE II: Mean average precision for different rescalings on a subset of the MED 2011 dataset using MBH features.

D. Text features from optical character recognition

Our fourth feature encodes high-level information extracted using an Optical Character Recognition (OCR) system. For each video frame (sampling rate of 5Hz), MSER [14] regions are extracted from the luminance channel (see Figure 9 a, b). Regions that do not have a suitable aspect ratio or weak gradients on their boundary are eliminated (Figure 9 c). Remaining ones are grouped into text lines, which are further segmented into words (Figure 9 d). Then, each region is



(a) Video frame (b) Extracted MSERs

(c) Filtered MSERs (d) MSER grouping

Fig. 9: Three stages of the OCR system, see text for details.

expressed in term of a HOG-based descriptor [8], and a RBF kernel SVM classifier (trained on standard Windows fonts) predicts the probability of each character. Those probabilities are combined using an English language model based on 4-grams over letters to yield the final OCR results at the word level.

From the OCR output, a sparse bag-of-word descriptor is formed for each video. To decrease the descriptor sparsity, we include in the words hypernyms (according to the Wordnet lexical database). We also found that including bi-words (i.e. pairs of words) improves their distinctiveness and the overall performance.

E. Classifier training and feature fusion strategies

For each feature type (visual static and motion, audio, and text) we learn linear Support Vector Machine (SVM) classifiers, which permit efficient training and testing. We compared different feature aggregation techniques. Early and late fusion techniques are applied to combine the information from the different features.

1) Feature aggregation techniques: We have explored two setups to train classifiers. In the first we compute a single signature for the complete video by averaging the frame-level FVs into a single descriptor as described above. The second option we considered was to segment the video in several chunks of a fixed duration, and to compute a FV for each chunk.

Using the first setup we train and use the SVM classifiers in a conventional manner. For the second setup we proceed as follows. During training all the chunks extracted from a video are treated as separate training videos that inherit the class label of the complete video. To classify a new video, we apply the classifier to the chunks, and assign the video the maximum classification score that was obtained among the chunks.

While the second chunk-based approach might be less sensitive to irrelevant portions of the video by taking the maximum, we do no find it to be more effective in practice. This might be understood by noting that during training this approach introduces a form of label noise, since irrelevant parts of a positive video are included in the training data for the classifier. The benefit of the complete-video approach is substantial, e.g. in initial experiments using the SIFT features we measured an improvement from 44% to 51% mean average precision over the 10 categories used in the evaluation presented in the next section.

2) Early and late fusion: In order to combine the different low-level features we consider an early fusion strategy, which consists in concatenating the signatures extracted from the different features. A relative scaling of the features is determined using a cross-validation technique. We used a discrete grid-search to set the weight of the feature signatures, but instead of exhaustively considering all possible weights on a pre-determined grid, we do a form of local search. A pool of optimal combinations is maintained, and at each iteration, the neighbors of the current optimum are evaluated, while the less-than-optimal solutions are progressively removed from the pool.

Second, we include a late-fusion stage in which we linearly combine the classifier scores computed from each of the three low-level features, the OCR features, as well as the early-fusion system. To find this late-fusion weights we experimented with training a logistic discriminant classifier, and using an exhaustive grid search which is feasible since at this stage each video is represented by a small set of classifier scores.

IV. Experiments

In the following we describe our experiments for TRECVid 2012, see [18] for an overview.

A. Known Item Search

The known-item search experiments were carried out in BBC London in September 2012. A total of 20 media professionals from BBC participated in the experiments. The experiments were carried out in sessions with between three and six participants per session. Before each session, the participants were briefed on the purpose of the experiment, and given a short 10-15 minute tutorial on how to operate the system. Each participant was assigned six topics and had five minutes to complete each topic. After each session, participants were asked to complete a brief survey and provide some free form feedback on the system.

We submitted four runs of our system for evaluation. Each run used an identical search system and user interface, varying only in the users that actually performed the search. Figure 10 shows the proportion of correct videos found in each of the runs submitted by all participants for evaluation. The AXES runs are highlighted in blue and the interactive runs in green. Users from our best run found 13 of the 25 topics, and the performed above the median out of the submitted interactive runs. It is clear from the table that interactive systems almost always outperformed fully automatic systems. Figure 11 shows the mean time taken by axes users to find the correct video in each of the runs.

