ITI-CERTH participation to TRECVID 2015

Foteini Markatopoulou[1,2], Anastasia Ioannidou[1], Christos Tzelepis[1,2], Theodoros Mironidis[1], Damianos Galanopoulos[1], Stavros Arestis-Chartampilas[1], Nikiforos Pittaras[1], Konstantinos Avgerinakis[1], Nikolaos Gkalelis[1], Anastasia Moumtzidou[1], Stefanos Vrochidis[1], Vasileios Mezaris[1], Ioannis Kompatsiaris[1], Ioannis Patras[2]

[1] Information Technologies Institute/Centre for Research and Technology Hellas, 6th Km. Charilaou - Thermi Road, 57001 Thermi-Thessaloniki, Greece
{markatopoulou, ioananas, tzelepis, mironidis, dgalanop, stav_ares, npittaras, koafgeri, gkalelis, moumtzid, stefanos, bmezaris, ikom}@iti.gr
[2] Queen Mary University of London, Mile end Campus, UK, E14NS
i.patras@qmul.ac.uk

Abstract

This paper provides an overview of the runs submitted to TRECVID 2015 by ITI-CERTH. ITI-CERTH participated in the Semantic Indexing (SIN), Multimedia Event Detection (MED), Instance Search (INS) and Surveillance Event Detection (SED) tasks. Our SIN task participation is based on the extraction of discriminative features and the improvement of the concept learning module. Specifically, DCNN-based descriptors are combined with local descriptors. Furthermore, a MTL approach is used to share knowledge between tasks and improve the concept detection accuracy. In the MED task, a kernel subclass version of our discriminant analysis method (KSDA) combined with a fast linear SVM, along with the RDSVM for exploiting "near-miss" samples, is employed. Motion descriptors as well as local and DCNN-based visual descriptors are used. Moreover, we use our zero-example event detection framework for the 000Ex training condition. The INS task is performed by employing VERGE, which is an interactive retrieval application that integrates multiple retrieval functionalities considering only visual information. Finally, the interactive SED task integrates a sophisticated activity detection algorithm into a simple and user-friendly graphical interface.

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 (also known as high-level feature extraction task - HLFE) under the MESH (TRECVID 2008) and K-SPACE (TRECVID 2007 and 2008) EU-funded research projects. In 2009 ITI-CERTH participated as a stand-alone organization in the SIN and Search tasks ([2]), in 2010 and 2011 in the KIS, INS, SIN and MED tasks ([3], [4]) and in 2012, 2013 and 2014 in the INS, SIN, MED and MER tasks ([5], [6], [7]) of TRECVID. 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: SIN, MED, INS and SED. In the following sections we will present in detail the employed algorithms and the evaluation for the runs we performed in the aforementioned tasks.

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

2 Semantic Indexing

2.1 Objective of the Submission

The goal in the TRECVID 2015 SIN task [8] is the development and use of concept detectors to retrieve for each concept a ranked list of 2000 test shots that are mostly related with it. The ITI-CERTH participation in the SIN task was based on the extraction of discriminative features and the improvement of the concept learning module. Our aim was to improve system's performance, in comparison to our SIN 2014 participation. To achieve our goal we focused on four different directions, which correspond to sub-components of one or more of our submitted runs: i) Features extracted with the use of pre-trained and fine-tuned deep convolutional neural networks (DCNN). ii) Dimensionality reduction using Kernel Subclass Discriminant Analysis (KSDA) [9, 10, 11]. iii) Cascades of local and DCNN-based descriptors [12, 13]. iv) Multi-task learning (MTL) using the Logistic-lasso algorithm [14].

2.2 System Overview

Similar to our TRECVID 2014 participation [7], we developed a two-layer concept detection system. The first layer builds multiple independent concept detectors. The second layer takes as input the output of the first layer, exploits concept correlations and refines the initial scores.

Specifically, in the first layer of our system, the video stream is initially sampled, generating one keyframe per shot. Each sample is represented using one or more types of i) hand-crafted features (e.g. local descriptors, such as SIFT, SURF, ORB etc., aggregated into global image descriptor vectors using the VLAD encoding) and ii) DCNN-based features. Subsequently, we experimented with different learning methods that use these features and build concept detectors:

- Each type of features serves as input to Linear Support Vector Machine (LSVM) or Logistic Regression (LR) classifier.
- An extended and speeded-up version of our KSDA method [9, 10, 11] for dimensionality reduction is used to reduce the dimensionality of the feature vectors; the reduced vectors similarly to the previous case serve as input to LSVMs or LR classifiers.
- We propose an improved way of ordering and combining independently trained LSVMs using a cascade.
- A MTL approach is used to share knowledge between tasks and improve the concept detection accuracy.

In the second layer of the stacking architecture, the fused scores from the first layer are aggregated in model vectors and refined by two different approaches that work sequentially. The first approach uses a multi-label learning algorithm that incorporates concept correlations [15]. The second approach is a temporal re-ranking method that re-evaluates the detection scores based on adjacent video segments as proposed in [16].

The key components of our concept detection experiments are presented below.

2.2.1 Hand-crafted Visual Features

We employed image representations that are based on hand-crafted local descriptors following the experimental setup of our TRECVID 2014 SIN submission [7]. More specifically, we calculate 128-SIFT, 128-SURF and 256-ORB grayscale descriptors. We also calculate two color extensions, in the RGB and Opponent color space, for each of the grayscale descriptors [15]. For each of these descriptors, the local feature vectors are subsequently aggregated using the VLAD encoding. The VLAD encodings are then compressed into 4000-element vectors by applying a modification of the random projection matrix [17].

2.2.2 DCNN-based Visual Features

We used features based on three different pre-trained DCNN networks: i) The 16-layer deep ConvNet network provided by [18], ii) the 22-layer GoogLeNet network provided by [19], and iii) the 8-layer

CaffeNet network described in [20]. We apply each of these networks on the TRECVID keyframes and we use as a feature i) the output of the last hidden layer of ConvNet (fc7), which results in a 4096-element vector, ii) the output of the last fully-connected (fc) layer of CaffeNet (fc8), which results in a 1000-element vector, iii) the output of the last fc layer of GoogLeNet (loss3), which results in a 1000-element vector. We refer to these features as CONV, CAFFE and GNET in the sequel, respectively.

