Deliverable D1.6 Intelligent hypervideo analysis evaluation, final results

Evlampios Apostolidis / CERTH
Fotini Markatopoulou / CERTH
Nikiforos Pitaras / CERTH
Nikolaos Gkalelis / CERTH
Damianos Galanopoulos / CERTH
Vasileios Mezaris / CERTH
Jaap Blom / Sound & Vision

02/04/2015

Work Package 1: Intelligent hypervideo analysis

LinkedTV
Television Linked To The Web
Integrated Project (IP)
FP7-ICT-2011-7. Information and Communication Technologies
Grant Agreement Number 287911

Dissemination level	PU
Contractual date of delivery	31/03/2015
Actual date of delivery	02/04/2015
Deliverable number	D1.6
Deliverable name	Intelligent hypervideo analysis evaluation, final results
File	`linkedtv-d1.6.tex`
Nature	Report
Status & version	Final & V1.1
Number of pages	64
WP contributing to the deliverable	1
Task responsible	CERTH
Other contributors	Sound & Vision
Author(s)	Evlampios Apostolidis / CERTH Fotini Markatopoulou / CERTH Nikiforos Pitaras / CERTH Nikolaos Gkalelis / CERTH Damianos Galanopoulos / CERTH Vasileios Mezaris / CERTH Jaap Blom / Sound & Vision
Reviewer	Michael Stadtschnitzer / Fraunhofer
EC Project Officer	Thomas Küpper
Keywords	Video Segmentation, Video Concept Detection, Video Event Detection, Object Re-detection, Editor Tool

Abstract (for dissemination)	This deliverable describes the conducted evaluation activities for assessing the performance of a number of developed methods for intelligent hypervideo analysis and the usability of the implemented Editor Tool for supporting video annotation and enrichment. Based on the performance evaluations reported in D1.4 regarding a set of LinkedTV analysis components, we extended our experiments for assessing the effectiveness of newer versions of these methods as well as of entirely new techniques, concerning the accuracy and the time efficiency of the analysis. For this purpose, in-house experiments and participations at international benchmarking activities were made, and the outcomes are reported in this deliverable. Moreover, we present the results of user trials regarding the developed Editor Tool, where groups of experts assessed its usability and the supported functionalities, and evaluated the usefulness and the accuracy of the implemented video segmentation approaches based on the analysis requirements of the LinkedTV scenarios. By this deliverable we complete the reporting of WP1 evaluations that aimed to assess the efficiency of the developed multimedia analysis methods throughout the project, according to the analysis requirements of the LinkedTV scenarios.

0 Content

1 Introduction

This deliverable describes the outcomes of the conducted activities for evaluating the performance of a number of developed methods for intelligent hypervideo analysis, and the usability of the implemented tool for supporting video annotation and enrichment. Following the evaluation results for a number of different implemented multimedia analysis technologies that were reported in D1.4, in this deliverable we present the findings of additional evaluation activities, where newer versions of these methods as well as completely new techniques, were tested in terms of accuracy and time efficiency. These assessments were made through in-house experiments, or by participating at international benchmarking activities such as MediaEval [1] and TRECVID [2]. Moreover, we report the results of the realized user trials concerning the developed Editor Tool, where groups of experts assessed its usability and the supported functionalities, and evaluated the usefulness and the accuracy of the implemented chapter segmentation approaches based on the analysis requirements of the LinkedTV scenarios.

The first section of the deliverable concentrates on two different approaches for higher-level video segmentation, namely a newly developed fine-grained chapter segmentation algorithm for the videos of the documentary scenario and an adaptation of a generic scene segmentation algorithm. The first was evaluated via a number of in-house experiments using LinkedTV data, while the second was assessed for its suitability in defining media fragments for hyperlinking, via participating to the Search and Hyperlinking task of MediaEval benchmarking activity. The findings of these evaluations are reported.

The next section is dedicated to the evaluation of the developed method for video annotation with concept labels. The results of our participation to the Semantic Indexing (SIN) task of TRECVID benchmarking activity are initially reported. Subsequently the findings of in-house experiments regarding the impact in algorithm's efficiency after modifying specific parts of the concept detection pipeline, such as the use of binary or non-binary local descriptors and the utilization of correlations between concepts, are presented.

Section 4 deals with our techniques for extended video annotation with event labels. This section is divided in two parts, where the first part is dedicated to our participation at the Multimedia Event Detection (MED) task of TRECVID benchmarking activity, and the second part is related to a new developed approach for video annotation with zero positive samples. The results of a number of evaluations regarding the performance of these methods are described.

Subsequently, in section 5 we focus on our method for object re-detection in videos. Based on the algorithm presented in section 8.2 of D1.2, we present our efforts in developing an extension of this method motivated by the need to further accelerate the instance-based spatiotemporal labeling of videos. Each introduced modification and or extension was extensively tested and the evaluation results are presented in this section. Moreover, the implemented approach was tested against a set of different methods for object re-detection and tracking from the literature, through a user study that was based on the use of an implemented on-line tool for object-based labeling of videos. The outcomes of the user study are also presented in detail.

Section 6 is related to the evaluation activities regarding the suitability of the developed Editor Tool. A group of video editors were participated in a user study, aiming to assess: (a) the usability of the tool and its supported functionalities, and (b) the accuracy and usefulness of the automatically defined video segmentations and the provided enrichments. The outcomes of this study are reported in this section.

The deliverable concludes in section 7, with a brief summary of the presented evaluations.

2 Evaluation of temporal video segmentation

2.1 Overview

The temporal segmentation of a video into storytelling parts, by defining the semantically coherent and conceptually distinctive segments of it, is the basis for the creation of self-contained, meaningful media fragments that can be used for annotation and hyperlinking with related content. During the project we developed and tested a variety of techniques for efficient segmentation of videos in different levels of abstraction. Specifically the methodology and the performance (in terms of time efficiency and detection accuracy) of the implemented shot segmentation algorithm (variation of [AM14]) and a set of different approaches for higher-level topic or chapter segmentation, were described in sections 2 and 3 of D1.4. According to the findings of the experiments that are reported there, the developed techniques achieve remarkably high levels of detection accuracy

[1] http://www.multimediaeval.org/

[2] http://trecvid.nist.gov/

(varying between 94% and 100%), while the needed processing time makes them several times faster than real-time analysis.

Besides these temporal segmentation approaches, we also evaluated the efficiency of two other methods. The first is an extension of the chapter segmentation algorithm described in section 3.2.2 of D1.4, which aims to define a more fine-grained segmentation of the videos from the LinkedTV documentary scenario. The efficiency of this method was assessed based on a set of in-house experiments. The second is an adaptation of the scene segmentation algorithm from [SMK$^+$11], that was presented in section 3.6 of D1.1. The performance evaluation of this technique was made by participating in the Search and Hyperlinking task of MediaEval benchmarking activity. The findings regarding the effectiveness of these video segmentation approaches are reported in the following subsections.

2.2 Fine-grained chapter segmentation of videos from the LinkedTV documentary scenario

Based on the already implemented and evaluated chapter segmentation approach for the videos of the LinkedTV documentary scenario, which are episodes from the "Tussen Kunst & Kitsch" show of AVRO [3] (see section 3.2.2 of D1.4), we developed an extended version that defines a more fine-grained segmentation of these videos. The motivation behind this effort was to find a way to perform a more efficient decomposition of these episodes into chapters, where each chapter - besides the intro, welcome and closing parts of the show - is strictly related to the presentation of a particular art object.

The already used chapter segmentation algorithm (the one from section 3.2.2 of D1.4) draws input from the developed shot segmentation method (see section 2.2 of D1.4) and performs shot grouping into chapters based on the detection of a pre-defined visual cue that is used for the transition between successive parts of the show (also termed as "bumper" in the video editing community). This visual cue is the AVRO [4] logo of Fig. 1, and its detection in the frames of the video is performed by applying the object re-detection algorithm of [AMK13], which was also presented in section 8.2 of D1.2. However, this methodology performs a coarse-grained segmentation of the show into (usually) 7-8 big chapters, where the majority of them - besides the intro and the closing parts - contain more than one art objects, as shown in Fig. 2.

Aiming to define a finer segmentation of the show into smaller chapters we tried to identify and exploit other visual characteristics that are related to the structure and the presentation of the show. By this observation-based study we saw that the beginning of the discussion about an art object is most commonly indicated by a gradual transition effect (dissolve). Based on this finding we extended the already used shot segmentation algorithm in order to get information about the gradual transitions between the detected shots of the video. However, this type of transition (dissolve) is also used - more rarely though - during the discussion about an art object, for the brief presentation of another similar or related object. Aiming to minimize the number of erroneously detected chapter boundaries due to the existence of these dissolves, and based on the fact that each chapter is strictly related to one expert of the show, we tried to indicate a timestamp within each chapter of the show, by re-detecting the same visual cue (see Fig. 1) in the appearing banners with the experts' names. The re-detection was based on the same algorithm, looking this time for significantly scaled-down instances of the visual cue.

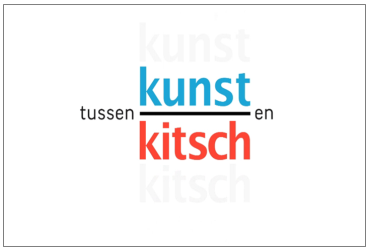


Figure 1 The AVRO logo that is used as visual cue for re-detecting "bumpers" and "banners" of the show.

Having all the visual features described above detected, and using prior knowledge regarding the structure of the show, we consider two consecutive dissolves as the starting and ending boundaries of a chapter of the show only in the case that an expert's appearance (i.e., the "banner" with the expert's name) was detected in

[3] http://www.avro.tv/
[4] http://www.avro.tv/

a timestamp between them. Based on this methodology we result in a more fine-grained segmentation of the show, defining chapters that focus on the presentation of a single art object, as presented in Fig. 3.

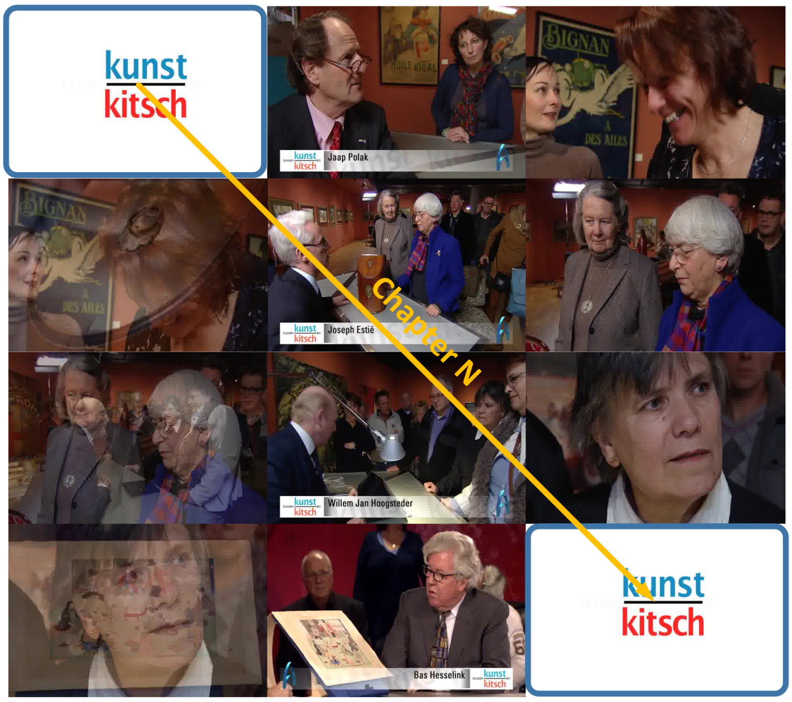


Figure 2 Chapter segmentation based on the re-detection of "bumpers" (highlighted by the blue bounding boxes).

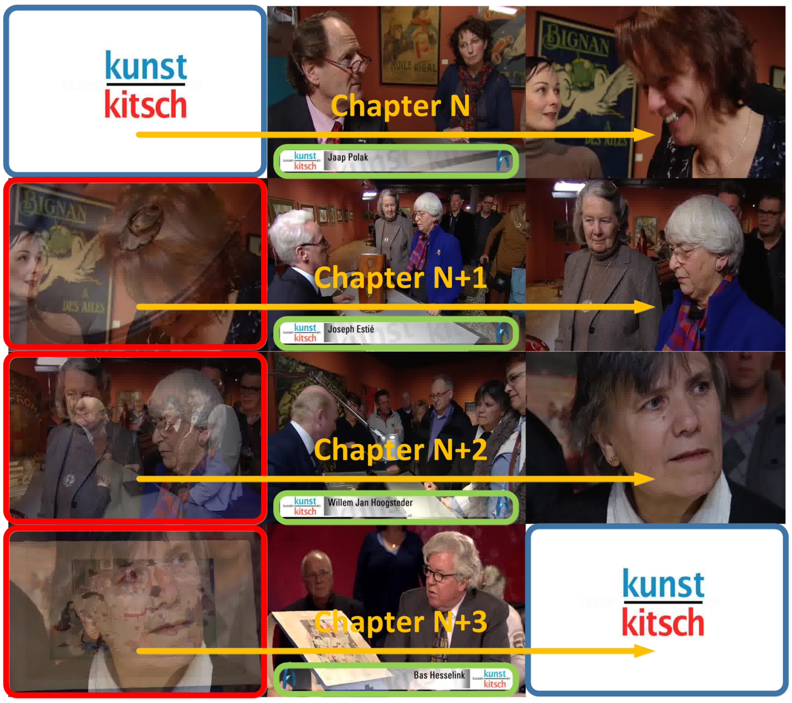


Figure 3 Chapter segmentation based on the re-detection of the "bumpers" (highlighted by the blue bounding boxes), "banners" (highlighted by the green bounding boxes) and dissolves (highlighted by the red bounding boxes).

For evaluating the performance of the new chapter segmentation approach, we tested its detection accuracy using a set of 13 videos from the LinkedTV documentary scenario with total duration 350 min. and 304 manually defined chapters based on human observation. The outcomes from this evaluation are presented in Tab. 1, where for each video we report: (a) the number of actual chapter boundaries based to the created ground-truth (2nd column), (b) the number of correctly detected chapter boundaries by the applied method (3rd column), (c) the number of misdetected chapter boundaries (4th column), (d) the number of erroneously detected chapter boundaries, and (e) the number of correctly defined chapters according to the requirements of the LinkedTV documentary scenario, i.e., parts of the show where a single art object is discussed.

By further elaborating on the computed total values (last row of Tab. 1), we result in the values of Tab. 2 that illustrate the overall performance of the developed algorithm. As can be seen the designed method performs reasonably well, achieving Precision, Recall and F-score values around 0.9, while almost 80% of the manually defined chapter segments have been successfully formed by the conducted automatic analysis.

Concerning the time efficiency, the algorithm runs 3 times faster than real-time processing and is a bit slower than the chapter segmentation approach described in section 3.2.2 of D1.4, which runs 4 times faster than real-time analysis. The latter is explained by the fact that the object re-detection algorithm, that requires 10% of video duration for processing, is applied twice in the new method for the re-detection of the "bumpers" and the "banners" of the show, while another 13.5% of the video duration is needed, as before, for performing the initial shot segmentation of the video.

Table 1 Evaluation results of the fine-grained chapter segmentation algorithm.

Video ID	Actual chapter boundaries	Detected chapter boundaries	Misdetected chapter boundaries	False alarms	Correctly formed chapters
1	25	24	1	1	23
2	23	21	2	4	20
3	24	21	3	0	19
4	23	20	3	0	18
5	24	20	4	0	17
6	24	24	0	1	24
7	24	21	3	3	18
8	22	18	4	2	15
9	22	18	4	3	15
10	24	20	4	2	16
11	25	23	2	2	21
12	22	18	4	3	16
13	22	21	1	1	20
Total	**304**	**269**	**35**	**22**	**242**

Table 2 Performance of the fine-grained chapter segmentation algorithm in terms of Precision, Recall, F-score and the percentage of appropriately formed chapter segments.

Evaluation Metrics			
Precision	Recall	F-Score	Correct Chapters
0.924	0.885	0.904	79.6%

2.3 Assessment of the suitability of different temporal granularities for hyperlinking - the MediaEval benchmarking activity

The LinkedTV team, composed by a group of organizations from the LinkedTV consortium that are responsible for providing multimedia analysis technologies (i.e., partners associated to WP1 and WP2), participated in the Search and Hyperlinking task of MediaEval 2013 benchmarking activity [5]. As briefly described by its name, this task aims to evaluate the performance of different technologies that can be utilized for searching video fragments and for providing links to related multimedia content, in a way similar to the widely applied text-based search and navigation to relevant information via following manually proposed hyperlinks. The

[5] http://www.multimediaeval.org/mediaeval2013/hyper2013/

only difference here (a reasonable one since we are looking for media items) is that the initial search relies on textual information that briefly describes both the semantics and the visual content of the needed fragments.

Through the Hyperlinking sub-task, where the goal was to search within a multimedia collection for content related to a given media fragment, we assessed the efficiency of the scene segmentation technique of [SMK$^+$11], in decomposing the videos into meaningful and semantically coherent parts that are closely related to the human information needs and can be used for establishing links between relevant media. The results of this evaluation are reported in [AMS$^+$14]. This technique takes as input the shot segments of the video (in our case defined automatically by the shot segmentation method that was presented in section 2 of D1.4) and groups them into sets that correspond to individual scenes of the video based on content similarity and temporal consistency among shots. Content similarity in our experiments means visual similarity, and the latter was assessed by computing and comparing the HSV (Hue-Saturation-Value) histograms of the keyframes of different shots. Visual similarity and temporal consistency are jointly considered during the grouping of the shots into scenes, with the help of two extensions of the well known Scene Transition Graph (STG) algorithm [YYL98]. The first extension reduces the computational cost of STG-based shot grouping by considering shot linking transitivity and the fact that scenes are by definition convex sets of shots. The second one builds on the former to construct a probabilistic framework that alleviates the need for manual STG parameter selection.

The organizers of the task provided a dataset composed by 1667 hours of video (2323 videos from BBC broadcasting company) of various content, such as news shows, talk shows, sitcoms and documentaries. For each video manually transcribed subtitles, created transcripts based on automatic speech recognition (ASR), textual metadata and automatically detected shot boundaries and keyframes were also given. Aiming to exploit all these different types of information we designed and developed a framework for multi-modal analysis of multimedia content and for indexing the analysis results in a way that supports the retrieval of the appropriate (i.e., relevant) media fragments. As illustrated in Fig. 4, the proposed framework consists of two main components. The analysis component includes all the utilized analysis modules, namely our methods for (a) shot segmentation, (b) scene segmentation, (c) optical character recognition, (d) visual concept detection and (e) keyword extraction. The storage component contains the data storage and indexing structures that facilitate the retrieval and linking of related media fragments. The technologies that form the analysis component of this framework correspond to different methods that were developed and extensively tested throughout the project (as reported in deliverables D1.2 and D1.4), in a bid to fulfil the analysis requirements of the LinkedTV scenarios. The produced analysis results, along with the subtitles and metadata of the videos, are indexed using the storage component of the framework, which is based on the Solr/Lucene[6] platform and creates indexes that contain data at two granularities: the video level and the scene level.

Besides the suitability and efficiency of the scene segmentation algorithm for the creation of well-defined media fragments for hyperlinking as will be explained below, there is another analysis module that improved the searching performance of the proposed framework when looking for a specific video segment, either within the scope of the Searching sub-task of the activity, or for finding related media. This module is the applied visual concept detection method, and the way that the analysis results were integrated into the media searching process of the implemented framework, as well as the achieved improvements from exploiting these results, are described in D2.7. The importance of extracting and exploiting information about visual concepts was also highlighted through our participation in the next year's Search and Hyperlinking task [7]. The used framework was similar to the one described above, integrating improved or extended versions of the same technologies. However, based on the description of the task no visual cues about the needed segments were provided by the organizers this year. As presented in [LBH$^+$14] this lack of information about the visual content of the fragments that need to be defined, either for searching or for hyperlinking purposes, resulted in a notable reduction of the performance.

For the needs of the Hyperlinking task the organizers defined a set of 30 “anchors” (media fragments described by the video's name and their start and end times; thus, no further temporal segmentation of them is necessary), which are used as the basis for seeking related content within the provided collection. For each “anchor”, a broader yet related temporal segment with contextual information about the “anchor”, called “context”, was also defined. For evaluating the hyperlinking performance, Precision @ k (which counts the number of relevant fragments within the top k of the ranked list of hyperlinks, with k being 5, 10 and 20) was used. Moreover, three slightly different functions were defined for measuring the relevance of a retrieved segment; the “overlap relevance”, which considers the temporal overlap between a retrieved segment and the actual one; the “binned relevance”, which assigns segments into bins; and the “tolerance to irrelevance”, which takes into account only the start times of the retrieved segments [AEO$^+$13].

[6] http://lucene.apache.org/solr/

[7] http://www.multimediaeval.org/mediaeval2014/hyper2014/

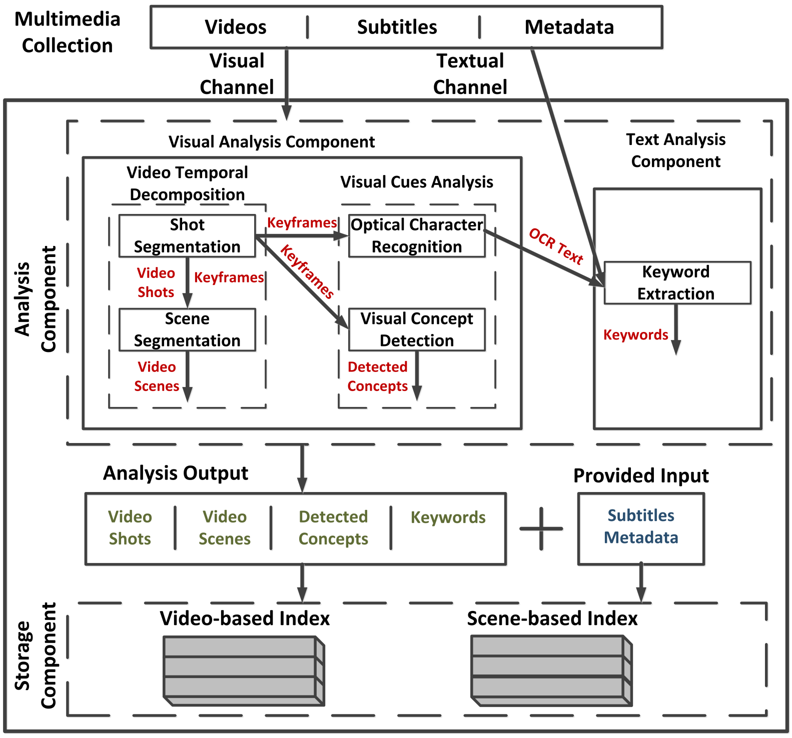


Figure 4 The proposed framework for multi-modal analysis and indexing, that supports the retrieval and hyperlinking of media fragments.

Given a pair of “anchor” and “context” fragments the proposed framework initially creates automatically two queries, one using only the “anchor” information and another one using both “anchor” and “context” information, which are going to be applied on the created indexes. These queries are defined by extracting keywords from the subtitles of the “anchor”/“context” fragments, and by applying visual concept detection. The latter is performed on the keyframes of all shots of the corresponding media fragment and its results are combined using max pooling (i.e., keeping for each concept the highest confidence score). Our framework then applies these queries on the video-level index; this step filters the entire collection of videos, resulting in a much smaller set of potentially relevant videos. Using the new limited set of videos, the same queries are applied on the scene-level index, and a ranked list with the scenes that were identified as the most relevant ones is returned, forming the output of the proposed system.

Fig. 5 to 7 illustrate the best mean performance of each participating team in MediaEval 2013 (the methodology of each team can be studied in detail by following the related citations in the Tables 3 to 5 below), in terms of Precision @ k using the “overlap relevance” metric when only the “anchor” or the “anchor” and “context” information is exploited, also indicating (by color) which were the segmentation units utilized by each approach. As shown, when only the “anchor” is known, our proposed approach exhibits the highest performance for k equal to 5 or 10, while it is among the top-2 highest performers when “context” information is also included or for k equal to 20. Specifically, the k-th first items (hyperlinks) proposed by our system to the user are very likely to include the needed media fragment (over 80% for the top-5, over 70% for the top-10 and over 50% for the top-20). Moreover, the comparison of the different video decomposition approaches shows that the visual-based segmentation techniques (scene or shot segmentation) are more effective than other speech-based, text-based or fixed-window segmentation methods.

The competitiveness of the developed hyperlinking approach is also highlighted in Tables 3, 4 and 5. These tables contain the best scores of each participating team for the Mean Precision @ 5, 10 and 20 measures, according to the different defined relevance functions (highest scores are in bold font; a dash means that no run was submitted to MediaEval 2013). As shown, the proposed framework achieves the best performance in 15 out of 18 cases, while it is among the top-3 in the remaining 3. Moreover, we also ran an experiment with a variation of our approach that used a simple temporal window (defined by grouping shots that are no more than 10 sec. apart) for determining the temporal segments used for hyperlinking, instead of the outcome of scene segmentation (last row of Tables 3 to 5). The comparison indicates once again that automatically detected scenes are more meaningful video fragments for hyperlinking, compared to simpler

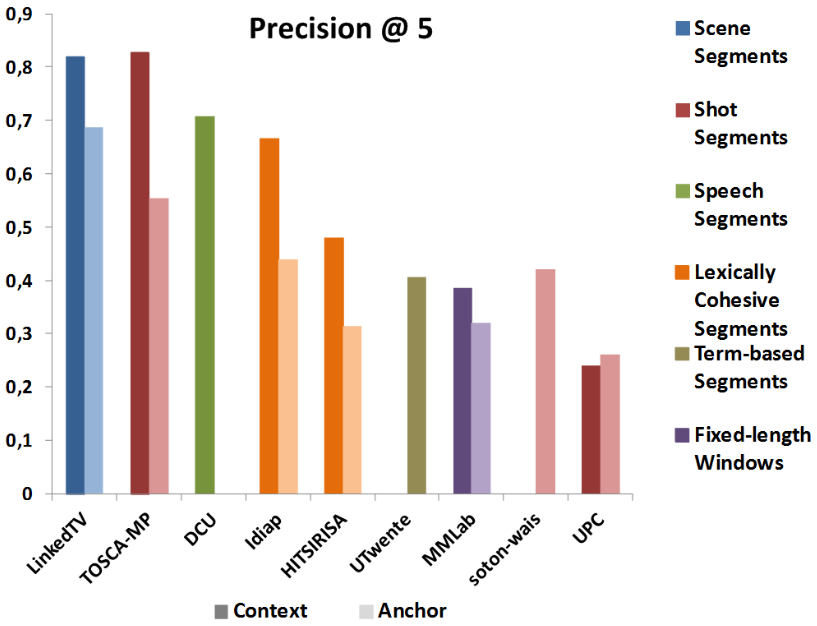


Figure 5 The best mean performance of each participating approach in the Hyperlinking sub-task of MediaEval 2013 in terms of Precision @ 5 using the "overlap relevance" metric, also in relation to the segmentation unit employed by each team (see legend on the right).

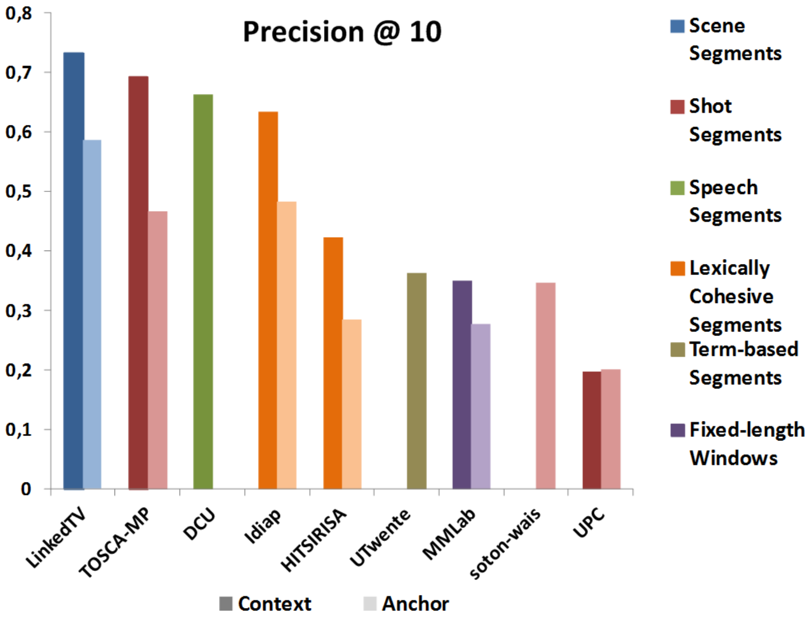


Figure 6 The best mean performance of each participating approach in the Hyperlinking sub-task of MediaEval 2013 in terms of Precision @ 10 using the "overlap relevance" metric, also in relation to the segmentation unit employed by each team (see legend on the right).

temporal segmentations (e.g., windowing).