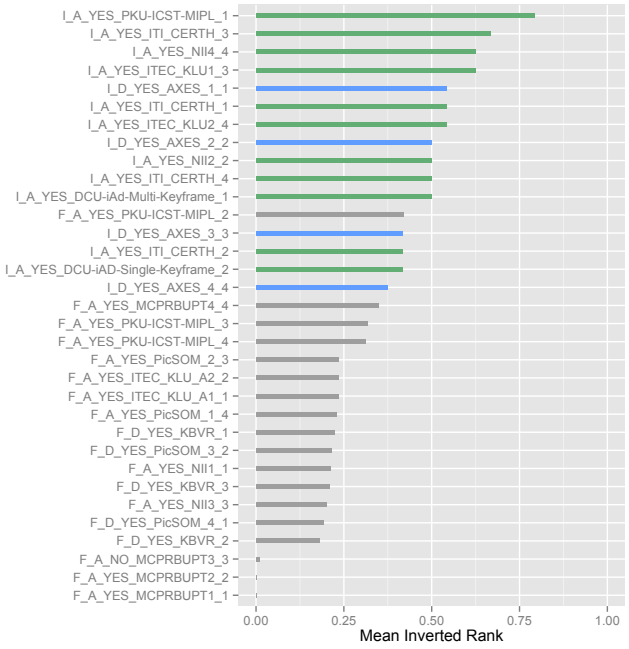


Fig. 10: Proportion of correct videos found by KIS participants in each of the KIS runs. The graph is ordered from top to bottom by the number of videos found. The blue bars represent the runs submitted by AXES, the green bars represent interactive runs submitted by other groups, and the gray bars represent automatic runs.

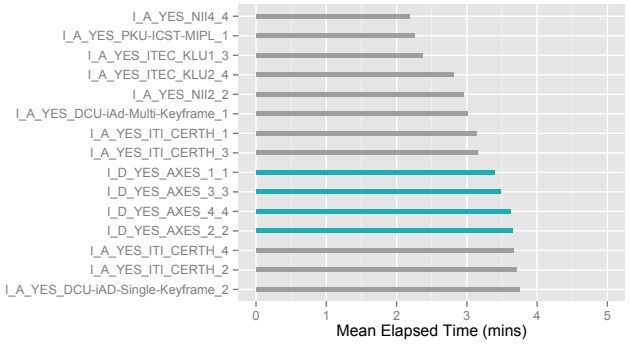


Fig. 11: Mean time (in minutes) required to find the correct video by AXES runs and other runs.

B. Instance Search

The instance search experiments were carried out at Dublin City University in August 2012. A total of 16 people participated in the experiments. Participants were primarily research assistants, students, and post doctoral researchers. Each participant was assigned five or six topics and had 15 minutes to complete each topic. Participants were briefed on the purpose of the experiment the day before it was run, and shown how to operate the user interface. They were also given a sample topic and some time to familiarize themselves with the interface before the experiment.

We submitted four runs of our system for evaluation. Each run trialled a different variant of the user interface. The user