We also fine-tuned the GoogLeNet and the CaffeNet network, on the training set of the TRECVID SIN 2015 positive samples for the 60 concepts. We used features based on three different fine-tuned DCNN networks: i) the GoogLeNet extended by one layer, ii) the CaffeNet extended by one layer, and iii) the GoogLeNet extended by two layers. With respect to the first two fine-tuned networks, the output classification layer for the 1000 ImageNet ISLVRC concepts of the pre-trained GoogLeNet and CaffeNet networks is discarded and replaced with a 1024-dimensional fc layer, along with RELU and dropout layers, followed by a 60-dimensional fc layer, which constitutes the classification layer for the 60 SIN test concepts. This approach extends the GoogLeNet and CaffeNet network by one layer (considering layers with weights only). Due to the fact that the GoogLeNet has three classifiers, one hidden layer is added before each classification fc layer. We use the output of the last classification layer for each of the two fine-tuned networks as a feature vector to train concept detectors separately for each concept. We refer to these features as GNET_E1, CAFFE_E1, for GoogleNet and CaffeNet fine-tuned networks, respectively. Furthermore, we use the output of the extended hidden layer corresponding to the second (Loss2_fci) and third classifier (Loss3_fci) of the GoogLeNet fine-tuned network, which results in a 1024-element vector for each of the layers. We refer to these features as GNET_E1_L2_FCI, GNET_E1_L3_FCI. With respect to the third fine-tuned network, we extend the pre-trained GoogLeNet network with two 512-dimensional fc layers, along with RELU and dropout layers, followed with a 60-dimensional fc classification layer. We use as a feature the output of the two hidden layers that correspond to the third classifier of the fine-tuned GoogLeNet (Loss3_fci, Loss3_fc_ii), which results in a 512-element vector for each layer. We refer to these features as GNET_E2_L3_FCI, GNET_E2_L3_FCII.

Both the reduced VLAD vectors and the DCNN-based features are used for learning concept detectors in one of the following ways: i) they serve as input directly to LSVM classifiers, ii) they are reduced using KSDA and serve as input to LSVM classifiers, iii) they are reduced using KSDA and serve as input to the Logistic-lasso algorithm for MTL.

2.2.3 Kernel Subclass Discriminant Analysis and LSVMs for Classification

While previous features can serve as input directly to LSVMs or LR classifiers, in some of our experiments we used KSDA+LSVM for classification [9, 10]. Specifically, the KSDA method is used to derive a lower dimensional embedding of the original feature vectors. Then, the features in the resulting subspace serve as input to LSVMs. For more details on an extended and speeded-up version of KSDA that was used for the SIN task please refer to [11].

2.2.4 Cascade of Classifiers

For each concept many base classifiers are trained on different types of features. The simplest way to combine the output of them for the same concept is late fusion, e.g., in terms of arithmetic mean. For some of our experiments we introduce a more elaborate approach that arranges the base classifiers on a cascade architecture. We use the cascade of classifiers proposed in [12], where the trained classifiers are arranged in stages using a search-based algorithm. A keyframe is classified sequentially by each stage and the next stage is triggered only if the previous one returns a positive prediction (i.e. that the concept appears in the keyframe). The rationale behind this is to rapidly reject keyframes that clearly do not match the classification criteria and focus on those keyframes that are more difficult and more likely to depict the sought concept or object. This procedure is able to reduce the number of classifier evaluations and consequently the number of classifiers to be fused.

2.2.5 Multi-task Learning

We used a MTL algorithm to share knowledge between tasks and improve the concept detection performance. Assuming that some groups of concepts are expected to be related through some underlying structure, we utilize the Logistic-lasso [21] implementation for MTL learning. Logistic-lasso algorithm can define this relatedness on the parameters of the independently trained concept detectors. The MALSAR MTL library [14] was used as the source of the Logistic-lasso algorithm. Each type of features serves as input to the Logistic-lasso algorithm to train MTL models. The scores returned from the MTL models for the same concept are fused using late fusion (arithmetic mean).

2.2.6 Stacking for Exploiting Concept Correlations

All the above techniques were part of the first layer of the stacking architecture in one or more runs. For the second layer of the stack in all runs we use a multi-label learning algorithm in order to capture concept correlations. More specifically, we obtain concept score predictions 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. We choose the LP [15] algorithm that models correlations among sets of more than two concepts. For more details, please refer to our previous submission [7].

2.3 Description of Runs

Four SIN runs were submitted in order to evaluate the potential of the aforementioned approaches on the TRECVID 2015 SIN dataset [8]. The submitted runs are based on different combinations of the following four developed systems:

- System A: A cascade of 11 different visual descriptors that have been arranged on four cascade stages; specifically, the cascade consists of one binary local descriptor and its two color variants (ORBx3), two types of non-binary local descriptors with their color variants (SURFx3, SIFTx3), the output of the CONV layer and the output of the GNET_E1 layer. We train one LSVM for each of the 11 visual descriptors. Therefore, 11 LSVMs per concept are trained and arranged on cascade stages. The scores returned from them are fused using the cascade.
- System B: A cascade of 11 different visual descriptors; specifically, the cascade consists of one binary local descriptor and its two color variants (ORBx3), two types of non-binary local descriptors with their color variants (SURFx3, SIFTx3), the output of the CONV layer and the output of the GNET_E1_L3_FCI layer. In contrast to A, each descriptor was firstly reduced to a lower dimensional feature space using the KSDA method.
- System C: Late fusion of the 11 descriptors of B using MTL; in this case the Logistic-lasso algorithm is applied separately to each descriptor in order to train one MTL model for each of the 11 descriptors. The corresponding descriptors that were used as stages of the cascade method in B, are firstly combined by averaging the MTL model's output scores and then the combined outputs of all stages are further fused together.
- System D: Late fusion of 7 DCNN-based descriptors; specifically, we train five LR models using *cross validated committees* method [7] for each of the following DCNN-based descriptors: i) CAFFE, ii) GNET, iii) CAFFE_E1, iv) GNET_E1_L2_FCI, v) GNET_E1_L3_FCI, vi) GNET_E2_L3_FCI, vii) GNET_E2_L3_FCII. Therefore 35 LR models per concept are generated. The scores returned from the 35 LR models are fused using the arithmetic mean.

For each of the above four systems, the fused scores per concept are refined in the second layer of the stacking architecture, firstly by applying the LP multi-label learning algorithm that captures concept correlations, and secondly by applying the temporal re-ranking method.

