ITI-CERTH participation to TRECVID 2013

Fotini Markatopoulou, Anastasia Moumtzidou, Christos Tzelepis, Konstantinos Avgerinakis, Nikolaos Gkalelis, Stefanos Vrochidis, Vasileios Mezaris, Ioannis Kompatsiaris

Information Technologies Institute/Centre for Research and Technology Hellas,
6th Km. Charilaou - Thermi Road, P.O. Box 60361, 57001 Thermi, Greece
{markatopoulou, moumtzid, tzelepis, koafgeri, gkalelis, stefanos, bmezaris, ikom}@iti.gr

Abstract

This paper provides an overview of the tasks submitted to TRECVID 2013 by ITI-CERTH. ITI-CERTH participated in the Semantic Indexing (SIN), the Event Detection in Internet Multimedia (MED), the Multimedia Event Recounting (MER) and the Instance Search (INS) tasks. In the SIN task, techniques are developed, which combine new video representations (video tomographs) with existing well-performing descriptors such as SIFT, Bag-of-Words for shot representation, ensemble construction techniques and a multi-label learning method for score refinement. In the MED task, an efficient method that uses only static visual features as well as limited audio information is evaluated. In the MER sub-task of MED a discriminant analysis-based feature selection method is combined with a model vector approach for selecting the key semantic entities depicted in the video that best describe the detected event. Finally, the INS task is performed by employing VERGE, which is an interactive retrieval application combining retrieval functionalities in various modalities, used previously for supporting the Known Item Search (KIS) task.

1 Introduction

This paper describes the recent work of ITI-CERTH[1] in the domain of video analysis and retrieval. Being one of the major evaluation activities in the area, TRECVID [1] has always been a target initiative for ITI-CERTH. In the past, ITI-CERTH participated in the search task under the research network COST292 (TRECVID 2006, 2007 and 2008) and in the semantic indexing (SIN) task (which is the similar to the old high-level feature extraction task) under the MESH[2] (2008) and K-SPACE (2007 and 2008) EU projects. In 2009 ITI-CERTH participated as a stand alone organization in the HLFE and Search tasks [2], in 2010 and 2011 in the KIS, INS, SIN and MED tasks [3], [4] and in 2012 in the KIS, SIN, MED and MER tasks [5] of TRECVID respectively. Based on the acquired experience from previous submissions to TRECVID, our aim is to evaluate our algorithms and systems in order to improve and enhance them. This year, ITI-CERTH participated in four tasks: semantic indexing, event detection in internet multimedia, multimedia event recounting and instance search. In the following sections we will present in detail the applied algorithms and the evaluation for the runs we performed in the aforementioned tasks.

[1] Information Technologies Institute - Centre for Research & Technology Hellas

[2] Multimedia sEmantic Syndication for enHanced news services, http://www.mesh-ip.eu/?Page=project

2 Semantic Indexing

2.1 Objective of the submission

In 2013, the ITI-CERTH participation in the SIN task [6] was based on an extension of our 2012 SIN system. The goal of this task is to use the concept detectors in order to retrieve for each concept a ranked list of 2000 test shots that are mostly related with it. The main idea of our submission is to optimally combine the output of Linear Support Vector Machine (LSVM) classifiers, based on multiple shot representations, descriptors, interest point detectors and assignment techniques, in order to achieve enhanced performance, both in terms of accuracy and computational cost. This year we continued this effort by enhancing our SIN 2012 system with more techniques as described in the following. For learning the 346 concepts we use the TRECVID annotated dataset to build concept detectors by adopting two different strategies:

- Training example manipulation [7]; we train many LSVMs, for the same concept, each time with a different subset of the training examples in order to create different hypotheses.
- Multiple feature extraction alternatives; we describe each video shot with several different features by combining different descriptors, detectors, assignment methods etc.

In addition to this, we accelerate our system using an approximation method for data scaling in SVM-based concept detection. Finally, we develop a stacking-based approach for score refinement. The objective of our submission in SIN task is to evaluate these techniques.

2.2 Concept Detection System Overview

We developed a concept detection scheme based on a two-layer stacking architecture. The first layer consists of multiple LSVM concept detectors. More specifically, similarly with the last year's system, for each shot a keyframe or a shot tomograph [8] is extracted. Tomographs are 2-dimensional slices with one dimension in time and one dimension in space. A set of 25 different feature extraction procedures are employed, following the methodology of [5], in order to generate 25 Bag-of-Words (BoW) feature vectors for each selected image of a shot (by image we mean here either a keyframe or a tomograph). A sampling strategy is utilized in order to partition the training set into five subsets. Five LSVMs, called a Bag of Models (BoMs) in the sequel, using a different subset of the same training set are trained separately for each feature extraction procedure and each concept. Therefore, in total 25 BoMs are created for each concept. A late-fusion strategy is applied in order to combine the LSVM models.

In the second layer of our stacking architecture the fused scores are further refined by two different approaches. The first approach uses a multi-label learning algorithm that incorporates concept correlations [9]. The second approach is a temporal re-ranking method that re-evaluates the detection scores based on video segments as proposed in [10]. The block diagram of a first-layer concept detector corresponding to one concept is given in Fig. 1. A detailed description of our system is presented in the following sub-sections.