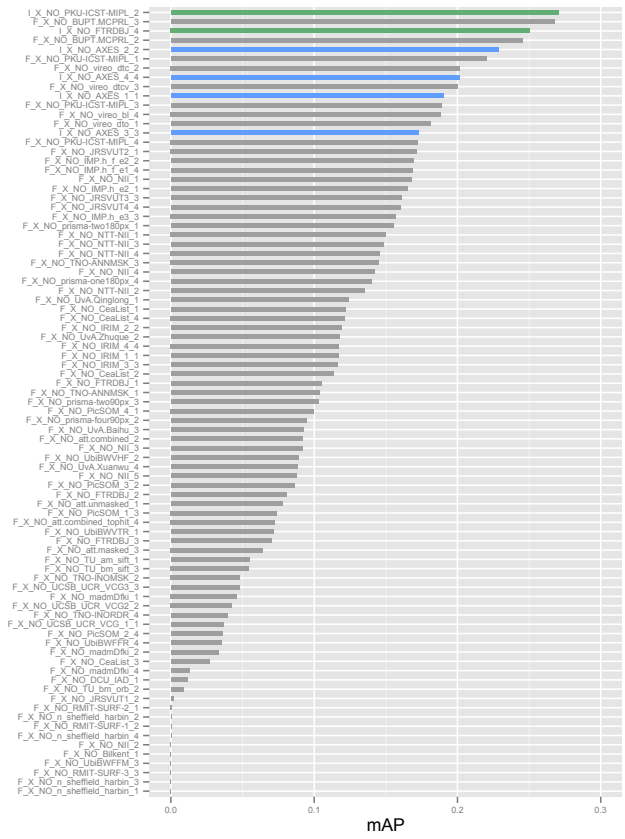


Fig. 12: Mean average precision over all submitted INS runs. AXES runs are shown in blue and other interactive runs in green.

interface variants for each of the runs were:

1) AXES_1_1: An un-tabbed user interface with a feedback mechanism for visual search;
2) AXES_2_2: A tabbed user interface with a feedback mechanism for visual search;
3) AXES_3_3: An un-tabbed user interface without a feedback mechanism for visual search;
4) AXES_4_4: A tabbed user interface without a feedback mechanism for visual search.

In the tabbed variations of the user interface, search results were displayed in a new in page tab each time the user clicked on the Find button, allowing users to keep many results around simultaneous available and start new queries while they browsed previously found results. The feedback mechanism allowed users to see the intermediate images that were returned from the external search engine when performing an on the fly visual concept search and select which ones that should be used to model the concept.

Figure 12 shows mean average precision for all submitted INS runs, with AXES runs shown in blue and other interactive runs shown in green. Based on mAP, we found that the tabbed version of the interface consistently outperformed the un-tabbed version. It is also clear based on mAP that variants with the feedback mechanism enabled outperformed their counterparts without a feedback mechanism, suggesting that

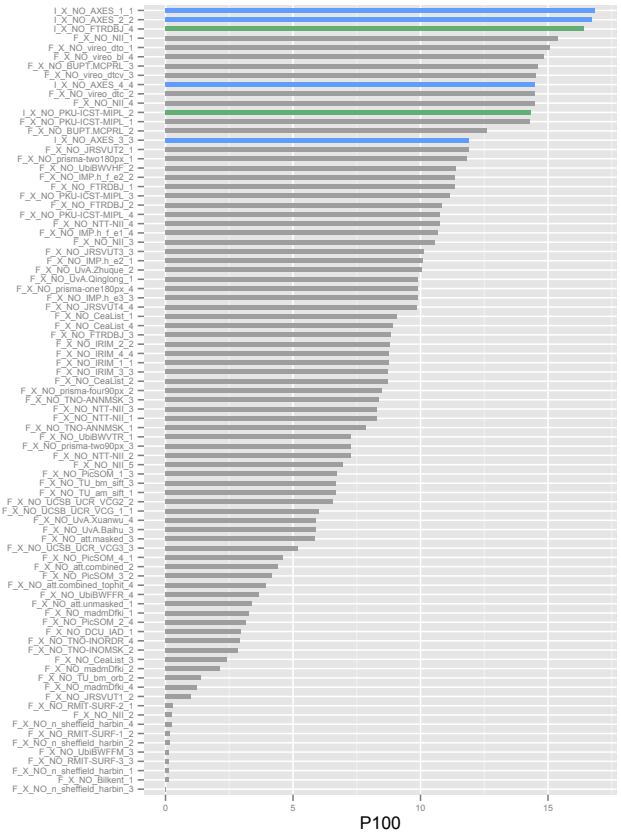


Fig. 13: Number of correct results out of the first 100 results returned for all submitted INS runs. AXES runs are shown in blue and other interactive runs in green.