The 4 submitted runs for the main task of the 2015 TRECVID SIN competition are briefly described in the following. When the combination of different systems is indicated, we mean a late fusion combination which takes the output scores from the combined systems and calculates the arithmetic mean of these scores:

- ITI-CERTH-Run1: "Combination". In this run we combine the output scores of the four systems presented above (Systems A to D).

- ITI-CERTH-Run2: "MTL-LogLasso". This run is a subset of the previous one ("Combination"), that combines only systems C and D.
- ITI-CERTH-Run3: "Cascade-KDA". This run is a subset of the "Combination" run that combines only systems B and D.
- ITI-CERTH-Run4: "Cascade". This run is a subset of the "Combination" run that combines only systems A and D.

2.4 Semantic Indexing Task Results

Table 1: Mean Extended Inferred Average Precision (MXinfAP) for all submitted runs, and for the four developed systems (System A to D) that contribute to these runs.

Submitted run:	ITI-CERTH 1	ITI-CERTH 2	ITI-CERTH 3	ITI-CERTH 4
MXinfAP (with the bug)	0.130	0.112	0.096	0.110
MXinfAP (without the bug)	0.263	0.260	0.250	0.252
Developed systems:	System A	System B	System C	System D
MXinfAP (without the bug)	0.239	0.258	0.219	0.232

Table 1 summarizes the evaluation results of the aforementioned runs in terms of the Mean Extended Inferred Average Precision (MXinfAP). Furthermore, we present the MXinfAP for each of the four developed systems that have been presented in the previous section. It should be noted that after submission we discovered a bug in the feature extraction process, which led to wrong results. We fixed this bug and Table 1 reports both the official results (with the bug), but also the correct results (without the bug). From the obtained results the following conclusions can be drawn:

The "Cascade" and the "Cascade-KDA" run (ITI-CERTH-Run-4 and ITI-CERTH-Run-3, respectively), present similar accuracy. It should be noted that the employed cascades lead to approximately 37% relative decrease in the amount of classifier evaluations compared to the late fusion alternative, without decreasing the overall concept detection accuracy. The "MTL-LogLasso" run (ITI-CERTH-Run-2) uses MTL to take advantage of the relations between different concepts. The improved accuracy achieved by this run demonstrates the significance of transferring learning among different concepts, rather than training each concept independently from the others. Finally, the "Combination" run (ITI-CERTH-Run-1) that combines the four developed systems can only slightly improve the previous runs, which shows that the different systems are not really complementary to each other. Overall, considering our best run (0.263 MXinfAP), our system performs above the median for 28 out of 30 evaluated concepts, as shown in Fig. 1.

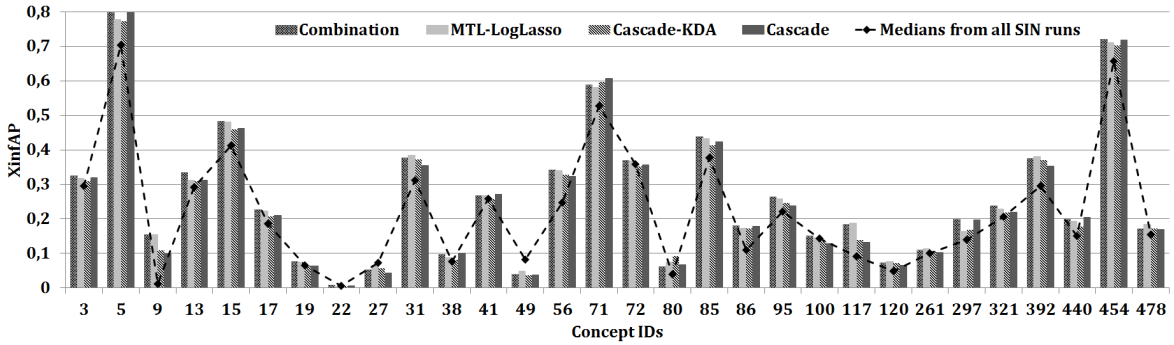

Figure 1: Extended Inferred Average Precision (XinfAP) per concept for our submitted runs (results without the bug).

3 Multimedia Event Detection

3.1 Objective of the Submission

In our submission we applied methods for learning i) solely from the textual description of an event class (000Ex task) and ii) from few (010Ex task) or from an abundance of training videos (100Ex task), while we also exploited "near-miss" video samples as weighted negative or weighted positive ones using an automatic weighting scheme.

3.2 System Overview

3.2.1 000Ex: Learning video event detectors from events' textual descriptions

In the 000Ex task we use our zero-example event detection framework presented in [22]. This framework uses only the textual description of each event class, namely the Event Kit. For linking this textual information with the visual content of the MED15–EvalSub video collection, we use a) a pool of 1000 concepts along with their titles and, in some cases, a limited number of subtitles (e.g. concept *bicycle-built-for-two* has the subtitles *tandem bicycle* and *tandem*), and b) a pre-trained detector for these concepts. The latter is the 16-layer deep ConvNet [18] trained on the ImageNet data [23].

Given the textual description of an event, our framework first identifies N words or phrases that most closely relate to the event; this word-set is called Event Language Model (ELM). Three different types of ELMs are constructed; The first type of ELM is based on the automatic extraction of word terms solely from the title of an event; in the second type, the visual cues of the event kit are used along with title of the event; and, the third type is the enrichment of the second type with audio cues.

In parallel, for each of the 1000 concepts of our concept pool, our framework similarly identifies M words or phrases: the Concept Language Model (CLM) of the corresponding concept. Six different types of CLMs, depending on the textual information used for each concept (the title of the concept or the Bag of Words (BoW) representation of the top-20 articles in Google or in Wikipedia), as well as the weighting technique (Tf-Idf or none) adopted for transforming this textual information in a BoW representation.

Subsequently, for each word in ELM and each word in each one of CLMs we calculate the Explicit Semantic Analysis (ESA) distance [24] between them. For each CLM, the resulting $N \times M$ distance matrix expresses the relation between the given event and the corresponding concept. In order to compute a single score expressing this relation, we apply to this matrix different operators, such as various matrix norms (l_2, Frobenius or l_∞) or distance measures (Hausdorff distance). Consequently, a score is computed for each pair of ELM and CLM. The 1000 considered concepts are ordered according to these scores (in descending order) and the K-top concepts along with their scores constitute our event detector. Multiple event detectors are produced as the result of different combinations of the above steps.

