ITI-CERTH participation in TRECVID 2016

Foteini Markatopoulou[1,2], Anastasia Moumtzidou[1], Damianos Galanopoulos[1], Theodoros Mironidis[1], Vagia Kaltsa[1], Anastasia Ioannidou[1], Spyridon Symeonidis[1], Konstantinos Avgerinakis[1], Stelios Andreadis[1], Ilias Gialampoukidis[1], Stefanos Vrochidis[1], Alexia Briassouli[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, moumtzid, dgalanop, mironidis, vagiakal, ioananas, spyridons, koafgeri, andreadisst, heliasgj, stefanos, abria, 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 2016 by ITI-CERTH. ITI-CERTH participated in the Ad-hoc Video Search (AVS), Multimedia Event Detection (MED), Instance Search (INS) and Surveillance Event Detection (SED) tasks. Our AVS task participation is based on a method that combines the linguistic analysis of the query and the concept-based annotation of video fragments. In the MED task, in 000Ex task we exploit the textual description of an event class in order retrieve related videos, without using positive samples. Furthermore, in 010Ex and 100Ex tasks, a kernel sub class version of our discriminant analysis method (KSDA) combined with a fast linear SVM is employed. The INS task is performed by employing VERGE, which is an interactive retrieval application that integrates retrieval functionalities that consider only visual information. For the SED task, we deploy a novel activity detection algorithm that is based on Motion Boundary Activity Areas (MBAA), dense trajectories, Fisher vectors and an overlapping sliding window.

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, in 2010 and 2011 in the KIS, INS, SIN and MED tasks and in 2012, 2013, 2014 and 2015 in the INS, SIN, MED and MER tasks ([2], [3], [4], [5]) 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 them. This year, ITI-CERTH participated in four tasks: AVS, 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 Ad-hoc Video Search

2.1 Objective of the Submission

The goal in the TRECVID 2016 AVS task [6] is the development of suitable techniques to retrieve for each ad-hoc query a ranked list of 1000 test shots that are mostly related with it. The ITI-CERTH participation in the AVS task was based on representing each query as a vector of related concepts. Specifically, a sequence of steps was followed starting with the ad-hoc query and transforming it to a vector of concepts. In addition, each video shot was annotated with semantic concepts using deep learning, which resulted to another vector representation that corresponds to the concepts that are depicted in the video shot. Finally, given a test query the query's concept vector was compared with the shot's concept vector and the 1000 video shots with the smallest distance from the query's concept vector were retrieved. Our aim was to investigate the way that each of the steps of constructing the query's concept vector reacts to the final retrieval accuracy.

2.2 System Overview

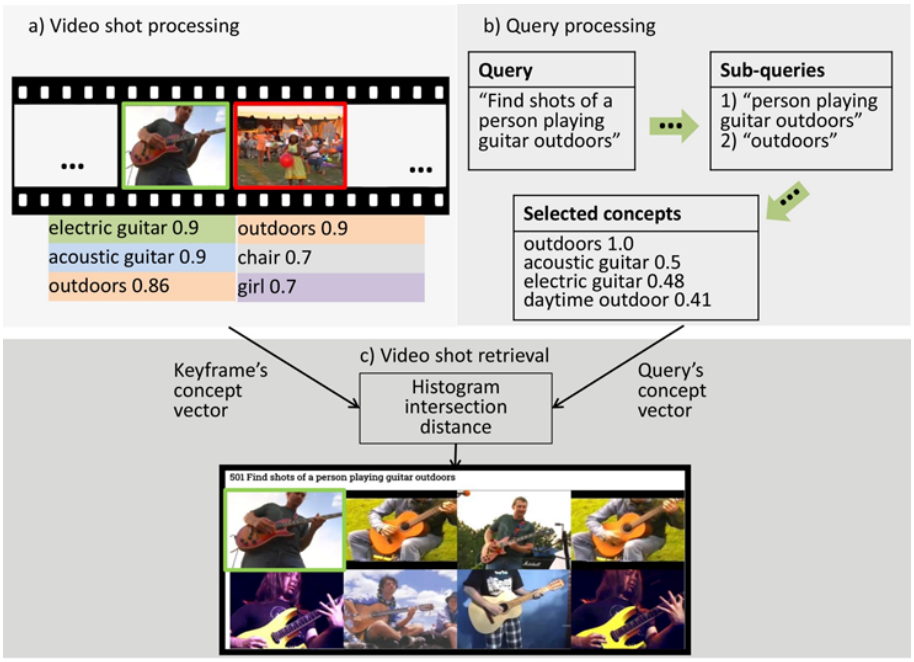

Figure 1: Developed AVS system.