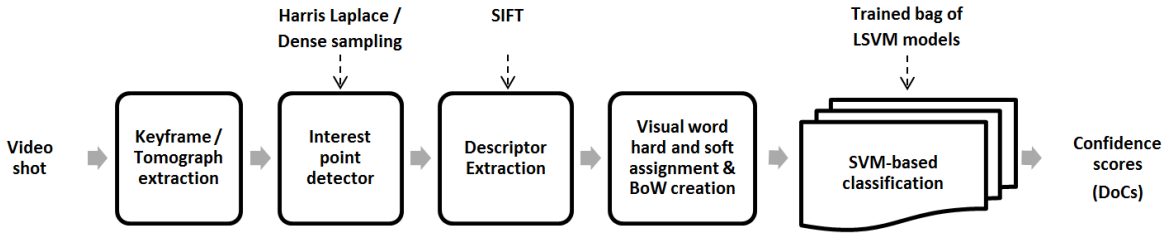

Figure 1: Block diagram of the concept detection approach utilizing 25 BoMs.

2.2.1 Training phase

During the training phase a sampling strategy is applied to partition the training set into 5 subclasses and for each subset a LSVM is trained to create a BoMs. The sampling strategy and consequently the training of BoMs are performed separately for each feature extraction procedure and each concept. The following major decisions were made in order to develop BoMs that show diversity and accuracy.

Regarding diversity we use a different subset of the training set to train each LSVM inside a bag. The negative annotated training examples are randomly divided into five non overlapping sets. For the positive annotated training examples, which are significantly fewer, we construct overlapping subsets. We follow a process similar to the *cross validated committees* method [11] in order to equally reuse the same positive samples. Particularly, the positive annotated training examples are randomly divided into 5 disjoint subsets. Then 5 overlapping training sets are constructed by dropping out different ones of these 5 subsets. Note here that depending on the number of available annotated positive samples for each concept, the proportion of the samples that have been reused changes. For instance we may use the 4/5, 3/5 etc. By applying this method we are able to use the majority of available training examples for constructing one aggregated predictor from many weak classifiers trained in different feature spaces. From our early experimentation, we concluded that exploitation of diversity during classifier design (i.e. avoiding to create almost identical models) can improve the generalization performance. Another decision is regarding the positive and negative ratio of training examples. As in our past participations, a variable proportion of positive to negative samples is used. This proportion ranges from 1:5 to 1:1. The maximum limit of 25000 positive and negative training examples for each concept is adopted due to limitations in computational resources.

With respect to accuracy we select LSVM base classifiers, which have shown promising classification accuracy in the concept detection domain. In order to exploit the correlations among concepts, in a second layer we construct model vectors by concatenating the responses of the LSVMs and train a ML-kNN model [12] using a separate validation set [9].

Finally, in terms of computational time, we use a new scaling method to accelerate the training and classification module. SVM models perform better if the input data is scaled within a specific region. However, a major issue in on-line classification of large-scale datasets with a large bag of classifiers is the high computational cost that may be required for scaling the test observations. To this end, we apply a new scaling strategy, where during the training phase the k-means algorithm is employed in order to learn 10 common ranges among concepts (instead of 346) and during testing the most suitable range for the specific concept is applied.

2.2.2 Classification phase

During the classification phase, a new unlabeled video shot is fetched by the 125 first-layer LSVMs (25 feature extraction procedures * 5 LSVMs per BoMs). Each LSVM returns one prediction score, in the range [0,1], expressing the degree of confidence (DoC) that the concept is depicted in the image. A late fusion strategy is then applied to combine these scores, and this process is repeated for all of the tested concepts.

In the second layer of our stacking architecture a post-processing step is performed in order to refine these initial DoCs. More specifically, we construct a model vector by concatenating the fused responses from the 125 LSVMs and the model vector is further refined by the ML-kNN model. The final step is to re-rank the scores per concept following the approach in [10].

The above process is finally iterated for 17 (11 applied on keyframes and 6 applied on tomographs) out of the 25 feature extraction procedures, as 8 of them have been removed due to their low performance on a separate validation set (a subset of the TRECVID 2013 development set [13]). The final results were sorted according to the resulted DoCs in descending order and the ranking list containing the top 2000 shots per concept was submitted to NIST.

2.3 Datasets and evaluation

We tested our framework on the TRECVID 2013 Semantic Indexing (SIN) dataset [13]. This set consists of an annotated development set [14] and a test set for blind evaluation. The above sets are approximately 800 and 200 hours of internet archive videos for training and testing, respectively. We

further used a part of the original development set in order to gather meta-learning information to train the ML-kNN algorithm.

2.4 Description of runs

Four SIN runs were submitted in order to evaluate the potential of the aforementioned approaches. All 4 runs were based on generating SIFT-based (SIFT and their color variants) features, and building ensembles of LSVM models. We opted to investigate the following issues: a) whether the BoMs, constructed by training set manipulation, perform better than a single LSVM model and b) if score refinement using the proposed multi-label algorithm can increase system's performance. The 4 submitted runs for the main task of the 2013 TRECVID SIN competition are briefly described in the following:

- ITI-CERTH-Run4: "Baseline"; Combination of 17 LSVM classifiers per concept.
- ITI-CERTH-Run3: "Bagging"; Combination of 85 LSVM classifiers (17 BoMs) per concept.
- ITI-CERTH-Run2: "Meta346"; Optimized combination of 85 LSVM classifiers per concept, score refinement for 60 concepts using ML-kNN algorithm trained on a meta-learning feature space constructed using detection scores for an extended set of 346 concepts.
- ITI-CERTH-Run1: "Meta60"; Optimized combination of 85 LSVM classifiers per concept, score refinement for 60 concepts using ML-kNN algorithm trained on a meta-learning feature space constructed using detection scores for the same 60 concepts.