3.2.2 010Ex, 100Ex: Learning video event detectors from positive and related video examples

The target of an event detection system is to learn a decision function $f\colon \mathcal{F} \to \{\pm 1\}$, where $\mathcal{F}$ denotes the space where the video representations lie in. f assigns a test video to the event class (labelled with the integer 1) or to the "rest of the world" class (labelled with the integer -1). For each event class, this is typically achieved using a training set $\mathcal{X} = \{(\mathbf{x}_i, y_i)\colon \mathbf{x}_i \in \mathcal{F}, y_i \in \{0, \pm 1\},\ i = 1, \ldots, N\}$, where $\mathbf{x}_i$ denotes the representation of the i-th training video and y_i denotes the corrsponding ground truth label. Labels -1, $+1$ correspond respectively to negative and positive training examples, while label $y_i = 0$ indicates that the i-th video example does not exactly fulfill the requirements to be characterized as a true positive example, nevertheless is closely related to the corresponding event class; we refer to these samples as "near-miss". In some runs we treat near-miss videos as either weighted negatives or weighted positives in conjunction with an automatic weighting selection scheme.

Our method exploits three types of visual information, i.e., motion features, local descriptors, and DCNN-based features. For the extraction of local visual descriptors and DCNN-based features, the procedure described in Section 2 is applied. We briefly describe the different visual modalities in the following:

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

Table 2: MED 2015 Pre-Specified (PS) events.

- Each video is decoded into a set of keyframes at fixed temporal intervals (1 keyframe every 6 seconds). Low-level feature extraction and encoding is performed as described in Section 2.2.1. Specifically, four different local descriptors (SIFT, OpponentSIFT, RGB-SIFT, RGB-SURF), are applied to extract local visual information for every keyframe. The extracted features for each local descriptor are encoded using VLAD, compressed using a modification of the random projection matrix [17] technique to $\mathbb{R}^{4000}$, and averaged over all keyframes of the video. The four feature vectors are then concatenated to provide a single feature vector in $\mathbb{R}^{16000}$ at video level, encoding static visual information.
- Each video is decoded into a set of keyframes at fixed temporal intervals (approximately 2 keyframes per second). Each keyframe is represented using the last one or two hidden layers (fc7, fc8) of the 16-layer deep ConvNet network [18] also presented in Section 3.2.1. Thus, each keyframe is represented either by a 1000-element vector or a (4096+1000)-element vector. Then, a video-level model vector is computed for each video by taking the average of the corresponding keyframe-level representations.
- For encoding motion information we use improved dense trajectories (DT) [25]. Specifically we employ the following four low-level feature descriptors: Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and Motion Boundary Histograms in both x (MBHx) and y (MBHy) directions. Hellinger kernel normalization is applied to the resulting feature vectors, followed by Fisher Vector (FV) encoding with 256 GMM codewords. Subsequently, the four feature vectors are concatenated to yield the final motion feature descriptor for each video in $\mathbb{R}^{101376}$.

The final feature vector representing a video is formed by concatenating the feature vectors derived for each visual modality (static local, motion, model vectors), yielding a new feature vector in $\mathbb{R}^{117722}$.

In our submission we used two different machine learning methods for building our event detectors:

- We utilized an extended and speeded-up version of our Kernel Subclass Discriminant Analysis [9, 10] for dimensionality reduction, followed by a fast linear SVM (KSDA+LSVM), as discussed in Section 2.2.3. The GPU-accelerated implementation of this method [11] was not used in our MED 2015 experiments; due to the limited number of training samples, it was not necessary for us to exploit the GPU computing capabilities.
- We also used Relevance Degree SVM (RDSVM) proposed in [26] for handling "near-miss" video samples as weighted negative or weighted positive ones using an automatic weighting selection scheme as in our previous works [22].

3.3 Dataset Description

For training our Pre-Specified (PS) event detectors we used the PS-Training video sets, consisting of 2000 (80 hours) positive (or near-miss) videos, and the Event-BG video set containing 5000 (200 hours) of background videos. The 20 PS event classes are shown in Table 2 for the shake of completeness.

For the evaluation of our systems we processed the MED15–EvalSub set consisting of 32000 videos (960 hours). We submitted runs for the 000Ex, the 010Ex, and the 100Ex evaluation conditions (i.e., 0, 10 or 100 positive exemplars, respectively, are used for learning the specified event detector).

3.4 Description of Runs

3.4.1 000Ex

For our 000Ex submission we experimented with 3 different CLMs, 3 different ELMs, 2 weighting techniques and 4 different matrix operators, as described in Section 3.2.1, which resulted in 72 different design choices for building an event detector. We chose the best of them (hereafter called the *best detector*), as well as the *top-10* of them, based on experiments on previous MED datasets, as in [22]. We submitted 5 different runs in the 0Ex task, one primary and four contrastives:

- **c-1oneCosine**: In the first contrastive run we use our best detector as the event detector and the cosine similarity as the similarity measure.
- **c-2avgCosine**: In this run, the arithmetic mean of the top-10 detectors is employed and the cosine similarity is used as the similarity measure.
- **c-3oneHist**: In the third run, the best detector is used as the event detector and the Histogram Intersection is used as the similarity measure.
- **c-4avgHist**: In the last contrastive run we use the arithmetic mean of the top-10 detectors and the Histogram Intersection is used as the similarity measure.
- **p-1Fusion**: As our primary run we use the fusion in terms of arithmetic mean of the above four contrastive runs.

3.4.2 010Ex & 100Ex

For each of the training conditions of 010Ex and 100Ex, we submitted 4 runs; one primary and three contrastives:

- **c-1KDALSVM**: In the first contrastive run, the KSDA+LSVM method is used to build the event detectors and perform the event search in the MED15–EvalSub set using motion, static local, and DCNN-based features, as discussed in Section 3.2.2.
- **c-2RDKSVM**: In the second contrastive run, RDSVM is used to build the event detectors by treating near-miss videos as weighted negatives or positives, as discussed in Section 3.2.2. Only the output of the last hidden layer (fc8) of the pre-trained DCNN described in Section 3.2.2 is used for representing each video.
- **c-3RDKSVM**: In the third contrastive run, similarly to previous run, RDSVM is used, together with the last two hidden layers of the DCNN-based feature representation (fc7+fc8).
- **p-1Fusion**: As our primary run we use the fusion in terms of arithmetic mean of the above three contrastive runs.

3.5 Multimedia Event Detection Results

In Table 3, the evaluation results of our 000Ex(3a), 010Ex(3b), and 100Ex(3c) systems for the MED task are shown in terms of InfAP@200 along the 20 target events for the PS task.