The system we developed for the AVS task consists of three components, as presented in Fig. 1:

2.2.1 Concept-based Video Shot Annotation

The first component of our system annotates each video shot with concepts from a predefined concept pool. The output of this component is one vector for each TRECVID AVS test video shot that indicates the probability that each of the concepts in the pool appears in the video shot. Specifically, one keyframe was extracted from each video shot of the TRECVID AVS test set and annotated based on 1000 ImageNet [7] and 345 TRECVID SIN [8] concepts (i.e., all the available TRECVID SIN concepts, except for one which was discarded because only 5 positive samples are provided for it).

To obtain scores regarding the 1000 ImageNet concepts, we applied five pre-trained ImageNet deep convolutional neural networks (DCNNs) on the AVS test keyframes: i) AlexNet [9], ii) GoogLeNet [10], iii) ResNet [11], iv) VGG Net [12] and v) a DCNN that we trained according to the 22-layer GoogLeNet architecture on the ImageNet "fall" 2011 dataset for 5055 categories (where we only

considered in AVS the subset of 1000 concepts out of the 5055 ones). The output of these networks was averaged in terms of arithmetic mean to obtain a single score for each of the 1000 concepts.

To obtain the scores regarding the 345 TRECVID SIN concepts we fine-tuned (FT) three of the above pre-trained ImageNet networks, (AlexNet, GoogLeNet and GoogLeNet trained on 5055 categories) on the 345 TRECVID SIN concepts using the TRECVID AVS development dataset [6]. We experimented with many FT strategies and finally selected the single best performing FT network. The complete set of these experiments can be found in [13]. To annotate the AVS test keyframes with the 345 concepts we evaluated two different approaches: i) The direct output of the FT network, and ii) Support Vector Machines (SVM) trained on DCNN-based features separately for each concept. Specifically, in the first case the TRECVID AVS test keyframes were forward propagated by the network and its output was used to annotate each keyframe. In the second case we applied the FT network on the TRECVID AVS development dataset and we used as a feature (i.e., a global keyframe representation) the output of the last hidden layer to train one SVM per concept. Subsequently, we applied this FT network on the TRECVID AVS test keyframes to extract features, and served them as input to the trained SVM classifiers in order to gather scores for each of the 345 concepts. In all cases, the final step of the concept-based video shot annotation was to refine the calculated detection scores by employing the re-ranking method proposed in [14].

The scores obtained from the two pools of concepts (1000 ImageNet, and 345 TRECVID SIN) were concatenated in a single vector. Consequently, a 1345-element concept vector was created for each test keyframe. Each element of this vector corresponds to one concept, from the 1345 available concepts, and indicates the probability that this concept appears in the video shot.

2.2.2 Linguistic Analysis of the Query

The second component of our system represents each query as a vector of related concepts. Given the above pool of 1345 concepts and the textual description of the query our method identifies those concepts that most closely relate to the query. Specifically, the selected concepts form a vector where each element of this vector indicates the degree that each concept is related to the query. To calculate this concept vector a sequence of steps was followed as presented below.

- Step one: The first step uses the complete textual description of the query to examine if one or more concepts in the concept pool can describe the query very well. The "semantic relatedness" between the query and each of the concepts in the concept pool is calculated by the Explicit Semantic Analysis (ESA) measure (a real number in the [0,1] interval) of [15]. If the score between a query and a concept is higher than a threshold then the concept is selected. For example, the query "a policeman where a police car is visible", and the concept "police car" are semantically close as the ESA measure returns a high value. If at least one concept is selected in this way, we assume that the query is very well described and the query processing stops; otherwise the query processing continues in step two.

- Step two: This step searches if any of the concepts in the pool appears in any part of the test query. Some (complex) concepts may describe the query quite well, but appear in a way that is difficult to detect them due to the subsequent linguistic analysis which is break down the query to sub-queries. So, in this step we search if any of the concepts appear in any part of the query, by string matching. Any concept that appears in the query is selected and the query processing continues in step three.