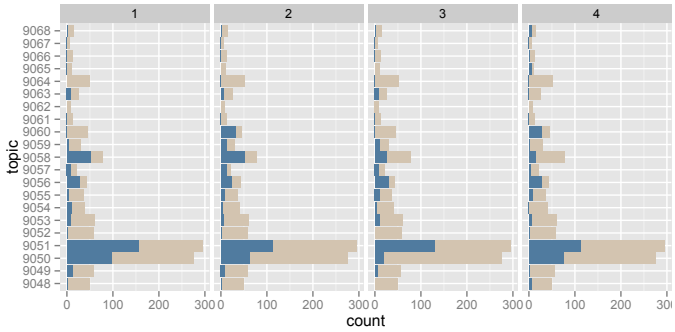


Fig. 14: Comparison of the number of relevant videos with the number of saved (returned) videos for each of the four AXES runs. The number of saved videos are shown as dark blue bars; the total number of relevant videos are shown as light brown bars.

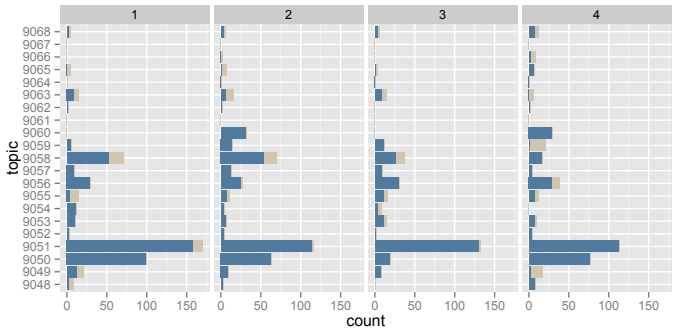


Fig. 15: Plot showing the relative proportions of relevant and non-relevant videos saved by each participant by topic. The dark blue bars represent the number of relevant videos; the light brown bars represent the number of non-relevant videos.

the allowing users to refine the images used to train the on the fly visual search models can often produce better results.

Figure 13 shows the number of correct results out of the first 100 results returned for all submitted INS runs. The two AXES runs that incorporated the feedback mechanism outperformed all other submitted runs, both interactive and automatic, when compared under this metric.

Figure 14 shows a more detailed plot of the proportion of relevant videos found by the experiment participants in each of four runs. Each bar in this plot represents the performance of a single user on a single topic. There was, in general, less relevant videos per topic than in TRECVid 2011. There was also clearly quite a few very difficult topics, particularly 9067, 9066, and 9061, in which none of our users were able to find any relevant examples.

Figure 15 shows the relative proportions of relevant and non-relevant videos saved by each participant by topic. In comparison with TRECVid 2011, the proportion of non-relevant videos that users believed were relevant (saved) has significantly reduced, indicating that there was less ambiguity in the topics this year.

C. Multimedia Event Detection

The setup of the MED 2012 evaluation is the same as in 2011, except that the training set is different, 10+5 more events are added, and the test set is larger (98000 videos, 4000 h).

1) Description of submitted runs: The final results we submitted are computed from two descriptor versions: "small" that we use in this section for validation, and "big" ones. We did not have time to validate the big version on the validation set. Therefore, we also submitted results on the small descriptors as a fallback. The descriptor sizes are summed up in table III:

- for MBH: the 96 D MBH[1] are reduced by PCA to $D = 64$, that are aggregated in FVs of $2KD$ dimensions (derivatives w.r.t. the mean and variance, each of dimension KD). For the small version: $K = 256$, big: $K = 1024$.
- for SIFT: the 128 D SIFT[2] descriptors are reduced by PCA to $D = 32$ leading to a FV of dimension $2KD$. For small: $K = 256$, big: $K = 1024$.
- for MFCC: the 13D MFCC is concatenated with its derivative and 2nd derivative, resulting in $D = 39$. For the small version, the FVs have dimension $2KD + K - 1$ (derivatives w.r.t. mean and variance + mixing weight for each Gaussian). For the big version, two vocabularies were used: one trained on audio data with speech, one

[1]The MBH implementation is available at: http://lear.inrialpes.fr/people/wang/dense_trajectories

[2]Code for SIFT and Fisher: http://lear.inrialpes.fr/src/inria_fisher

modality	descriptor	small version		big version	
		dim	× RT	dim	× RT
Motion	MBH	32768	2.42	131072	3.0
Image	SIFT	16384	2.54	65536	6.6
Audio	MFCC	40447	0.15	80894	0.2
Text	BoW	200k	1.42	200k	1.42
Total		289599	6.53	477502	11.22

TABLE III: Descriptor dimension and processing time (as a slowdown factor with respect to the real video time). The sparse bag-of words text descriptor comes in only one version.

run	signature size	late-fusion weights
c-LFdnsmall	small	grid search
c-LFjrlrsmall	small	logistic regression
p-LFdnbig	big	grid search
c-LFjrlrbig	big	logistic regression

TABLE IV: Description of our submitted runs.

on non-speech. FVs for the two were concatenated, hence the size doubled. In both cases, $K = 512$.