Table 3: ITI-CERTH results

(a) 000Ex

Run ID	mInfAP@200
p-1Fusion	**0.0617**
c-1oneCosine_1	0.0478
c-2avgCosine_1	0.0473
c-3oneHist_1	0.0474
c-4avgHist_1	0.0592

(b) 010Ex

Run ID	mInfAP@200
p-1Fusion	0.211
c-1KDALSVM	**0.2493**
c-2RDKSVM	0.1588
c-3RDKSVM	0.2026

(c) 100Ex

Run ID	mInfAP@200
p-1Fusion	0.3649
c-1KDALSVM	**0.4111**
c-2RDKSVM	0.2894
c-3RDKSVM	0.2367

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

- Concerning the 000Ex task, it seems that the exploitation of a large number of detectors, produced by our various design strategies described in Section 3.2.1, gives a boost to the performance of our method. Furthermore, late fusion led to better detection results.

- Concerning the 010Ex and 100Ex training conditions, we observe that our KSDA method (Section 3.2.2) achieved the best results (24.93% and 41.11%, respectively), while the contrastive runs that exploited "near-miss" samples as weighted negatives/positives using RDSVM (Section 3.2.2) achieved worse results than KSDA+LSVM, due to the fact that they used only the DCNN-based features, in contrast with the KSDA+LSVM run, where very high dimensional features capturing motion, static local and DCNN-output information were used for training the event detectors.
- Our run based on KSDA+LSVM using all the features presented in Section 3.2.2 (run c-1KDALSVM) achieved mInfAP@200=0.4111, which is the second-best result among all participants' runs validated on the MED15–EvalSub set.

4 Instance Search

4.1 Objective of the Submission

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. The searcher is provided with visual examples of the specific query object in order to commence with the searching. 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).

ITI-CERTH's participation in the TRECVID 2015 instance search (INS) task aimed at studying and drawing conclusions regarding the effectiveness of a more object-oriented visual search approach in comparison with our last year's concept-oriented approach. Our system is integrated in VERGE[1] interactive video search engine. Three runs were submitted this year that differ slightly on the algorithmic steps for building the BoW model which was used in our system.

4.2 System Overview

The system employed for the Instance Search task was VERGE, which is an interactive retrieval application that combines retrieval functionalities considering visual information, accessible through a friendly Graphical User Interface (GUI), as shown in Fig. 2. The following modules are integrated in the developed search application:

- Object-based Visual Search Module;
- High Level Visual Concept Retrieval;

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

The existence of a friendly and smartly designed graphical interface (GUI) plays a vital role in the procedure of searching. The interface comprises of four main components:

- the video search toolbar
- the main results area
- the topics drawer
- the visual similarity search panel

The main results area of the interface includes a shot-based representation of the available videos in a grid-like interface. When the user hovers over a shot keyframe, three options appear on its corners as shown in Fig. 3. A selection tool that lets the user select the current keyframe, a cross tool that expands a view of the temporarily related shots given from the scene segmentation, and a magnifier tool that opens up a frame that contains a larger preview of the image and gives the user the opportunity to crop an object in order to make an object-based search. All the shots from the main results area are draggable. When a user starts to drag an image, the *Visual Similarity Search Panel* that resides at the lower side of the screen slides up and he can drop the image on this area in order to make a search with one or more images/object instances. On the left side of the interface resides a list of the given topics. When the user selects a topic, the *Visual Similarity Search Panel* slides up and loads the four related given frames and their masked versions in order to let the user run a visual similarity search with these images. The aforementioned magnifier tool is also available in this case to give the user the choice to crop any part of these frames in order to make a more targeted search.

Below we demonstrate a usage scenario, supposing that a user is interested in finding shots which contain instances of a specific topic named "this yellow VW beetle with roofrack" (Fig. 2). When the user selects the topic, the images given for the topic and their masked versions are loaded to the *Visual Similarity Search Panel*, giving the option of initiating a search based on these images. From the results appearing, the user selects the shots illustrating the desired object, which in this case is the yellow car, and uses them in order to make a more effective query. When a shot with the desired object is spotted, the user can expand it using the cross tool with the aim to find more related shots. It is very possible for a scene to have more than one shots with the desired object. Finally, the user can choose from the concept list the one which is closer to the described topic in order to complement the visual search. In this example, the related concept *car* could be selected.

It should be noted, that the search system is built on open source web technologies and more specifically the PHP, HTML5, JavaScript and MongoDB database.

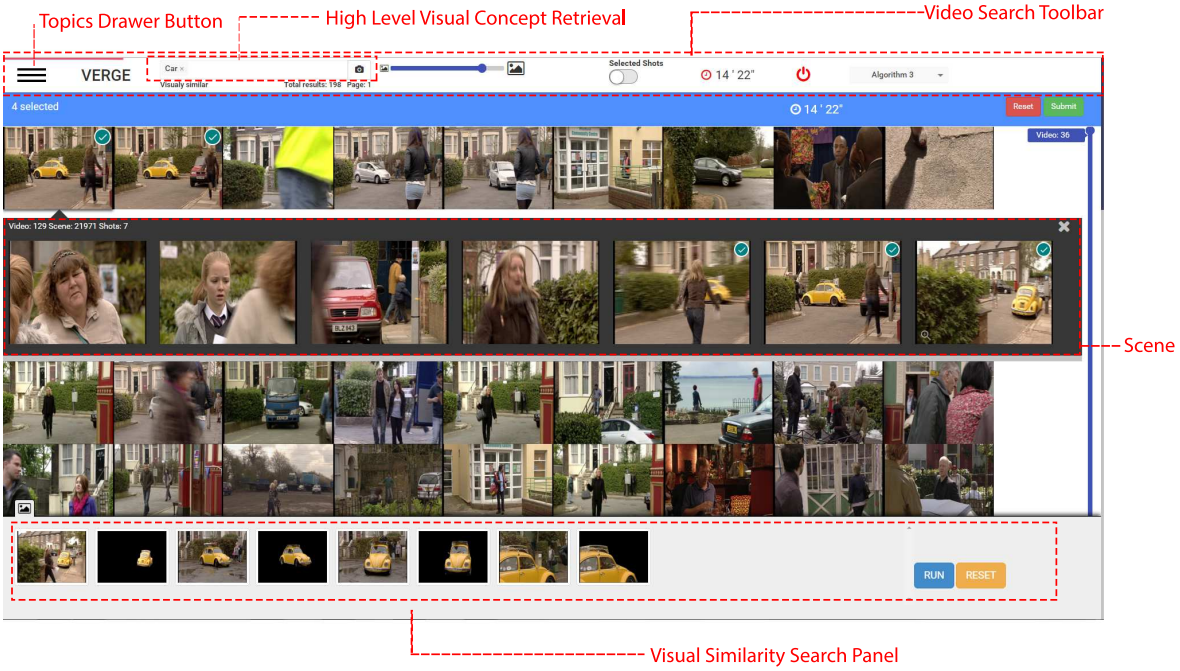

Figure 2: User interface of the interactive search platform.