The scores in Runs 3 and 4 were fused using the harmonic mean. On the other hand, the scores in Runs 1 and 2 were fused using the arithmetic mean because the ML-kNN algorithm during the post-processing step cannot work properly with the very small values that the harmonic mean generates. Finally, the temporal re-ranking method was included in all four runs (as the last processing step) as we observed that it improves the detection performance.

2.5 Results

Table 1: Mean Extended Inferred Average Precision (MXinfAP) for all single concepts and runs.

Run IDs	MXinfAP
Run4	0.144
Run3	0.15
Run2	0.158
Run1	0.155
Run3 using just keyframes	0.115
Run3 using just tomographs	0.06

Table 2: Number of improved concepts per run.

from/to	Run3	Run2	Run1
Run4	25	21	24
Run3	-	20	19
Run2	-	-	18

Table 1 summarizes the evaluation results of the aforementioned runs in terms of the Mean Extended Inferred Average Precision (MXinfAP). Moreover, in Table 2 we count the number of improved concepts between pairs of runs. For instance, when we go from Run4 (our Baseline) to Run3 we see that detection rates (XinfAP) are improved for 25 concepts. From the obtained results the following conclusions can be drawn:

- The "Baseline" run (ITI-CERTH-Run-4) combines 17 LSVM-based models per concept, and utilizes temporal re-ranking for score refinement. That is, one model has been trained for each feature extraction procedure without performing any training set manipulation. We observe that all the following runs perform better than this baseline run.

- The "Bagging" run (ITI-CERTH-Run-3) exploiting the use of BoMs. In this run we train 5 models using training-set manipulation for each of the 17 feature extraction procedures. Therefore 17 BoMs (or 85 LSVM models) per concept are retrieved. Again only temporal re-ranking is used for score refinement. This technique shows a performance gain improvement of 4.2% over the baseline run.

- The ITI-CERTH-Run-2 and ITI-CERTH-Run-1 exploiting correlations of 346 and 60 concepts, respectively, provides the best performance among all our runs.

Overall, taking into account our best run (0.158 MXinfAP), ITI-CERTH ranks in 12th place among 26 participants in the full run evaluation. Considering also that our system almost always performs better than the median of all SIN runs (in 36 out of 38 evaluated concepts as shown in Fig. 2), and taking into account that we made design choices that favor speed of execution over accuracy (use of LSVMs, range scaling reduction), this performance is judged satisfactory.

After submitting the above runs a post-analysis of our results was performed. In particular we investigated in what extend tomographs can improve the performance of our system. The last two lines of Table 1 correspond to ITI-CERTH-Run3 and show the results of our system when only tomograph-based or keyframe-based classifiers are fused. Although keyframe-based classifiers (Table 1:Run3 using just keyframes) present significantly better performance than tomograph-based classifiers (Table 1:Run3 using just tomographs), the combination of both (Table 1:Run3) leads to improved performance. Moreover, we investigated the effect of tomographs at the individual concept level. Fig. 2 shows the MXinfAP for each of the evaluated concepts. In the horizontal axis of this figure, the motion related concepts are presented first followed by the static one, and we report results from the fusion of tomograph-based classifiers, keyframe-based classifiers and the combination of these two. Based on this diagram we can see that our system performs very well for the motion related concepts achieving performance better than the median of all SIN runs for each concept. At the same time we observe the importance of the tomograph-based classifiers for three specific motion-related concepts (concepts 15, 80 and 120).

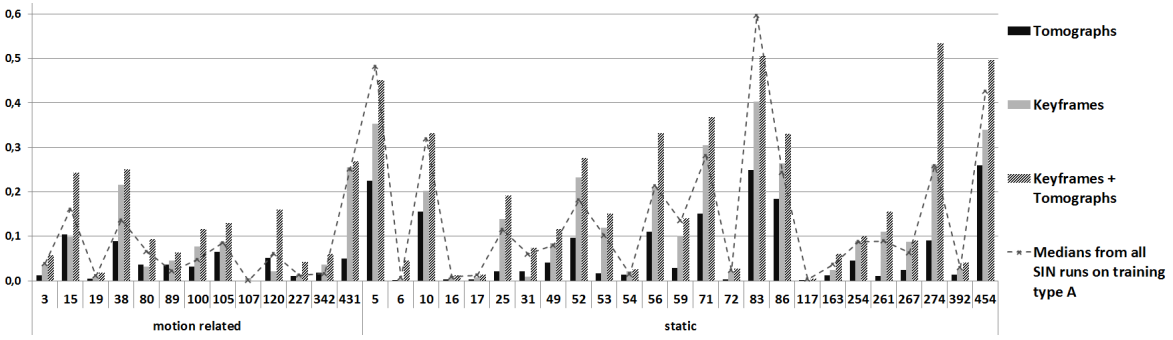

Figure 2: Mean Extended Inferred Average Precision (MXinfAP) per concept for different classifier combinations. For each concept the best results across runs is reported. Motion-related concepts are presented first followed by static ones.

Adult, Airplane, Airplane Flying, Anchorperson, Animal, Apartments, Baby, Basketball, Beach, Bicycling, Boat Ship, Boy, Bridges, Bus, Car, Car Racing, Chair, Charts, Cheering, Civilian Person, Classroom, Clearing, Computers, Dancing, Demonstration Or Protest, Door Opening, Doorway, Event, Explosion Fire, Face, Female-Human-Face-Closeup, Female Human Face, Female Person, Fields, Flags, Flowers, Forest, George Bush, Girl, Glasses, Government Leader, Greeting, Hand, Head And Shoulder, Highway, Hill, Indoor, Instrumental Musician, Kitchen, Lakes, Landscape, Male Human Face, Male Person, Man Wearing A Suit, Meeting, Military Airplane, Motorcycle, Mountain, News, News Studio, Nighttime, Oceans, Office, Old People, Overlaid Text, People Marching, Press Conference, Quadruped, Reporters, Roadway Junction, Running, Scene Text, Singing, Sitting Down, Skating, Skier, Sky, Soldiers, Speaking, Speaking To Camera, Sports, Stadium, Streets, Studio With Anchorperson, Swimming, Table, Teenagers, Telephones, Text, Throwing, Traffic, Two People, Urban Scenes, Walking, Walking Running.