Compared to the classification step, the descriptor extraction is by far the most expensive operation. Of this, the local descriptor extraction is the most expensive part.

Our four submitted runs are summerized in table IV. There two runs for small descriptors and two for big ones. The two variants differ in the way the late-fusion weights are obtained, either by grid search or by logistic regression.

2) Results: The official results consist in a single score (the average actual NDC over all events). This does allow only a very coarse-grained analysis. Table V shows that on the pre-specified events, we arrive second, and this ranking does not depend on whether we use small or big descriptors. We can observe that the big descriptors improve over the small ones. Estimating the late-fusion weights with a logistic regression is slightly better than brute-force grid search. We only report results for the better variant. On the ad-hoc events, for which ECNU did not submit, we get the best scores. The performance of the big descriptors is on par with the one on the pre-specifed run. The small descriptors perform significantly worse on the ad-hoc events. An analysis of the cross-validation scores indicate that this is due to a bug in the SIFT descriptor computation.

3) Evaluation on MED 2011: To obtain a more detailed analysis we evaluated our system using the TRECVID MED 2011 data set, and present results for the 10 event categories that were also used in MED 2012. For each category between 100 and 300 training videos are available, while the null class contains 9600 videos. The test set consists of 32,000 videos totaling 1,000 hours of video. We used a 30-fold cross-validation approach to set hyper-parameters, such as the SVM regularization parameter, and the balancing between positive and negative examples. Below, we use settings corresponding to the `c-LFdnsmall` run submitted for TRECVID MED 2012. The error measure we report is the NDC measure for the optimal threshold.

run	pre-specified		ad-hoc	
	run/group	NDC	run/group	NDC
best group	ECNU	0.4841	-	-
our big	c-LFjrlrbig	0.5154	c-LFjrlrbig	0.5357
our small	c-LFjrlrsmall	0.5361	c-LFdnsmall	0.7112
best after us	TokyoTechCanon	0.5698	BBNVISER	0.6873

TABLE V: Official MED results. The actual NDC is averaged over all events.

Among the results reported in Table VI we included the results of the best system that entered in the 2011 edition [16] (first row), and our 2011 submission (second row, see [5] for details). Our current system is similar that of last year; the most important differences are (i) the use of Fisher Vector encoding for the MBH and MFCC features instead of BOV encoding, in combination with linear SVMs instead of non-linear RBF-χ^2 kernel SVMs, (ii) the use of per-video aggregated SIFT features, instead of per frame classification and taking the maximum score, and (iii) the inclusion of OCR-based text features.

From the results it is clear that each of the visual features alone is quite competitive with last year's system. In many cases either the SIFT or MBH feature alone already performs on par with last year's submission. Late fusion of the low-level audio-visual features seems slightly more effective than early fusion, and provides a system that is outperforming our system from 2011 on all categories. On average the result of 0.434 is also comparable to last year's best entry (0.436), although per category results can differ significantly. Surprisingly, adding the early fusion of the visual features on top of the separate features in the late fusion step improves performance on 7 of the 10 categories. Finally, adding the text features further improves results to outperform the winning system of last year [16] on 6 out of 10 categories, as well as on average over all 10 classes.

Compared to [16] our system is relatively efficient, since the latter computed more features (which represents the main computational bottleneck), including:

- 4 static visual features (we only use dense SIFT),
- 3 static color features (we use none),
- 2 motion features (we only use MBH),
- 2 audio features (we only use MFCC),
- object detector results (we do not use),
- automatic speech recognition (we do not use).

From the additional features used in [16], the automatic speech recognition is one that is of most potential value to our system. First it is complementary to the other ones, and second our text features based on OCR are quite sparse in the sense that in many videos no or little text is detected.