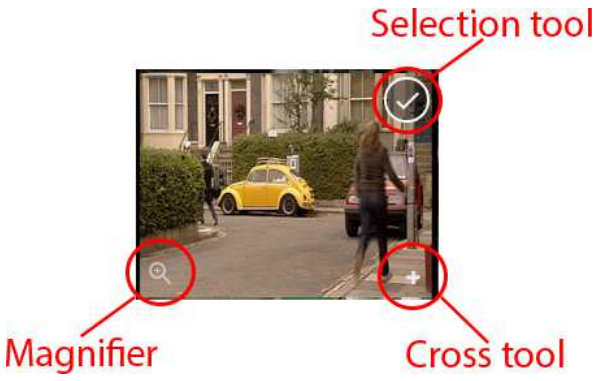

Figure 3: Shot options appearing over the shot.

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

4.2.1 Object-based Visual Search Module

The Object-based Visual Search module performs instance-based object retrieval and is based on the widely used Bag-Of-Word (BoW) model. Visual similarity is determined based on local features. In particular, fast Hessian detector and SIFT descriptor are exploited in order to extract local feature vectors from every keyframe of the dataset. For computational reasons, each keyframe's size was reduced at a pre-processing step to 75% of its original size.

The features detected in the whole image dataset are randomly sampled and afterwards, clustered using Repeated Bisecting K-Means [27] and a 2-layer visual vocabulary is constructed. Vocabularies of various sizes up to 150K were explored and the final representation of each keyframe is the result of a hard assignment. The size of the vocabulary used in the official runs is 100K.

In order to accelerate the online search time, an inverted index is built off-line using the open-source Apache Lucene[2] software for fast online search of the image database BoW vectors. The index is accessed every time a query is made in order to return the indices of the most similar keyframes. For querying the system, the whole keyframe (containing the object of interest) or any object/cropped part of the image can be selected. The similarity score between a query image and a video frame is obtained based on Lucene's scoring function (which exploits the tf-idf weighting scheme) and the ranking position of the frame in the retrieved list, i.e. Borda Count is computed for all frames in the list in order to form the final ranking.

After experimentation with the topics used in INS 2014 and based on the observation that the context around an object of interest (i.e. the background information) is sometimes helpful (e.g. some of the topics always appear in the same room/place), during our official experiments, we encouraged the usage of 8 images per query, i.e. the 4 original frames containing the object of interest and the 4 frames extracted after applying the provided masks. In this way, the most similar results for each query image (the object of interest but also its probably useful background) would be part of the initial retrieved list letting the user decide which of the results are best to use in order to re-query the system. The top-100 results are returned for each query image.

The BoW vector is computed online for each query image except for the case where the query is a video keyframe (i.e. the BoW vector is already stored in the index). The ranking lists returned for all query images are finally fused in order to display the retrieval results to the user. For the frames appearing in more than one ranking lists, we keep the maximum score (i.e. Borda count) and discard the others (MQ-Max method [28]).

In order to boost the final accuracy, the user is prompted to expand the visual search by using any number of "relevant images" from the initial ranking list. The user could use the whole images or could crop the depicting object of interest. In the first case, the retrieval is significantly faster (since the BoW vectors of the video frames are already computed and stored in the index), while in the second case, the computation of the BoW vectors for the cropped queries is performed online and the retrieval process requires more time to complete.

4.2.2 High Level Visual Concept Retrieval

This module facilitates search by indexing the video shots based on high level visual concept information, such as water, aircraft, landscape and crowd. The concepts that are incorporated into the system are the 346 concepts studied in the TRECVID 2014 SIN task using the techniques and the algorithms described in detail in [7] Section 2 (Semantic Indexing).

4.3 Instance Search Task Results

We submitted three runs to the interactive INS task. In our first submission (I_A_ITI_CERTH_2), the baseline BoW model described above was utilized. In the second submission (I_A_ITI_CERTH_3), we examined the impact of employing a saliency detection algorithm [29] before the feature extraction and the visual vocabulary construction on the retrieval results. Finally, our third run (I_A_ITI_CERTH_1) includes the results from the simple late fusion of the two aforementioned submissions (two different modalities). The two modalities run in parallel for each query and their results are fused using the MQ-Max method.

[2] `https://lucene.apache.org/core/`

According to the TRECVID guidelines the time duration for each run was set to fifteen minutes. Regarding the users that participated in INS runs, eight users were engaged in total. Although the educational and age profile of the users is similar, their experience with similar systems differs (e.g some users were expert users while others were moderate users). The mean average precision as well the recall for all runs are illustrated in Table 4 along with the results from our last year's participation.

Table 4: Evaluation of instance search task results.

Run IDs	Mean Average Precision	Recall
I_A_ITI_CERTH_1	0.064	831/8817
I_A_ITI_CERTH_2	0.053	651/8817
I_A_ITI_CERTH_3	0.046	525/8817
I_NO_ITI_CERTH_1 (INS 2014)	0.032	532/9336
I_NO_ITI_CERTH_2 (INS 2014)	0.028	315/9336

Based on the first three rows of table 4, we can conclude that despite the fact that the saliency-based BoW model performed worse than the baseline BoW model, their fusion resulted in higher Mean Average Precision and Recall values which suggests that the study of the video frames' saliency can benefit the system's retrieval performance at least for some objects. Comparing this year's results with the ones obtained from the 2014 submissions, it can be reported that the BoW representation seems to work better than VLAD for the INS task. Even though the efficiency of our system is still low compared to the other competing systems, it should be noted that we increased our system's performance considerably.