In total, our participation to the Hyperlinking sub-task of MediaEval 2013 showed that the proposed framework that relies on a subset of LinkedTV technologies for multimedia analysis and storage, exhibited competitive performance compared to the approaches of the other participants. Moreover, the evaluation results clearly indicate that video scene segmentation can provide more meaningful segments, compared to other decomposition methods, for hyperlinking purposes.

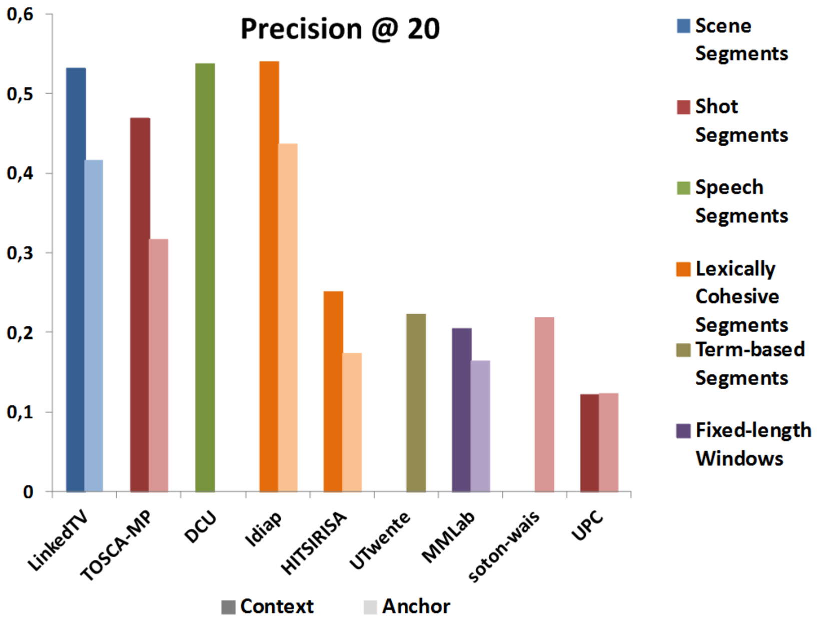


Figure 7 The best mean performance of each participating approach in the Hyperlinking sub-task of MediaEval 2013 in terms of Precision @ 20 using the “overlap relevance” metric, also in relation to the segmentation unit employed by each team (see legend on the right).

Table 3 The best Mean Precision @ 5 scores (for the different relevance measures) for the teams participating to the Hyperlinking sub-task of MediaEval 2013, using “anchor” and “context” information.

Precision @ 5	Overlap Relevance		Binned Relevance		Tolerance to Irrelevance	
	Context	Anchor	Context	Anchor	Context	Anchor
LinkedTV	0.8200	**0.6867**	**0.7200**	**0.6600**	**0.6933**	**0.6133**
TOSCA-MP [BLS14]	**0.8267**	0.5533	0.5400	0.5333	0.5933	0.5133
DCU [CJO13]	0.7067	-	0.6000	-	0.5333	-
Idiap [BPH+14]	0.6667	0.4400	0.6333	0.5000	0.4600	0.3867
HITSIRISA [GSG+13]	0.4800	0.3133	0.4600	0.3400	0.4667	0.3133
Utwente [SAO13]	-	0.4067	-	0.3933	-	0.3600
MMLab [NNM+13]	0.3867	0.3200	0.3867	0.3267	0.3667	0.3067
soton-wais [PHS+13]	-	0.4200	-	0.4000	-	0.3400
UPC [VTAiN13]	0.2400	0.2600	0.2400	0.2600	0.2333	0.2467
Windowing	0.5733	0.4467	0.6067	0.5000	0.4600	0.3467

2.4 Conclusion

A variety of different methods, either generic, such as the shot segmentation algorithm of [AM14] and the scene segmentation method of [SMK+11], or scenario-specific, such as the chapter/topic segmentation approaches for the LinkedTV content, presented in section 3 of D1.4 and here, have been developed and extensively tested for their performance throughout the project. The findings of the conducted evaluations, that were based on in-house experiments and participations at a strongly-related task of an international benchmarking activity, clearly indicate their suitability for decomposing videos into different levels of abstraction and for the definition of semantically meaningful media fragments that can be considered as self-contained entities for media annotation and hyperlinking.

Table 4 The best Mean Precision @ 10 scores (for the different relevance measures) for the teams participating to the Hyperlinking sub-task of MediaEval 2013, using "anchor" and "context" information.

Precision @ 10	Overlap Relevance		Binned Relevance		Tolerance to Irrelevance	
	Context	Anchor	Context	Anchor	Context	Anchor
LinkedTV	**0.7333**	**0.5867**	**0.6333**	**0.5467**	**0.6367**	**0.5133**
TOSCA-MP [BLS14]	0.6933	0.4667	0.3867	0.4333	0.4433	0.4200
DCU [CJO13]	0.6633	-	0.5667	-	0.4667	-
Idiap [BPH$^+$14]	0.6333	0.4833	0.5167	0.4867	0.4433	0.4033
HITSIRISA [GSG$^+$13]	0.4233	0.2833	0.4100	0.3000	0.4100	0.2733
Utwente [SAO13]	-	0.3633	-	0.3500	-	0.3267
MMLab [NNM$^+$13]	0.3500	0.2767	0.3500	0.2800	0.3233	0.2600
soton-wais [PHS$^+$13]	-	0.3467	-	0.3267	-	0.2900
UPC [VTAiN13]	0.1967	0.2000	0.1967	0.1900	0.1933	0.1900
Windowing	0.4833	0.3200	0.5333	0.3733	0.4000	0.2533

Table 5 The best Mean Precision @ 20 scores (for the different relevance measures) for the teams participating to the Hyperlinking sub-task of MediaEval 2013, using "anchor" and "context" information.

Precision @ 20	Overlap Relevance		Binned Relevance		Tolerance to Irrelevance	
	Context	Anchor	Context	Anchor	Context	Anchor
LinkedTV	0.5317	0.4167	**0.4900**	**0.3983**	**0.4333**	**0.3400**
TOSCA-MP [BLS14]	0.4683	0.3167	0.2517	0.2800	0.2667	0.2783
DCU [CJO13]	0.5383	-	0.4100	-	0.3200	-
Idiap [BPH$^+$14]	**0.5400**	**0.4367**	0.3850	0.3917	0.3267	0.2900
HITSIRISA [GSG$^+$13]	0.2517	0.1733	0.2450	0.1900	0.2300	0.1650
Utwente [SAO13]	-	0.2233	-	0.2183	-	0.1950
MMLab [NNM$^+$13]	0.2050	0.165	0.2050	0.1667	0.1867	0.1517
soton-wais [PHS$^+$13]	-	0.2183	-	0.2083	-	0.1683
UPC [VTAiN13]	0.1217	0.1233	0.1267	0.1183	0.1133	0.1133
Windowing	0.2767	0.2067	0.3217	0.2450	0.2167	0.1517

3 Evaluation of video annotation with concept labels

3.1 Overview

Visual concept detection was used for extracting the high level semantics from the defined media fragments after analyzing the LinkedTV content with the developed methods for temporal segmentation of videos into shots and chapters. The detected concepts were used for the semantic annotation of these fragments (an outcome of WP1 analysis), while via the LinkedTV serialization process they were integrated in the created RDF metadata files and employed for finding enrichments (an outcome of WP2 analysis). In this section we evaluate many different computationally efficient methods for video annotation with concept labels on large-scale datasets. Firstly, we present the experimental results from our participation in the Semantic Indexing (SIN) task of TRECVID 2014 benchmarking activity, where we use methods described in section 7.2 of D1.4. Then, to improve video concept detection accuracy we enrich our system with more local descriptors, binary and non-binary, but also color extensions of them. In addition, we update our concept detectors with more powerful models by developing context-based algorithms that exploit existing semantic relations among concepts (e.g., the fact that *sun* and *sky* will often appear together in the same video shot). Finally, considering that the state-of-the-art systems in the SIN task achieve high concept detection accuracy by utilizing Deep Convolutional Neural Networks (DCNN) we also introduce DCNN-based features and we develop a cascade architecture of classifiers that optimally combines local descriptors with DCNN-based descriptors and improves computational complexity in training and classification.

3.2 Results of the TRECVID SIN benchmarking activity

Our participation in the TRECVID 2014 SIN task [Gea14] was based on the methods presented in section 7.2 of D1.4. Four runs were submitted and our best run included the following:

1. Keyframes and tomographs as shot samples.
2. Seven local descriptors: SIFT [Low04] and two color extensions of SIFT [VdSGS10], namely RGB-SIFT and OpponentSIFT; SURF [BETVG08] and two color extensions for SURF, inspired by the two color extensions of SIFT [VdSGS10], namely RGB-SURF and OpponentSURF [MPP$^+$15]; a binary local descriptor namely ORB (Oriented FAST and Rotated BRIEF) [RRKB11].
3. The aggregation of the local descriptors using the VLAD encoding [JPD$^+$12] and dimensionality reduction via Random Projection variant.
4. A methodology for building concept detectors, where an ensemble of five Logistic Regression (LR) models, called a Bag of Models (BoMs) in the sequel, is trained for each local descriptor and each concept [Mea13].
5. The introduction of a multi-label classification algorithm in the second layer of the stacking architecture to capture label correlations [MMK14].
6. Temporal re-ranking to refine the scores of neighboring shots [SQ11].

The other runs were variations of our best run (e.g., using less local descriptors, removing the second layer etc.). Fig. 8 shows the overview of results of the TRECVID 2014 SIN task in terms of MXinfAP [YKA08], which is an approximation of MAP, suitable for the partial ground truth that accompanies the TRECVID dataset [Oea13]. Our best run scored 0.207 in terms of MXinfAP, and was ranked 27th among all runs and 10th among the 15th participating teams. The other runs achieved MXinfAP of 0.205, 0.190, and 0.081, respectively. It should be noted that TRECVID categorizes the submitted runs based on the type of training data they use (i.e., category A runs: training data provided by TRECVID, category D runs: category A data and also other external data, category E runs: only external training data). Our team submitted only category A runs and our best run was ranked 3rd out of 11 participating category A teams.

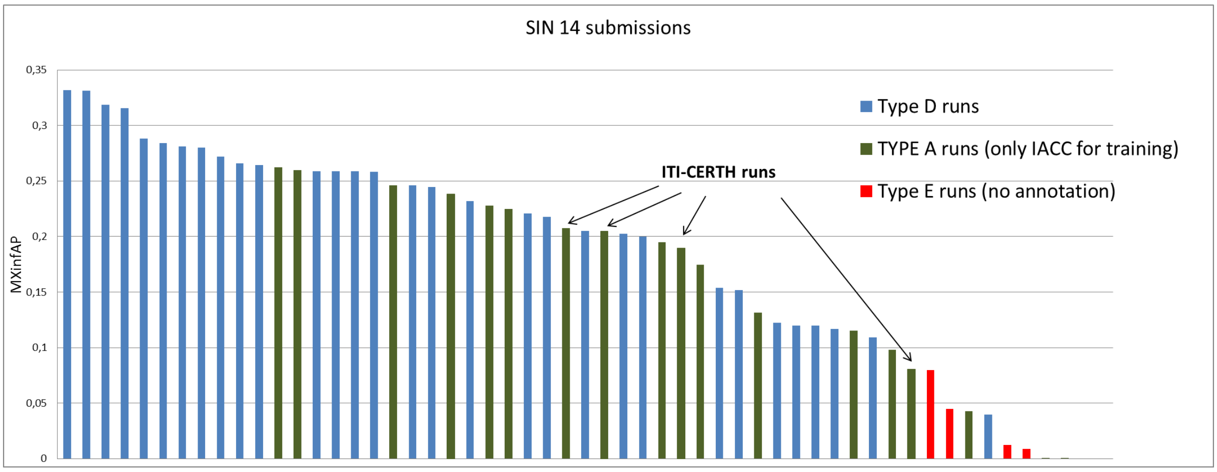


Figure 8 Mean Extended Inferred Average Precision (MXinfAP) by run submitted to TRECVID 2014 SIN task.

The reason for this moderate performance is mainly because we made design choices that favor speed of execution over accuracy. First of all, with respect to category A runs, while we perform dimensionality reduction to the employed VLAD vectors, both [Iea14] and [Jea14], the two teams that outperform us, utilized Fisher Vectors and GMM supervectors, respectively, without performing dimensionality reduction. Secondly, we use linear LR to train our concept detectors in contrast to [Iea14] and [Jea14] that employed more accurate but also more expensive kernel methods. Finally, we only utilize visual local descriptors in contrast to [Jea14] that also uses motion features. With respect to category D runs, all the runs that outperform us use DCNN. Specifically, features extracted from one or more hidden layers were used as image representations, which significantly improves concept detection accuracy. Finally, it should be noted that after submission we discovered a bug in the normalization process of the VLAD vectors and on the training of the multi-label learning algorithm. After fixing this bug the MXinfAP for each run is expected to be increased by approximately 0.03. Overall, considering our best run, our system performed slightly above the median for 23 out of 30 evaluated concepts, as shown in Fig. 9. This good result was achieved despite the fact that we made design choices that favor speed of execution over accuracy (use of linear LR, dimensionality reduction of VLAD vectors).

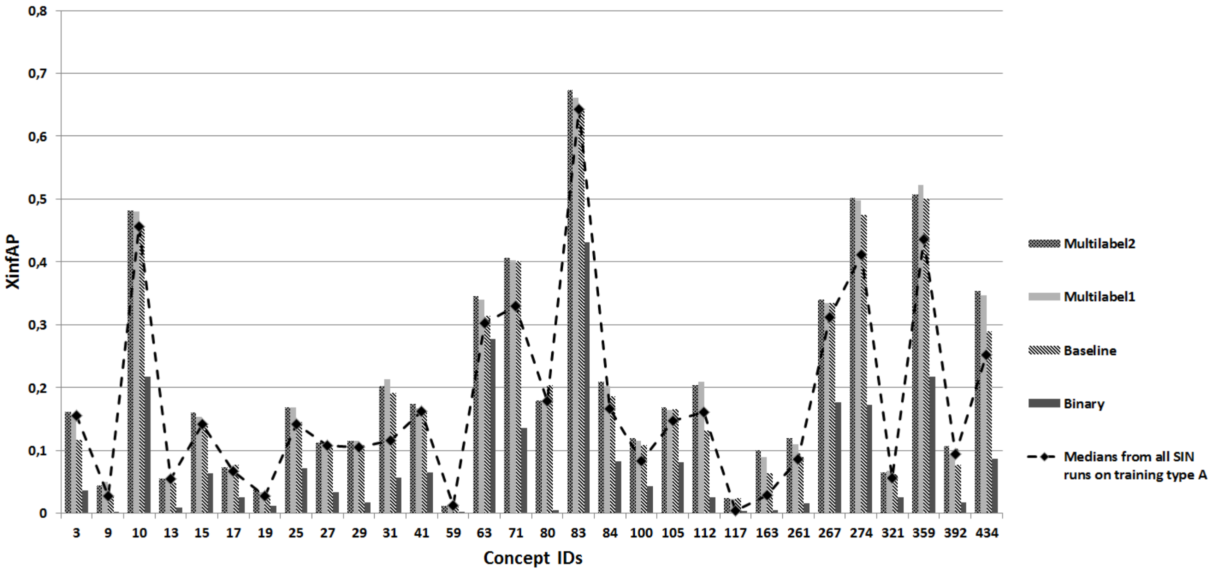


Figure 9 Extended Inferred Average Precision (XinfAP) per concept for our submitted runs.

3.3 Binary vs. non-binary local descriptors and their combinations

In this section we show how binary local descriptors can be used for video concept detection. Specifically, besides the ORB descriptor [RRKB11] that was also presented in section 7.2.2 of D1.4, we evaluated one more binary descriptor, termed BRISK [LCS11], and we examined how the two state-of-the-art binary local descriptors can facilitate concept detection. Furthermore, based on the good results of the methodology for introducing color information to SURF and ORB [MPP$^+$15], which has been described in section 7.2.2 of D1.4, we examined the impact of using the same methodology for introducing color information to BRISK. Finally, we present many different combinations of binary and non-binary descriptors, and distinguish those that lead to improved video concept detection. Generally, this section reports the experimental evaluation of the independent concept detectors that were presented in D1.4, covering different possibilities for using these descriptors (e.g., considering the target dimensionality of PCA (Principal Component Analysis) prior to the VLAD encoding of binary descriptors), evaluating several additional descriptor combinations, and extending the evaluation methodology so as to cover not only the problem of semantic video indexing within a large video collection, as in our previous conference paper [MPP$^+$15], but also the somewhat different problem of individual video annotation with semantic concepts.

Our experiments were performed on the TRECVID 2013 Semantic Indexing (SIN) dataset [Oea13], which consists of a development set and a test set (approx. 800 and 200 hours of internet archive videos, comprising more than 500000 and 112677 shots, respectively). We evaluated all techniques on the 2013 test set, for the 38 concepts for which NIST provided ground truth annotations [Oea13].

Our target was to examine the performance of the different methods both on the video indexing and on the video annotation problem. Based on this, we adopted two evaluation strategies: i) Considering the video indexing problem, given a concept we measure how well the top retrieved video shots for this concept truly relate to it. ii) Considering the video annotation problem, given a video shot we measure how well the top retrieved concepts describe it. For the indexing problem we calculated the Mean Extended Inferred Average Precision (MXinfAP) at depth 2000 [YKA08], which is an approximation of the Mean Average Precision (MAP) that has been adopted by TRECVID [Oea13]. For the annotation problem we calculated the Mean Average Precision at depth 3 (MAP@3). In the latter case, our evaluation was performed on shots that are annotated with at least one concept in the ground truth.

Table 6 Performance (MXinfAP, %, and MAP@3, %) for the different descriptors and their combinations, when typical and channel-PCA is used for dimensionality reduction. In parenthesis we show the relative improvement w.r.t. the corresponding original grayscale local descriptor for each of the SIFT, SURF, ORB, BRISK color variants.

		MXinfAP (indexing)			MAP@3 (annotation)		
Descriptor	Descriptor size in bits	Keyframes, typical-PCA	Keyframes, channel-PCA	Boost(%) w.r.t typical-PCA	Keyframes, typical-PCA	Keyframes, channel-PCA	Boost(%) w.r.t typical-PCA
SIFT	1024	14.22	14.22	-	74.32	74.32	-
RGB-SIFT	3072	14.97 (+5.3%)	14.5 (+2.0%)	-3.1%	74.67 (+0.5%)	74.07 (-0.3%)	-0.8%
OpponentSIFT	3072	14.23 (+0.1%)	14.34 (+0.8%)	+0.8%	74.54 (+0.3%)	74.53 (+0.3%)	0.0%
All SIFT (SIFTx3)	**-**	**19.11 (+34.4%)**	**19.24 (+35.3%)**	**+0.7%**	**76.47 (+2.9%)**	**76.38 (+2.8%)**	**-0.1%**
SURF	1024	14.68	14.68	-	74.25	74.25	-
RGB-SURF	3072	15.71 (+7.0%)	15.99 (+8.9%)	+1.8%	74.58 (+0.4%)	74.83 (+0.8%)	+0.3%
OpponentSURF	3072	14.7 (+0.1%)	15.26 (+4.0%)	+3.8%	73.85 (-0.5%)	74.07 (-0.2%)	+0.3%
All SURF (SURFx3)	**-**	**19.4 (+32.2%)**	**19.48 (+32.7%)**	**+0.4%**	**75.89 (+2.2%)**	**76.12 (+2.5%)**	**0.3%**
ORB 256 (no PCA)	256	10.36	10.36	-	71.05	71.05	-
RGB-ORB 256	768	13.02 (+25.7%)	13.58 (+31.1%)	+4.3%	72.86 (+2.6%)	73.21 (+3.0%)	+0.5%
OpponentORB 256	768	12.61 (+21.7%)	12.73 (+22.9%)	+1.0%	72.66 (+2.3%)	72.46 (+2.0%)	-0.3%
All ORB 256	**-**	**16.58 (+60.0%)**	**16.8 (+62.2%)**	**+1.3%**	**74.32 (+4.6%)**	**74.20 (+4.4%)**	**-0.2%**
ORB 80	256	11.43	11.43	-	72.02	72.02	-
RGB-ORB 80	768	13.79 (+20.6%)	13.48 (+17.9%)	-2.2%	73.20 (+1.6%)	72.96 (+1.3%)	-0.3%
OpponentORB 80	768	12.81 (+12.1%)	12.57 (+10.0%)	-1.9%	72.56 (+0.7%)	72.01 (0.0%)	-0.8%
All ORB 80 (ORBx3)	**-**	**17.48 (+52.9%)**	**17.17 (+50.2%)**	**-1.8%**	**74.64 (+3.6%)**	**74.58 (+3.6%)**	**-0.1%**
BRISK 256	512	11.43	11.43	-	72.36	72.36	-
RGB-BRISK 256	1536	11.78 (+3.1%)	12 (+5.0%)	+1.9%	72.74 (+0.5%)	72.67 (+0.4%)	-0.1%
OpponentBRISK 256	1536	11.68 (+2.2%)	11.96 (+4.6%)	+2.4%	72.42 (+0.1%)	72.35 (0.0%)	-0.1%
All BRISK 256 (BRISKx3)	**-**	**16.4 (+43.5%)**	**16.47 (+44.1%)**	**+0.4%**	**74.56 (+3.0%)**	**74.58 (+3.1%)**	**0.0%**
BRISK 80	512	10.73	10.73	-	71.79	71.79	-
RGB-BRISK 80	1536	12.21 (+13.8%)	11.6 (+8.1%)	-5.0%	72.70 (+1.3%)	72.29 (+0.7%)	-0.6%
OpponentBRISK 80	1536	11.05 (+3.0%)	11.15 (+3.9%)	+0.9%	72.10 (+0.4%)	71.49 (-0.4%)	-0.9%
All BRISK 80	**-**	**16.43 (+53.1%)**	**15.95 (+48.6%)**	**-2.9%**	**74.51 (+3.8%)**	**74.39 (3.6%)**	**-0.2%**

We started by assessing the performance of detectors in relation to the indexing problem. Table 6 shows the evaluation results for the different local descriptors and their color extensions that were considered in this work, as well as combinations of them. First, by comparing the original ORB and BRISK descriptors with the non-binary ones (SIFT, SURF), we saw that binary descriptors performed a bit worse than their non-binary counterparts but still reasonably well. This satisfactory performance was achieved despite ORB, BRISK and their extensions being much more compact than SIFT and SURF, as seen in the second column of Table 6. Second, concerning the methodology for introducing color information to local descriptors, we saw that the combination of the original SIFT descriptor and the two known color SIFT variants that we examined ("All SIFT" in Table 6) outperforms the original SIFT descriptor alone by 34.4% (35.3% for channel-PCA). The similar combination of the SURF color variants with the original SURF descriptor, is shown in Table 6 to outperform the original SURF by 32.2% (which increases to 32.7% for channel-PCA), and even more pronounced improvements were observed for ORB and BRISK. These results indicate that this relatively straightforward way for introducing color information is in fact generally applicable to heterogeneous local descriptors.

We also compared the performance of each binary descriptor when it is reduced to 256 and to 80 dimensions. By reducing ORB and its color variants to 80 dimensions and by combining them performs better than reducing them to 256 dimensions (both when applying typical- and channel-PCA). On the other hand, by reducing BRISK and its two color variants to 256 dimensions and by combining them gave the best results (in combination with channel-PCA).

In D1.4 (section 7.2.2) we presented two alternatives for performing PCA to local descriptors, namely channel-PCA and typical-PCA. In Table 6 we also compare the channel-PCA with the typical approach of applying PCA directly on the entire color descriptor vector for more local descriptors. In both cases PCA was applied before the VLAD encoding, and in applying channel-PCA we kept the same number of principal components from each color channel (e.g., for RGB-SIFT, which is reduced to $l' = 80$ using typical-PCA, we set $p_1 = p_2 = 27$ for the first two channels and $p_3 = 26$ for the third color channel; $p_1 + p_2 + p_3 = l'$). According to the relative improvement data reported in the fifth column of Table 6 (i.e., for the indexing problem), performing the proposed channel-PCA in most cases improves the concept detection results, compared to the typical-PCA alternative, without introducing any additional computational overhead.

According to Table 6, for each local descriptor, the combination with its color variants that presents the highest MXinfAP is the following: SIFTx3 with channel-PCA, SURFx3 with channel-PCA, ORBx3 with typical-PCA, BRISKx3 with channel-PCA. In Table 7 we further combine the above to examine how heterogeneous descriptors would work in concert. From these results we can see that the performance increases when pairs of local descriptors (including their color extensions) are combined (i.e., SIFTx3+SURFx3, SIFTx3+ORBx3, SIFTx3+BRISKx3 etc.), which shows a complementarity in the information that the different local descriptors capture. The performance further increases when triplets of different descriptors are employed, with the best combination being SIFTx3+SURFx3+ORBx3. Combining all four considered local descriptors and their color variants did not show in our experiments to further improve the latter results.

Table 7 Performance (MXinfAP, % ; MAP@3, %) of pairs and triplets of the best combinations of Table 6 descriptors (SIFTx3 channel-PCA, SURFx3 channel-PCA, ORBx3 typical-PCA, BRISKx3 channel-PCA).

(a) Descriptor pairs	+SURFx3	+ORBx3	+BRISKx3
SIFTx3	**22.4**; 76.64	21.31; **76.81**	20.71; 76.53
SURFx3		21.6; 76.43	21.13; 76.68
ORBx3			19.08; 75.34

(b) Descriptor triplets	+ORBx3	+BRISKx3
SIFTx3+SURFx3	**22.9**; 77.29	22.52; **77.39**
SIFTx3+ORBx3		21.5; 76.61
SURFx3+ORBx3		21.76; 76.56

In Table 8 we present the improved scores of selected entries from Tables 6 and 7 after exploiting the method of video tomographs [SMK14], a technique that was firstly described in section 4.2.1 of D1.2 and was also reported among the employed components of the concept detection approach that was presented in section 7.2.2 of D1.4 (for simplicity, these tomographs are described using only SIFT and its two color extensions). The results of Table 8 indicate that introducing temporal information (through tomographs) can give an additional 7.3% relative improvement to the best results reported in Table 7 (MXinfAP increased from 22.9 to 24.57).

Table 8 Performance (MXinfAP, %, and MAP@3, %) for the best combinations of local descriptors (SIFTx3 channel-PCA, SURFx3 channel-PCA, ORBx3 typical-PCA, BRISKx3 channel-PCA). (a) When features are extracted only from keyframes, (b) when horizontal and vertical tomographs [SMK14] are also examined.

	MXinfAP (indexing)			MAP@3 (annotation)		
Descriptor	(a) Keyframes	(b) Keyframes+ Tomographs	Boost (%) w.r.t (a)	(a) Keyframes	(b) Keyframes+ Tomographs	Boost (%) w.r.t (a)
SIFTx3	19.24	20.28	+5.4%	76.38	76.30	-0.1%
SURFx3	19.48	19.74	+1.3%	76.12	75.98	-0.2%
BRISKx3	16.47	19.08	+15.8%	74.58	75.26	+0.9%
ORBx3	17.48	19.24	+10.1%	74.64	75.16	+0.7%
SIFTx3+SURFx3+ORBx3	22.9	24.57	+7.3%	77.29	77.79	+0.7%

Concerning the performance of independent detectors with respect to the annotation problem, for which results are also presented in Tables 6, 7 and 8, similar conclusions can be made regarding the usefulness of ORB and BRISK descriptors, and how color information is introduced to SURF, ORB and BRISK descriptors. Concerning channel-PCA, in this case it does not seem to affect the system's performance: the differences between detectors that use the typical-PCA and channel-PCA are marginal. Another important observation is that in Tables 6, 7 and 8 a significant improvement of the MXinfAP (i.e., of the indexing problem results) does not lead to a correspondingly significant improvement of results on the annotation problem.