In conclusion, the evaluation shows a marked improvement of our system with respect to last year's one, underlining the benefit of (i) the FV encoding over BOV, and (ii) using the sparse but high-level OCR text features. Moreover, by using a small set of state-of-the art low-level descriptors we obtain a system with performance that is better or comparable to the winning entry in the 2011 edition of TRECVID MED which was based on a multitude of features.

	E006	E007	E008	E009	E010	E011	E012	E013	E014	E015	Mean
Baselines											
Best 2011 result [16]	0.442	0.438	0.263	0.379	0.622	0.561	0.446	0.308	0.331	0.575	0.436
INRIA-LEAR 2011 [5]	0.716	0.732	0.433	0.565	0.803	0.857	0.551	0.448	0.509	0.802	0.642
Individual features											
MBH	0.766	0.785	0.338	0.590	0.754	0.768	0.523	0.254	0.531	0.652	0.596
SIFT	0.713	0.627	0.400	0.452	0.746	0.693	0.713	0.570	0.611	0.768	0.629
Audio	0.645	0.926	0.704	0.767	0.959	0.936	0.800	0.944	0.553	0.823	0.806
Text	0.950	0.941	0.914	0.992	0.927	0.845	0.951	0.995	0.683	0.884	0.908
Early fusion: MBH + SIFT + Audio											
Non-weighted early fusion	0.496	0.445	0.257	0.385	0.582	0.630	0.432	0.250	0.376	0.578	0.443
Weighted early fusion	0.517	0.454	0.253	0.390	0.577	0.639	0.433	0.296	0.375	0.584	0.452
Late fusion											
MBH + SIFT + Audio	0.488	0.480	0.261	0.377	0.586	0.646	0.414	0.214	0.351	0.517	0.434
MBH + SIFT + Audio + NW	0.484	0.456	0.265	0.391	0.566	0.641	0.405	0.220	0.345	0.515	0.429
MBH + SIFT + Audio + W	0.479	0.454	0.270	0.379	0.553	0.644	0.405	0.227	0.350	0.511	0.427
MBH + SIFT + Audio + NW + Text	0.458	0.451	0.257	0.383	0.538	0.550	0.391	0.226	0.344	0.507	0.411

TABLE VI: NDC performance on the 2011 MED dataset (w: weighted early fusion of MBH+SIFT, nw: idem, non-weighted).

V. Summary

This paper described the AXES participation in the interactive KIS and INS tasks, and the MED task for TRECVid 2012. Our interactive INS and KIS systems used nearly identical backends and user interfaces and focused using classifiers trained at query time with positive examples collected from external search engines.

Our MED system used a relatively simple approach based on four low-level descriptors of audio and visual content, aggregated to produce high-dimensional video-level signatures, which were then used to train a set of linear classifiers. All our systems performed relatively well in our experiments. Our best KIS run found 13 of the 25 topics, our best INS runs outperformed all other submitted runs in terms of P@100, and our MED system achieved the second-best score of all submitted runs in the main track, and best score in the ad-hoc track. The results show that this configuration can perform well, provided the descriptors are large enough and tuned carefully. Their large dimension also makes it possible to use linear classifiers only, reducing the computational cost and memory consumption at test time.

Acknowledgements

We would also like to acknowledge everyone that participated in the experiments, both the media professionals and the visiting students. This work was funded by the EU FP7 Project AXES ICT-269980 and the QUAERO project supported by OSEO. Furthermore, we are grateful to the UK EPSRC and ERC grant VisRec no. 228180 for financial support.

References

[1] R. Aly, C. Hauff, W. Heeren, D. Hiemstra, F. de Jong, R. Ordelman, T. Verschoor, and A. de Vries. The lowlands team at TRECVid 2007. In *Proceedings of the 7th TRECVid Workshop*, Geithesburg, U.S., February 2007. NIST.

[2] R. Aly, D. Hiemstra, A. P. de Vries, and H. Rode. The lowlands team at TRECVid 2008. In *Proceedings of the 8th TRECVid Workshop*, 2008.

[3] R. Arandjelović and A. Zisserman. Multiple queries for large scale specific object retrieval. In *Proceedings of the British Machine Vision Conference*, 2012.