5 Surveillance Event Detection

5.1 Objective of the Submission

Surveillance Event Detection (SED) task addresses the case where observations of specific events need to be detected in a collection of surveillance video data files. The interactive event detection task involves human interaction with the built system, though, for no more than 25 minutes. TRECVID iSED 2015 provides approximately 100-hour videos for development (2008 DevSet and EvalSet, Gatwick data) and a 9 hour subset of the multi-camera data for the main evaluation. A new Group Dynamic Subset (SUB15) using only 2 hours of this video and limited to the Embrace, PeopleMeet and PeopleSplitUp events is also introduced this year.

Since this is the first ITI-CERTH's participation to the iSED task, we aimed at building a generic system for the detection of events involving solely people. Therefore, we took under consideration a subset of five (out of seven) events of interest and only one run was submitted.

5.2 System Overview

The interface developed for the Interactive SED task shown in Fig. 4, allows the user to detect visual events that would be important for airport security management. The user selects the event of interest from the drop-down list placed at the upper left corner of the screen and the system instantly returns a list of video segments that are more possible to meet the event's definition. The interface makes use of the HTML5 *video* tag and its feature that allows playing a certain video segment. When the user hovers a video appeared in the results returned, the specific video segment is being looped. Thus, the user can easily evaluate if the particular video segment belongs to the event that needs to be detected. Two options are given for each video during the manual annotation, an option for indicating the right detections (check symbol at the upper right corner of each video segment) and a second one for noting the false alarms (x symbol at the upper left corner of each video segment). When the user finishes with the evaluation, he submits the results he previously marked as correct detections by clicking on the corresponding button (upper right corner of the screen).

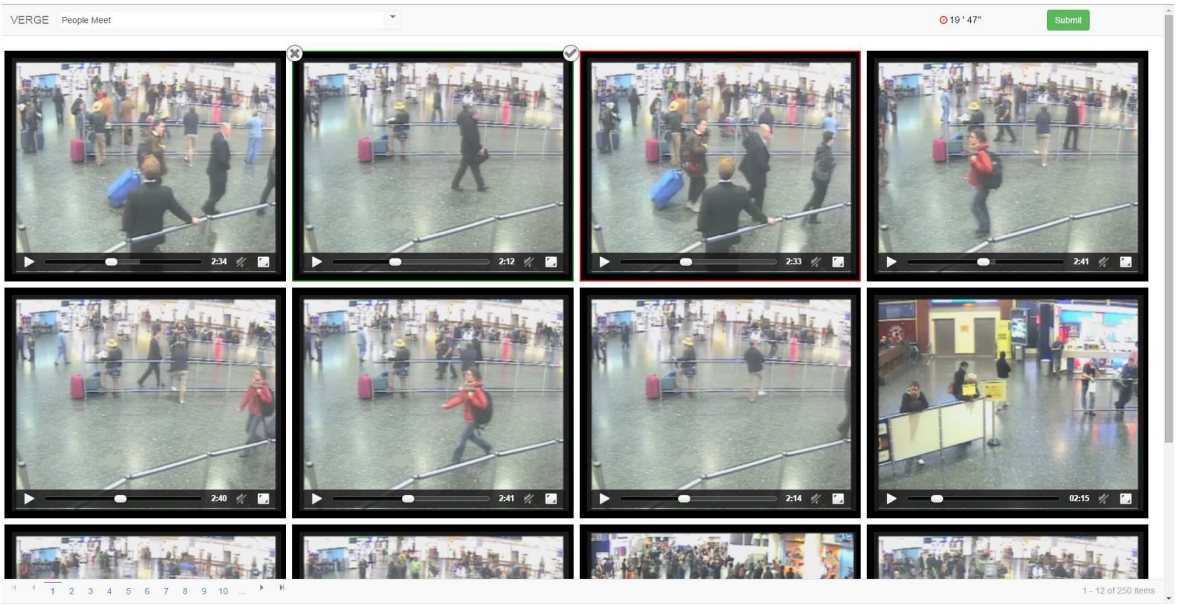

Figure 4: User interface for the SED task.

5.2.1 Surveillance Event Detection System

A generic event detection system targeting on a subset of 5 events of interest, i.e. PersonRuns, PeopleMeet, PeopleSplitUp, Embrace and Pointing, was designed. The events involving a single person plus an object (i.e. CellToEar and ObjectPut) are considered more challenging and we intend to incorporate them to our system in a future version.

Our interactive surveillance event detection system is based on the following three steps [30]:

1. Low-level feature extraction, which is performed by using Motion Boundary Activity Areas (MBAA) to sample dense trajectories and represented with HOG/HOF descriptors.
2. Applying Gaussian Mixture Model (GMM) on the computed training descriptors, as an intermediate representation level, for the construction of a thorough visual vocabulary and Fisher encoding to represent each activity.
3. High level representation with the use of linear SVMs for event learning and classification.

Our activity detection system entails two separate stages. One offline that is used to build a discriminative visual vocabulary and linear SVM models from the training videos and one online that uses our detection algorithm to localize in a spatio-temporal manner the desired activities inside the test videos.

The first offline/training stage, uses Motion Boundary Activity Areas in the videos of the training set in order to sample trajectory points inside them and build dense trajectories [31] over them. HOGHOF [32] descriptors are then extracted around the trajectory points in order to capture appearance and motion information and concatenated in a common spatio-temporal descriptor in order to form the action descriptor. Trajectory coordinates are also concatenated to the vector to include global spatial information in the final descriptor, as proposed in [30]. A visual vocabulary is subsequently constructed by using a Gaussian Mixture Model (GMM) with 64 clusters and Fisher vector encoding framework is deployed in order to characterize each video segment. Finally, the 5 considered event models are learned by 5 linear SVMs. Training data from cameras 1,2,3 and 5 were used in this offline stage to build the event models (the videos of CAM4 were discarded since they contain a very limited number of events).

In the online/testing stage, a 50-frame sliding window traverses every video sequence of the test set with a temporal step of 15 frames in order to process all video content and detect the specified events of interest.

The overall process is depicted in Fig. 5.