Table 3: The 95 concepts of the SIN 2013 task used by the event detection systems.

3 Multimedia Event Detection and Recounting

3.1 Objective of the submission

High level event detection in video is a challenging task that has practical applications in several domains such as news analysis, video surveillance, and multimedia organization and retrieval. Applications in these domains are often time-critical and the use of systems that provide low-latency responses is highly desired. The objective of our submission is the evaluation of a number of efficient algorithms that exploit only sparsely sampled keyframes, static visual features and limited audio information for the task of event detection and event recounting.

3.2 System Overview

The target of the event detection and recounting system is to learn a decision function $f(\mathbf{X}) \rightarrow \mathcal{Y}$, $\mathcal{Y} = \{1, 2\}$ that assigns the test video $\mathbf{X}$ to the event class (labeled with the integer one) or to the "rest of the world" class (labeled with the integer two). Additionally for each positive detection to produce a MER document recounting the key semantic entities of the detected event depicted in the video. This is typically achieved using a training set $\mathcal{X} = \{\mathbf{X}_i^p | p = 1, \ldots, L_i, i = 1, 2\}$, where, $\mathbf{X}_i^p$ denotes the p-th video of the i-th class and L_i is the number of videos belonging to the i-th class.

3.2.1 Subsystem exploiting visual information

Our method exploits only static visual information extracted from selected video frames following the procedure described in Section 2. More specifically, a video signal is first decoded and one frame every 6 seconds is selected uniformly to represent the video with a sequence of keyframes. Then, dense and Harris-Laplace sampling strategy is combined with a spatial pyramid approach to extract salient image points at different pyramid levels, the SIFT, rgbSIFT and opponentSIFT descriptor is used to derive 384-dimensional (color descriptors) and 128-dimensional (SIFT descriptor) feature vectors at those points, and a bag-of-words (BoW) method is applied to construct a visual codebook. Subsequently, a video frame is described with a feature vector in $\mathbb{R}^{4000}$ using soft and hard assignment of visual words to image features at each pyramid level. Therefore, in total 12 low-level feature vectors are derived for each keyframe. The final feature vectors are used as input to a set $\mathcal{G} = \{h_\kappa() | \kappa = 1, \ldots, F\}$ of F fused semantic concept detectors $h_\kappa() \rightarrow [0, 1]$ for associating each keyframe with a model vector [15]. In particular, we used 11 fused keyframe-based concept detectors derived from our participation in the SIN task (i.e., the procedures described in Section 2, excluding the sift-harris-hard and opponentSIFT-harris-hard procedures, and including a global-image feature (color histograms)) referring to the 95 concepts depicted in Table 3.

Following the above procedure, the p-th video of the i-th class, consisting of o_p keyframes, is represented as $\mathbf{X}_i^p = [\mathbf{x}_i^{p,1,1}, \ldots, \mathbf{x}_i^{p,o_p,1}]$, where $\mathbf{x}_i^{p,q,1} \in \mathbb{R}^F$ is the model vector of the q-th keyframe of the video. Note that the κ-th element of the above model vector expresses the degree of confidence (DoC) that the κ-th semantic concept is depicted in the keyframe. Furthermore, we represent each video keyframe with two additional low-level features (BoW features) $\mathbf{x}_i^{p,q,r}$, $r = 2, 3$ corresponding to the opponentSIFT+dense+soft ($r = 2$), and rgbSIFT+dense+soft ($r = 3$) feature extraction procedures, respectively. We retrieve the feature vector $\mathbf{x}_i^{p,r}$ for representing the whole video with

respect to the r-th feature by averaging the respective feature vectors along all video keyframes $\mathbf{x}_i^{p,r} = \sum_{q=1}^{o_p} \mathbf{x}_i^{p,q,r}$.

For each feature and event one KSVM is learned, i.e., in total 3 KSVMs are trained for each event. For KSVMs the LIBSVM implementation is employed [16]. During testing, the 3 DoCs of the event detectors corresponding to the same event are fused using the Geometric Mean operator.

3.2.2 Subsystem exploiting audio information

In parallel to the visual features described in section 3.2.1, we additionally exploit low-level audio information. In particular, short-time frequency analysis of audio is performed and linear frequency cepstral coefficients (LFCC) are extracted following the method described in [17], as explained briefly in the following.

For each audio frame we extract 20 static LFCCs and their first and second order derivative coefficients (delta and double delta respectively), leading to a 60-element feature vector. The training of event detectors is based on Gaussian Mixture Models (GMM) and consists of two phases [18]. First, we train a so-called Universal Background Model GMM (UBM-GMM) using negative examples of the event. We also represent the "rest of the world" category with an equal number of videos selected from the negative videos regarding the target event. In the second step, the training set is used to train an event GMM model via a Maximum a Posteriori (MAP) process from the UBM-GMM. For all the test videos we extract the same features as the ones for training, apply normalization and sampling with energy detection.