3.4 Use of concept correlations in concept detection

In this section we evaluate our stacking approach, described in D1.4 (section 7.2.2), that introduces a second layer on the concept detection pipeline in order to capture concept correlations [MMK14]. According to this approach concept score predictions are obtained from the individual concept detectors in the first layer, in order to create a *model vector* for each shot. These vectors form a meta-level training set, which is used to train a multi-label learning algorithm. Our stacking architecture learns concept correlations in the second layer both from the outputs of first-layer concept detectors and by modeling correlations directly from the ground-truth annotation of a meta-level training set. Similarly to the previous section, our experiments were performed on the TRECVID 2013 Semantic Indexing (SIN) dataset [Oea13], where again we wanted to examine the performance of the different methods both on the video indexing and on the video annotation problem. We further used the TRECVID 2012 test set (approx. 200 hours; 145634 shots), which is a subset of the 2013 development set, as a validation set to train algorithms for the second layer of the stack.

We instantiated the second layer of the proposed architecture with four different multi-label learning algorithms and will refer to our framework as P-CLR , P-LP, P-PPT and P-MLkNN when instantiated with CLR [FHLB08], LP [TKV10], PPT [Rea08] and ML-kNN [ZZ07] respectively. The value of l for P-PPT was set to 30. We compared these instantiations of the proposed framework against five SoA stacking based methods: BCBCF [JCL06], DMF [SNN03], BSBRM [Tea09], MCF [WC12] and CF [HMQ13]. For BCBCF we used the concept predictions instead of the ground truth in order to form the meta-learning dataset, as this was shown to improve its performance in our experiments; we refer to this method as CBCFpred in the sequel. Regarding the concept selection step we selected these parameters: λ = 0.5, θ = 0.6, η = 0.2, γ = the mean of Mutual Information values. For MCF we only employed the spatial cue, so temporal weights have been

Table 9 Performance, (MXinfAP (%), MAP@3 (%) and CPU time), for the methods compared on the TRECVID 2013 dataset. The meta-learning feature space for the second layer of the stacking architecture is constructed using detection scores for (I) 346 concepts and (II) a reduced set of 60 concepts. CPU times refer to mean training (in minutes) for all concepts, and application of the trained second-layer detectors on one shot of the test set (in milliseconds). Columns (a) and (c) show the results of the second layer detectors only. Columns (b) and (d) show the results after combining the output of first and second layer detectors, by means of arithmetic mean. "Baseline" denotes the output of the independent concept detectors that constitute the first layer of the stacking architecture (i.e., the best detectors reported in Table 8). In parenthesis we show the relative improvement w.r.t. the baseline.

Method	MXinfAP (indexing)		MAP@3 (annotation)		
	2nd layer (a)	1st and 2nd layer combination (b)	2nd layer (c)	1st and 2nd layer combination (d)	Mean Exec. Time Training/Testing (e)
Baseline	24.57	24.57	77.79	77.79	N/A
	(I) Using the output of 346 concepts' detectors for meta-learning				
DMF [SNN03]	23.97 (-2.4%)	25.38 (+3.3%)	78.71 (+1.2%)	79.12 (+1.7%)	27.62/0.61
BSBRM [Tea09]	24.7 (+0.5%)	24.95 (+1.5%)	79.31 (+2.0%)	79.06 (+1.6%)	1.02/0.08
MCF [WC12]	24.33 (-1.0%)	24.53 (-0.2%)	76.14 (-2.1%)	77.31 (-0.6%)	1140.98/0.22
CBCFpred [JCL06]	24.32(-1.0%)	24.56 (0%)	78.95 (1.5%)	78.39 (0.8%)	26.84/0.27
CF [HMQ13]	23.34 (-5.0%)	25.27 (+2.8%)	78.13 (+0.4%)	78.81 (+1.3%)	55.24/1.22
P-CLR	14.01 (-43.0%)	24.52 (-0.2%)	79.17 (+1.8%)	79.26 (+1.9%)	49.40/9.85
P-LP	**25.23 (+2.7%)**	**25.6 (+4.2%)**	**80.88 (+4.0%)**	79.06 (+1.6%)	549.40/24.93
P-PPT	23.8 (-3.1%)	24.94 (+1.5%)	79.39 (+2.1%)	78.3 (+0.7%)	392.49/0.03
P-MLkNN	19.38 (-21.1%)	24.56 (0.0%)	77.55 (-0.3%)	**79.64 (+2.4%)**	607.40/273.80
	(II) Using the output of a subset of the 346 concepts' detectors (60 concepts) for meta-learning				
DMF [SNN03]	24.32 (-1.0%)	25.04 (+1.9%)	79.47 (+2.2%)	79.19 (+1.8%)	2.64/0.30
BSBRM [Tea09]	24.71 (+0.6%)	24.96 (+1.6%)	79.82 (+2.6%)	79.26 (+1.9%)	0.65/0.08
MCF [WC12]	**24.85 (+1.1%)**	24.74 (+0.7%)	77.84 (+0.1%)	77.88 (+0.1%)	466.69/0.18
CBCFpred [JCL06]	15.66 (-36.3%)	22.41 (-8.8%)	79.58 (+2.3%)	79.01 (+1.6%)	2.42/0.25
CF [HMQ13]	24.8 (+0.9%)	25.18 (+2.5%)	79.02 (+1.6%)	79.04 (+1.6%)	5.28/0.60
P-CLR	16.16 (-34.2%)	24.44 (-0.5%)	78.85 (+1.4%)	79.12 (+1.7%)	6.32/5.82
P-LP	23.85 (-2.9%)	**25.28 (+2.9%)**	**80.22 (+3.1%)**	79.04 (+1.6%)	208.9/41.43
P-PPT	24.12 (-1.8%)	24.96 (+1.6%)	79.6 (+2.3%)	78.45 (+0.8%)	90.13/0.31
P-MLkNN	22.21 (-9.6%)	24.94 (+1.5%)	77.68 (-0.1%)	**79.42 (+2.1%)**	167.40/72.54

set to zero. The ϕ coefficient threshold, used by BSBRM, was set to 0.09. Finally, for CF we performed two iterations without temporal re-scoring (TRS). We avoided using TRS in order to make this method comparable to the others. For implementing the above techniques, the WEKA [WF05] and MULAN [TSxVV11] machine learning libraries were used as the source of single-class and multi-label learning algorithms, respectively. The reader can refer to [MMK14] for a detailed description of the above algorithms.

In Table 9 we report results of the proposed stacking architecture and compare with other methods that exploit concept correlations. As first layer of the stack we used the best-performing independent detectors of Table 8 (i.e., the last line of Table 8, fusing keyframes and tomographs). We start the analysis with the upper part of Table 9, where we used the output of such detectors for 346 concepts.

In relation to the indexing problem (Table 9:(a),(b)), we observed that the second layer concept detectors alone do not perform so well; in many cases they are not able to outperform the independent first layer detectors (baseline). However, when the concept detectors of the two layers are combined (Table 9:(b)), i.e., the second layer concept detection scores are averaged with the initial scores of the first layer, the accuracy of all the methods is improved. More specifically, P-LP outperforms all the compared methods, reaching a MXinfAP of 25.6. LP considers each subset of labels (label sets) presented in the training set as a class of a multi-class problem, which seems to be helpful for the stacking architecture. PPT models correlations on a similar manner, however, it prunes away label sets that occur less times than a threshold. Modeling different kinds of correlations (e.g., by using ML-kNN, CLR) exhibits moderate to low performance. To investigate the statistical significance of the difference of each method from the baseline we used a two-tailed pair-wise sign test [BdVBF05] and found that only differences between P-LP and the baseline are significant (at 1% significance level).

In relation to the annotation problem (Table 9:(c),(d)) the results show again the effectiveness of the proposed stacking architecture when combined with P-LP, reaching a MAP@3 of 80.88 and improving the baseline results by 4.0%. In this problem also P-MLkNN presents good results, reaching top performance when combined with the detectors of the first layer. Also, for P-LP the relative boost of MXinfAP with respect to the baseline is of the same order of magnitude as the relative boost of MAP@3 (which, as we recall from section 3.3, is not the case when examining independent concept detectors).

To assess the influence of the number of input detectors in the second layer we also performed experiments where the predictions of a reduced set of 60 concept detectors (the 60 concepts that NIST pre-selected for the TRECVID SIN 2013 task [Oea13]) are used for constructing the meta-level dataset (Table: 9:(II)). Results show that usually a larger input space (detectors for 346 concepts instead of 60) is better, increasing both MXinfAP and MAP@3.

To investigate the importance of stacking-based methods separately for each concept, we closely examined the four best-performing methods of column (b) in Table 9:(I). Fig. 10 shows the difference of each method from the baseline. From these results it can be shown that the majority of concepts exhibit improved results when any of the second-layer methods is used. The most concepts benefit from the use of P-LP (29 of the 38 concepts), while the number of concepts that benefit from DMF, BSBRM and CF, compared to the baseline, is 25, 21, and 25 respectively. One concept (5:animal) consistently presents a great improvement when concept correlations are considered, while there are 3 concepts (5:anchorperson, 59:hand and 100:running) that are negatively affected regardless of the employed stacking method.

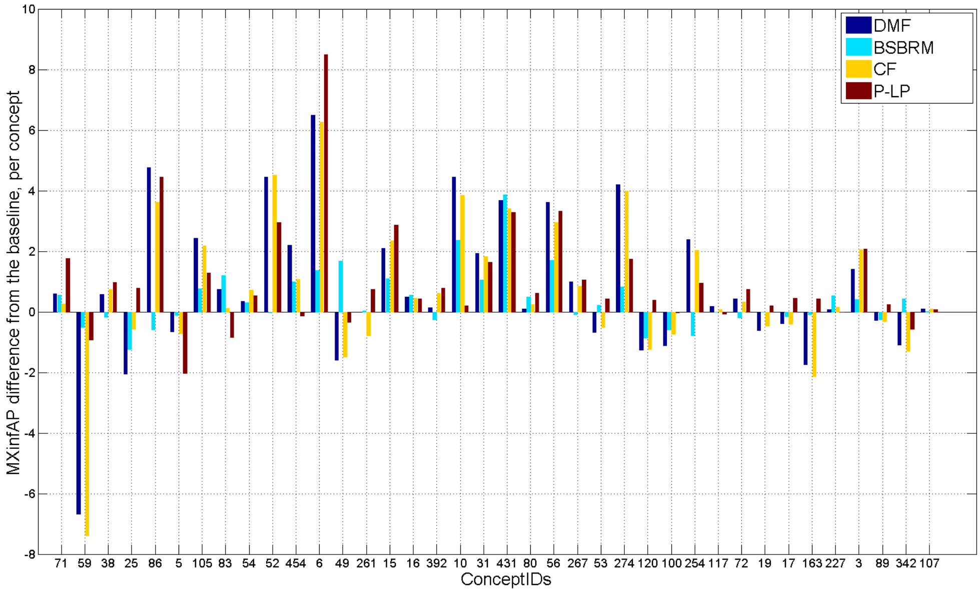


Figure 10 Differences of selected second layer method from the baseline per concept with respect to the indexing problem when a meta-learning set of 346 concepts is used. Concepts ordered according to their frequency in the test set (in descending order). Concepts on the far right side of the chart (most infrequent concepts) seem to be the least affected, either positively or negatively, by the second-layer learning.

Finally, we counted the processing time that each method requires (Table 9:(e)). One could argue that the proposed architecture that uses multi-label learning methods requires considerably more time than the typical BR-stacking one. However, we should note here that the extraction of one model vector from one video shot using the first-layer detectors for 346 concepts, requires approximately 3.2 minutes in our experiments, which is about three orders of magnitude slower than the slowest of the second-layer methods. As a result of the inevitable computational complexity of the first layer of the stack, the differences in processing time between all the second-layer methods that are reported in Table 9 can be considered negligible. This is, in sharp, contrast to building a multi-label classifier directly from the low-level visual features of video shots, where the high requirements for memory space and computation time that the latter methods exhibit make their application to our dataset practically infeasible.

Specifically, the computational complexity of BR, CLR, LP and PPT when used in a single-layer architecture depends on the complexity of the base classifier, in our case the Logistic Regression, and on the parameters of the learning problem. Given that the training dataset used in this work consists of more than 500.000 training examples, and each training example (video shot) is represented by a 4000-element low-level feature vector for each visual descriptor, the BR algorithm, which is the simplest one, would build N models for N concepts; CLR, the next least complex algorithm, would build N BR-models and $N*(N-1)/2$ one-against-one models. LP and PPT, would build a multi-class model, with the number of classes being equal to the number of distinct label sets in the training set (after pruning, in the case of PPT); this is in order of N^2 in our dataset. Finally ML-kNN would compare each training example with all other (500.000) available examples; in all these cases, the 4000-element low-level feature vectors would be employed. Taking into account the dimensionality of these feature vectors, the use of any such multi-label learning method in a single-layer architecture would require several orders of magnitude more computations compared to the BR alternative that we employ as the first layer in our proposed stacking architecture. In addition to this, typically, multi-label learning algorithms require the full training set to be loaded on memory at once (e.g., [TSxVV11]), which would be practically infeasible in a single-layer setting, given the dimensionality of the low-level feature vectors. We conclude that the two major obstacles of using multi-label classification algorithms in a one-layer architecture are the high memory space and computation time requirements, and this finding further stresses the merit of our proposed multi-label stacking architecture.

3.5 Local descriptors vs. DCNNs and their combinations

Features based on Deep Convolutional Networks (DCNNs) is a popular category of visual features that presented SoA results in the SIN task of TRECVID 2014. One or more hidden layers of a DCNN are typically used as a global image representation [SZ14]. DCNN-based descriptors present high discriminative power and generally outperform the local descriptors [SDH$^+$14], [SSF$^+$14]. There is a lot of recent research on the complementarity of different features, focusing mainly on local descriptors [MPP$^+$15]. However, the combination of DCNN-based features with other state-of-the-art local descriptors has not been thoroughly examined [SDH$^+$14]. In this section we report our efforts in building concept detectors from DCNN-based features and we show optimal ways of combining them with other popular local descriptors.

The typical way of combining multiple features for concept detection is to separately train supervised classifiers for the same concept and each different feature. When all the classifiers give their decisions, a fusion step computes the final confidence score (e.g., in terms of averaging); this is known as late fusion. Hierarchical late fusion [SBB$^+$12] is a more elaborate approach; classifiers that have been trained on more similar features (e.g., SIFT and RGB-SIFT) are firstly fused together and then, more dissimilar classifiers (e.g., SURF) are sequentially fused with the previous groups. A problem when training supervised classifiers is the large-scale and imbalanced training sets, where for many concepts negative samples are significantly more than positives. The simplest way to deal with this is to randomly select a subset of the negative examples in order to have a reasonable ratio of positive-negative examples [SQ10]. Random Under-Sampling is a different technique, which can somewhat balance the loss of negative samples of random selection [SQ10] [Mea13]. Different subsets of the negative examples are given as input to train different classifiers that are finally combined by late fusion. To achieve fast detection and reduce the computational requirements of the above process, linear supervised classifiers such as linear SVMs or Logistic Regression (LR) models, are typically preferred.

While late fusion is one reasonable solution, there are other ways that these classifiers can be trained and combined in order to accelerate the learning and detection process [GCB$^+$04], [Bea11]. Cascading is a learning and fusion technique that is useful in large-scale datasets and also accelerates training and classification. We developed a cascade of classifiers where the classifiers are arranged in stages, from the less accurate to the most accurate ones. Each stage is associated with a rejection threshold. A video shot is classified sequentially by each stage and the next stage is triggered only if the previous one returns a prediction score which is higher than the stage threshold (i.e., indicating that the concept appears in the shot). The rationale behind this is to rapidly reject shots (i.e., keyframes) that clearly do not present a specific concept and focus on those shots that are more difficult and more likely to depict the concept. The proposed architecture is computationally more efficient than typical state-of-the-art video concept detection systems, without affecting the detection accuracy. Furthermore, we modified the off-line method proposed by [CSS06] for optical character recognition, in order to refine the rejection thresholds of each stage, aiming to improve the overall performance of the cascade. The rest of this section compares the proposed cascade with other late fusion schemes and presents a detailed study on combining descriptors based on DCNNs with other popular local descriptors, both within the proposed cascade and when using different late-fusion schemes.

Once again, our experiments were performed on the TRECVID 2013 Semantic Indexing (SIN) dataset [Oea13]. Regarding feature extraction, we followed the experimental setup of 3.3. More specifically, we extracted three binary descriptors (ORB, RGB-ORB and OpponentORB) and six non-binary descriptors (SIFT, RGB-SIFT and OpponentSIFT; SURF, RGB SURF and OpponentSURF). All the local descriptors, were compacted using PCA and were subsequently aggregated using the VLAD encoding. The VLAD vectors were reduced to 4000 dimensions and served as input to LR classifiers, used as base classifiers on the cascade or trained independently as described in the next paragraph. In all cases, the final step of concept detection was to refine the calculated detection scores by employing the re-ranking method proposed in [SQ11]. In addition, we used features based on the last hidden layer of a pre-trained DCNN. Specifically, to extract these features we used the 16-layer pre-trained deep ConvNet network provided by [SZ14]. The network has been trained on the ImageNet data only [DDS$^+$09], and provides scores for 1000 concepts. We applied the network on the TRECVID keyframes and similar to other studies [SDH$^+$14] we used as a feature the output of the last hidden layer (fc7), which resulted to a 4096-element vector. We refer to these features as DCNN in the sequel.

To train our algorithms, for each concept, 70% of the negative annotated training examples was randomly chosen for training and the rest 30% was chosen as a validation set for the offline cascade optimization method. Each of the two negative sets was merged with all positive annotated samples, by adding in every case three copies of each such positive sample (in order to account for their, most often, limited number). Then the positive and negative ratio of training examples was fixed on each of these sets by randomly rejecting any excess negative samples, to achieve a 1:6 ratio (which is important for building a balanced classifier). To train the proposed cascade architecture, the full training set was given as input to the first stage, while each later stage was trained with the subset of it that passed from its previous stages. We compared the proposed cascade with a typical late-fusion scheme, where one LR classifier was trained for each type of features on the same full training set (that included three copies of each positive sample), denoted in the sequel as overtraining scheme. We also compared with another late-fusion scheme, where one LR classifier was trained on different subsets of the training set for each type of features. To construct different subsets of training sets we followed the Random Under-Sampling technique described earlier; for each classifier, trained on a different type of features, a different subset of negatives was merged with all the positives (just one copy of each positive sample, in this case) and was given as input. The ratio of positive/negative samples was also fixed to 1:6. This scheme is denoted in the sequel as undersampling. For the off-line cascade optimization we used quantization to ensure that the optimized cascade generalizes well to unseen samples. In these lines, instead of searching for candidate thresholds on all the M examples of a validation set, we sorted the values by confidence and split at every M/Q example (Q was set to 200).

Tables 10 and 11 present the results of our experiments in terms of Mean Extended Inferred Average Precision (MXinfAP) [YKA08]. Starting from Table 10 we present the ten types of features that have been extracted and used by the algorithms of this study. For brevity, for SIFT, ORB and SURF we only show the MXinfAP when the original grayscale descriptor is combined with other two corresponding color variants by means of late fusion (averaging) (see [MPP$^+$15] for indicative fine-grained results).

Table 10 Performance (MXinfAP. %) for base classifiers or combinations of them trained on different features.

Descriptor	MXinfAP	Base classifiers (ordered in terms of accuracy)
ORBx3	18.31	ORB, OpponentORB, RGB-ORB
SIFTx3	18.98	SIFT, OpponentSIFT, RGB-SIFT
SURFx3	19.34	SURF, OpponentSURF, RGB-SURF
DCNN	23.84	Last hidden layer of DCNN

Table 11 Performance (MXinfAP. %) and relative computational complexity for different architectures/schemes: (a) cascade architecture (in parenthesis we show results for the offline cascade optimization [CSS06]), (c) overtraining, (d) undersampling.

			Cascade (Offline cascade optimization)			Late fusion-overtraining		Late fusion-undersampling		
Run ID	Features	# of Base detectors/ Stages	MXinfAP (%)	amount of training data (%)	amount of classifier evaluations (%)	MXinfAP (%)	amount of training data (%)	MXinfAP (%)	amount of training data (%)	amount of classifier evaluations (%); same for both late fusion schemes
R1	ORBx3;DCNN	4 / 2	27.25 (27.31)	39.3	37.3 (38.4)	27.28	40.0	25.31	15.6	40.0
R2	SIFTx3;DCNN	4 / 2	27.42 (27.47)	38.8	36.8 (38.1)	27.41	40.0	25.89	15.6	40.0
R3	SURFx3;DCNN	4 / 2	26.9 (27.3)	39.3	37.6 (38.3)	27.01	40.0	25.98	15.6	40.0
R4	ORBx3;SIFTx3;DCNN	7 / 3	**27.82 (27.88)**	65.3	57.9 (61.3)	27.76	70.0	26.42	27.3	70.0
R5	ORBx3;SURFx3;DCNN	7 / 3	27.16 (27.66)	65.8	58.4 (61.3)	27.63	70.0	26.5	27.3	70.0
R6	SIFTx3;SURFx3;DCNN	7 / 3	27.69 (27.71)	64.5	56.7 (61.4)	27.7	70.0	26.8	27.3	70.0
R7	ORBx3;SIFTx3; SURFx3;DCNN	10 / 4	27.52 (27.52)	90.8	75.4 (83.0)	27.61	100.0	26.62	39.0	100.0
R8	ORBx3;DCNN	4 / 4	24.43 (24.53)	36.6	30.4 (33.7)	24.55	40.0	22.46	15.6	40.0
R9	SIFTx3;DCNN	4 / 4	24.42 (24.43)	35.6	29.1 (32.8)	24.5	40.0	23.05	15.6	40.0
R10	SURFx3;DCNN	4 / 4	24.49 (24.49)	36.9	31.7 (34.0)	24.46	40.0	23.66	15.6	40.0
R11	ORBx3;SIFTx3;DCNN	7 / 7	24.66 (24.79)	60.5	44.7 (51.5)	24.82	70.0	23.47	27.3	70.0
R12	ORBx3;SURFx3;DCNN	7 / 7	23.02 (24.77)	61.8	46.9 (54.0)	24.72	70.0	23.96	27.3	70.0
R13	SIFTx3;SURFx3;DCNN	7 / 7	23.53 (25.24)	60.0	44.1 (53.1)	25.16	70.0	24.32	27.3	70.0
R14	ORBx3;SIFTx3; SURFx3;DCNN	10 / 10	23.55 (25.06)	82.5	57.0 (67.3)	25.09	100.0	24.28	39.0	100.0

Table 11 presents the performance and computational complexity of the proposed cascade architecture and the overtraining and undersampling schemes. The second column shows the features on which base classifiers have been trained for each run, and the number of stages (column three) indicates how the features have been grouped in stages. Runs $R1$ to$R7$ use stages that combine many base classifiers, in terms of late fusion (averaging). Specifically, stages that correspond to SIFT, SURF and ORB consist of three base classifiers (e.g., for the grayscale descriptor and the two color variants), while the last stage of DCNN features contains only one base classifier. Runs $R8$ to $R14$ use stages made of a single base classifier (trained on a single type of features). We sort the stages on each cascade according to the accuracy of the employed individual base classifiers or combinations of them (according to Table 10) from the less accurate to the most accurate. Stages do not refer only to cascade but also affect the way that late fusion has been performed by the overtraining and undersampling schemes. For example, for stages that consist of many features, the corresponding base classifiers per stage were firstly combined by averaging the classifier output scores and then the combined outputs of all stages were further fused together.

Table 11 shows that both cascade and late fusion-overtraining outperform the most commonly used approach of late fusion-undersampling, which uses less negative training examples to train each base classifier. The best results for cascade and overtaining are achieved by $R4$, reaching a MXinfAP of 27.82 and 27.76 respectively. The cascade reaches this good accuracy while at the same time is less computationally expensive than overtraining, both during training and during classification. Specifically, the cascade employed for $R4$ achieves 17.3% relative decrease in the number of classifier evaluations. Considering that training is performed off-line only once, but classification will be repeated many times for any new input video, the latter is more important and this makes the observed reduction in the amount of classifier evaluations significant. Table 11 also presents the results of the cascade when the offline cascade optimization technique of [CSS06] for threshold refinement is employed. We observe that in many cases MXinfAP increases. The amount of necessary classifier evaluations also increases in this case, but even so the cascade is more computationally efficient than the two late fusion schemes.

The second fold of this work is to examine how we can effectively combine DCNN-based features with handcrafted local descriptors. According to Table 10, the DCNN performs better than the combinations of SIFT, SURF and ORB with their color variants. It should be noted that each of the base classifiers of these groups (e.g., RGB-SIFT) is rather weak, achieving MXinfAP that ranges from 11.68 to 15.04 (depending on which descriptor is used). Table 11 shows that the way that the stages of a cascade are constructed but also the way that late fusion is performed on the overtraining and undersampling schemes affects the combination of DCNN with the other descriptors. Generally, it is better to merge weaker base classifiers to a more robust one (e.g., grouping grayscale SIFT with its color variants) in order to either use them as a cascade stage or to combine their decisions in terms of late fusion ($R1$-$R7$: MXinfAP ranges from 22.6 to 25.09), than treating each of them independently from the others (i.e., using one weak classifier per stage or fusing all of them with equal weight; $R8$-$R14$: MXinfAP ranges from 25.31 to 27.82). The best way to combine DCNN with other local descriptors is $R4$, where ORBx3, SIFTx3 and DCNN are arranged in a cascade, increasing the MXinfAP from 23.84 (for DCNN alone) to 27.82.

Finally, we compared this optimized video concept detection system that combines DCNN with other binary and non-binary descriptors with the system developed for the SIN task of TRECVID 2014 (section 3.2). A relative boost of 34.7% was observed (MXinfAP increasing from 20.7 to 27.88). Our TRECVID 2014 system was evaluated on the TRECVID 2014 SIN dataset and 30 semantic concepts, while the optimized system that uses DCNN features was evaluated on the TRECVID 2013 SIN task and 38 concepts, where some of the concepts are common. The two systems were evaluated on different datasets although, these results are comparable as the two datasets are similar to size and constructed by similar videos. As a result this increase of concept detection accuracy can be considered as significant.

3.6 Conclusion

The detection of visual concepts that describe the high level semantics of the video content was used in the context of LinkedTV for guiding the semantic annotation and enrichment of media fragments. During the project, and as reported in D1.2, D1.4 and here, there was a continuous effort for improving the accuracy of our method, where several different algorithms have been developed and extensively tested via in-house experiments and by participating at international benchmarking activities. The last part of these evaluation activities was described in this section. Specifically we showed that two binary local descriptors (ORB, BRISK) can perform reasonably well compared to their state-of-the-art non-binary counterparts in the video semantic concept detection task and furthermore their combination can improve video concept detection. We subsequently showed that a methodology previously used for defining two color variants of SIFT is a generic one that is also applicable to other binary and non-binary local descriptors. In addition, we presented a useful method that takes advantage of concept correlation information for building better detectors. For this we proposed a stacking architecture, that uses multi-label learning algorithms in the last level of the stack. Finally, we presented effective ways to fuse DCNN-based features with the other local descriptors and we showed significant increase on video concept detection accuracy.

4 Evaluation of extended video annotation

4.1 Overview

The concept-based video annotation can be extended to event-based annotation; the event labels can then be used similarly to the concept labels. In this section we provide an overview of our evaluations for extended video annotation. Firstly, we present the evaluation results regarding the performance of our new machine learning method, after participating in the Multimedia Event Detection (MED) task of TRECVID benchmarking activity. This method utilizes a discriminant analysis (DA) step, called spectral regression kernel subclass DA (SRKSDA), to project the data in a lower-dimensional subspace, and a detection step using the LSVM classifier. More specifically, SRKSDA-LSVM extends the GSDA-LSVM technique presented in section 8.2 of D1.4 offering an improved training time.

By observing that intensive parts of SRKSDA-LSVM are fully parallelizable (e.g., large-scale matrix operations, Cholesky factorization), SRKSDA-LSVM is implemented in C++ taking in advantage the computing capabilities of the modern multi-core graphics cards (GPU). The GPU accelerated SRKSDA-LSVM is further evaluated using annotated MED and SIN datasets for event and concept detection.

The techniques discussed above are known as supervised machine learning approaches. That is, it is assumed that an annotated set of training observations is provided. However, in many real world retrieval applications only a textual target class description is provided. To this end, in our initial attempts of learning high level semantic interpretations of videos using textual descriptions that were presented in section 7.2.1 of D1.4, ASR transcripts were generated from spoken video content and used for training SVM-based concept detectors. As reported in D1.4 the combination of text- and visual-based trained classifiers resulted in slightly improved performance. Here, we propose and evaluate a novel “zero-example” technique for event detection in video. This method combines event/concept language models for textual event analysis, techniques generating pseudo-labeled examples, and SVMs in order to provide an event learning and detection framework. The proposed approach is evaluated in a subset of the recent MED 2014 benchmarking activity.