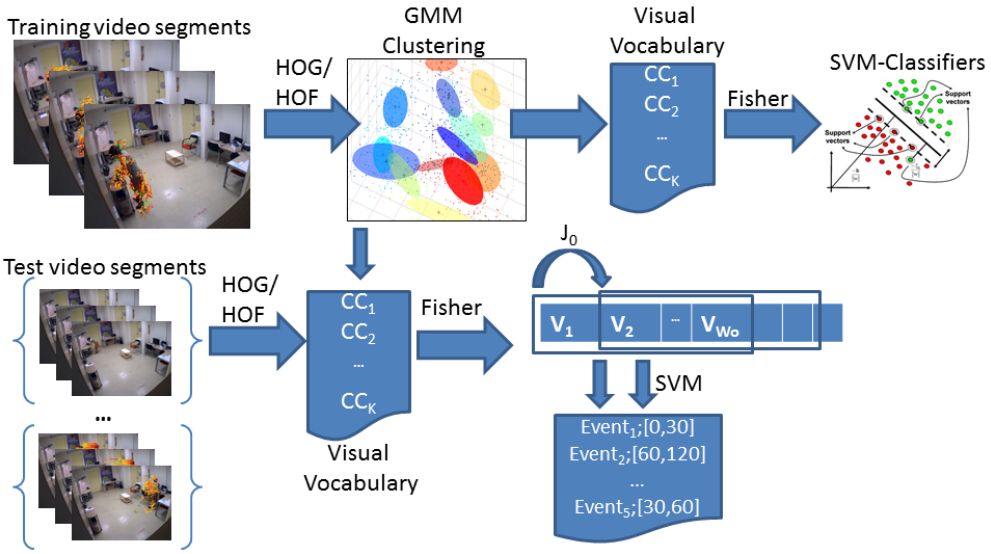

Figure 5: Block diagram of the surveillance event detection system.

5.3 Interactive Surveillance Event Detection (iSED) Results

As already reported, we submitted one run to the iSED 2015 task. Two users participated in the official run which was performed on a 64-bit Windows PC with Intel Core i7 3.50 GHz and 32 GB RAM. The performance of our generic system is reported in Table 5. As it can be seen from the results, we achieved the 2nd best performance in three out of five events, i.e. PeopleMeet, PersonRuns and Pointing. Even though the efficiency of our system is not optimal yet, we believe that it has the potential to develop in order to become more competitive.

Table 5: The Actual DCR and Minimum DCR of the 2015 interactive result.

Event	Rank	ADCR	MDCR	#CorDet	#FA	#Miss
Embrace	3	0.9855	0.9855	2	0	136
PeopleMeet	2	0.9990	0.9984	1	5	255
PeopleSplitUp	3	0.9868	0.9868	2	0	150
PersonRuns	2	0.9834	0.9823	1	6	49
Pointing	2	1.0054	1.0006	3	16	791

6 Conclusions

In this paper we reported the ITI-CERTH framework for the TRECVID 2015 evaluation [8]. ITI-CERTH participated in the SIN, MED, INS and SED tasks in order to evaluate new techniques and algorithms. Regarding the SIN task, the combination of visual DCNN-based descriptors with local descriptors significantly improved our TRECVID 2014 SIN results. Furthermore, Multi-task learning seems to be an important direction that can further improve concept detection accuracy. Concerning the MED task, a new algorithm, combining discriminant analysis and LSVMs (KSDA+LSVM) was evaluated, providing very good performance in terms of both accuracy and training time. Our run based on KSDA+LSVM that uses all the features presented in Section 3.2.2 (run c-1KDALSVM) achieved mInfAP@200 equal to 0.4111, which is the second-best result among all participants' runs validated on the MED15–EvalSub set. As far as INS task is concerned, the results reported were significantly better than last year results but there is still a lot of room for improvement in order for the system to become competitive against the other systems. The most important conclusions from this year runs was that the BoW model is more suitable for the INS task than VLAD encoding and

that saliency detection can be exploited in order to boost the retrieval performance. Finally, with respect to the SED task, despite the fact that this is our first participation to the task, we achieved satisfactory results which inspire us to continue evolving our system.

7 Acknowledgements

This work was partially supported by the European Commission under contracts, FP7-600826 ForgetIT, FP7-610411 MULTISENSOR and FP7-312388 HOMER.

References

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

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

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

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

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

[6] F. Markatopoulou, A. Moumtzidou, and C. Tzelepis et al. ITI-CERTH participation to TRECVID 2013. In *TRECVID 2013 Workshop*, Gaithersburg, MD, USA, 2013.

[7] N. Gkalelis, F. Markatopoulou, and A. Moumtzidou et al. ITI-CERTH participation to TRECVID 2014. In *TRECVID 2014 Workshop*, Gaithersburg, MD, USA, 2012.

[8] O. Paul, A. George, and M. et al. Martial. Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA, 2015.

[9] 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, Jan 2013.

[10] 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.

[11] S. Arestis-Chartampilas, N. Gkalelis, and V. Mezaris. Gpu accelerated generalised subclass discriminant analysis for event and concept detection in video. In *ACM Multimedia 2015*, Brisbane, Australia, 2015.

[12] F. Markatopoulou, V. Mezaris, and I. Patras. Ordering of visual descriptors in a classifier cascade towards improved video concept detection. In *MultiMedia Modeling Conf. (MMM 2016)*, Florida, Jan 2016. Springer.

[13] F. Markatopoulou, V. Mezaris, and I. Patras. Cascade of classifiers based on binary, non-binary and deep convolutional network descriptors for video concept detection. In *IEEE Int. Conf. on Image Processing (ICIP 2015)*, Canada, 2015. IEEE.

[14] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University, 2011.

[15] F. Markatopoulou, V. Mezaris, N. Pittaras, and I. Patras. Local features and a two-layer stacking architecture for semantic concept detection in video. *IEEE Trans. on Emerging Topics in Computing,*, 3(2):193–204, 2015.

[16] 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.

[17] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 245–250, NY, 2001. ACM.

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

[19] C. Szegedy and et al. Going deeper with convolutions. In *CVPR 2015*, 2015.

[20] A. Krizhevsky, S. Ilya, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, Series B:267288, 1996.

[22] C. Tzelepis, D. Galanopoulos, V. Mezaris, and I. Patras. Learning to detect video events from zero or very few video examples. *Image and Vision Computing Journal, Elsevier, accepted for publication.*, 2015.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, pages 248–255. IEEE, 2009.

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

[25] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[26] C. Tzelepis, N. Gkalelis, V. Mezaris, and I. Kompatsiaris. Improving event detection using related videos and relevance degree support vector machines. In *Proceedings of the 21st ACM Int. Conf. on Multimedia*, pages 673–676. ACM, 2013.

[27] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*, 2000.

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

[29] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 1155–1162, 2013.

[30] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris. Activity detection using sequential statistical boundary detection (ssbd). *To appear in Computer Vision and Image Understanding (CVIU).*

[31] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176, 2011.

[32] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*, pages 1–8, 2008.