- Step three: Queries are complex sentences; this step decomposes queries to understand and process better their parts. Specifically, the test query is automatically transformed into a set of elementary *sub-queries*; then, each of the *sub-queries* is processed and translated to concept vectors. We define a *sub-query* as a meaningful smaller phrase or term that is included in the original query, and we automatically decompose the query to sub-queries. For example, the query "Find shots of one or more people at train station platform" is split into the following four sub-queries: "people", "train station platform", "persons" and "train station". To infer sub-queries, conventional natural language processing procedures (NLP), e.g., part-of-speech tagging, stop-word removal etc., are used, together with a task-specific set of NLP rules. For example, if the original query contains a sequence in the form of "Noun - Verb - Noun", this

triad is considered to be a sub-query. The motivation is that such a sequence is much more characteristic of the original query than any of the single terms alone (e.g., considering each of the three terms as a different sub-query).

Then, the ESA measure is calculated between each sub-query and each of the concepts in the pool. If the score between a sub-query and a concept is higher than a threshold then the concept is selected. In the case that for all of the sub-queries at least one concept has been selected, we assume that the query has been very well analysed and we proceed to step six. If for a subset of the sub-queries no concepts have been selected then these sub-queries are propagated to step four. Finally, if for all of the sub-queries no concepts have been selected then the test query and all of the sub-queries are propagated to step five.

- Step four: For a subset of the sub-queries no concepts were selected due to their small semantic relatedness (i.e., in terms of ESA measure their relatedness is lower than the utilised threshold). For these sub-queries the concept with the higher value of ESA measure is selected, and then we proceed to step six.

- Step five: For some queries neither step three nor step four are able to select concepts. In this case, the original query and the sub-queries are served as input to the zero-example event detection pipeline of [16], which returns a ranked list of the most relevant concepts in accordance with a relatedness score again based on the ESA measure. Then, we proceed to video shot retrieval.

Finally, the query's concept vector is formed by the corresponding scores of the selected concepts. If a concept has been selected in steps 1, 3, 4 or 5 the corresponding vector's element is assigned with the relatedness score (calculated using the ESA measure) and if it has been selected in step 2 it is set equal to 1. It should be noted that every time we use a threshold in order to take a decision regarding selecting a concept or not this threshold equals to 0.8.

2.2.3 Video Shot Retrieval

The third component of our system retrieves for each query the 1000 test shots that are mostly related with it. Specifically, the distance between the query's concept vector (Section 2.2.2) and the keyframe's concept vector(Section 2.2.1) for each of the test AVS keyframes is calculated and the 1000 keyframes with the smallest distance from query's concept vector are retrieved. In our experiments the histogram intersection distance was used.

2.3 Description of Runs

Four AVS runs were submitted in order to evaluate the potential of the aforementioned approaches on the TRECVID 2016 AVS dataset [6]. The submitted runs are briefly described below:

- ITI-CERTH 1: Complete pipeline using five pre-trained ImageNet networks for annotating the test keyframes with 1000 ImageNet concepts and SVM-based concept detectors for annotating the test keyframes with 345 TRECVID SIN concepts, including linguistic analysis of the query, multiple steps of matching the queries and sub-queries inferred from it with concepts from our concept detectors' pool, and a histogram intersection distance for matching the keyframe's concept vector with the query's concept vector.

- ITI-CERTH 2: This run is a variation of ITI-CERTH 1 run that uses the direct output of the FT network in order to annotate the test keyframes with the 345 TRECVID SIN concepts, instead of using SVM-based concept detectors. It should be noted that we evaluated our concept detectors in the TRECVID SIN 2013 test set and we found that using the SVM-based classifiers of ITI-CERTH 1 run outperforms the direct output of the FT network. Furthermore, in one of our previous works [17] we found that SVM-based detectors have different strengths when they are used on the video annotation and on the video retrieval problem. In this run we want to examine if SVM detectors trained on DCNN-based features for the 345 SIN concepts will make a difference compared to the direct output of a DCNN network when used for the ad-hoc video search problem.

- ITI-CERTH 3: This run is a subset of ITI-CERTH 1 run that ignores step four (Section 2.2.2). I.e., if at least one of the sub-queries presents high semantic relatedness with one or more concepts in our pool, then these concepts are selected. However, the rest of the sub-queries that do not present semantic relatedness with any of the concepts in the pool are ignored.

- ITI-CERTH 4: This run is a subset of ITI-CERTH 1 run that excludes step two (Section 2.2.2).

2.4 Ad-hoc Video Search Task Results

Table 1: Mean Extended Inferred Average Precision (MXinfAP) for all submitted runs both for fully-automatic and manually-assisted runs of the AVS task.