The remaining of this section is organized as follows. In subsection 4.2 we present the experimental results of SRKSDA+LSVM in the TRECVID MED benchmarking activity, while in subsection 4.3 the C++ GPU implementation of SRKSDA+LSVM is evaluated using recent TRECVID MED and SIN datasets for the tasks of event and concept detection, respectively. Then subsection 4.4 describes our novel zero-sample event detection method and presents experimental results using a subset of the MED 2014 dataset. Finally, conclusions are drawn in the last subsection 4.5.

4.2 Results of the TRECVID MED benchmarking activity

The video collections provided by the TRECVID MED benchmarking activities are among the most challenging in the field of large-scale event detection. In the TRECVID MED 2014 we participated in the pre-specified (PS) and ad-hoc (AH) event detection tasks. We submitted runs for the 010Ex condition, which is the main evaluation condition, as well as for the optional 100Ex condition. The 010Ex and 100Ex evaluation conditions require that only 10 or 100 positive exemplars, respectively, are used for learning the specified event detector. In the following, we briefly describe the MED 2014 dataset, the preprocessing of videos for extracting feature representations of them, the event detection system and the evaluation results obtained from our participation in MED 2014.

4.2.1 Datasets and features

For training our PS and AH event detectors we used the PS-Training, AH-Training and Event-BG sets, while for evaluation the EvalSub set was employed. The amount of videos contained in each dataset are shown in Table 12. The PS and AH events were 20 and 10 respectively, and are listed below for the shake of completeness:

– PS events: “E021: Bike trick”, “E022: Cleaning an appliance”, “E023: Dog show”, “E024: Giving directions”, “E025: Marriage proposal”, “E026: Renovating a home”, “E027: Rock climbing”, “E028: Town hall meeting”, “E029: Winning race without a vehicle”, “E030: Working on a metal crafts project”, “E031: Beekeeping”, “E032: Wedding shower”, “E033: Non–motorized vehicle repair”, “E034: Fixing a musical instrument", "E035: Horse riding competition", "E036: Felling a tree", "E037: Parking a vehicle", "E038: Playing fetch", "E039: Tailgating", "E040: Tuning a musical instrument".

- AH events: "E041: Baby Shower", "E042: Building a Fire", "E043: Busking", "E044: Decorating for a Celebration", "E045: Extinguishing a Fire", "E046: Making a Purchase", "E047: Modeling", "E048: Doing a Magic Trick", "E049: Putting on Additional Apparel", "E050: Teaching Dance Choreography".

Table 12 MED 2014 dataset.

	# videos	hours
PS-Training	2000	80
AH-Training	1000	40
Event-BG	5000	200
EvalSub	32000	960

Our method exploits three types of visual information, i.e., static, motion, and model vectors. For the extraction of static visual features and model vectors the procedure described in section 3 is applied. We briefly describe the different visual modalities in the following:

- Static visual information: Each video is decoded into a set of keyframes at fixed temporal intervals (one keyframe every six seconds). Low-level feature extraction and encoding is performed as described in section 3. Specifically, local visual information is extracted using four different descriptors (SIFT, OpponentSIFT, RGB-SIFT, RGB-SURF) at every keyframe. The extracted features are encoded using VLAD, projected in a 4000-dimensional subspace using a modification of the random projection technique, and averaged over all keyframes of the video. A single 16000-dimensional feature vector is then constructed by concatenating the four individual feature vectors described above.
- Visual model vectors: Model vectors of videos are created following a three-steps procedure [MHX+12, GMK11]: a) extraction of low-level visual features at keyframe level, b) application of external concept detectors at keyframe level, and, c) application of a pooling strategy to retrieve a single model vector at video level. In more detail, the four low-level feature representations at keyframe level described above are directly exploited. A pool of 1384 external concept detectors (346 concepts × 4 local descriptors), the ones derived in section 3 concept labeling), is then used to represent every keyframe with a set of model vectors (one model vector for each feature extraction procedure). The model vectors referring to the same keyframe are aggregated using the arithmetic mean operator, and subsequently, the model vectors of a video are averaged to represent the video in $\mathbb{R}^{346}$.
- Motion visual information: Motion information is extracted using improved dense trajectories (DT) [WS13]. The following four low-level feature descriptors are employed: Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histograms in both x (MBHx) and y (MBHy) directions. The resulting feature vectors are first normalized using Hellinger kernel normalization, and then encoded using the Fisher Vector (FV) technique with 256 GMM codewords. Subsequently, the individual feature vectors are concatenated to yield a single motion feature vector for each video in $\mathbb{R}^{101376}$.

The final representation of a video is a 117722-dimensional feature vector, formed by concatenating the individual feature vectors (static, motion, model vectors).

4.2.2 Event detection

A two stage framework is used to build an event detector. In the first stage a nonlinear discriminant analysis (NLDA) method is used to learn a discriminant subspace $\mathbb{R}^D$ of the input space $\mathbb{R}^{101376}$. We utilized a novel NLDA method recently developed in our lab, based on our previous methods KMSDA and GSDA [GMKS13b, GMKS13a, GM14] called spectral regression kernel subclass discriminant analysis (SRKSDA). Given the training set in $\mathbb{R}^{101376}$ for a specific event, SRKSDA learns a D-dimensional subspace, $D \in [2,3]$, which is discriminant for the specific event in question. Experimental evaluation has shown that SRKSDA outperform other DA approaches in both accuracy and computational efficiency (particularly at the learning stage). In the second stage, a linear support vector machine (LSVM) is applied in $\mathbb{R}^D$ to learn a maximum margin hyperplane separating the target from the rest-of-the-world events. During training, a 3 cycle cross-validation (CV) procedure was employed, where at each CV cycle the training set was divided to 70% learning and 30% validation set for learning the SRKSDA-LSVM parameters.

During detection, an unlabelled set of observations in the input space is initially projected in the discriminant subspace $\mathbb{R}^D$ using the SRKSDA transformation matrix. Subsequently, the trained LSVM is applied to provide a degree of confidence (DoC) score for each unlabelled observation.

4.2.3 Results

The MED 2014 evaluation results of our four runs are shown in Table 13 in terms of Mean Average Precision (MAP). The different cells of this table depict our MAP performance in our four different runs: PS task, AH task along 010Ex and 100Ex conditions. A bar graph depicting the overall performance of the of the different submissions along all participants in the AH 010Ex task is shown in Fig. 11. From the analysis of the evaluation results we can conclude the following:

- In comparison to most of the other submissions we still employ only a small number of visual features. Nevertheless, among the submissions that processed only the EvalSub set our system provides the best performance. Moreover, considering all submissions, our runs were ranked above the median of the submissions.
- In comparison to our previous year submission we observed a large performance gain of more than 12% and 20% in the AH and PS tasks respectively. This is due to improvements at all levels of our system, such as, the use of improved visual static feature descriptors and concept detectors, exploitation of motion information, application of the new DA preprocessing step to extract the most discriminant features, and an overall application of a faster detection method that allows for a more effective optimization.
- A major advantage of SRKSDA+LSVM is that it can learn automatically useful dimensions of the feature space without the need to e.g., manually select the concepts that are most relevant with the target event. This can be seen from the good results in the AH subtask (Fig. 11), where we did not use any knowledge about the PS events when building our system.

Table 13 Evaluation results (% MAP) for MED 2014.

	010Ex	100Ex
PS	15.1	30.3
AH	18.3	33.1

4.3 Accelerated Discriminant Analysis for concept and event labeling

The main computational effort of the detection algorithm described in the previous section has been “moved” to the proposed DA method. The most computational intensive parts of SRKSDA is the Cholesky factorization algorithm and the computation of several large-scale matrix operations (multiplications, additions, etc.). However, the above parts are fully parallelizable and, thus, the overall algorithm can be considerably accelerated by exploiting appropriate machine architectures and computing frameworks, such as multicore CPUs and GPUs. To this end, we have evaluated different SRKSDA implementations in C++ based on OpenCV, eigen, OpenMP, Intel MKL and Nvidia CUDA GPU libraries (e.g., CULA, Thrust, CUBLAS). The evaluation was based on time and accuracy performance in different machine learning problems. From these experiments, we have selected to base the implementation of the SRKSDA algorithm in C++ CUDA GPU libraries such as CUBLAS.

4.3.1 Datasets and features

Two datasets were used for the evaluation of the different implementations of SRKSDA+LSVM and its comparison with LSVM. Particularly, we utilized subsets of the MED 2012 and SIN 2013 video corpora for event and concept detection, respectively.

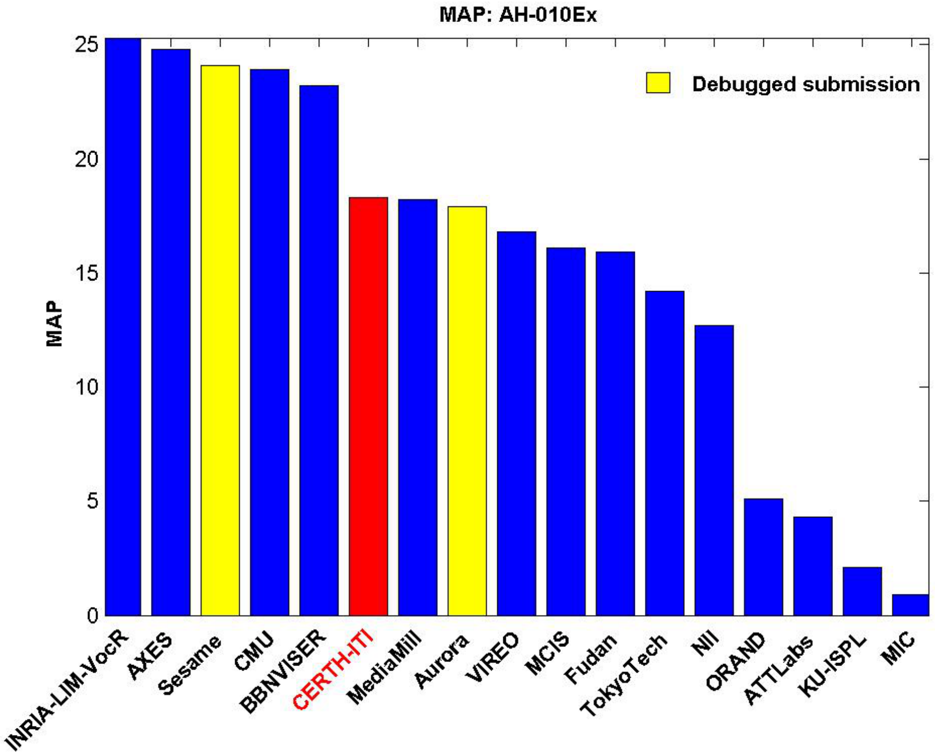


Figure 11 Evaluation results along all MED14 submissions for the ad-hoc 010Ex task.

MED-HBB: For the evaluation of SRKSDA+LSVM for event detection we used the publicly available partitioning of MED 2012 provided by the authors of [HvdSS13]. It utilizes a subset of the MED 2012 video corpus [P. 13], comprising of 13274 videos, and is divided to a training and evaluation partition of 8840 and 4434 videos, respectively. It contains 25 target events, specifically the events E21-E30 described in the previous section, and the events E01-E15 listed below:

– "E01: Attempting a board trick", "E02: Feeding an animal", "E03: Landing a fish", "E04: Wedding ceremony", "E05: Working on a woodworking project", "E06: Birthday party", "E07: Changing a vehicle tire", "E08: Flash mob gathering", "E09: Getting a vehicle unstuck", "E10: Grooming an animal", "E11: Making a sandwich", "E12: Parade", "E13: Parkour", "E14: Repairing an appliance", "E15: Working on a sewing project".

A feature extraction procedure is applied in the above videos in order to represent them using motion visual information. Particularly, DT features are used as described in the previous section yielding a single motion feature vector for each video in $\mathbb{R}^{101376}$.

SIN: For concept detection evaluation we used the SIN 2013 dataset, consisting of 800 and 200 hours of development and test set, respectively. Specifically, for training we used the training observations concerning a subset of 38 concepts of SIN 2013, while for evaluation 112677 videos were utilized. The IDs along with the number of positive and negative samples used for training are depicted in the first three columns of Table 14. Two feature extraction procedures were used for representing the videos:

a) *SIN-LOCAL*: In a first approach, local visual information is exploited as explained in the following. A video is represented with a sequence of keyframes extracted at uniform time intervals. Subsequently, SIFT, SURF and ORB descriptors are applied to extract local features at every keyframe. The extracted features are encoded using VLAD, and compressed by utilizing a modification of the random projection matrix technique to provide a single feature vector in $\mathbb{R}^{4000}$ for each video.

b) *SIN-CNN*: In a second approach, ConvNet network (CNN) features are employed for video representation. Particularly, the pre-trained network provided in [SZ14] is employed, created using the 2009 ImageNet dataset [DDS^{+}09]. In our case, the 16-layer output is utilized providing a 1000-dimensional model vector representation for each keyframe. Average pooling along the keyframes is performed to retrieve a CNN feature vector for each SIN video.

Table 14 SIN 2013 concept IDs along with number of positive and negative training observations (First to third column). Evaluation results (MXinfAP) of LSVM and SRKSDA-1 for each concept using the 1000-dimensional CNN features.

ID	# Positive	# Negative	LSVM	SRKSDA+LSVM	Improvement
26	9962	14236	13.67	**21.79**	8.12
170	3304	17451	34	**35.49**	1.49
358	2778	16909	27.9	**29.56**	1.66
401	5186	13988	**13.1**	12.23	-0.87
248	4355	14425	**59.52**	58.32	-1.2
407	3445	14217	0.1	**0.14**	0.04
319	2818	14619	34.33	**35.9**	1.57
24	2868	14388	48.14	**56.75**	8.61
307	2482	14896	70.35	**72.93**	2.58
52	2171	13182	**53.12**	51.31	-1.81
125	2156	12713	**19.44**	16.42	-3.02
63	1961	11961	4.63	**5.34**	0.71
190	1940	11760	7.93	**12.97**	5.04
204	1928	11016	25.73	**28.15**	2.42
176	2035	10985	10.2	**13.06**	2.86
198	1479	9870	3.51	**3.9**	0.39
437	1608	9426	54.79	**62.77**	7.98
408	1530	9137	48.66	**54.79**	6.13
88	2117	8065	**23.79**	22.64	-1.15
163	2173	7923	14.38	**15.65**	1.27
220	3283	6826	37.36	**37.85**	0.49
108	1761	7935	29.9	**34.88**	4.98
330	1362	8078	2.6	**3.16**	0.56
186	1300	7804	**30.1**	24.7	-5.4
59	1193	7099	18.52	**29.18**	10.66
15	1009	6020	14.45	**17.44**	2.99
261	1179	5840	**22.2**	21.38	-0.82
386	1018	5976	**13.18**	10.68	-2.5
197	1200	5922	32.54	**38.79**	6.25
183	1573	4559	21.16	**22.77**	1.61
65	903	4482	5.26	**6.79**	1.53
41	942	4641	33.04	**34.75**	1.71
140	671	3340	3.5	**4.57**	1.07
299	600	2961	15.26	**18.3**	3.04
463	599	2947	17.81	**21.41**	0.36
455	531	2625	2.35	**5.09**	2.74
290	513	2550	7.23	**8.18**	0.95
68	298	1469	2.72	**7.5**	4.78
AVG	2059	9006	23.07	**25.2**	2.13

4.3.2 Event and concept detection

For event and concept detection an accelerated version of the SRKSDA+LSVM algorithm (section 4.2.2) using Nvidia GPU C++ libraries is used. Specifically, Thurst, CULA and other relevant GPU libraries are used to speed-up the computation of intensive parts of SRKSDA such large-scale matrix operations, Cholesky factorization, and so on. The acceleration offered by this implementation is quantified using two different Nvidia graphics cards, namely the GeForce GT640 and Tesla K40. The accelerated version of SRKSDA+LSVM was compared against the liblinear implementation of LSVM. For the evaluation in terms of training time the C++ implementation of SRKSDA+LSVM was compared with our initial Matlab implementation and with LSVM.

The training of SRKSDA+LSVM was performed using the overall training set concerning the target concept/event, and a common Gaussian kernel parameter was utilized for all concepts/events. During detection, an unlabelled set of observations in the input space is initially projected in the discriminant subspace $\mathbb{R}^D$ ($D \in [2,3]$) using the SRKSDA transformation matrix. Subsequently, the trained LSVM is applied to provide a degree of confidence (DoC) score for each unlabelled observation. MXinfAP and MAP were used as the evaluation metrics for concept and event detection respectively. For convenience, we provide the following naming convention concerning the different algorithms, implementations and graphic cards used in the experiments:

a) SRKSDA-1: the accelerated C++ implementation of SRKSDA+LSVM is used, running in a machine equipped with the Nvidia GeForce GT 640 graphics card,

b) SRKSDA-2: the accelerated C++ implementation of SRKSDA+LSVM is employed, running in a machine utilizing the Nvidia Tesla K40 card,

c) SRKSDA-3: the Matlab implementation of SRKSDA+LSVM is used,

d) LSVM: LSVM is used (a preprocessing step with SRKSDA is not applied), implemented in C++ using the liblinear library.

4.3.3 Results

The evaluation results in terms of MXinfAP for concept detection and MAP for event detection concerning SRKSDA+LSVM (specifically the C++ GPU implementation) and LSVM for the task of event and concept detection using the MED-HBB, SIN-CNN and SIN-LOCAL datasets are shown in Table 15. Moreover, the performance of SRKSDA+LSVM (SRKSDA-1) and LSVM for each individual concept evaluated using the CNN features (SIN-CNN) is shown in Table 14. In this table, the concept IDs are sorted in descending order based on the total number of training observation (positive plus negative number of observations). The corresponding time complexities of the evaluation in the above datasets (MED-HBB, SIN-CNN and SIN-LOCAL) for different implementations of LSVM and SRKSDA+LSVM (Matlab, C++ GPU) and along different graphic cards (GeForce GT640 and Tesla K40) are shown in Table 16, whereas the specifications of the different machines used in the evaluation are shown in Table 17. Finally, the minimum memory requirements for each method along the different datasets are shown in Table 18.

Table 15 Evaluation results (MXinfAP) of LSVM and SRKSDA+LSVM (SRKSDA-1: C++ GPU implementation).

	LSVM	SRKSDA+LSVM
SIN-CNN	23.07	**25.20**
SIN-LOCAL	12.60	**14.60**
MED-HBB	38.73	**41.24**

Table 16 Training times (in minutes) of LSVM and different implementations of SRKSDA+LSVM.

	LSVM	SRKSDA-1	SRKSDA-2	SRKSDA-3
MED-HBB	49.97	29,7	**19.8**	565.4
SIN-LOCAL	9	11.8	**4.5**	14.8
SIN-CNN	9.5	30.9	**5.3**	24.2

Table 17 Specifications of workstations used for the evaluation of LSVM and the different implementations of SRKSDA+LSVM.

	LSVM	SRKSDA-1	SRKSDA-2	SRKSDA-3
RAM	32 GB	16 GB	16 GB	32 GB
CPU	Intel I7-3.5GHz	Intel I5-3.4GHz	Intel I5-3.4GHz	Intel I7-3.5GHz
OS	Win 7 64bit	Win 7 64bit	Win 7 64bit	Win 7 64bit
Graphics card	GeForce GT640	GeForce GT640	Tesla K40	GeForce GT640

From these results we note the following:

Table 18 Minimum memory requirements (GB) for LSVM and different implementations of SRKSDA+LSVM.

	LSVM	SRKSDA-1 & SRKSDA-2	SRKSDA-3
MED-HBB	30	**15**	43.5
SIN-LOCAL	15.1	**10.1**	30.5
SIN-CNN	**3.7**	9	27.2

- As shown in Table 15, SRKSDA+LSVM outperforms LSVM in terms of retrieval performance for both event and concept detection. Particularly, an absolute MAP improvement of more than 2% is observed in all datasets when SRKSDA is used prior to the application of LSVM.
- In Table 14 we observe that the proposed method provides a better retrieval performance from LSVM in 30 out of 38 concepts. Specifically, for certain concepts a very large improvement is observed. For instance, an absolute MXinfAP improvement of 8.12%, 8.61%, 5.04%, 7.98%, 6.13%, 4.98%, 10.66%, 6.25%, 4.78% is observed for concept IDs 26, 24, 190, 437, 408, 108, 59, 197 and 68, respectively. In contrary, for the concepts that LSVM outperforms SRKSDA+LSVM a relatively small performance deference is observed.
- Concerning time complexity, the C++ GPU implementation of SRKSDA+LSVM running in conventional GeForce GT640 graphics card (SRKSDA-1) offers a significant speed-up over the equivalent implementation in Matlab (SRKSDA-3), and an equivalent detection response in comparison to LSVM. Furthermore, when the Tesla K40 card is employed (SRKSDA-2), SRKSDA+LSVM offers a speed-up over LSVM of $1.5\times$ to $2\times$.
- The C++ GPU implementation of SRKSDA+LSVM provides an improved performance in memory complexity as well. Specifically, we observe that the C++ GPU implementation requires 1.5 to 3 times fewer memory than the corresponding Matlab implementation of SRKSDA+LSVM (depending on the dataset). Additionally, for the largest datasets (MED-HBB, SIN-LOCAL) it also outperformes LSVM in terms of memory requirements.

4.4 Video annotation with zero positive samples

Video retrieval using only textual information is a very challenging and relatively unexplored research field. Taking into account that in many real-world applications the only available information concerning the target event is the textual event definition alone or a more elaborate event description, a growing interest on "zero-sample" learning techniques is recently observed. Motivated by the above discussion, in this section we present and evaluate a novel "zero-example" event detection method that combines event/concept textual modelling methods, pseudo-labeled example generation techniques and SVMs.

4.4.1 Dataset and features

For evaluating our system, the PS-Training, AH-Training and Event-BG video datasets of TRECVID MED 2014 dataset (Table 12) were used. In overall a dataset consisting of 8000 videos, 30 target events was created. The above sets were divided to a training and evaluation set as shown in Table 19. The distribution of target event and background videos in each partition are shown in the following:

- Training Set: 50 positive and 25 near-miss (i.e., related but clearly positive) observations per target event, 2496 background observations (i.e., negative for all event classes),
- Evaluation Set: ~ 50 positive and ~ 25 near-miss observations per target event, 2496 background observations.

The developed framework is illustrated in Fig. 12. Its input consists of a textual description of the target event along with a set of a number of concepts. The titles of the provided set of concepts are utilized to generate queries for Google search engine and Wikipedia. A so-called Concept Language Model (CLM) is then constructed using the top-M list of words and phrases for each concept and query type. That is, three different types of CLMs are constructed for each concept, namely, the "Google" "Wikipedia" and "Title" type CLMs, corresponding to using Google, Wikipedia or the title of the concept alone. A BoW technique and BoW

Table 19 Subset of MED 2014 dataset used for zero-example event detection.

	Training	Evaluation
# positive videos	1500	~ 1500
# near-miss videos	750	~ 750
# negative videos	750	~ 750
# target-events	30	30

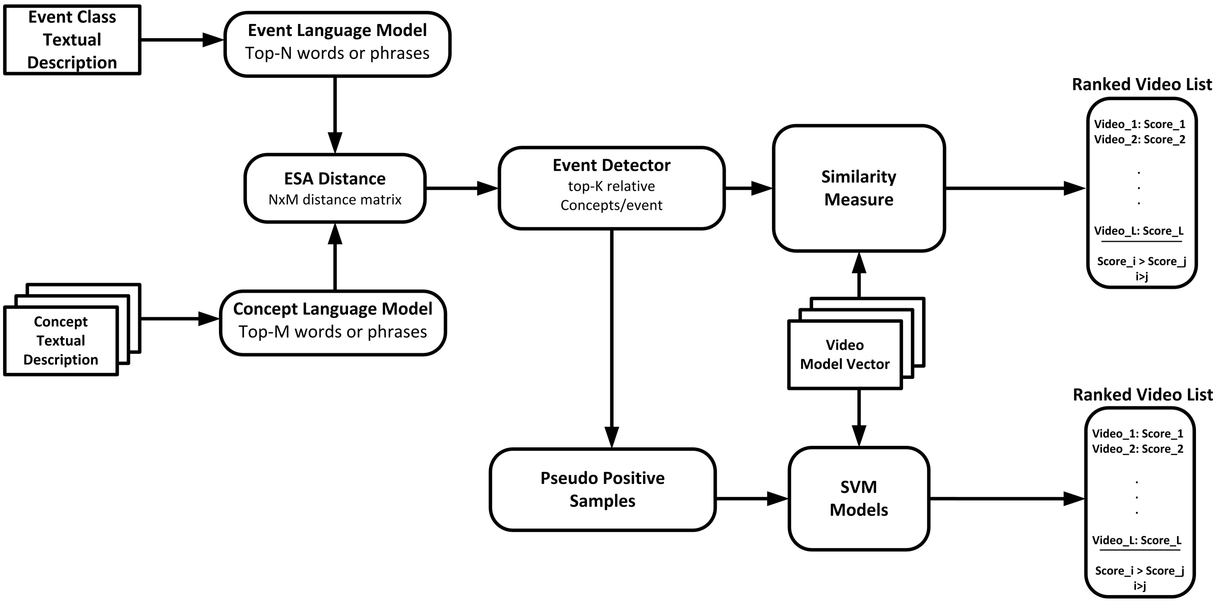


Figure 12 Zero example pipeline.

combined with term frequency-inverse document frequency (Tf-Idf) weighting is then utilized to quantize the CLM histograms, yielding 6 CLMs per concept.

Given the textual description of the event class, our framework first identifies N words or phrases that most closely related to the event class; this word-set we call Event Language Model (ELM). To this end we consider 3 different choices for creating ELM:

- Title: the title of an event is used
- Visual: the title with a set of visual cues are used
- Minimum: as Visual among with audio cues

Subsequently, for a given ELM and a CLM, an Explicit Semantic Analysis (ESA) distance is computed [GM07], for each word contained in the ELM and CLM, yielding an $N \times M$ distance matrix. Note that each matrix denotes the relation between each pair of event class-concept. A single score expressing the relation between each concept and the underlying event is then computed using different matrix operations (ℓ_2 Norm, Hausdorff Distance, Maximum entry etc.). For each event class, we thus end up with a list of scores to the corresponding concept in descending order and we choose the top-K scores, which are considered to be the event detectors of the proposed framework

In order to retrieve a consistent representation of test videos, a pool of pre-trained concept detectors is utilized (e.g., pre-trained CNN networks), and scores related to the top-K concepts of the target event prototype model vector (PMV) are selected to represent a test video with a corresponding model vector (MV).

4.4.2 Event detection

As shown in Fig. 12 the output of the implemented method is a ranked list of videos relevant to the target event. Two methods were evaluated for event learning and detection using the model vector representation described in section 4.4.1:

- Test video MVs are directly compared with target event PMV using an appropriate measure, such as, cosine similarity, histogram intersection kernel, kullback leibler divergence. The derived similarity or distance scores are then exploited to create a rank list of the test videos.
- A pseudo-labeling technique is initially applied to retrieve a labelled learning set. Specifically, pseudo-positive observations are generated based on the event PMV, while pseudo-negative observations are created based on "background" videos or pseudo-positive observations derived from other event PMVs. Event detectors may then be constructed using the pseudo-labeled set and conventional supervised learning techniques.

4.4.3 Results

The main target here is to identify the optimal configuration of the developed framework by evaluating different framework configurations. Specifically, there are 5 parameterizable components, namely, a) ELM, b) CLM, c) CLM weighting scheme, d) ELM - CLM distance matrix computation, e) event detector, yielding a total of 450 different configurations.

For the evaluation, test videos were represented using a pre-trained Deep CNN (DCNN). Particularly, the 16-layer pre-trained DCNN of [SZ14], trained on the ImageNet data [DDS^{+}09], was used. Test MVs were created using the following steps: a) each video was decoded yielding 2 keyframes per second, b) the DCNN was used to provide a MV for each keyframe in $\mathbb{R}^{1000}$, and, c) average pooling was applied to retrieve a 1000-dimensional model vector for each test video.