The test and train feature vectors are then used to derive a log-likelihood ratio (LLR) score using the event GMMs [17]. The derived LLR score values are in the range $(-\infty, +\infty)$. To this end, we floor and ceil the values that are below -1 and above $+1$ respectively, and scale the resulting values to the range $[0, 1]$ to retrieve a DoC score for the test video. These DoCs are then concatenated in the visual model vectors derived in the previous section to provide an extended model vector for each video in the training and testing set. Subsequently, these model vectors are used to train one KSVM for each event. During testing the DoCs derived from the event detectors corresponding to the visual model vectors and the extended model vectors are averaged. The final DoC for each video is retrieved using the geometric mean of the above DoC with the DoCs derived from the event detectors corresponding to the two low-level features (rgbSIFT, opponentSIFT) and the visual model vectors.

3.2.3 Multimedia event recounting

Additionally to event detection, our system provides an event recounting of the detected event. To this end, we exploited the MSDA-based feature selection method described in [19, 20], and the model vector representations derived in Sections 3.2.1. Using the above technique for each event a transformation matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2] \in \mathbb{R}^{F \times H-1}$ is derived, where we set $H = 3$. The derived transformation matrix is then used for selecting the $I = 15$ most discriminant concepts concerning the target event. This is done by firstly computing the weighted video model vector $\mathbf{y}^\tau = [y_1^\tau, \ldots, y_F^\tau]^T$ using

$$\mathbf{y}^\tau = \text{argmax}(\mathbf{w}_1 \circ \mathbf{x}^\tau, \mathbf{w}_2 \circ \mathbf{x}^\tau) \tag{1}$$

where y_f^τ express the DoC concerning the f-th concept weighted with a significance value regarding the target event, $\mathbf{x}^\tau$ is the video model vector, and the operator $\circ$ is used to denote element-wise vector multiplication. The I most discriminant concepts are then selected according the following rule

$$\{c_1, \ldots, c_I\} = \text{argmax}_I(y_1^\tau, \ldots, y_F^\tau)\,. \tag{2}$$

A MER output file is generated in the required format using a Python script. Specifically, this script receives as input a text file containing the names, ids and DoCs of the detected concepts, and generates as output an XML file according to the DTD schema provided by NIST.

3.3 Dataset description

The following 6 video collections of the TRECVID MED 2013 track are used for evaluating our system for the Pre-Specified (PS) and AdHoc (AH) MED tasks and the MER task:

- PS EVENTS: This collection contains approximately 200 videos for each Pre-Specified event[1]. It is the union of the MED11 Training event collection (10 event kits referring to events E06-15) and the MED12 Pre-Specified events (10 event kits referring to events E21-E30).
- AH EVENTS: For the AdHoc event task a video collection consisting of approximately 150 videos for each of the 10 AdHoc events[2] is provided.
- OTHER-EVENTS: The MED11 Training event collection containing approximately 820 videos belonging to one of the 5 training events E01-05.
- DEV-T: The MED11 transparent data collection (DEV-T) contains 10273 videos ($\sim$ 350 hours) belonging to one of the events E01-05 or to the "rest of the world" category.
- MED11TEST: The MED11 Test collection contains 32061 videos ($\sim$ 1000 hours) belonging to one of the events E06-15 or to the "rest of the world" category.
- PROGSub: A subset of the MED13 PROGAll collection containing 32768 videos belonging to one of the events E06-15, E21-E30 or to the "rest of the world" category.

The first 5 sets described above (PS EVENTS, AH EVENTS, OTHER-EVENTS, DEVT, MED11TEST) are designated for training purposes, while the last set (PROGSub) is used for the blind evaluation of the Pre-Secified and AdHoc event detectors.

3.4 Description of runs

We submitted 4 runs in the TRECVID MED 2013 evaluation track for each of the Pre-Specified and AdHoc tasks, namely, VisualSys_100Ex, FullSys_100Ex, VisualSys_10Ex, FullSys_10Ex using the PROGSub for blind evaluation. That is, in total 8 runs were submitted. In the 100Ex and 10Ex conditions, 100 or 10 positive videos were used during training, respectively. With regard to the 10Ex condition, the VisualSys and FullSys submissions are the same. For both the Pre-Specified and AdHoc tasks we followed the same cross-validation strategy for identifying the optimal parameters for our algorithms. A brief explanation of each run is provided below.

- VisualSys_100Ex (PS, AH): In this run only static visual features were used for training the event detectors, as described in Section 3.2.1.
- FullSys_100Ex (PS, AH): In this run visual and audio features, as described in Sections 3.2.1 and 3.2.2, were used for training the event detectors. Moreover, the event detectors used 100 positive videos for training. Additionally, a MER document was provided by our system for each video that was evaluated as belonging to a target event.
- VisualSys_10Ex (PS, AH): This run is similar to the VisualSys_100Ex run described above, but only 10 positive videos for each event were used for training the event detectors.
- FullSys_10Ex (PS, AH): This run is exactly the same as VisualSys_10Ex, as no audio information was exploited in the training of the event detectors in the conditions of the 10Ex subtask.

[1]The Pre-Specified events are: 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, E21: Attempting a bike trick, E22: Cleaning an appliance, E23: Dog show, E24: Giving directions to a location, E25: Marriage proposal, E26: Renovating a home, E27: Rock climbing, E28: Town hall meeting, E29: Winning a race without a vehicle, E30: Working on a metal crafts project

[2]E31: Beekeeping, E32: Wedding shower, E33: Non-motorized vehicle repair, E34: Fixing musical instrument, E35: Horse riding competition, E36: Felling a tree, E37: Parking a vehicle, E38: Playing fetch, E39: Tailgating, E40: Tuning musical instrument

Run		MAP	
		VisualSys	FullSys
PS	100Ex	10.2	10.5
PS	10Ex	3.0	3.0
AH	100Ex	8.2	7.8
AH	10Ex	3.0	3.0

(a) MED.