[4] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[5] M. Ayari, J. Delhumeau, M. Douze, H. Jégou, D. Potapov, J. Revaud, C. Schmid, and J. Yuan. INRIA@TRECVID'2011: Copy Detection & Multimedia Event Detection. In *TRECVID*, Gaithersburg, United States, December 2011.

[6] K. Chatfield and A. Zisserman. Visor: Towards on-the-fly large-scale object category retrieval. In *Asian Conference on Computer Vision*, 2012.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop Statistical Learning in Computer Vision*, 2004.

[8] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, 2006.

[9] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 2009.

[10] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 2005.

[11] C. Foley, J. Guo, D. Scott, P. Wilkins, C. Gurrin, A. F. Smeaton, P. Ferguson, K. McCusker, E. S. Diaz, X. Giro-i-Nieto, F. Marques, K. McGuinness, and N. E. O'Connor. TRECVID 2010 Experiments at Dublin City University. In *Proceedings of the 10th TRECVid Workshop*, Gaithersburg, USA, 2010.

[12] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[13] H. Jégou, M. Douze, G. Gravier, C. Schmid, and P. Gros. INRIA LEAR-TEXMEX: Video copy detection task. In *Proc. of the TRECVid 2010 Workshop*, Gaithersburg, United States, 2010.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[15] K. McGuinness, R. Aly, S. Chen, M. Frappier, M. Kleppe, H. Lee, R. J. F. Ordelman, R. Arandjelovic, M. Juneja, C. V. Jawahar, A. Vedaldi, J. Schwenninger, S. Tschöpel, D. Schneider, N. E. O'Conner, A. Zisserman, A. Smeaton, and H. Beunders. AXES at TRECVid 2011. In *TREC 2011 Video Retrieval Evaluation Online Proceedings (TRECVid 2011)*,, Geithesburg, U.S., December 2011. NIST.

[16] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan. Multimodal feature fusion for robust event detection in web videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[17] M. Osian and L. V. Gool. Video shot characterization. In *Proceedings of the 1th TRECVid Workshop*, Gaithersburg, USA, November 2003.

[18] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quéenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[19] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quéenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[20] O. M. Parkhi, A. Vedaldi, and A. Zisserman. On-the-fly specific person retrieval. In *International Workshop on Image Analysis for Multimedia Interactive Services*. IEEE, 2012.

[21] M. Perdoch. http://cmp.felk.cvut.cz/∼perdom1/code/index.html.
[22] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
[23] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.
[24] F. Perronnin, J.Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the European Conference on Computer Vision*, 2010.
[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
[26] J. A. Shaw, E. A. Fox, J. A. Shaw, and E. A. Fox. Combination of multiple searches. In *The Third Text REtrieval Conference (TREC-3)*, pages 243–252, 1994.
[27] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1470–1477, October 2003.
[28] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.
[29] L. C. D. Team. *Lucene 3.2.*
[30] S. Vempati, M. Jain, O. M. Parkhi, C. V. Jawahar, M. Marszalek, A. Vedaldi, and A. Zisserman. Oxford-IIIT TRECVID 2009 - Notebook Paper. In *Proceedings of the 5th TRECVid Workshop*, 2009.
[31] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2011.
[32] P. Wilkins, D. Byrne, G. J. F. Jones, H. Lee, G. Keenan, K. McGuinness, N. E. O'Connor, N. O'Hare, A. F. Smeaton, T. Adamek, R. Troncy, A. Amin, R. Benmokhtar, E. Dumont, B. Huet, B. Merialdo, G. Tolias, E. Spyrou, Y. Avrithis, G. T. Papadopoulous, V. Mezaris, I. Kompatsiaris, R. Mörzinger, P. Schallauer, W. Bailer, K. Chandramouli, E. Izquierdo, L. Goldmann, M. Haller, A. Samour, A. Corbet, T. Sikora, P. Praks, D. Hannah, M. Halvey, F. Hopfgartner, R. Villa, P. Punitha, A. Goyal, and J. M. Jose. K-Space at TRECVid 2008. In *Proceedings of the 8th TRECVid Workshop*, Gaithersburg, USA, 2008.
[33] R. Yan. *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval*. PhD thesis, Carnegie Mellon University, 2006.