Table 20 depicts the performance of the 10 best configurations in terms of MAP along the 30 target events. We can see that considering the difficulty of the task a quite good performance of more than 11% MAP is achieved. From these results, it is also clear that the best configurations consists of the Hausdorff distance measure for ELM - CLM distance matrix computation, the utilization of Google search engine for CLM construction, and without Tf-IDf weighting in the BoW technique.

Table 20 Top-10 best configurations (in terms of % MAP) of the proposed framework.

Matrix Operation	Weighting	ELM	CLM	Distance	% MAP
Hausdorff	no Tf-Idf	Minimum	Google	Cosine	11.11
Hausdorff	no Tf-Idf	Minimum	Google	Histog_Inter	11.09
Hausdorff	no Tf-Idf	Minimum	Google	Kullback	10.54
-	-	Title	Title	Histog_Inter	10.45
Hausdorff	no Tf-Idf	Visual Only	Google	Cosine	10.05
Hausdorff	no Tf-Idf	Visual Only	Google	Histog_Inter	9.91
-	-	Title	Title	Cosine	9.88
Hausdorff	Tf-Idf	Minimum	Google	Histog_Inter	9.78
Hausdorff	no Tf-Idf	Visual Only	Google	Kullback	9.56
Hausdorff	Tf-Idf	Minimum	Google	cosine	9.33

In order to access the significance of the different components in the overall performance of the framework, we evaluated different variations of the best configuration by changing the settings of only one component at a time. The corresponding evaluation results are shown in Tables 21, 22, 23, 24, 25, where we alternate among different event detection distance measures, CLM types, ELM types, BoW weighting schemes, and ELM - CLM matrix computation distance measures, respectively.

Table 21 Evaluation of different event detection measures.

Matrix Operation	Weighting	ELM	CLM	Distance	% MAP
Hausdorff	no Tf-Idf	Minimum	Google	Cosine	11.11
				Histogram Inter	11.09
				Kullback	10.54
				X^2	8.32
				Euclidean	6.90

From the obtained results the following conclusions are drawn:

- From Table 21, we observe that the best event detection performance is achieved with the cosine similarity measure, closely followed by the Histogram intersection distance. Surprisingly, the Euclidean

Table 22 Evaluation of different Concept Language Models.

Matrix Operation	Weighting	ELM	CLM	Distance	% MAP
Hausdorff	no Tf-Idf	Minimum	Google	Cosine	11.11
			Wikipedia		8.50
			Title		7.60

Table 23 Evaluation of different Event Language Models.

Matrix Operation	Weighting	ELM	CLM	Distance	% MAP
Hausdorff	no Tf-Idf	Minimum	Google	Cosine	11.11
		Visual			10.05
		Title			7.60

Table 24 Evaluation of different Weighting schemes.

Matrix Operation	Weighting	ELM	CLM	Distance	% MAP
Hausdorff	no Tf-Idf	Minimum	Google	Cosine	11.11
	Tf-Idf				10.05

Table 25 Evaluation of different matrix computation distance measures.

Matrix Operation	Weighting	ELM	CLM	Distance	% MAP
Hausdorff	no Tf-Idf	Minimum	Google	Cosine	11.11
l_2					8.33
Frobenius					8.27
l_∞					8.28
Max					8.04

distance provides the worst performance, resulting in a performance loss of more than 4% MAP.

- CLMs created using the Google search engine are of superior quality to the one build using Wikipedia (Table 22). As expected, CLMs utilizing only the concept title are less robust, however, still providing a 7% MAP.
- The configuration utilizing the Minimum type of ELM outperforms the rest (Table 23). This may be explained considering that the Minimum ELM describes the corresponded event more accurately by adding audio cues and short explication phrases.
- Concerning Tf-Idf weighting, although it has been proven beneficial in several applications, its use in our framework yields small decrease in the overall performance. (Table 24).
- From Table 25, we see the superiority of Haussdorf distance in the ELM - CLM distance matrix computation, where an approximately 3% MAP gain is achieved in comparison to the other methods. Concerning the rest of the methods a rather equivalent performance is observed.

Finally, an evaluation of the proposed supervised learning technique framework was also performed. For each event, a set of pseudo-positive observations were created using the target event PMVs generated from the different choices of the parameterizable components, while pseudo-negative observations were generated using Event-BG videos or pseudo-positive observations of other events. Subsequently, SVM-based detectors were created using the pseudo-set described above and the libsvm library. For the evaluation the following methods were compared:

- ES: The best framework configuration using PMV alone (i.e., without using a supervised learning algorithm).
- SVM - PMV: The best framework configuration with an SVM-based classifier for event detection, and a training pseudo-set created using the PMVs of other events.
- SVM - BG: The best framework configuration with an SVM-based classifier for event detection, and a training pseudo-set created using videos of the Event-BG dataset.

– ES - SVM: Linear late fusion of the scores derived after the application of ES and SVM - PMV.

From the obtained results, as depicted in Table 26, it can be seen that concerning the individual methods, the best performance is achieved using the non-supervised configuration (ES), closely followed by the configuration utilizing supervised learning and other events PMV-based pseudo-positive observations to create a training dataset (SVM - PMV). On the other hand, the method utilizing pseudo-negative observations created using real world videos (SVM - BG) clearly underperforms the other two by more than 4% MAP. Finally, the results of the combined method (ES - SVM) provide a rather significant gain of more than 2% MAP.

Table 26 Performance evaluation of the proposed framework using different event detection methods.

ID	ES	SVM-PMV	SVM-BG	ES+SVM
E021	0.1709	0.1304	0.0948	0.1612
E022	0.1340	0.0971	0.0083	0.1367
E023	0.1412	0.1143	0.0207	0.1501
E024	0.0247	0.0141	0.0106	0.0161
E025	0.0254	0.0202	0.0189	0.0213
E026	0.0241	0.0304	0.0334	0.0272
E027	0.0262	0.0583	0.0177	0.0256
E028	0.0300	0.0308	0.0164	0.0309
E029	0.1174	0.0571	0.0148	0.1027
E030	0.1177	0.0603	0.0271	0.1217
E031	0.7270	0.8320	0.5706	0.8317
E032	0.0664	0.0482	0.0263	0.0737
E033	0.0817	0.0433	0.0571	0.0753
E034	0.1892	0.0203	0.0242	0.0952
E035	0.2156	0.1303	0.0226	0.1932
E036	0.0383	0.0403	0.0193	0.0359
E037	0.2350	0.2799	0.1316	0.4583
E038	0.1102	0.0260	0.0076	0.0986
E039	0.0265	0.0128	0.0085	0.0155
E040	0.2085	0.3936	0.3379	0.3477
E041	0.0317	0.0139	0.0067	0.0282
E042	0.0436	0.0658	0.0220	0.0440
E043	0.0134	0.0137	0.0126	0.0133
E044	0.0688	0.0236	0.0085	0.0475
E045	0.2365	0.1354	0.1470	0.2322
E046	0.1497	0.2048	0.0394	0.2494
E047	0.0302	0.0664	0.0463	0.0293
E048	0.0136	0.0201	0.0083	0.0145
E049	0.0258	0.0388	0.0092	0.0278
E050	0.0096	0.0154	0.0151	0.0087
MAP	11.11%	10.13%	5.94%	12.38%

4.5 Conclusion

In this section we presented the results of the conducted evaluations (both in-house experiments and participations at benchmarking activities) of the developed methods for video event detection that can be used for extended video annotation. Specifically, accelerated versions of our GSDA-LSVM method that was presented in section 8.2 of D1.4, have been developed using C++ GPU libraries, and evaluated during our participation in the MED task of TRECVID 2014 benchmarking activity for the task of event detection, and on several subsets of MED and SIN datasets for both tasks of event and concept detection. In a parallel effort, acknowledging the increased interest on textual analysis-based techniques for pattern detection, a zero-example learning technique was implemented and tested providing promising evaluation results in a MED 2014 video subset.

5 Evaluation of object re-detection

5.1 Overview

For instance-based labelling of videos and for the creation of object-specific spatiotemporal media fragments that can be used for hyperlinking, we developed the object re-detection algorithm introduced in [AMK13]. This method was described in section 8.2 of D1.2, while after its first implementation, an extension that takes multiple instances of an object as input and supports the re-detection of 3-dimensional objects that appear under varying viewing positions was also developed. Within the scope of LinkedTV this method is mainly used as an internal analysis component of the developed chapter segmentation algorithm for the videos from the documentary scenario, i.e., for the re-detection of a visual cue that temporally demarcates the beginning of the different chapters of the "Tussen Kunst & Kitsch" show of AVRO [8], as described in section 2.2.

However, the analysis capabilities and the usefulness of the developed algorithm are much wider, as reported in section 8.3 of D1.2 and at [AMK13]. The designed method exhibited remarkable performance, both in terms of detection accuracy and time efficiency, enabling the detection of a scaled and rotated/occluded instances of the given object in the frames of the video, requiring running time that allows faster than real-time analysis. Based on this, and motivated by our goal for building a tool for real-time object-specific spatiotemporal labeling of videos we tried to investigate potential modifications or extensions of this method that would lead to further reduction of the needed processing time, ensuring nevertheless similar levels of detection accuracy. The conducted study and the performed evaluations (using CPU-based processing only) are presented in details in the following subsections.

5.2 Accelerated object re-detection

Starting from the algorithm of [AMK13], we initially tried to indicate the most time consuming parts of the analysis. As described in section 8.2 of D1.2 the processing steps performed when matching a pair of images (also illustrated in the block diagram of Fig. 13) are: (a) the detection of interest points from each image, (b) the extraction of descriptor vectors for the detected interest points, (c) the pair-wise matching of the computed descriptor vectors and (d) the filtering of the outliers based on geometric validation. After this, a final decision about the matching is taken based on a simple thresholding of the number of matched descriptors. During the object re-detection analysis the parts related to the detection and description of interest points from the given object are applied only once, while the entire chain of analysis is performed for every frame of the video that is matched against the object.

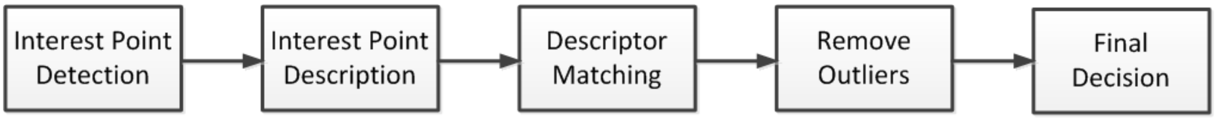


Figure 13 Block diagram of the core analysis steps that are applied by the object re-detection algorithm of [AMK13].

Our first evaluation aimed to measure the processing time required by each of these analysis steps (i.e., the steps (a) to (d) described above). In our experiments we used a set of 5 objects and we matched each one of them against three video frames that contain scaled and rotated/occluded instances of it. The employed dataset is depicted in Fig. 14 and the experimental results are presented in Fig. 15. From this evaluation it was shown that the most computationally intensive part of the analysis is the one related to the extraction of the descriptor vectors which takes almost 46% of the overall execution time. The interest point detection and the matching of the extracted descriptors follow consuming 28% and 24% of the entire analysis time respectively. The applied geometric validation for filtering-out erroneous matches corresponds to a very small fraction (only 2%) of the total processing time. Based on these findings, we decided to concentrate on the investigation of alternative techniques for implementing the three most computationally demanding parts of the image matching process. In every case, the efficiency of each tested combination was evaluated based on the required processing time, while the object re-detection accuracy was also taken under consideration.

For the interest point detection part of the analysis we compared the used SURF detector that is employed in [AMK13] against a set of recently proposed methods. Specifically, in our experiments we considered (a) the MSER [MCUP04] detector which detects regions that remain stable after a number of intensity thresholdings

[8] http://www.avro.tv/

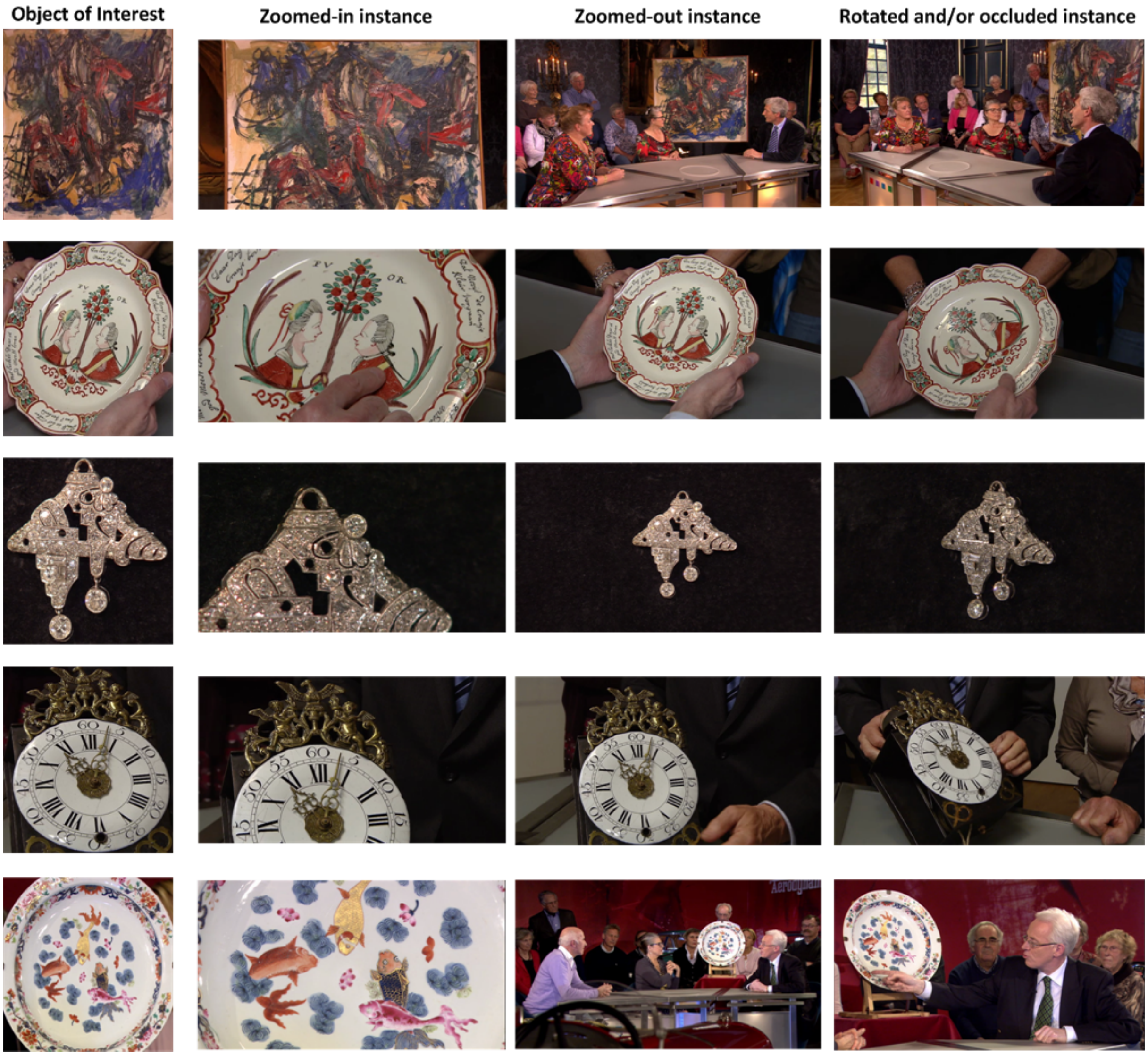


Figure 14 The set of images used in our experiments for evaluating the time efficiency of each step of the object re-detection analysis pipeline.

of the image, (b) the ORB [RRKB11] detector which relies on the FAST corner detection algorithm and enhances it with the use of an image scale pyramid and the Harris corner measure, and (c) the BRISK [LCS11] detector which uses another variation of the FAST algorithm (i.e., the AGAST corner detector) and detects interest points in a scale-space pyramid. The results of this assessment are presented in Tables 27 and 28.

Table 27 reports the average number of the detected interest points from each image of the utilized dataset (first row), as well as the time needed for their detection (second row). Moreover, the required detection time per interest point for each method was also computed (third row). The reported time values are expressed in milliseconds. The Hessian parameter of the SURF detector was set to 400, as in the method of [AMK13]. For the rest of the evaluated detectors the default values, as proposed by the employed OpenCV[9] library, were used. As shown in Table 27 the most time efficient methods are the ORB and the BRISK detectors. These techniques have similar performance, requiring approximately 10 msec for analysing a frame and 0.018 msec for the detection of an interest point. SURF is almost 4 times slower compared to BRISK and 5 times slower compared to ORB, however this difference is smaller if the number of detected interest points is taken under consideration. In particular the number of the SURF-based detected interest points is 2 and 2.5 times larger than the number of the detected points using the BRISK and the ORB method respectively. The latter highlights the discriminative ability of the SURF algorithm and indicates that this method is less than two times slower compared to BRISK and ORB in terms of the needed processing time per interest point. Finally,

[9] http://opencv.org/

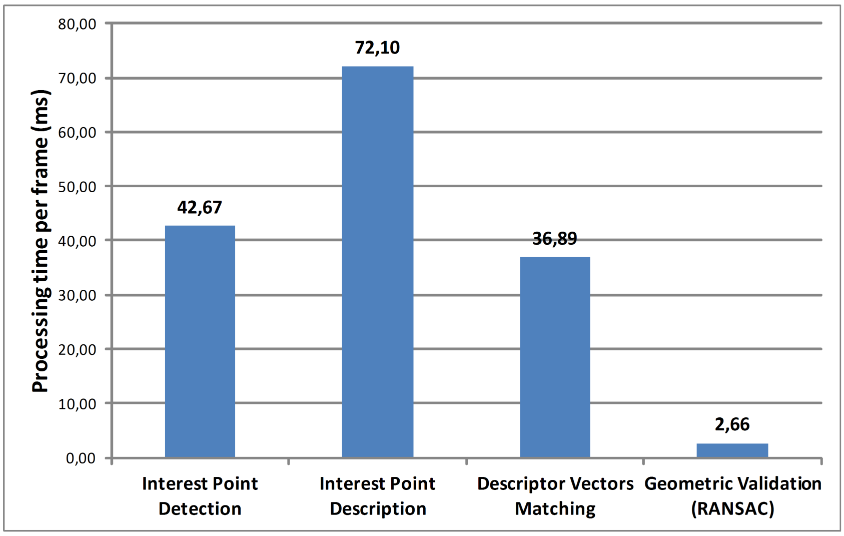


Figure 15 Required processing time (in msec) of each step of the algorithm proposed in [AMK13].

MSER exhibited the worst time performance among the tested approaches, detecting at the same time the smallest number of interest points per frame.

Table 27 Time performance of the evaluated approaches for interest point detection.

	SURF	**MSER**	**BRISK**	**ORB**
Average Number of Interest Points	1340	378	688	500
Average Time/Frame (msec)	42.67	90.23	12.14	8.83
Average Time/Interest Point (msec)	0.032	0.24	0.018	0.018

Table 28 illustrates the cases where each examined approach succeeded (✓) or failed (**X**) to detect the given object in the corresponding frames that included scaled (i.e., zoomed in (zi) or zoomed out (zo)) and rotated/occluded (r/o) instances of it. These data indicate that the SURF detector clearly outperforms the other evaluated methods, resulting in successful re-detection of the object in all considered cases. The other approaches led to a number of misdetections which varies between 20% and 53% (for MSER and BRISK detector respectively). Based on the outcome of this evaluation and motivated by our goal to accelerate the analysis without compromising the detection accuracy of the algorithm, we concluded to the use of the SURF detector, despite the fact that this method is not the fastest among the tested ones.

Table 28 Object re-detection accuracy of the tested approaches for interest point detection.

	Interest Point Detectors											
	SURF			MSER			BRISK			ORB		
	zi	zo	r/o	zi	zo	r/o	zi	zo	r/o	zi	zo	r/o
Object 1	✓	✓	✓	✓	✓	✓	✓	**X**	✓	✓	**X**	✓
Object 2	✓	✓	✓	✓	✓	✓	**X**	**X**	**X**	✓	**X**	**X**
Object 3	✓	✓	✓	**X**	✓	**X**	**X**	✓	**X**	**X**	✓	✓
Object 4	✓	✓	✓	✓	✓	**X**	**X**	✓	✓	✓	✓	✓
Object 5	✓	✓	✓	✓	✓	✓	✓	✓	**X**	✓	✓	✓

Having defined the most effective method for interest point detection, we then tried to investigate alternatives for the extraction of descriptor vectors. Considering the current research trend in using binary descriptors for computer vision applications, we decided to evaluate the performance of some of these new descriptors for the task of object re-detection. In particular, we examined the BRISK [LCS11], the ORB [RRKB11], the

FREAK [AOV12] and the BRIEF [CLO$^+$12] descriptors. As before, we compared both time performance and re-detection accuracy of these binary descriptors with the performance of the already utilized SURF algorithm. Concerning time performance, besides the processing time needed for computing the descriptor vectors we also measured the elapsed time for their matching, since this is another factor that also affects the overall time efficiency of these methods. The experimental results from these experiments are presented in Tables 29 and 30. Similarly as before, Table 29 contains information about the time efficiency of these methods, considering the time consumed for both extracting and matching the descriptor vectors, while Table 30 reports the achieved object re-detection performance of each evaluated approach.

Specifically, Table 29 presents the size of the computed vector from each algorithm (first row), which directly affects the time needed for its extraction (second row) and matching (third row). As before, the reported time values are expressed in milliseconds. From these measurements the BRIEF algorithm seems to be the fastest one requiring only 4 msec per frame for descriptor extraction, while the ORB and BRISK methods follow being almost 2 times slower. The most time consuming binary approach is the FREAK algorithm which is several times (5 to 10) slower compared to the other ones, while the floating point representation used by the SURF method makes the latter by far the most computationally expensive, and thus the slowest approach. However, taking into consideration the time required for matching the computed descriptor vectors (since this part is directly affected by the employed descriptor) it occurs that the three fastest approaches mentioned above (i.e., the BRIEF, the ORB and the BRISK method) exhibit similar overall time performance, being significantly faster (approximately 5 times) compared to the SURF method used in [AMK13].

Table 29 Performance comparison of the tested approaches for interest point description.

	SURF	**BRISK**	**ORB**	**FREAK**	**BRIEF**
Size	128	64	32	64	32
Description Time/Frame	72.10	7.77	7.93	44.04	4.07
Matching Time/Frame	26.83	12.74	9.27	7.37	13.68
(Description + Matching) Time/Frame	98.93	20.51	17.20	51.41	17.75

The findings related to the object re-detection accuracy of these methods are presented in Table 30. As can be seen, the SURF method achieves the highest accuracy while competitive performance is also exhibited by the BRISK and the FREAK algorithms. The remaining techniques (i.e., the BRIEF and the ORB methods) appear to be the most inefficient ones, resulting in a remarkable number of misdetections. Based on the outcome regarding the time efficiency of the evaluated approaches (as reported in Table 29), it can be easily concluded that the BRISK method is the most suitable for this kind of analysis. This technique ensures high levels of object re-detection accuracy, similar to the one obtained by SURF and FREAK methods, while being the least time consuming among them (up to 5 times faster).

Table 30 Performance comparison of the tested approaches for interest point description.

	SURF/SURF			**SURF/BRISK**			**SURF/ORB**			**SURF/FREAK**			**SURF/BRIEF**		
	zi	zo	r/o	zi	zo	r/o	zi	zo	r/o	zi	zo	r/o	zi	zo	r/o
Object 1	✓	✓	✓	✓	✓	✓	**X**	**X**	**X**	✓	**X**	✓	**X**	**X**	**X**
Object 2	✓	✓	✓	✓	✓	✓	**X**	**X**	✓	✓	✓	✓	**X**	✓	✓
Object 3	✓	✓	✓	✓	✓	**X**	**X**	✓	**X**	✓	✓	✓	**X**	✓	✓
Object 4	✓	✓	**X**	✓	✓	**X**	**X**	✓	**X**	✓	✓	**X**	**X**	✓	✓
Object 5	✓	✓	✓	✓	✓	✓	**X**	✓	✓	✓	✓	✓	✓	✓	**X**

Based on the findings of our evaluations so far, we decided on methods for the detection (i.e., SURF) and description (i.e., BRISK) of interest points from images. So, in the next step we focused on techniques for matching the extracted descriptor vectors. Specifically, we examined an entirely different approach for fast yet approximate nearest neighbour search that relies on the use of the Locality Sensitive Hashing (LSH) algorithm. After a set of experiments for tuning the LSH algorithm's parameters, a configuration that uses 3 hash tables and 5 bits for the hash key was finally selected, as exhibited the best trade-off between matching accuracy and required processing time.

For each tested combination of methods (i.e., SURF and Brute-Force, BRISK and LSH), we initially evaluated the descriptor matching efficiency by counting the number of defined correspondences (i.e., matches) between the paired sets of descriptors before and after filtering outliers via the applied geometric verification step. Specifically, based on the evaluation procedure described in [Khv12] we measured (a) the number of initially matched descriptors (denoted as IM from now on) after applying the 2-NN search and the distance ratio test, and (b) the number of filtered matches (denoted as FM from now on) after the geometric verification via applying the RANSAC algorithm [FB81]. Then, we computed the percentages of initially matched pairs of descriptors and the percentages of correct matches that passed the RANSAC test, using the formulas below:

$$\begin{aligned} \text{Match Rate [\%]} &= \frac{\text{IM}}{\min(O_p, F_p)} * 100 \\ \text{Correct Match Rate [\%]} &= \frac{\text{FM}}{\text{IM}} * 100 \end{aligned} \tag{1}$$

where O_p and F_p is the number of detected interest points by the SURF algorithm in the object of interest (O) and the video frame (F) respectively.

The results of this experiment are reported in Table 31. As shown, the extraction of BRISK descriptors and their matching using the LSH indexes result in equivalent or higher percentages (up to 10% in the most challenging case of rotated and/or occluded instances) of both initial and correct matches compared to the corresponding approach that relies on SURF descriptors and Brute-Force matching. This finding enhances the belief that BRISK descriptors are more suitable for image matching purposes compared to SURF, a conclusion that becomes more important considering the improvement in terms of required processing time, as will be reported below (see Table 32).

The low percentages of matched descriptors reported in Table 31 are explained by the fact that only a restricted subset of the descriptors extracted from the object of interest can be matched when pairing the object with a video frame that includes a scaled/rotated/occluded instance of it, and not with a transformed instance of it, as is the case in [Khv12]. Specifically, a zoomed-in instance may missing some areas of the object of interest, and so the descriptors extracted from these areas of the object cannot be matched with any of the descriptors extracted from the zoomed-in instance. The same applies for the video frames containing occluded instances of an object. On the other hand, a zoomed-out instance covers only a small area of the overall video frame, so the descriptors extracted from the object of interest can be matched with a very small set of descriptors that were extracted from this specific area of the video frame.

Following, we assessed the effectiveness of each approach in terms of processing time and provided accuracy for object re-detection. For evaluating the time efficiency of each configuration we used the same dataset as before (i.e., the 5 objects and 3 frames for each of them, depicting scaled and rotated/occluded instances of the objects) and we measured the time needed for matching the computed descriptors. Moreover, the object re-detection performance of each method was also evaluated. The outcomes of these experiments that are presented in Table 32 indicate that the combination of BRISK descriptors with the LSH-based matching strategy can achieve competitive performance compared to [AMK13] in terms of object re-detection accuracy. However, the remarkable speed-up of the descriptor matching process (2.5 times faster analysis) justifies the employment of LSH as a suitable approach for matching the extracted BRISK descriptors.

Specifically, we considered the scenario where interest point detection and description can be applied for the entire set of video frames during an off-line step, storing the computed data for each frame in a file. Then, during the on-line step of the analysis, where the re-detection of instances of the given object in the frames of the video is performed, these pre-computed files are loaded and used for matching. For this purpose, we explored several options regarding the storage of the computed BRISK descriptors. Among a variety of file formats, such as text files, binary files and xml files, and driven by the size of the created files which strictly affects the storage cost of the algorithm we ended up to the use of binary files due to their smallest file size. Moreover, aiming to further improve the computation efficiency of the off-line step of this approach we serialized the computed matrices of interest points and descriptor vectors before storing them.

After the configuration of the off-line procedure we used 6 videos from the dataset and we counted the average number of frames that can be processed per second during the re-detection of the object throughout the video. The considered methodologies were: (a) the use of the SURF algorithm for both detection and description of interest points, (b) an accelerated version of (a) using GPU-based (where GPU stands for Graphics Processing Unit) parallel computing, (c) a combination of GPU-based parallelized SURF interest point detector and CPU-based BRISK descriptor extractor, and (d) the loading of binary files with the pre-computed BRISK descriptors. Aiming to measure only the time needed for interest point detection and

Table 31 Performance comparison of the different methodologies, in terms of percentages of matched pairs of descriptors before and after filtering outliers.