Accuracy	ObsTextScore	PRRT
43.87	1.06	129.96

(b) MER.

Table 4: Evaluation results for MED (a) and MER (b) tasks.

3.5 Results

The evaluation results of our 8 runs in the TRECVID MED 2013 Pre-Specified and AdHoc event tasks are shown in Table 4a, in terms of MAP along the 20 and 10 target events for the Pre-Specified and AdHoc tasks, respectively. Moreover, our results for our MER evaluation along all events are depicted in Table 4b. For this task the following metrics were defined in the MER evaluation plan: a) Accuracy: percentage of MER outputs which the assessment by the judges agree with the MED ground truth, b) ObsTextScore: the mean of the scores provided by the judges on the question "how well does the text of the observation describe the video snippet", c) PRRT: percentage of clip time the judges took to evaluate it.

From the analysis of the detection results we can conclude the following:

- The overall performance of our PS and AH runs compared to the rest of the submissions is rather average. However, taking into account that in comparison to the other submissions we use only limited static visual features (opponentSIFT, rgbSIFT descriptors in sparsely sampled keyframes) and LFCCs (short-term audio features), the performance of our detection algorithms can be considered good.

- In particular, for the VisualSys-PROGSub-PS-100Ex task among the 14 runs that exploit only visual information our respective run ranks 9th (Fig. 3). The systems ranking higher than ours utilize among others motion features (e.g., MoSIFT, STIP, etc.), in contrast to our submission that exploits only static visual features. It is well known that motion features can offer higher recognition rates, however, they are much more computationally expensive. Among the submissions that use only static visual information, our system provides the best performance.

- The utilization of short-term audio information did not improve considerably our system's performance. This contrasts our previous year submission where the combination of short- and long-term (periodogram) features provided a small performance gain [5]. Based on both submissions, we can conclude that the combination of short- and long-term audio information may contain significant information for the task of event detection.

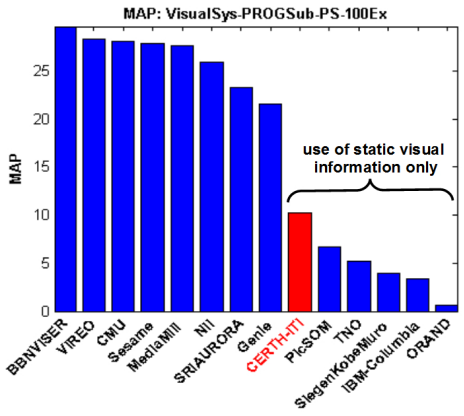

Figure 3: MAP for all systems that use visual information.

- Concerning the MER task, from the attained results we conclude that the performance of our run is average. This was rather expected because only a small set of semantic concepts are used, which additionally exploit only limited static visual information.
- The event agent execution time of our algorithms for processing the whole PROGSub dataset for one event is in the order of a few minutes. Therefore, our event detection system offers real-time performance.

4 Instance Search

4.1 Objective of the submission

ITI-CERTH's participation in the TRECVID 2013 instance search (INS) task aimed at studying and drawing conclusions regarding the effectiveness of different retrieval modules, which are integrated in VERGE[4] interactive video search engine, in the retrieval procedure. According to the TRECVID guidelines, the INS task represents the situation, in which the user is searching for video segments of a specific person, object, or place contained in a video collection. It should be noted, that the searcher is provided with visual examples of the specific query object in order to commence with the searching. In this context, the representation mode that was used was the standard shot-based video retrieval. Three runs are submitted, each combining several indexing and retrieval modules in a different way, for evaluation purposes. Finally, it should be noted that the videos used in the INS task are provided by BBC and they are part of the EastEnders tv series (Programme material BBC).

4.2 System Overview

The system employed for the Known-Item search task was VERGE, which is an interactive retrieval application that combines basic retrieval functionalities in various modalities, accessible through a friendly Graphical User Interface (GUI), as shown in Fig. 4. The following basic modules are integrated in the developed search application:

- Scene Segmentation Module;
- Visual Similarity Search Module;

The search system is built on open source web technologies and more specifically Apache server, PHP, JavaScript and mySQL database.

Besides the basic retrieval modules, VERGE integrates a set of complementary functionalities, which aim at improving retrieved results. To begin with, the system supports basic temporal queries such as the shot-segmented view of each video. The selected shots by a user could be stored in a storage structure that mimics the functionality of the shopping cart found in electronic commerce sites. Finally, a history bin is supported, in which all the user actions are recorded.

A detailed description of the aforementioned modules is presented in the following sections.

4.2.1 Scene Segmentation Module

The employed technique for scene segmentation is based on the algorithm introduced in [21]. In general, scenes are temporal segments usually defined as Logical Story Units (LSU), which are higher-level temporal segments, each covering either a single event or several related events taking place in parallel. This method groups the shots of the video detected by performing shot segmentation[5] into sets that correspond to individual scenes of the video, based on the visual similarity and the temporal consistency among them. Specifically, one representative key-frame is extracted from each shot of the video and the visual similarity between pairs of key-frames is estimated via HSV histogram comparison. The grouping of shots into scenes is then performed, by utilizing the two proposed extensions of the well known Scene Transition Graph (STG) method [22], which clusters shots into scenes by examining whether a link, between two shots, exists.