	SURF/BF					
	zi		zo		r/o	
	Match Rate [%]	Correct Match Rate [%]	Match Rate [%]	Correct Match Rate [%]	Match Rate [%]	Correct Match Rate [%]
Object 1	8.02	91.67	7.35	100.00	6.68	96.67
Object 2	10.51	86.32	1.77	100.00	7.30	84.85
Object 3	0.88	66.67	14.62	98.00	1.61	54.55
Object 4	6.26	66.04	25.97	94.55	6.26	58.49
Object 5	17.19	82.24	29.07	94.55	3.96	42.86
Average	8.57	78.58	15.76	97.42	5.16	67.48
	BRISK/LSH					
	zi		zo		r/o	
	Match Rate [%]	Correct Match Rate [%]	Match Rate [%]	Correct Match Rate [%]	Match Rate [%]	Correct Match Rate [%]
Object 1	10.91	91.84	8.69	100.00	7.57	100.00
Object 2	11.06	91.00	1.99	83.33	7.19	87.69
Object 3	0.88	66.67	20.32	98.56	1.02	71.43
Object 4	7.08	58.33	32.00	95.57	8.15	52.17
Object 5	19.57	87.86	37.90	97.61	4.52	77.50
Average	9.90	79.14	20.18	95.02	5.69	77.76

Table 32 Performance evaluation, in terms of run time (in msec), of the two tested configurations for object re-detection.

Configuration	SURF-SURF-BF		SURF-BRISK-LSH	
Object	Matching Time	Correct Detection	Matching Time	Correct Detection
1 zoomed in	17.72	✓	14.45	✓
1 zoomed out	17.64	✓	12.61	✓
1 rotated/occluded	19.04	✓	12.55	✓
2 zoomed in	21.12	✓	9.98	✓
2 zoomed out	17.26	✓	7.57	✓
2 rotated/occluded	13.26	✓	8.75	✓
3 zoomed in	64.5	✓	9.42	✓
3 zoomed out	5.82	✓	5.56	✓
3 rotated/occluded	44.50	✓	6.17	**X**
4 zoomed in	46.61	✓	19.13	✓
4 zoomed out	17.96	✓	13.21	✓
4 rotated/occluded	40.24	**X**	13.13	**X**
5 zoomed in	23.51	✓	14.20	✓
5 zoomed out	17.59	✓	13.16	✓
5 rotated/occluded	37.19	✓	13.15	✓
Average	26.93		11.54	
SD (σ)	15.95		3.59	

description we did not take into account neither the time needed to load the video frames or for transferring data between CPU and GPU.

According to the findings that are reported in Table 33, the use of GPU-based parallel processing can accelerate the corresponding CPU-based implementation 6 times, resulting in a faster-than-real-time analysis. Moreover, the replacement of the GPU-based parallelized SURF algorithm for descriptor extraction by the related CPU-based BRISK method accelerates further (by a factor of 1.5) the analysis, making it over 2 times faster than real-time processing. However, the loading of binary files with the pre-computed descriptors from the entire set of video frames increases rapidly the processing speed, making the analysis 13 times faster than real-time processing. However, as an aftereffect of this off-line analysis we must report that the time required for pre-processing a video with 720×480 resolution, 45 minutes duration and 25 fps frame-rate (i.e., approximately 68000 frames) is around 24 minutes, while the required storage space for the created binary files is about 3GB.

Table 33 Average processing frame rate for different on-line interest point detection and description configurations in comparison to loading files with pre-computed descriptors.

Video Resolution	SURF (frames/sec)	GPU_SURF (frames/sec)	GPU_SURF & BRISK (frames/sec)	Loading Bin. Files (frames/sec)
720x480	6	36	55	326

After this clear evidence about the effectiveness of this pre-processing step, in terms of re-detection time, we then made a number of experiments regarding the time performance of the entire object re-detection pipeline, using again the dataset of Fig. 14. In particular we compared the required processing time of the following four approaches: (a) the method of [AMK13] when only CPU-based processing is employed, (b) the new designed method that combines SURF interest point detector with BRISK descriptor extractor and LSH-based matching, (c) the method of (b) when a prior video analysis is performed and (d) the method proposed in [AMK13] which takes advantage of GPU-based parallel processing and also enables faster-than-real-time-analysis. The results of this evaluation are presented in Table 34, where the run time is expressed as a factor of real-time processing (a value below 1 indicates faster than real-time analysis).

Table 34 Performance evaluation in terms of run time, where the latter is expressed as a factor of real-time processing (a value below 1 indicates faster than real-time analysis), of the four tested configurations for object re-detection.

		Processing Time (as a factor of real-time processing)			
	# frames depicting the object	SURF/SURF BF/CPU	SURF/BRISK BF/CPU	SURF/BRISK/LSH (Pre-computed data)	SURF/SURF BF/GPU
Object 1	4914	0.455	0.198	0.065	0.090
Object 2	4256	0.312	0.208	0.027	0.077
Object 3	1819	0.289	0.119	0.026	0.062
Object 4	1648	0.277	0.136	0.036	0.089
Object 5	1428	0.592	0.091	0.038	0.074
Average	-	0.385	0.150	0.038	0.078
SD (σ)	-	0.126	0.047	0.015	0.011

As shown in Table 34, the findings of our study regarding the time performance of the core analysis parts of the object re-detection pipeline led to a new configuration that employs the SURF algorithm for interest point detection, the BRISK algorithm for interest point description and the LSH algorithm for descriptor matching. When only CPU-based processing is used the developed method is over 2 times faster than the corresponding method that relies on the SURF algorithm for interest point detection and description and the Brute-Force approach for matching them. Moreover, our idea about pre-processing the video frames during an off-line step, and the loading of the pre-computed data (i.e., interest points and descriptor vectors) for re-detecting the object in the video frames during an on-line step has proven to be quite effective, resulting in a remarkable reduction of the time required for re-detection. Specifically, the needed processing time is 4 to 5 times smaller compared to the time needed if this pre-processing step was not performed, while it is 2 times faster than the algorithm of [AMK13] which employs GPU-based parallel processing.

From the outcomes of the evaluations presented above, we ended-up to a new framework for object re-detection that is depicted in Fig. 16. The developed approach was realized using the OpenCV[10] (ver. 2.4.9) library for visual analysis and the Boost [11] (ver. 1.55.0) library for the serialization of the computed descriptor vectors, and consists of two steps. During the off-line step, interest point detection and description are

[10] http://opencv.org/

[11] http://www.boost.org

performed for every frame of the video using the SURF and BRISK methods respectively. The computed data are stored in the disk as serialized binary files. Then, during the on-line step a number of manually selected instances of the object of interest are given as input to the algorithm, and zoomed-out versions of them are created by shrinking the original images into smaller ones using nearest neighbour interpolation as performed in [AMK13]. These scaled-down instances will be used for the detection of extremely downsized instances of the object that may appear in the frames of the video, while contrary to the algorithm of [AMK13] there are no zoomed-in instances created, since no significant advantage of the algorithm's detection accuracy was observed in our experiments. When these instances are available, SURF-based interest points and BRISK-based descriptor vectors are also extracted from each one of them. Afterwards, we again exploit information about the shot-level structure of the selected video (which is also given as input in the on-line step of the framework), by applying the same video-structure-based frame sampling strategy described in [AMK13] for reducing the number of frames that need to be checked. The pre-computed data (i.e., the stored interest points and descriptor vectors) of these frames only, are loaded and used by the LSH algorithm which performs fast matching of descriptors based on the applied approximate nearest neighbour search. For the selection of the best matches the distance ratio criterion described in [AMK13] is used, while outliers that may occur from the matching process are filtered out by applying a geometric verification process which relies on the RANSAC algorithm.

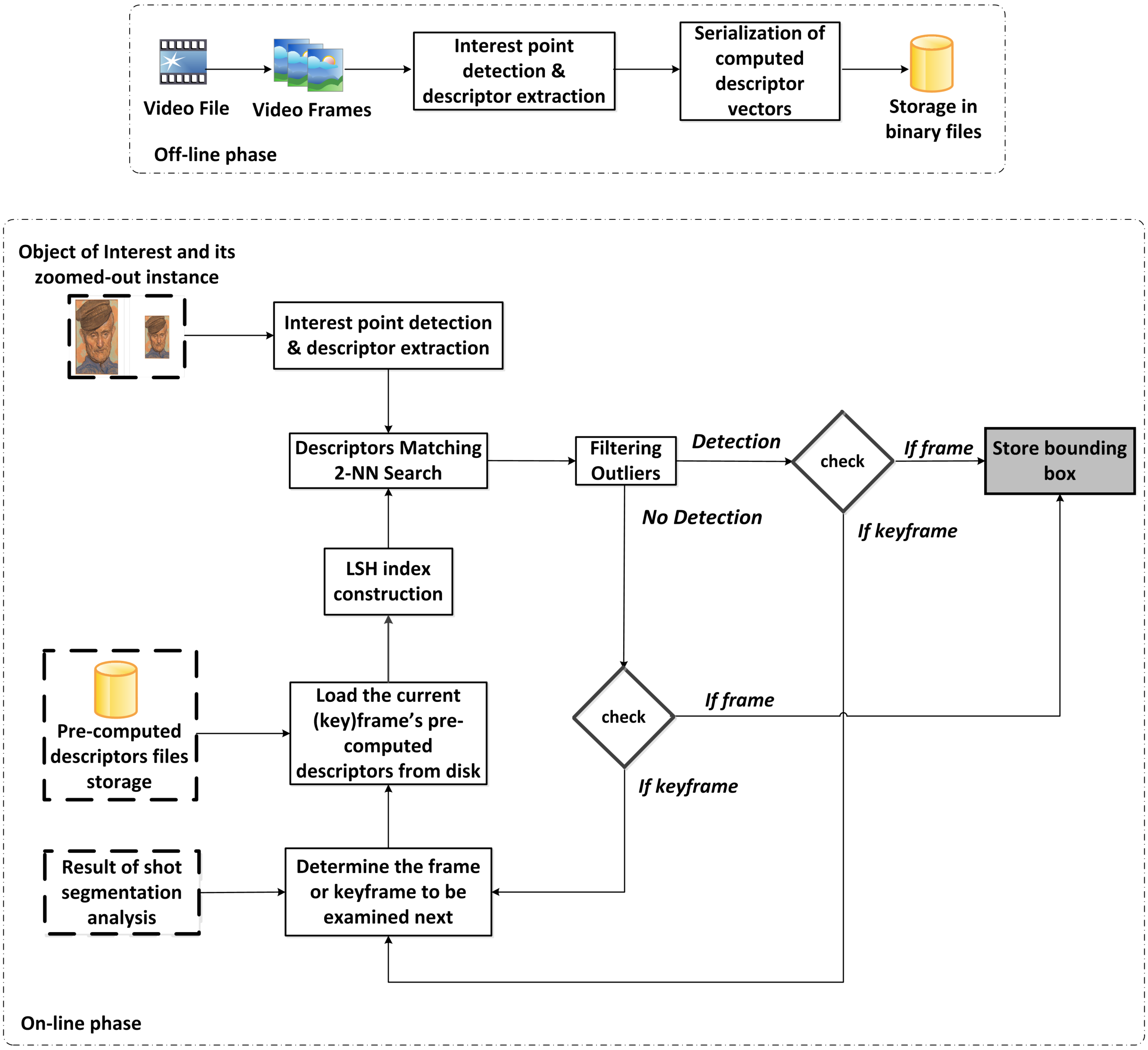

Video File
Video Frames
Interest point detection & descriptor extraction
Serialization of computed descriptor vectors
Storage in binary files
Off-line phase
Object of Interest and its zoomed-out instance
Interest point detection & descriptor extraction
Descriptors Matching 2-NN Search
Filtering Outliers
Detection
check
If frame
Store bounding box
No Detection
If keyframe
LSH index construction
Pre-computed descriptors files storage
Load the current (key)frame's pre-computed descriptors from disk
check
If frame
If keyframe
Result of shot segmentation analysis
Determine the frame or keyframe to be examined next
On-line phase

Figure 16 The proposed object re-detection framework. The dashed line boxes indicate the algorithm's input, while the gray shaded one represents the output.

Having decided about the different core analysis components of the new object re-detection framework, we evaluated more extensively its performance, both in terms of time efficiency and detection accuracy, using a large set of objects and videos. Specifically, the employed dataset is twice the size of the dataset reported in [AMK13] and consists of (a) 12 episodes of the cultural heritage show “Tussen Kunst & Kitsch” of the Dutch public broadcaster AVRO [12] with total duration 545 minutes and (b) 60 manually selected objects that appear in these videos. The selected objects include paintings, cards, plates and teapots, small carpets, pieces of jewellery and clocks. Some of them are 2-dimensional, such as paintings, carpets, cards and posters, while others are 3-dimensional which are exhibited under different viewing positions such as clocks, books, art boxes and jars, or on top of a rotating disk, such as small statues or collections of glasses. Indicative examples of objects from the used dataset are illustrated in Fig. 17.

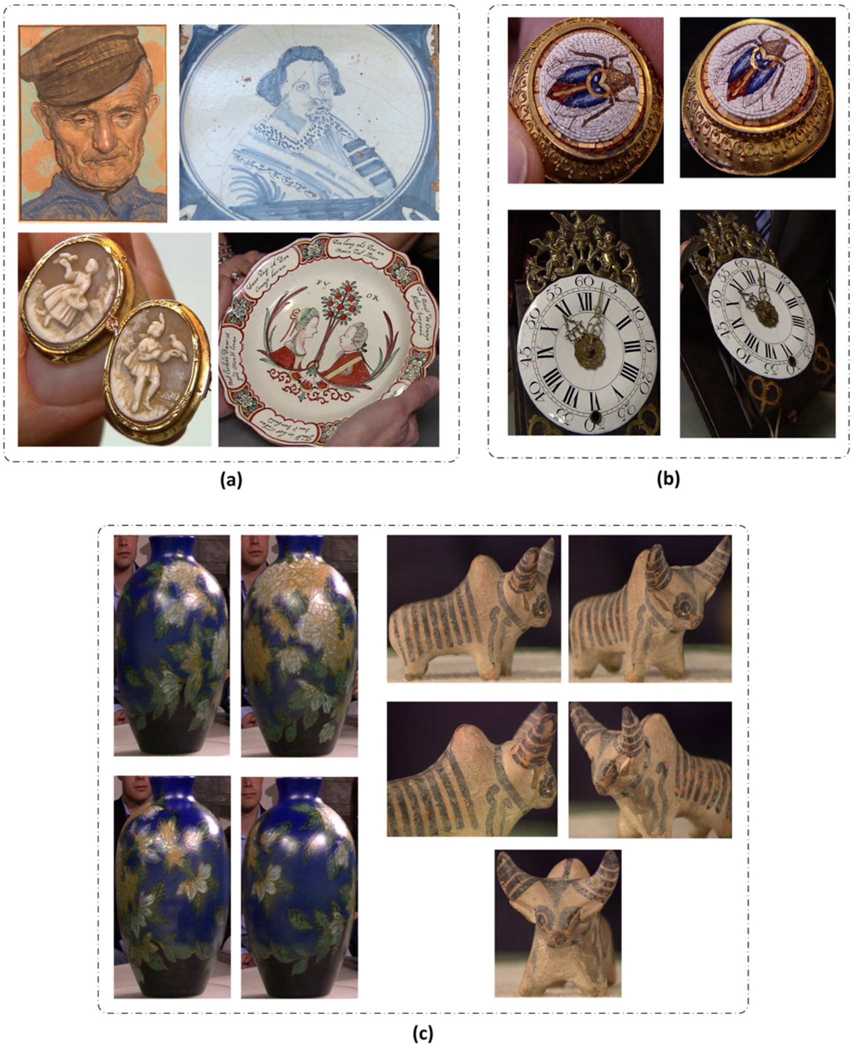


Figure 17 Examples of objects of interest used in our experiments. (a) 2-dimensional objects, (b) 3-dimensional objects taken from different points of view, (c) multiple instances of 3-dimensional objects which are demonstrated on a rotating disk.

Based on the ground truth for this dataset (which was created via human observation), 127.764 video frames contain at least one instance of the considered objects, whereas none of the selected objects appears in the remaining 689.469 frames. The used metrics in our evaluations were (a) the Precision, (b) the Recall and (c) the F-score, while the time efficiency of each tested approach was evaluated by expressing the needed processing time as a factor of real-time processing i.e., comparing these times with the actual duration of the processed videos (thus, similarly as before, a factor below 1 indicates faster-than-real-time processing). The experiments were conducted on a system with an Intel Core i7-2600K processor (with 4 cores, 8 threads and 3.4 GHz base frequency), 8 GB RAM memory and an NVIDIA GeForce GTX560 graphics card (with 336 CUDA cores). Parallel (multi-core) computing was used only in the case where GPU-based processing was

[12] `http://avro.nl`

employed for accelerating the analysis.

The overall performance of the new developed algorithm was compared against the performance of the method from [AMK13]. The experimental results that are shown in Table 30, indicate that both of these methods exhibit great re-detection accuracy. The new method is slightly lacking (approximately by 5%) in terms of Recall but still its accuracy is remarkably high. The algorithm successfully detects the given object under a variety of different viewing conditions as illustrated in Fig. 18. However, as can be seen from the last column of Table 35, the proposed method achieves a remarkable reduction of the time needed for analysis despite the fact that the entire re-detection process is performed using CPU-based processing only. Specifically, the proposed approach is 5 to 7 times faster compared to the method of [AMK13], requiring for processing only 3% of the video duration (average value). The wide range of time values reported in the last column of Table 35 is explained by the varying number of frames that each object appears in, which affects the number of shots and video frames where a more detailed analysis must be performed (according to the frame-sampling strategy that is applied by the algorithms), while other factors that affect the processing time are the number and size of the object's instances that are given as input to the algorithm.

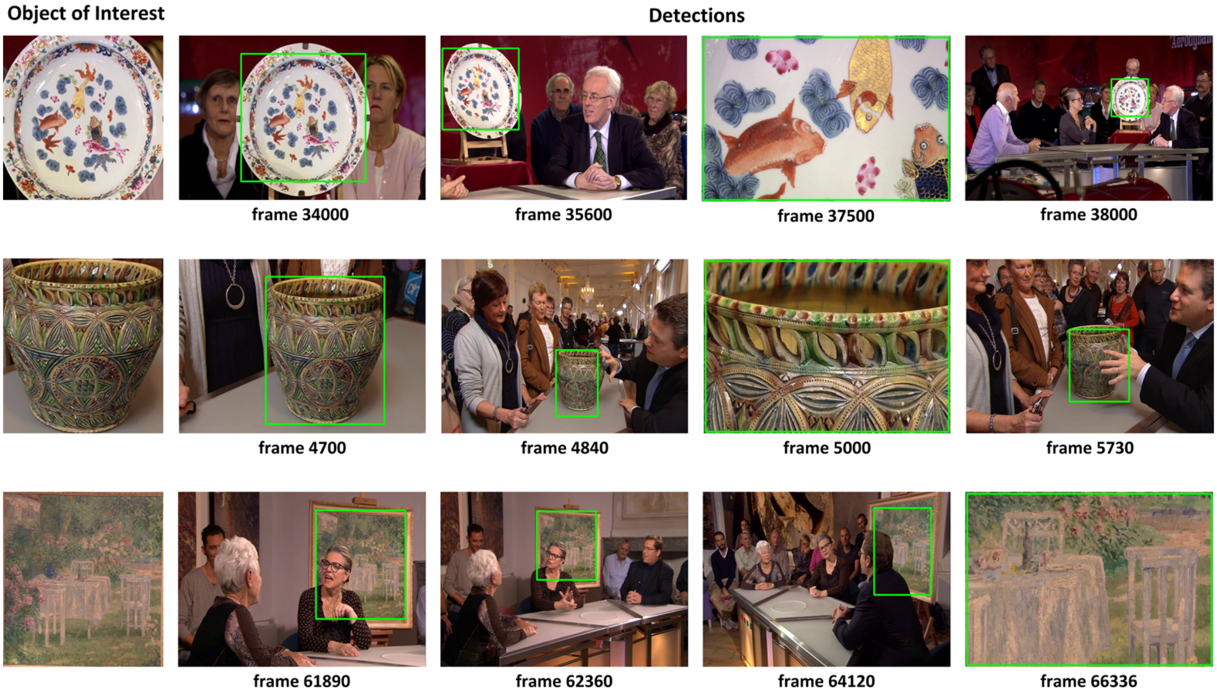


Figure 18 Objects of interest (left column) and their detected instances (in green bounding boxes) under different conditions such as zoomed in/out state, occlusion, occlusion-rotation.

Table 35 Performance comparison between the algorithm of [AMK13] and the new developed framework.

	Precision	Recall	F-Score	Time (× Real-Time)
method of [AMK13]	0.997	0.909	0.951	0.039 - 0.445
new approach	0.999	0.851	0.919	**0.006 - 0.085**

5.3 Results of user study on instance-level video labeling

Aiming to evaluate the performance of the object re-detection algorithm as a method for instance-based spatiotemporal labeling of videos, we developed a tool that enables users to annotate a video based on the appearance of a specific object of interest. This tool has a web interface and can be used as a standalone tool or can be integrated into the Editor Tool. Specifically, it allows the user to select one or more object(s) or region(s) of interest that appear in the video, and then it performs the re-detection of the selected area(s) throughout the whole video. No prior knowledge about the underlying object re-detection algorithm is required for using the tool. The implemented interface is interactive and simple to use, while brief instructions-of-use are also provided within the web interface of the tool (see Fig. 19).

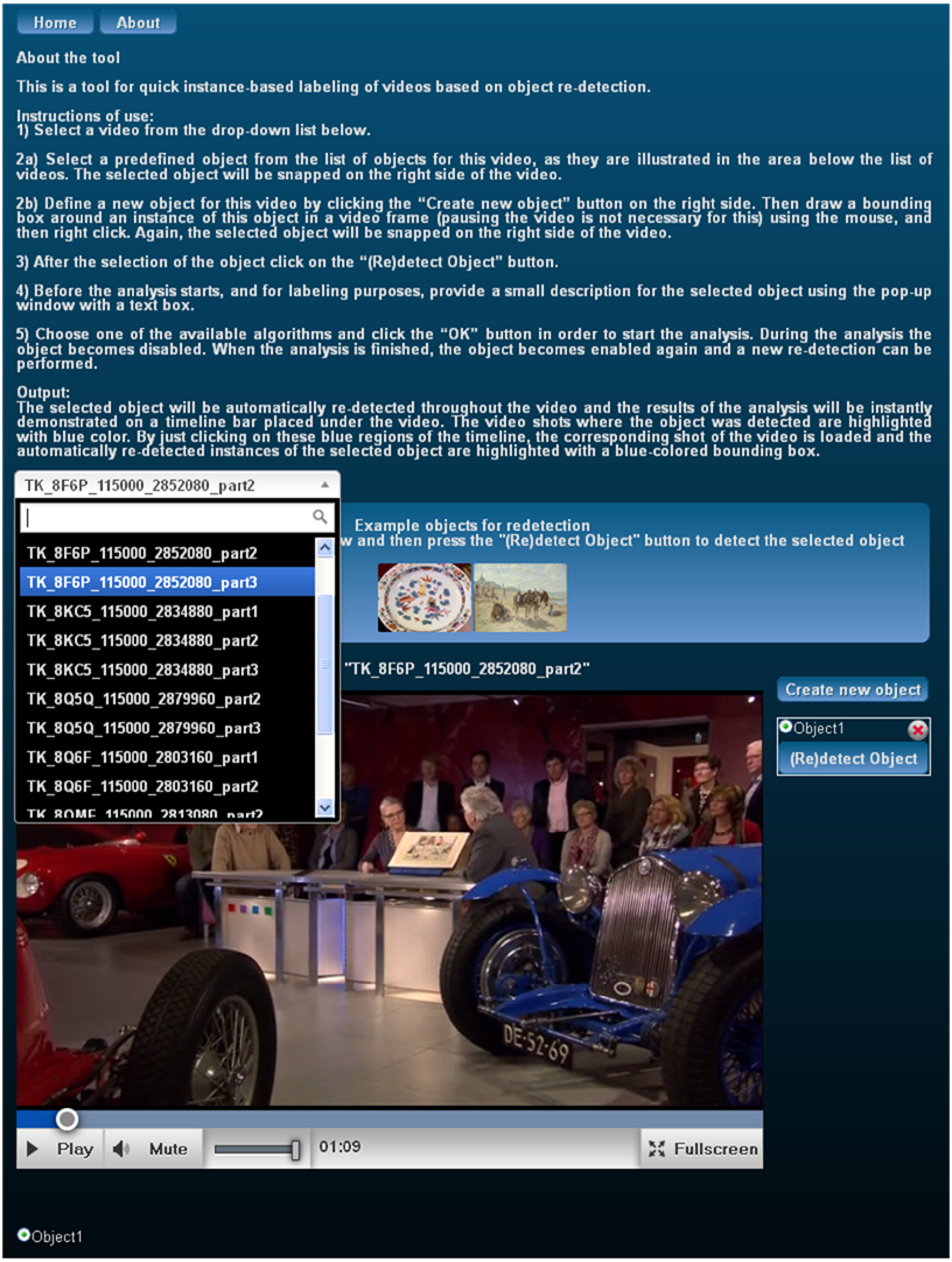
Initially the user is prompted to select a video from a drop-down list (see Fig. 19), which is then played by the video player of the tool. Afterwards, the user can select an object that appears in the video, by drawing a bounding box around an instance of it in one of the video frames, either during playing the video or after pausing it (as shown in Fig. 20). When this initial selection of the area around the object is done, the tool allows the user to adjust the position and the size of the bounding box in order to end up with the most appropriate (i.e., the most accurate) spatial demarcation of the object's instance. After the spatial re-arrangement of the bounding box ends, the selection of the object's instance is performed simply by right-clicking on it, so the selected area is snapped on the right size of the video player and a pop-up window appears asking the user to enter a brief description about the object (as presented in Fig. 21). This description can be used as tag during the video labelling process. The instance selection process can be repeated as many times as the user wants in order to define additional instances of the object of interest that are necessary for its re-detection. Finally, the re-detection of the manually selected object using the entire set of the user-defined instances of it, is initiated when the user presses the "OK" button in the tag-related pop-up window (see Fig. 21).

The analysis is performed in a shot-by-shot manner, starting from the shot where the last instance of the object was selected from. When the algorithm finishes with the processing of the final shot of the video, it moves to the first shot and continues the processing until reaching the shot right before the first analysed shot of the video. This shot-by-shot processing can be seen via the color change in parts of this timeline bar, since during the analysis the parts of this bar that correspond to video shots with detected instances of the object are highlighted with dark blue color, while the parts with no detected instances of the object within them are colored by grey color (see Fig. 22). By selecting any of these dark blue regions of the timeline bar the user moves to the corresponding shot of the video. After pressing the play button of the video player the re-detected instances of the object in the frames of this shot are shown highlighted by a blue bounding box, while a tag including the user-defined description for this object appears in the upper right corner of the player whenever a re-detected instance of the object appears (see Fig. 22).

For the user study we extended the tool by adding implementations of different approaches for object re-detection or tracking, allowing the participants to make a direct comparison between the performance of the proposed approach against other techniques from the relevant literature. In particular, besides the algorithm presented in this study (denoted as method 1 in the sequel) and the method introduced in [AMK13] (denoted as method 2 in the sequel), the tool was extended by integrating a Sparse Flow tracker which is a pyramidal KLT tracker (denoted as method 3 in the sequel), the Circulant tracker from [HCMB12] (denoted as method 4 in the sequel) and a variant of the recently proposed TLD algorithm of [KMM12] (denoted as method 5 in the sequel). Moreover, based on the questionnaire developed by Chin et. al. [CDN88] we prepared a questionnaire for user interface satisfaction (QUIS) in order to record and evaluate the participants' opinion regarding the performance of the different tested object re-detection techniques, as well as their viewpoint on a variety of aspects that are strictly related to the usability of the tool. By usability we refer at (a) the ease-of-use and the interactivity of the developed interface, (b) the accuracy of the object re-detection algorithm and (c) the overall performance of the tool. We adjusted the original questionnaire by removing parts of it that were not applicable to our tool and by keeping only the questions that were relevant in order to meet the needs of our tool's evaluation purposes. Additionally, we modified the 10 scales used in QUIS to 5 in order to make it easier for our user study participants to answer the questions.