[4]VERGE: http://mklab.iti.gr/verge

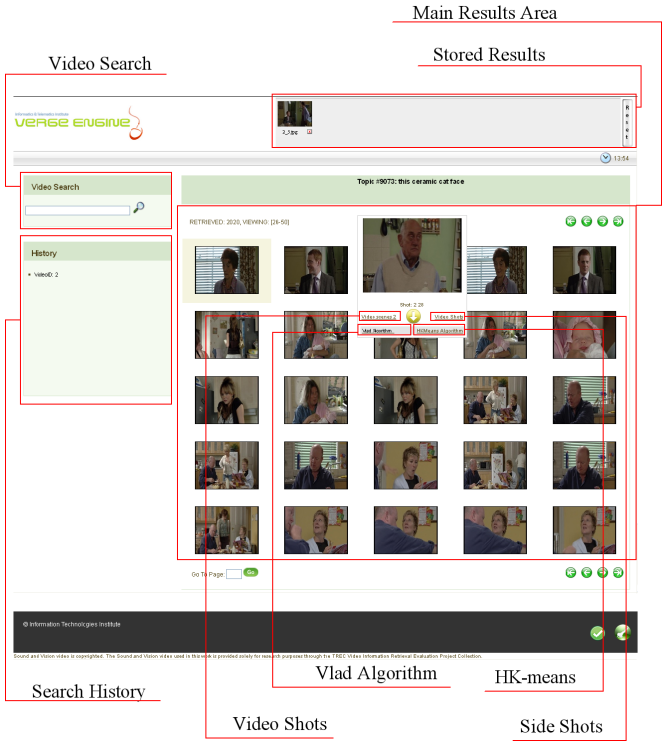

Figure 4: User interface of the interactive search platform.

The first extension, called Fast STG, reduces the computational cost of shot grouping, by considering shot linking transitivity and the fact that scenes are by definition convex sets of shots, thus limiting the number of shot pairs whose possible linking needs to be evaluated. The latter allows for faster detection of the scene boundaries, while maintaining the same performance with the original STG algorithm. The second extension, called Generalized STG, builds on the former in order to construct a probabilistic framework, towards multiple STGs combination, alleviating the need for manual STG parameter selection. As described in [21], this probabilistic framework can also be used for the realization of a multi-modal approach for scene segmentation, allowing the fusion of STGs built by considering different forms of information extracted from the video, such as low level audio or visual features, visual concepts and audio events.

The scene segmentation module is applied in order to allow the system deal with the large amount of data provided (∼470.000 shots). Therefore, the searching techniques were applied only to representative shots of each scene and thus the total number of shots was minimized down to 1/4 compared to the original size. Thus, the application of scene segmentation aids at the indexing of the data and improves the scalability of the search engine.

4.2.2 Visual Similarity Search Module

The visual similarity search module performs image content-based retrieval with a view to retrieving visually similar results. Given that the input considered is video, these images are obtained by representing each shot with its temporally middle frame, called the representative keyframe. Visual similarity is realized using local information as formulated by vector quantizing the local descriptors obtained by computing SURF descriptors around specific interest points within the scene [23].

Inspired from recent success that local patch descriptors has achieved when combined with Bag of Visual Words (BoVW) schema [24] and other vector representations of an image, such as VLAD descriptor proposed in [25], we choose to adopt a local based approach for computing visual similarities among images. In this case, appearance histograms are computed around specific interest points

[5]In TRECVID 2013 INS the shot boundaries have been provided by the organizers

(either densely or sparsely) and then local information is aggregated into a single fixed-length vector representation using a well-designed quantization technique (i.e. BoVW, VLAD).

In our work, we compute 128-bin SURF [23] descriptors around each detected interest point (i.e. corner, edge, sampled grid point) for local patch description and then two holistic representation techniques follow up for representing, comparing and retrieving similar image samples adequately. On the first technique, we follow a hierarchical K-Means clustering for constructing a vocabulary tree for partitioning the high dimensional space of our image features. Inspired from high accurate results at low computational cost presented in [24], TF-IDF quantization and efficient indexing are adopted for computing approximate distances among images. On the second adopted technique, we followed K-Means clustering and VLAD encoding for representing images. PCA is applied on the resulting feature vectors for fast and discriminative acquisition while for indexing ADC approach was followed as in [25].

4.3 Instance Search Task Results

The system developed for the instance search task includes the aforementioned modules. We submitted three runs to the INS task. These runs employed different combinations of the visual similarity modules and are described in Table 5.

Table 5: Modules incorporated in each run.

Modules	Run IDs I_A_YES_ITI-CERTH_x		
	x=1	x=2	x=3
Scene Segmentation	yes	yes	yes
Hierachical K-means Clustering (BoVW)	yes	no	yes
K-Means clustering and VLAD encoding (VLAD)	yes	yes	no

It should be mentioned that the complementary functionalities (i.e. shot-segmented view of each video, storage structure and history bin) were available in all runs. According to the TRECVID guidelines the time duration for each run was set to fifteen minutes. The number of topics and the mean inverted rank for each run are illustrated in Table 6.

Table 6: Evaluation of search task results.

Run IDs	Mean Average Precision	Number of correctly recognized shots
I_NO_IT_CERTH_1	0.007	140/12422
I_NO_ITI_CERTH_2	0.010	150/12422
I_NO_ITI_CERTH_3	0.006	108/12422

Before proceeding with the analysis of the results, it should be noted that the low MAP values are due to an error that occurred during the extraction of keyframes from shots. Specifically, there was a misalignment between shots and keyframes, which resulted in submitting wrong shots that in most of the times are temporally adjacent with the correct ones.