The participants in this user study were 10 research assistants (8 male, 2 female) between 24 and 33 years old, from the Information Technologies Institute of the Centre for Research and Technology, Hellas. Five of them reported no previous knowledge or experience with object re-detection or tracking algorithms, two of them mentioned small experience, while the rest were familiar with this research field.

During the first part of the user study, each participant was given time for testing the tool in order to get familiarized with it, and then was requested to perform three different pairs of runs. In each run the participant had to select a video from the drop-down list of the tool and re-detect an object from this video. For re-detecting the object throughout the video two different approaches had to be used in each run; the first was the algorithm proposed in this study, i.e., method 1 of the tool, while the second was one of methods 3 to 5 of the tool. By performing this blind (the algorithms appeared in the interface with the names "Algorithm 1", "Algorithm 2" etc. and not with their actual names in order to make sure that the participants will not be influenced from previous relevant experience or knowledge) pairwise testing approach, the participants were able to make a direct performance comparison between the considered pair of techniques, since these techniques were used for addressing the exact same task. Due to the fact that some of these techniques fail to re-detect the object after analysing a shot that does not contain any instance of it or includes an occurrence



About the tool

This is a tool for quick instance-based labeling of videos based on object re-detection.

Instructions of use:
1) Select a video from the drop-down list below.

2a) Select a predefined object from the list of objects for this video, as they are illustrated in the area below the list of videos. The selected object will be snapped on the right side of the video.

2b) Define a new object for this video by clicking the "Create new object" button on the right side. Then draw a bounding box around an instance of this object in a video frame (pausing the video is not necessary for this) using the mouse, and then right click. Again, the selected object will be snapped on the right side of the video.

3) After the selection of the object click on the "(Re)detect Object" button.

4) Before the analysis starts, and for labeling purposes, provide a small description for the selected object using the pop-up window with a text box.

5) Choose one of the available algorithms and click the "OK" button in order to start the analysis. During the analysis the object becomes disabled. When the analysis is finished, the object becomes enabled again and a new re-detection can be performed.

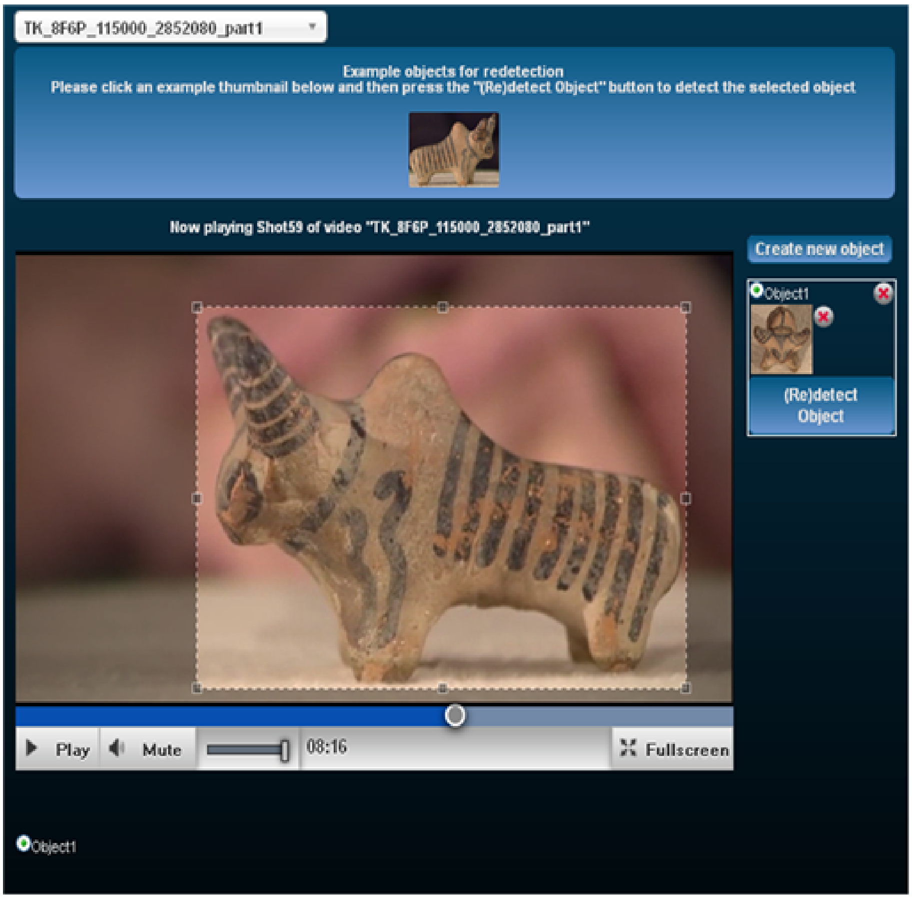
Output:
The selected object will be automatically re-detected throughout the video and the results of the analysis will be instantly demonstrated on a timeline bar placed under the video. The video shots where the object was detected are highlighted with blue color. By just clicking on these blue regions of the timeline, the corresponding shot of the video is loaded and the automatically re-detected instances of the selected object are highlighted with a blue-colored bounding box.

Example objects for redetection
w and then press the "(Re)detect Object" button to detect the selected object

"TK_8F6P_115000_2852080_part2"

TK_8F6P_115000_2852080_part2
TK_8F6P_115000_2852080_part2
TK_8F6P_115000_2852080_part3
TK_8KC5_115000_2834880_part1
TK_8KC5_115000_2834880_part2
TK_8KC5_115000_2834880_part3
TK_8Q5Q_115000_2879960_part2
TK_8Q5Q_115000_2879960_part3
TK_8Q6F_115000_2803160_part1
TK_8Q6F_115000_2803160_part2
TK_8QMF_115000_2813080_part2
Create new object
Object1
(Re)detect Object
DE-52-69
Play
Mute
01:09
Fullscreen
Object1

Figure 19 Screenshot of our web-based tool illustrating the drop-down list with the available videos, the video player and the region with the example objects placed on the upper side of the video player.



TK_8F6P_115000_2852080_part1

Example objects for redetection
Please click an example thumbnail below and then press the "(Re)detect Object" button to detect the selected object

Now playing Shot59 of video "TK_8F6P_115000_2852080_part1"

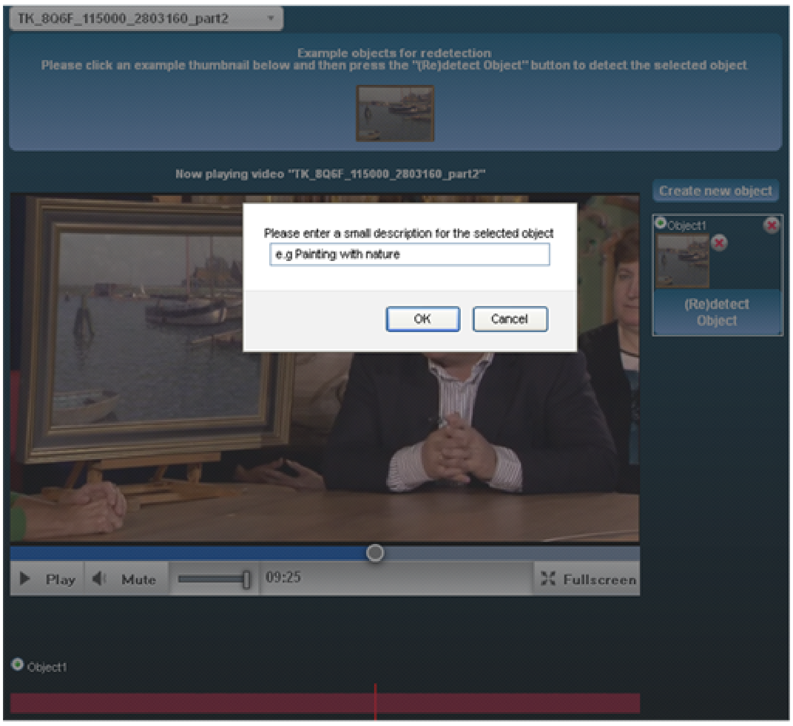
Figure 20 Manual selection of an instance of the object by drawing a bounding box around it.

of the object shown under different viewing conditions, we enabled the participants to re-apply the analysis for the object as many times as needed in order to re-detect all the different instances of it throughout the video. In this case, the result of every new "run" of the selected algorithm was added to the result of the previous run(s), while the shots of the video where the object was previously found were not examined in the following runs. Finally, the number of runs that each user performed for re-detecting each particular object using a specific approach was recorded and used in the evaluation.

The second phase of the user study included the answering of the designed QUIS questionnaire where each participant had to fill-in a set of questions about his/her experience using the tool and the integrated algorithms.

As illustrated in Fig. 23, the new developed framework got the highest scores both for speed and detection accuracy. Specifically regarding time performance it was scored (on average) with 4.7 in the range 1 to 5. At a substantial distance from this performance, the Sparse Flow and the Circulant trackers followed, rated by 2.8, while the most time consuming one was the TLD tracker with a average score of 2.3. Concerning detection accuracy the developed method was judged as the most efficient one getting an average score of 4.4 in the range 1 to 5. The Sparse Flow tracker was rated as the second best with a mean score of 2.9, the TLD tracker was rated by 2.4, while the Circulant tracker got the lowest score of 1.6.

The low scores that were assigned by the participants to the Sparse Flow, the Circulant and TLD tracker are explained by the fact that, as mentioned before, after detecting one of the object's instances these methods were very sensitive in the detection of other instances that appear in non-consecutive parts of the video or under different viewing positions, either failing to re-detect them or leading to false alarms. This means that the participants had to re-run these algorithms as many times as the number of the different instances of the object in the video in order to re-detect all of them. Specifically, the average number of iterations for the re-detection of the entire set of instances of the selected objects in the user study was 6 for the Sparse Flow tracker, 4 for the Circulant tracker and 4 for the TLD tracker, resulting nevertheless to a large number of false alarms and a significant number of misdetections which caused the dissatisfaction of the users regarding the effectiveness of these methods.



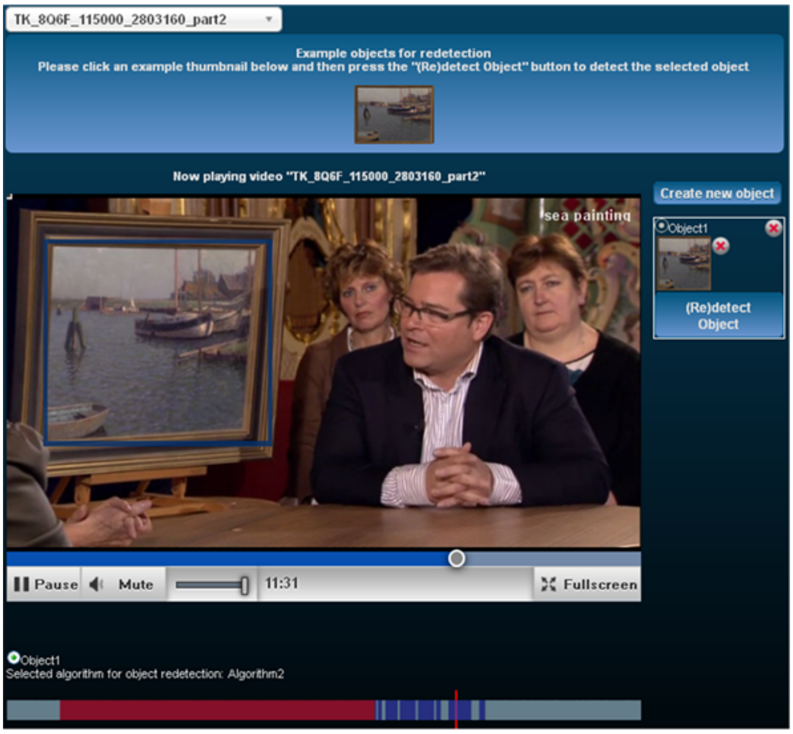
Example objects for redetection
Please click an example thumbnail below and then press the "(Re)detect Object" button to detect the selected object

Now playing video "TK_8Q6F_115000_2803160_part2"

Please enter a small description for the selected object

e.g Painting with nature

Figure 21 Pop-up window that appears before the re-detection starts, prompting the user to enter a brief description about the selected object.



Example objects for redetection
Please click an example thumbnail below and then press the "(Re)detect Object" button to detect the selected object

Now playing video "TK_8Q6F_115000_2803160_part2"

Selected algorithm for object redetection: Algorithm2

Figure 22 The color of the timeline bar of the video player indicates the current status of the analysis: blue regions indicate shots with detected instances of the object, gray regions correspond to shots that do not contain the object while shots that have not been processed yet are highlighted with red color. The re-detected instances of the given object are highlighted in the video frames by a blue bounding box around them.

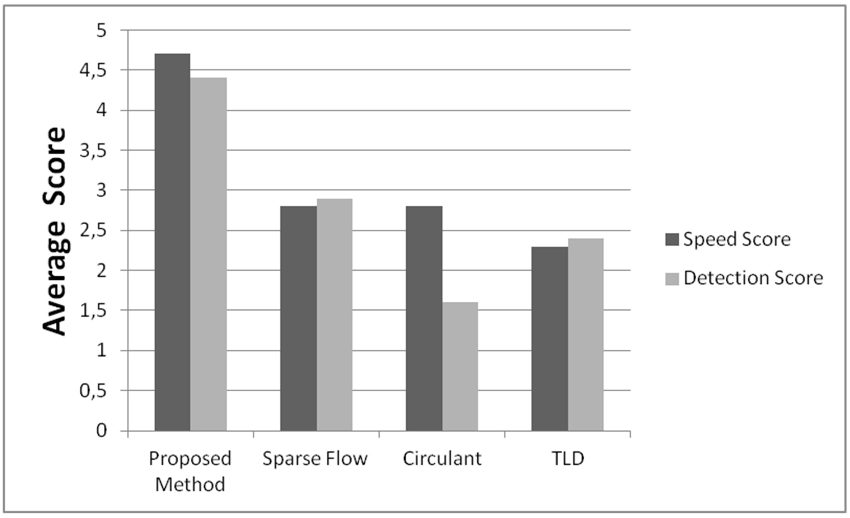


Figure 23 The average scores that got each tested method by the participants in the user study, regarding the speed and the accuracy of the analysis.

These results indicate that the algorithm proposed in this study clearly outperforms the remaining compared approaches from the relevant literature. The participants' scores for this algorithm show its supremacy against the other tested approaches both in time performance and detection accuracy. Aiming to project these results from the restricted sample of our user study to a more general framework we defined confidence intervals around the computed average values. The alpha value was set to 0.05, so the confidence interval was 95%. The standard deviations for the obtained scores regarding the speed and the detection accuracy of the algorithm was 0.48 and 0.52 respectively. So, the 95% confidence interval for the time efficiency of the algorithm is 4.7 ± 0.29 and the corresponding calculated range for its detection accuracy is 4.4 ± 0.32. These numbers clearly show that both factors that form the overall performance of the proposed technique were judged as absolutely satisfying by the users.

5.4 Conclusion

Starting from the method of [AMK13] and targeting to implement a faster technique for instance-based labeling of videos we conducted a number of in-house evaluation activities where we were able to perform a number of modifications to the object re-detection pipeline, assessing each time their impact on the algorithm's performance, in terms of detection accuracy and time efficiency. The outcomes from each step of this experimental procedure where fused, resulting in a new approach for object re-detection. Based on evaluations using an extended dataset of objects and videos, and after conducting a user study for comparison with other state-of-the-art techniques, we show that the new developed method ensures the same high levels of detection accuracy as the algorithm of [AMK13] while requiring only a very small fraction of the video's duration for analysis, being able to be used as a tool for faster-than-real-time object-specific spatiotemporal labeling of videos.

6 Evaluation of the Editor Tool

The evaluation of the Editor Tool (ET) was part of the so-called editor trials being part of LinkedTV's overall evaluation efforts, which mainly included:

- WP3 and WP6 for the evaluation of the end-user applications.
- WP1 and WP2 for the evaluation of LinkedTV technologies for automatic content analysis.

The editor trials can be positioned within the overall LinkedTV evaluation as the media professional's perspective on LinkedTV technology. Before reading on, it is advised to read up on the functionalities of the ET in D1.5. On some occasions certain functionalities will be briefly explained, but in general the reader is expected to have knowledge on the functionalities of the ET.

6.1 Overview and goals

The editor trials were held with media professionals from The Netherlands Institute for Sound and Vision (NISV) and Rundfunk Berlin Brandenburg (RBB). The main goal is to evaluate the usability of the ET and its supported functionalities from the viewpoint of media professionals; see section 6.2 on how the participants were involved. In section 6.3 the definition of methodologies is described, which were all made in close collaboration with the LinkedTV partners involved in organizing the evaluation of the end-user applications [13], namely LinkedCulture [14] and LinkedNews [15]. Section 6.4 reports on the analysis and results. Each of these subsections discusses a single feature of the ET and since each feature is dependant on the output of the automatically extracted information from WPs 1 and 2 and how this output is presented by the user interface (UI) of the ET, each subsection aims to address the following (if applicable):

1. Usability of the automatically produced data as it was presented in the ET.
2. Usability of the automatically produced data in general.
3. Relevance of the offered functionality for the participants' work.

In order to aid WP2 with their evaluation of the usability of their enrichment services, a logging system was implemented. See section 6.5 for a full description on this work.

6.2 Finding participants

Both RBB and NISV put effort in finding participants with preferably a background in (at least) one of the following fields:

1. Video editing: e.g., usage of video editing software for TV productions.
2. Video publishing: e.g., annotation of video for publishing to an online platform.
3. Video archiving: e.g., addition of metadata to video (segments) for archival purposes.
4. Media research: e.g., participation in projects evolving around audiovisual content.

As shown in Table 36, NISV found seven staff members, who are either involved in archiving, publishing or research related to audiovisual content. Furthermore two NISV members of the LinkedTV consortium used the ET extensively to curate the content for the LinkedCulture trials [16]. RBB found one participant from their news team available to participate. Besides the single participant from RBB, two RBB members from the LinkedTV consortium have used the ET extensively to curate the content for a longitudinal test, a.k.a. the LinkedNews trials, where the usability of the LinkedNews application was tested for a whole week with several users [17]. For further reference, Table 36 includes the RBB and NISV LinkedTV members who did use the ET extensively, but are not participants to the trials described in the following section, because of their inherent bias being a LinkedTV project member.

6.2.1 Contacting the AVROTROS

Lastly NISV also contacted the AVROTROS, the broadcaster of "Tussen Kunst & Kitsch", and had a meeting with the head of their new media team, NISV's main contact for LinkedTV, to show the end results of the project and discuss possible exploitation of the ET. [18]. After some discussion, showing the ET and explaining its possible uses, the AVROTROS mostly appreciated the ET's adaptability to run on different systems and was interested in possibilities of setting up the ET to work with AVROTROS specific services, enabling them to search for related content and links within their own systems. However because of upcoming future developments of the ET within other projects [19] the consensus was to wait for further improvements first and keep in touch until possibly a good opportunity arrives for both parties to pursue a collaboration.

[13] The results of this investigation and all partners involved are reported in D6.5

[14] `http://pip.ia.cwi.nl/culture`

[15] `http://pip.ia.cwi.nl/news2`

[16] Reported in D6.5

[17] Reported in D6.5

[18] The AVROTROS is interested in the ET and LinkedTV, but already mentioned that it would not be possible to have their TKK team participate in any user trials

[19] See D8.8

Table 36 Participants of the editor trials.

User code	Occupation	Category	Organization
ET1	Media manager	Archivist	NISV
ET2	Media manager	Publishing	NISV
ET3	Project assistant cultural heritage	Research	NISV
ET4	Inflow manager	Archivist	NISV
ET5	Media manager	Publishing	NISV
ET6	Media manager	Publishing	NISV
ET7	Media history specialist	Publishing	NISV
ET8	News editor	Programme editor	RBB
LTV1	Media researcher	LinkedTV member	NISV
LTV2	Media researcher	LinkedTV member	NISV
LTV3	Media researcher	LinkedTV member	RBB
LTV4	Media researcher	LinkedTV member	RBB

6.3 Methodology

Needing to get detailed feedback from participants on how useful they think the ET is for their work, it was decided to have evaluation sessions with one person at a time. In each of these sessions a participant was asked to carry out a number of tasks using the ET and tell the observant host(s) whatever he/she was doing as well as state whatever, positive or negative, was worth mentioning about the usability [20]. Closing each session the participants were asked to fill out a questionnaire. The aim was to have sessions no longer than 1.5 hours, so the participants would not have to sacrifice too much time from their own work. However, depending on each participant [21] or the host(s) needing to gain experience on how to prevent the sessions from going overtime, on some occasions a session took about 15-30 minutes longer.

The following sections describe each part of the NISV sessions in detail. The RBB session mostly followed the same methodology, but had some differences which are described in the final subsection.

6.3.1 Welcome and introduction

At the beginning of each session the participant is asked what he or she knows about LinkedTV and the LinkedCulture application. With the answers of the participant in mind the context of LinkedTV and the LinkedTV workflow, from processing content, to curation and finally to publication, is explained.

6.3.2 Short demonstration of the end-user application

To prepare for the role of editor for the LinkedCulture application, the application is briefly demonstrated. This also helps the participant in understanding what the aim of the editor's work would be.

6.3.3 Short demonstration of the Editor Tool

For the following reasons the ET is briefly demonstrated to the user:

- In a real life situation, editors would also be offered a training for using the ET.
- To avoid getting lots of feedback on uninteresting user interface issues.
- To have a bigger chance of finishing the session within the scheduled 1.5 hours.

The demonstration involves the host showing how to use the most important functionalities of the ET, including those parts the participant will also have to go through while carrying out the tasks.

[20] Using the well known think aloud protocol: `http://en.wikipedia.org/wiki/Think_aloud_protocol`

[21] Some participants are less used to new technology and need more time for explanation; some participants are more talkative

6.3.4 Consent form

The participant is informed about the fact that audio recordings will be made during the sessions and that he/she needs to sign a consent form to give the LinkedTV researchers the permission to use this for their analysis. The consent form informs the participant about the following:

- All recordings will be only used for the purpose of the evaluation within the LinkedTV project.
- No recordings will be shared with any other (third) party.
- All recordings will be kept until a maximum of two months after the end of the LinkedTV project.

All participants have signed the consent form after reading it.

6.3.5 Carrying out tasks

The participant is handed out an assignment sheet and is asked to read it and carry out all of the assignments. To speed up the process and for the convenience of the participants, the hosts helped out, if desired, by explaining each task in detail.

For NISV, the assignments, involved the following:

1. Curate chapters: the participant was asked to create two chapters involving the discussion of an art object by an expert [22] in the following way:
   (a) Create a chapter by using the shot selection functionality[23].
   (b) Verify the boundaries of the created chapter by using the player and, if needed, adjust the boundaries of the chapter by filling in the start and end times manually.
2. Annotate first chapter: the participant was asked to annotate the first (curated) chapter by adding annotations for each configured dimension, i.e., annotation layer.
3. Annotate second chapter: to give the participant a bit more experience, he/she is asked to annotate the second chapter that was created in the first assignment.
4. Delete everything: as a last assignment, the participant was asked to delete all of the curated chapters and annotations. This was also useful for making sure the next participant could start with a clean slate, using the same video.

6.3.6 Filling out the questionnaire

The participant is asked to fill out a questionnaire. To increase the likelihood of a useful response and to make it less tedious for the participant, the host tries to turn each open question into a small interview. In the analysis of the results it is taken into account that the participants' disposition could have been influenced because of the hosts' presence and involvement while filling in the survey. When the participant has finished, the hosts give their thanks and promise the participant to share the findings based on the evaluation outcome.

6.3.7 RBB session differences

The RBB session mostly followed the methodology described, with few differences:

- Welcome and Introduction: the context of LinkedTV and the LinkedNews application was described.
- A short demonstration of the tablet version of the LinkedNews application, showing the editor what the upcoming editing task would be for.
- Demonstration of the ET: the host and participant went through the functionalities step by step.
- Consent form: no consent form needed to be signed as no audio recordings were made. Also for the questionnaire the participant's name was an optional field to fill in.
- Carrying out the tasks: the host assisted the participant while carrying out the tasks. The tasks itself were customized to be relevant for the LinkedNews application.

[22] To avoid the risk of exceeding the time reserved for the session, the participant was asked to only use scenes of TKK where the art object is briefly discussed

[23] For creating chapters the ET includes a functionality for selecting a starting and an ending shot from a list of subsequent shots from the program

6.4 Evaluation results and analysis

This section describes the outcome of the analysis of each of the investigated features in different subsections, alternating between NISV and RBB results.

6.4.1 Chapter segmentation - NISV

For the TKK program, the automatic chapter segmentation is based on the detection of certain “bumpers”, i.e., a logo of sorts, that appear in the show to divide in certain scenes 2.2. During the demonstration of the ET, the hosts explained to each participant that the segmentation based on the appearance of these bumpers do not exactly correspond to the chapters that need to be created for the LinkedCulture application. Despite the fact that the automatic chapter segmentation was not optimized for defining the intended LinkedCulture chapters, 6 out of 7 participants stated they appreciated having an initial course-grained segmentation, as it offers a useful starting point for further segmentation.

Related ET features

Concerning the chapter segmentation features as presented in the ET, the participants most notably remarked on the following:

– Using the ET there is no way to conveniently determine the ending of a created chapter (7/7 participants). After defining the boundaries of a chapter using either a selection of shots (see 6.4.3) or manually filling in the time, there is no way to either play the chapter isolated (2/7) or otherwise being able to “skip to the end” (7/7).

– Given the aforementioned deficiency, a user must use the default player controls to search for the end; 4 out of 7 participants thought that the default controls are too sensitive for accurately determining a point in the video.

– Based on the feedback from users, most participants (5/7) thought that the chapter segmentation functionality of the ET was fairly intuitive and not too hard to use and was mostly lacking because of the aforementioned points.

Relation to personal work or NISV

From the questionnaire we can deduce that 5 out of 7 participants would appreciate having access to LinkedTV chapter segmentation in one form or another. Two participants mentioned being especially interested in this possibility for their work as an archivist/media manager.

Conclusion

While the automatic chapter segmentation, known to be too course-grained for LinkedCulture, was deemed to be a useful starting point, the chapter segmentation functionality of the ET was lacking in the sense that it was impossible to quickly determine the ending of (automatically) created chapters. In general the chapter segmentation functionality was considered to be intuitive and easy to use. Lastly it seems that for either their work or NISV, automatic chapter segmentation is considered to be a possibly useful asset by most (5/7) participants.

6.4.2 Chapter segmentation - RBB

For RBB content the automatic chapter segmentation is based on the detection of the anchorperson who does the introduction at the start of each news item. Based on the feedback of the RBB editor, the accuracy of this chapter segmentation does not closely resemble the intended chapter (2 out of a 5 point Likert scale), but is very useful (4 out of 5). The main reason for the low score on accuracy was the fact that the thumbnails used for representing the start time of automatically detected chapters were not always accurate. The result of this is that sometimes the thumbnail would show a shot belonging to the end of the previous chapter, while the actual start time of the detected chapter would be accurate when playing the video. It still remains unclear where this bug originated. However if this problem would be solved the RBB editor noted that fairly accurate chapter segmentation could be useful to save the editor some time while segmenting the video.

Concerning the chapter editing functionality the RBB editor suggested that the following two features would be highly appreciated:

– The possibility to choose a thumbnail image for each chapter.

– The possibility to write a short description for each chapter.

Conclusion

An unfortunate bug related to displaying the thumbnail of chapters, sometimes caused chapters to display a thumbnail representing one of the last frames from the previous chapter. Because of this, using the chapter segmentation functionality of the ET caused some frustration and was thus not well appreciated. Fortunately however, the RBB editor mentioned that in case of accurate segmentations and without this bug, LinkedTV's automatically detected chapters for RBB News could possible save editors time when segmenting the program.

6.4.3 Shot segmentation - NISV

LinkedTV's automatic shot segmentation is utilized in the ET in two different ways:

1. The editor can create chapters by selecting a starting and ending shot from an list of subsequent shots of the entire program.
2. When creating information cards, see 6.4.7, an editor can select a shot from a list of shots related to the currently selected chapter to use as a thumbnail to represent the created information card.

The first mentioned functionality is a prominent one and was specifically addressed in the questionnaire. Based on the outcome of this, all seven users considered the shot segmentation to be useful for defining the boundaries of a chapter (rating 4/5). To put this into further perspective, a similar functionality has been part of the NISV cataloguing system for many years, so most participants already were used to this functionality.