However, despite this error and given the fact that temporally adjacent shots of the correct ones were submitted, we can still compare the values of Table 6 and draw some first conclusions regarding the effectiveness of the applied visual similarity search techniques and their relative performance. First, by comparing runs 2 and 3, it is obvious that the VLAD technique achieves higher performance compared to the BoVW technique. The same conclusion is drawn when comparing runs 1 and 2, since it is clear that the involvement of BoVW technique doesn't improve the results of run 2.

5 Conclusions

In this paper we reported the ITI-CERTH framework for the TRECVID 2013 evaluation. ITI-CERTH participated in the SIN, INS, MED and MER tasks in order to evaluate existing techniques and algorithms.

Regarding the SIN task, the fact that a large number of annotated training examples is available gave us the opportunity to increase the number of the concept detectors by adopting a sampling strategy in order to train models with different subsets of the whole training set. The large pool of the concept detectors in combination with a stacking-based score refinement approach, using the ML-kNN algorithm, resulted to an improvement of 9.7% over the baseline approach.

As far as INS task is concerned, the results reported weren't satisfactory due to an error introduced during the keyframe extraction procedure. Despite this error, some first conclusions are drawn. First, the VLAD technique performed better than the BoVW and that a way to allow massive submission of shots would improve the results.

Finally, concerning the MED and MER tasks a number of efficient algorithms were evaluated providing satisfactory performance. In particular, for the former task a method exploiting limited static visual and/or short-time audio information was employed, while for the latter a feature selection method was applied combining discriminant analysis and a model vector video representation.

6 Acknowledgements

This work was partially supported by the European Commission under contracts FP7-287911 LinkedTV, FP7-318101 MediaMixer, FP7-600826 ForgetIT and FP7-610411 MULTISENSOR.

References

[1] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR '06: 8th ACM Int. Workshop on Multimedia Information Retrieval*, pages 321–330, NY, USA, 2006. ACM Press.

[2] A. Moumtzidou, A. Dimou, P. King, and S. Vrochidis et al. ITI-CERTH participation to TRECVID 2009 HLFE and Search. In *TRECVID 2009 Workshop*, pages 665–668. 7th TRECVID Workshop, Gaithersburg, USA, November 2009.

[3] A. Moumtzidou, A. Dimou, N. Gkalelis, and S. Vrochidis et al. ITI-CERTH participation to TRECVID 2010. In *TRECVID 2010 Workshop*. 8th TRECVID Workshop, Gaithersburg, MD, USA, November 2010.

[4] A. Moumtzidou, P. Sidiropoulos, S. Vrochidis, N. Gkalelis, and S. Nikolopoulos et al. ITI-CERTH participation to TRECVID 2011. In *TRECVID 2011 Workshop*. 9th TRECVID Workshop, Gaithersburg, MD, USA, December 2011.

[5] A. Moumtzidou, N. Gkalelis, P. Sidiropoulos, M. Dimopoulos, and S. et al. Nikolopoulos. ITI-CERTH participation to TRECVID 2012. In *TRECVID 2012 Workshop*, Gaithersburg, MD, USA, 2012.

[6] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In Ajay Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[7] T. Dietterich. Ensemble methods in machine learning. In *1st Int. Workshop in Multiple Classifier Systems*, volume 1, pages 1–15, 2000.

[8] P. Sidiropoulos, V. Mezaris, and I. Kompatsiaris. Video tomographs and a base detector selection strategy for improving large-scale video concept detection. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–14, 2014.

[9] 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 Cathal Gurrin, Frank Hopfgartner, Wolfgang Hurst, Hvard Johansen, Hyowon Lee, and Noel OConnor, editors, *MultiMedia Modeling*, volume 8325 of *Lecture Notes in Computer Science*, pages 1–12. Springer International Publishing, 2014.

[10] B. Safadi and G. Quénot. Re-ranking by Local Re-Scoring for Video Indexing and Retrieval. In C. Macdonald, I. Ounis, and I. Ruthven, editors, *CIKM*, pages 2081–2084. ACM, 2011.

[11] B. Parmanto, P. W. Munro, and H. R. Doyle. Improving committee diagnosis with resampling techniques. In D.S. Touretzky, M.C. Mozer, and M. E. Hesselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 882–888, Cambridge, MA., 1996. MIT press.

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

[13] P. Over, G. Awad, M. Michel, J.Fiscus, G.Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID 2013*. NIST, USA, 2013.

[14] S. Ayache and G. Quenot. Video Corpus Annotation using Active Learning. In *European Conference on Information Retrieval (ECIR)*, pages 187–198, Glasgow, Scotland, March 2008.

[15] N. Gkalelis, V. Mezaris, and I. Kompatsiaris. High-level event detection in video exploiting discriminant concepts. In *9th Int. Workshop on Content-Based Multimedia Indexing (CBMI 2011)*, pages 85–90, Madrid, Spain, June 2011.

[16] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[17] R. Mertens, L. Lei, H.and Gottlieb, G. Friedland, and A. Divakaran. Acoustic super models for large scale video event detection. In *2011 joint ACM workshop on Modeling and representing events*, J-MRE '11, pages 19–24, NY, USA, 2011. ACM.

[18] J.-F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. Evans, B. Fauve, and J. Mason. Alize/spkdet: A state-of-the-art open source software for speaker recognition. *Odyssey-The Speaker and Language Recognition Workshop*, 2008.

[19] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki. Video event recounting using mixture subclass discriminant analysis. In *IEEE Int. Conference on Image Processing (ICIP 2013)*, Melbourne, Australia, September 2013.

[20] 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.

[21] 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.

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

[23] 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.

[24] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[25] J. Hervé, D. Matthijs, S. Cordelia, and P. Patrick. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, June 2010.