Considering the accuracy of using selected shots, i.e., the start time of the starting shot and the end time of the closing shot, not all participants were equally positive (three rated 4/5, two rated 3/5 and two more rated 2/5). Most likely this is because of the fact that on most occasions the boundaries of the chapter, after creating it using the shot selection (see 6.3.5), needed modifying. One user, who rated a 2, noted that these modifications were quite significant. Another user indicated it was difficult to select shots because of he/she was not very familiar with the program.

Although it was not very prominently addressed in the questionnaire or a major part of the tasks, all participants mentioned appreciating the functionality of selecting a shot to be used as a thumbnail for created information cards. Solely from the perspective of the ET and how it is presented by its UI, all users appreciated the shot selection functionality as it offered them a visual overview of the video, making it fairly easy to spot the different scenes in the program for creating chapters. Most participants (5/7) however had some difficulty using the shot selection functionality either because of:

– Having difficulty interpreting what would be the exact ending of the resulting chapter (4/7).

– Having difficulty using the UI (2/7).

Related to personal work or NISV

Looking at the questionnaire 4 out of 7 participants indicated that being able to use LinkedTV automatic shot detection would be useful to their work or NISV in general.

Conclusion

To summarize, all participants appreciate using automatically generated shots for quickly being able to briefly inspect the visual content of a program. Moreover, despite some imperfections of either the UI or the accuracy of the detected shots, all users appreciated the shot selection functionality of the ET to define the boundaries of a chapter. Lastly in respect to their personal work or the benefit of NISV as a whole, most users indicated that having access to shot segmentations for television programs is considered to be useful.

6.4.4 Shot segmentation - RBB

RBB's editors appreciated (5/5) the shot segmentation and the related feature of making rough cuts within seconds, but they also made clear that the possibility to edit and adjust the segment borders, for chapters as well as for shots, is a helpful and necessary feature. Even in cases where the technical accuracy was sufficient they felt the need to exclude the first words of a moderation, because they referred back to the previous chapter and that would seem strange in the non-linear mode of the LinkedNews application where users select individual chapters.

6.4.5 Named entities - NISV

Within the ET, automatically detected named entities can be used for two different purposes:

1. To select as a basis for the creation of an information card [24].
2. To use as input for a search to one of the enrichment services[25].

Due to the misalignment of subtitles with the TKK video that was used for the NISV sessions, the usability of the automatically detected named entities could not be properly addressed in the NISV sessions. In the days leading up to the NISV sessions, this problem was identified, but unfortunately could not be solved in time.

While giving each participant a tour of the ET, the situation was explained and we did give participants an opportunity to give feedback. The following shows the highlights from this feedback:

- Detected entities can help an editor in coming up with search terms when searching for enrichments or filling in the information card template (for art objects).
- It should somehow be clear which detected entities were used.

Conclusion

Besides receiving some comments on the usability of the automatically detected entities, it is hard to draw any conclusions due to the aforementioned problem of having misaligned subtitles, making the detected entities also misaligned with the chapters that are created using the ET.

6.4.6 Named entities - RBB

For the RBB setup an additional source of named entities has been configured, namely the so-called EntityExpansion service that has the ability to fetch entities from other media websites providing the editor with entities from a broader context than just the news program itself. The EntityExpansion service is called, for each curated chapter, using the subtitles of each corresponding chapter as input to further detect entities.

In general, the expansion feature was appreciated but its added value was estimated low. While the EntityExpansion service found a few further relevant entities that had apparently been filtered from the subtitles, it also doubled some of the entities that were already there. All in all RBB's editors mostly appreciated using the entities in the information card panel (see 6.4.8) and estimated that using free-text search, thus disregarding the option to search by selecting entities, to be sufficient for searching enrichments (see 6.4.9)

6.4.7 Information cards - NISV

In the ET, the information card functionality basically allows editors to annotate video with elaborate descriptions, consisting of key/value pairs, where the value can be either a text or a named entity. The creation of information cards can be done by:

1. Filling out a predefined template consisting of key/value pairs. In this template, a user can also quickly fill in terms by using a DBpedia autocompletion field.
2. Create a list of key/value pairs based on a selected named entity [26].
3. Manually define a list of key/value pairs.

[24] In the information card panel, the user can see all detected entities that fall within the boundaries of the selected chapter, and select one to enable displaying related information from DBpedia (if any). This information can then be used as the basis of the information card to be created

[25] These services, such as IRAPI or TVNewsEnricher were developed in WP2

[26] This entity can either be selected from the results of automatically detected entities or can be manually added by searching it online in DBpedia, using an autocompletion input field

To connect to the LinkedCulture application, NISV participants were asked to only use the template option. Specifically the participants were asked to fill out the art object template as completely as possible. Most users (5/7) mentioned appreciating the templating option as it offers editors a steady guideline for annotating a television program. Not being part of the evaluation tasks, but being shown in the short training before starting off, 3 out of 7 participants considered the ability to create an information card based on the properties of entities a useful feature for possibly quickly adding annotations. Moreover, one participant also mentioned that he really appreciated the possibility of being able to manually fill out information cards in case nothing can be found automatically. Two users noted that being able to fill in terms by using the DBpedia autocompletion input field is very useful, although two users also mentioned that it can be a challenge to find the right term, especially in case of more art specific terminology.

Related to personal work or NISV

Based on the outcome of the questionnaire we can see that 5 out of 7 participants would consider an option to create information cards based on predefined templates a useful functionality to have in one form or another. Also 3 out of 7 say the same about a similar feature using named entities. Due to the vagueness of the question however it is hard to specifically define in what form these users would see this functionality.

Conclusion

To summarize the usability of the different aspects of information cards and the information card screen, we can say that from a user interface perspective a lot needs to be done in order to help the user understand what the possibilities are. However on a conceptual level the possibility to create "rich annotations" in the form of information cards is appreciated by most NISV participants.

6.4.8 Information cards - RBB

For the RBB editors the information card panel was used to add additional information on the main "named entities" per news item, e.g., politicians, locations, organizations, and so on. Unlike NISV editors who mainly focussed on filling out the art object template, RBB editors mainly aimed to find suitable entities that were either automatically detected or manually retrieved by using the DBpedia lookup input field.

Based on using these functionalities, the RBB editor indicated:

- Considering the automatically generated entities to be very useful as suggestions for the editor.
- Sometimes fetching additional information for entities [27] does not yield anything.
- When using the lookup input field, it would be useful to have other sources, i.e., vocabularies, to look through besides DBpedia to improve the chance of finding a relevant entity.

Conclusion

RBB editors appreciated automatically generated entities in the form of topical suggestions. Moreover, annotating audiovisual content with information related to named entities, seems to be a desirable form of annotation for RBB. The final remark of the participant, concerning the extension of the amount of online vocabularies connected to the search field, is an interesting one, which would be applicable for other use-cases as well. In fact, this issue was already conceived by the members of the LinkedTV consortium, but could not be addressed so far.

6.4.9 Searching enrichments - NISV

In version 1 of the ET one of the tasks of the editor, would be to go through a list of automatically detected enrichments and select those enrichments that would be considered useful for an editor. Because of the low usability of these enrichments [28], the final version of the ET no longer shows automatically detected enrichments. Instead the editor can request enrichments on demand by either providing:

- A number of automatically generated entities.

[27] By clicking on an entity, the ET tries to fetch, from DBpedia, additional information from the selected entity

[28] Each enrichment was based on a single automatically generated entity, which often result in enrichments that are either too unspecific or incorrect

- A free text string.
- A number of curated enrichments and/or information cards.

In this semi-automatic enrichment workflow, LinkedTV's enrichment services can be utilized more effectively as editors are in control of assembling the queries and thus increase the chance of finding relevant results. More details on this can be found in D1.5. With the aforementioned in mind to clarify the context, the following feedback was received: most users (5/7) understood the functionalities of the enrichment search panel quite well (by rating 4/5) and also thought it was quite easy to use (also 4/5, except one person who rated 3/5). The two remaining participants gave both aspects a lower rating of 2/5, indicating to have a bit more trouble working with this functionality.

Since, unfortunately, the entities were misaligned (see 6.4.6), users generally used the free text search option, to search for enrichments. Concerning the performance of the search, in terms of speed, users generally remarked that it is important that searching must be quick, in order to reduce the amount of time it would take to fully curate a program. During the tests it became apparent that the searches within the "Background" dimension, i.e., IRAPI, generally took a fair amount of time more [29] than the other two dimensions, namely "Related artworks" and "Related chapters", which both generally return results after a couple of seconds.

With respect to inspecting the results and judging the relevance of the enrichments found, most users (5/7) noted that it was very inconvenient to have results without thumbnails and/or titles. On the positive side, 4 users mentioned appreciating the option to inspect descriptions of each result, by triggering its tooltip [30]. One user noted he did not mind inspecting each enrichment by clicking on its hyperlink and consulting the resulting web page.

One point that was generally considered confusing, and should be improved in the user interface, is the fact that after a search, not all results are shown on a single page. Instead all results are grouped by source [31] and, right after searching, only the results of one of these sources are shown. To see results from other sources, the user needs to select another source from the, wrongly labelled, "filter" drop-down box, which was considered not user friendly by most users (6/7).

Related to personal work or NISV

4 out of 7 users considered having a similar search functionality for annotating video could be useful for contextualizing video with links to e.g., other sources of cultural heritage or background information. 5 out of 7 users considered it useful to link to other content, i.e., audiovisual material, as well.

Conclusion

To summarize, most participants found their way around the enrichment search panel quite easily and generally appreciated how this could be useful to them in their work at NISV for contextualizing audiovisual content from the archive with external sources of information or content. Considering the speed of the searching functionality participants generally thought it was acceptable [32], but generally stressed the fact that it should be as fast as possible. With respect to the inspection of retrieved results, most users disliked seeing results that did not have a thumbnail or title for quick insight into its relevance. Furthermore the "filter" option was not appreciated as it initially hides results, where instead it was expected to be used for refining results displayed on the page.

6.4.10 Searching enrichments - RBB

In the same manner as the NISV setup, RBB used the enrichment search panel to search for relevant enrichments in two dimensions namely, background information (labelled "Hintergrund") or related chapters from RBB News (labelled "RBB-Beiträge"). From RBB the following feedback was obtained:

- Using the enrichment search panel was quite intuitive and easy to use.
- Searching for enrichments was too slow (rated a 1 out of 5 in the questionnaire). Like for NISV, in this case IRAPI was the enrichment service that was deemed slow.

[29] This was not measured, but in our experience this service often takes 10 seconds or more to complete a request

[30] Hovering the mouse cursor on top of a search results triggers a tooltip with extra information about the enrichment

[31] For example: when searching through the Europeana API, underlying the related artworks dimension, results are grouped by the owner of each collection

[32] Except for the IRAPI searches, which were quite slow

- Like NISV, the RBB editor disliked that certain search results did not have a thumbnail or a title.
- Especially in cases of irrelevant results, it was sometimes unclear, because of the absence of provenance information, how certain results could have been found.

Related to her own work, the RBB editor mentioned that annotating video with external links is currently not something that RBB does in any regular workflow. Finding external information however is part of the work of the website publishing staff. However she mentioned that using the ET in its current state for this would probably take more time than simply using web search.

Conclusion

Although not having any trouble understanding how to use the enrichment search panel, it seems that the RBB editor had quite a hard time finding good results, both because of the slowness of the IRAPI system as well as the difficulty of interpreting the relevancy of results that often did not have any thumbnail or proper description. Lastly the RBB editor mentioned that manually searching for related links online would be probably faster than using the ET in its current state.

6.5 Evaluation of enrichment usability

To enable the members of WP2 to gain insight into which enrichments users of the ET consider useful, a logging system was implemented that kept track of all of the enrichments a user selects after each time querying an enrichment service. Specifically the log file contained the following per enrichment query issued:

- Time when the log entry was created.
- Video ID that can be used to locate the video in the LinkedTV platform.
- User: either “sv”, i.e., Sound & Vision, or “rbb”, which is RBB.
- Title of the chapter the enrichments are meant for.
- URLs of the HTTP requests sent to query the intended enrichment service.
- All enrichments returned by the HTTP requests.
- The enrichment the user eventually selected to be saved.

The resulting analysis based on this log file are reported in D2.7.

6.6 Conclusion

With 7 participants from NISV and 1 participant from RBB the ET trials have been concluded. Looking at the received feedback, it seems that in general NISV users see quite some potential in the different possibilities the ET offers for their work or for NISV:

- Most participants appreciated the automatic “bumper” detection to be useful as a starting point for further segmentation.
- All participants appreciated automatic shot segmentation as a means to have a quick overview of a program.
- All participants appreciated the functionality in the ET that enables a user to create a chapter by selecting a starting and ending shot from a list of subsequent shots.
- Most participants considered the information card template useful as a guideline for creating program specific annotations.
- Most participants indicated that the enrichment search functionalities could be useful for their work, for contextualizing programs from the archive with external links and/or content.

Moreover the feedback from the different staff members from NISV showed that the ET potentially could be positioned in several different departments within NISV, namely:

- Editing team that works on the content for the so-called channels website of NISV[33].
- Editing team that works on video dossiers and publishes these on the NISV website[34].
- A possible extension to NISV's educational platform where e.g., teachers could prepare contextualized/enriched video for students.

From RBB's perspective the following aspects of the current version of the ET were considered useful:

- Automatically generated entities serve as useful suggestions for editors while annotating video.
- Automatically generated chapters give a first impression on the probable segments of RBB News.
- The functionality in the ET that enables a user to create a chapter by selecting a starting and ending shot from a list of subsequent shots.

In general RBB always supported the concept of the ET where editors remain in control of what enrichments are appropriate for broadcasting alongside the news. Moreover, being interested in the capabilities of the ET, RBB, NISV and Noterik participated together at a workshop for the Europeana Space[35] project and built a HbbTV prototype, using the multi-screen toolkit[36] together with the ET. Comparing NISV and RBB, it seems that in general the NISV staff appreciated the ET somewhat more. It could be argued that archivists take more time for curation and are aware of the value of adding metadata to the content. An important circumstance that should be taken into account for explaining this difference is, that it was quite stressful for RBB editors to use the ET to curate content for the longitudinal trials, simulating a "real life" workflow where the curated content had to be finished by the end of each day. For NISV this stress factor was not there as the curated content for the LinkedCulture trials has been prepared well in advance. Besides this circumstantial difference, it is likely that the RBB use case of live news broadcasting is not the most suitable for using the ET. Instead, letting broadcasters enrich their archived material for rebroadcast may be a better approach.

Finally, the feedback from both RBB and NISV was also very useful for discovering several notable issues:

- There is no convenient way to check the end point of a created chapter.
- There is an issue (with RBB only) with the chapters showing a thumbnails that actually belong to the previous chapter.
- Most participants from RBB and NISV thought the information card panel was quite complicated on first sight.
- Using the DBpedia search field, participants from both NISV and RBB occasionally have had trouble finding the entities they were looking for.
- There is no way to watch the video as the same time as e.g., creating chapters or filling in information cards or searching for enrichments.
- Most participants did not appreciate there being found enrichments without a thumbnail or proper title.
- Most participants were confused by the "filter" option in the enrichment search panel.

Fortunately most of the issues from this list were already identified and in most cases possible solutions have already been thought of. Since the ET will be further used by NISV in future projects the outcomes of this evaluation will most likely be addressed in the near future. All improvements made to the ET will be made available on the public GitHub [37] repository as well.

[33] http://in.beeldengeluid.nl
[34] http://www.beeldengeluid.nl
[35] http://www.europeana-space.eu/
[36] http://www.noterik.nl/products/multiscreentoolkit
[37] https://github.com/beeldengeluid/linkedtv-editortool

7 Summary

Based on the findings regarding the performance of the developed LinkedTV technologies for multimedia analysis that were presented in D1.4, we continued our efforts for further improvement of these methods aiming to fulfil as much as possible the analysis requirements of the LinkedTV scenarios. The results of the conducted evaluation activities for assessing the efficiency of new versions or extensions of the implemented LinkedTV analysis techniques were reported in this deliverable.

Specifically, section 2.2 presented the outcomes of in-house experiments for evaluating an extension of the developed chapter segmentation algorithm for videos of the documentary scenario that performs a more fine-grained (and thus much closer to the analysis needs) fragmentation of these videos into chapters. Moreover, the findings regarding the performance of an adaptation of the scene segmentation approach described in D1.1, after participating at an international benchmarking activity were also presented in this section. The following sections 3 and 4 concentrated on the evaluation of video annotation using the developed methods for concept and event detection. Both in-house experiments and participations to benchmarking activities were also described there, indicating the extensive efforts for assessing the performance of the implemented technologies. Then, section 5 described our strategy for evaluating the performance of our developed method for object re-detection and the performed experiments for designing and developing a new one that could be used for fast object-based spatiotemporal annotation of videos.

The other major outcome (besides the set of methods for multimedia analysis) of the work done within the WP1 of the LinkedTV project is the developed Editor Tool for supporting the annotation and enrichment of video content. For evaluating the usability and the provided functionality of the tool, and for assessing the accuracy and the usefulness of the automatic analysis results from a subset of techniques (mainly for video segmentation and enrichment), a user study was performed by a group of professionals from the area of video editing and the findings of this study were also reported in section 6 of the deliverable.

As an extension of the work reported in D1.4, with this deliverable we completed the documentation and reporting of the conducted evaluations that aimed to assess the efficiency of the delivered methods and tools that were delivered by the WP1 of the LinkedTV project. The findings of these assessments indicate that a number of different multimedia analysis technologies have been developed during the project fulfilling effectively the analysis needs of the LinkedTV scenarios. Moreover the created Editor Tool that builds on the outcomes of the automatic analysis for performing video annotation and enrichment has proven to be a good starting point for building a professional tool that could be used for supporting many tasks of video production, editing and archiving organizations.

[AEO$^+$13] Robin Aly, Maria Eskevich, Roeland Ordelman, et al. Adapting binary information retrieval evaluation metrics for segment-based retrieval tasks. Technical report, ArXiv e-prints, 2013.

[AM14] E. Apostolidis and V. Mezaris. Fast shot segmentation combining global and local visual descriptors. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6583–6587, May 2014.

[AMK13] E. Apostolidis, V. Mezaris, and I Kompatsiaris. Fast object re-detection and localization in video for spatio-temporal fragment creation. In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6, July 2013.

[AMS$^+$14] Evlampios Apostolidis, Vasileios Mezaris, Mathilde Sahuguet, Benoit Huet, Barbora Červenková, Daniel Stein, Stefan Eickeler, José Luis Redondo Garcia, Raphaël Troncy, and Lukás Pikora. Automatic fine-grained hyperlinking of videos within a closed collection using scene segmentation. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 1033–1036, New York, NY, USA, 2014. ACM.

[AOV12] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517, June 2012.

[BdVBF05] Henk M. Blanken, Arjen P. de Vries, Henk Ernst Blok, and Ling Feng. *Multimedia Retrieval*. Springer Berlin Heidelberg, NY, 2005.

[Bea11] L. Bao et al. Informedia@TRECVID 2011. In *TRECVID 2011 Workshop*, Gaithersburg, MD, USA, 2011.

[BETVG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[BLS14] Werner Bailer, Michal Lokaj, and Harald Stiegler. Context in video search: Is close-by good enough when using linking? In *ACM ICMR*, Glasgow, UK, April 1-4 2014.

[BPH$^+$14] Chidansh A. Bhatt, Nikolaos Pappas, Maryam Habibi, et al. Multimodal reranking of content-based recommendations for hyperlinking video snippets. In *ACM ICMR*, Glasgow, UK, April 1-4 2014.

[CDN88] J.P. Chin, V.A. Diehl, and K.L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 213–218, 1988.

[CJO13] Shu Chen, Gareth J. F. Jones, and Noel E. O'Connor. DCU linking runs at MediaEval 2013: Search and Hyperlinking task. In *MediaEval*, 2013.

[CLO$^+$12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1281–1298, July 2012.

[CSS06] K. Chellapilla, M. Shilman, and P. Simard. Combining multiple classifiers for faster optical character recognition. In *7th International Conference on Document Analysis Systems*, DAS'06, pages 358–367, Berlin, 2006. Springer.

[DDS$^+$09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *ACM Communications*, 24(6):381–395, June 1981.

[FHLB08] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[GCB+04] H. P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, and V. Vapnik. Parallel support vector machines: The cascade SVM. In *Neural Information Processing Systems*, pages 13–18, 2004.

[Gea14] N. Gkalelis et al. ITI-CERTH participation to TRECVID 2014. In *TRECVID 2014 Workshop*, Gaithersburg, MD, USA, 2014.

[GM07] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[GM14] N. Gkalelis and V. Mezaris. Video event detection using generalized subclass discriminant analysis and linear support vector machines. In *International Conference on Multimedia Retrieval, ICMR '14, Glasgow, United Kingdom - April 01 - 04, 2014*, page 25, 2014.

[GMK11] N. Gkalelis, V. Mezaris, and I. Kompatsiaris. High-level event detection in video exploiting discriminant concepts. In *Proc. CBMI*, pages 85–90, Madrid, Spain, June 2011.

[GMKS13a] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki. Mixture subclass discriminant analysis link to restricted Gaussian model and other generalizations. *IEEE Trans. Neural Netw. Learn. Syst.*, 24(1):8–21, January 2013.

[GMKS13b] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki. Video event recounting using mixture subclass discriminant analysis. In *IEEE International Conference on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013*, pages 4372–4376, 2013.

[GSG+13] Camille Guinaudeau, Anca-Roxana Simon, Guillaume Gravier, et al. HITS and IRISA at MediaEval 2013: Search and Hyperlinking task. In *MediaEval*, 2013.

[HCMB12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, pages 702–715, 2012.

[HMQ13] A. Hamadi, P. Mulhem, and G. Quenot. Conceptual feedback for semantic multimedia indexing. In *11th Int. Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 53–58, 2013.

[HvdSS13] A. Habibian, K. E. A. van de Sande, and C. G. M. Snoek. Recommendations for video event recognition using concept vocabularies. In *Proc. ACM ICMR*, pages 89–96, Dallas, Texas, USA, 2013.

[Iea14] N. Inoue et al. TokyoTech-Waseda at TRECVID 2014 Nakamasa. In *TRECVID 2014 Workshop*, Gaithersburg, MD, USA, 2014.

[JCL06] W. Jiang, S.-F. Chang, and A. C. Loui. Active context-based concept fusion with partial user labels. In *IEEE Int. Conf. on Image Processing*, NY, 2006. IEEE.

[Jea14] L. Jiang et al. CMU-Informedia @ TRECVID 2014. In *TRECVID 2014 Workshop*, Gaithersburg, MD, USA, 2014.

[JPD+12] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[Khv12] E. Khvedchenya. A battle of three descriptors: SURF, FREAK and BRISK, 2012. [Online; accessed December-2014].

[KMM12] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34(7):1409–1422, July 2012.

[LBH+14] Hoang An Le, Q.M Bui, Benoît Huet, B Cervenková, J Bouchner, E. Apostolidis, F Markatopoulou, A Pournaras, V Mezaris, D Stein, S Eickeler, and M Stadtschnitzer. LinkedTV at MediaEval 2014 search and hyperlinking task. In *MEDIAEVAL 2014, MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop, October 16-17, 2014, Barcelona, Spain*, Barcelona, SPAIN, 10 2014.

[LCS11] S. Leutenegger, M. Chli, and R.Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555, Nov 2011.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[MCUP04] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. British Machine Vision Computing 2002.

[Mea13] F. Markatopoulou et al. ITI-CERTH participation to TRECVID 2013. In *TRECVID 2013 Workshop*, Gaithersburg, MD, USA, 2013.

[MHX$^+$12] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev. Semantic model vectors for complex video event recognition. *IEEE Trans. Multimedia*, 14(1):88–101, February 2012.

[MMK14] F. Markatopoulou, V. Mezaris, and I. Kompatsiaris. A comparative study on the use of multi-label classification techniques for concept-based video indexing and annotation. In *MultiMedia Modeling*, volume 8325 of *LNCS*, pages 1–12. Springer, 2014.

[MPP$^+$15] F. Markatopoulou, N. Pittaras, O. Papadopoulou, V. Mezaris, and I. Patras. A study on the use of a binary local descriptor and color extensions of local descriptors for video concept detection. In *2015 MultiMedia Modeling Conference*, volume 8935 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2015.

[NNM$^+$13] Tom De Nies, Wesley De Neve, Erik Mannens, et al. Ghent University-iMinds at MediaEval 2013: An unsupervised named entity-based similarity measure for search and hyperlinking. In *MediaEval*, 2013.

[Oea13] P. Over et al. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[P. 13] P. Over et al. TRECVID 2012 - an introduction to the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. TRECVID 2012 Workshop*, Gaithersburg, MD, USA, November 2013.

[PHS$^+$13] John Preston, Jonathon S. Hare, Sina Samangooei, et al. A unified, modular and multimodal approach to search and hyperlinking video. In *MediaEval*, 2013.

[Rea08] J. Read. A pruned problem transformation method for multi-label classification. In *2008 New Zealand Computer Science Research Student Conference (NZCSRS)*, New Zealand, 2008.

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *IEEE Int. Conf. on Computer Vision*, pages 2564–2571, 2011.

[SAO13] Kim Schouten, Robin Aly, and Roeland Ordelman. Searching and Hyperlinking using word importance segment boundaries in MediaEval 2013. In *MediaEval*, 2013.

[SBB$^+$12] S. T. Strat, A. Benoit, H. Bredin, G. Quenot, and P. Lambert. Hierarchical late fusion for concept detection in videos. In *European Conference on Computer Vision (ECCV) 2012. Workshops and Demonstrations*, volume 7585 of *Lecture Notes in Computer Science*, pages 335–344. Springer, 2012.

[SDH$^+$14] B. Safadi, N. Derbas, A. Hamadi, M. Budnik, P. Mulhem, and G. Qu. LIG at TRECVid 2014 : Semantic Indexing tion of the semantic indexing. In *TRECVID 2014 Workshop*, Gaithersburg, MD, USA, 2014.

[SMK$^+$11] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163 –1177, August 2011.

[SMK14] P. Sidiropoulos, V. Mezaris, and I Kompatsiaris. Video tomographs and a base detector selection strategy for improving large-scale video concept detection. *IEEE Trans. on Circuits and Systems for Video Technology*, 24(7):1251–1264, 2014.

[SNN03] J.R. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *2003 Int. Conf. on Multimedia and Expo. (ICME)*, pages 445–448, NY, 2003. IEEE.

[SQ10] B. Safadi and G. Quénot. Evaluations of multi-learner approaches for concept indexing in video documents. In *RIAO'10*, pages 88–91, 2010.

[SQ11] B. Safadi and G. Quénot. Re-ranking by local re-scoring for video indexing and retrieval. In *20th ACM Int. Conf. on Information and Knowledge Management*, pages 2081–2084, NY, 2011. ACM.

[SSF+14] C. G. M. Snoek, K. E. A. Van De Sande, D. Fontijne, S. Cappallo, J. Van Gemert, and A. Habibian. MediaMill at TRECVID 2014 : Searching Concepts , Objects , Instances and Events in Video. In *TRECVID 2014 Workshop*, Gaithersburg, MD, USA, 2014.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv technical report*, 2014.

[Tea09] G. Tsoumakas et al. Correlation-Based Pruning of Stacked Binary Relevance Models for Multi-Label learning. In *ECML/PKDD 2009 Workshop on Learning from Multi-Label Data (MLD'09)*, pages 101–116, Berlin, 2009. Springer-Verlag.

[TKV10] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–686. Springer, Berlin, 2010.

[TSxVV11] G. Tsoumakas, E. Spyromitros-xioufis, J. Vilcek, and I. Vlahavas. MULAN : A Java Library for Multi-Label Learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

[VdSGS10] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[VTAiN13] Carles Ventura, Marcel Tella-Amo, and Xavier Giró i Nieto. UPC at MediaEval 2013 Hyperlinking task. In *MediaEval*, 2013.

[WC12] M.-F. Weng and Y.-Y. Chuang. Cross-Domain Multicue Fusion for Concept-Based Video Indexing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(10):1927–1941, 2012.

[WF05] I. Witten and E. Frank. *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, second edition, 2005.

[WS13] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[YKA08] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *31st ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*, pages 603–610, USA, 2008. ACM.

[YYL98] M. Yeung, B.-L. Yeo, and B. Liu. Segmentation of video by clustering and graph analysis. *Computer Vision Image Understanding*, 71(1):94–109, July 1998.

[ZZ07] M.-L. Zhang and Z.-H. Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.