Submitted run:	ITI-CERTH 1	ITI-CERTH 2	ITI-CERTH 3	ITI-CERTH 4
MXinfAP (fully-automatic)	**0.051**	0.042	**0.051**	**0.051**
MXinfAP (manually-assisted)	0.043	0.037	0.037	0.043

Table 1 summarizes the evaluation results of the aforementioned runs in terms of the Mean Extended Inferred Average Precision (MXinfAP). Our team submitted both fully-automatic and manually-assisted runs. In the latter case a member of our team that was not involved in the development of our AVS system took a look at each query and manually suggested sub-queries for it, without knowledge of the automatically-generated ones. The manually defined sub-queries were added to the automatically-generated ones, and our automatic AVS system was applied.

According to table 1 we conclude as follows:

- Improving the accuracy of the individual concept detectors by training SVMs on DCNN-based features separately for each concept (ITI-CERTH 1 run) instead of just using DCNNs as standalone classifiers (ITI-CERTH 2 run), i.e., using the direct output of the FT network, also improves the overall accuracy of our ad-hoc video search system. I.e., ITI-CERTH 1 run outperforms ITI-CERTH 2 run both regarding the fully-automatic and the manually-assisted runs.

- Sub-queries that do not present high semantic relatedness with any of the concepts can be ignored when for at least one sub-query one or more concepts have been selected. I.e., ITI-CERTH 1 and ITI-CERTH 3 runs present the same accuracy. However, this does not hold in the manually-assisted runs where ignoring the former sub-queries slightly reduces the retrieval accuracy. This indicates that some of the sub-queries that the user has introduced into the system could be useful even if they were not detected to have high semantic relatedness with any of the concepts in the pool.

- String match between the test query and each of the concepts does not improve the system's retrieval accuracy indicating that the rest of the steps that are based only on the semantic relatedness of the query and sub-queries with the concepts are adequate to create a descriptive query concept vector. I.e., ITI-CERTH 1 and ITI-CERTH 4 runs present the same accuracy both with respect to the fully-automatic and the manually-assisted runs.

- Fully-automatic runs always outperform the manually-assisted runs. We discovered that the user often defines sub-queries that are too general, e.g., given the query "Find shots of a man indoors looking at camera where a bookcase is behind him", the sub-queries returned by the user were as follows: "man, male, indoors, looking at camera, man looking at camera, bookcase". Terms like "man", "male" and "indoors" constitute over-simplifications of the query and the retrieval accuracy was reduced when these terms were fed to the system.

- Overall, our fully automatic runs performed very well in this challenging task, compared to the runs of the other participating. Specifically, our best run was ranked 2nd-best, achieving an MXinfAP of 0.051 (compared to 0.054 reached by the best-performing participant in the fully-automatic category, and 0.040 reached by the 3rd best-performing one).

- Our fully-automatic runs also compared favorably to the manually-assisted runs that were submitted to AVS: with an MXinfAP of 0.051, our best fully automatic run also outperformed the runs of all but one participant in the manually-assisted run category.

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).

3.2 System Overview

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

In the 000Ex task we use a modification of our MED15 zero-example event detection framework [5], also presented in [16], along with an enriched concept pool of 14525 concepts, compared to our previous submission that uses 1000 concepts. 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 MED16–EvalSub video collection, we use a) the pool of 14525 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) pre-trained concept detectors for these concepts in order to annotate each video in the MED16–EvalSub with semantic labels (i.e, a concept vector that indicates the probability that each concept appears in the video).

We used five different concept sets so as to construct our overall concept pool: i) 12988 concepts from the ImageNet "fall" 2011 dataset [7], ii) 345 concepts from the TRECVID SIN dataset [8] (i.e., all the available TRECVID SIN concepts, except for one which was discarded because only 5 positive samples are provided for it), iii) 500 event-related concepts from the EventNet dataset [18], iv) 487 sport-related concepts from the Sports-1M dataset [19] and v) 205 place-related concepts from the Places dataset [20].

Given the textual description of an event, our framework first identifies N words or phrases which are the most closely relate to the event; this word-set is called Event Language Model (ELM). The ELM is based on the automatic extraction of word terms from the visual and audio cues of the event kit along with the title of the event. In parallel, for each of the 14525 concepts of our concept pool, our framework similarly identifies M words or phrases: the Concept Language Model (CLM) of the corresponding concept using the top-20 articles in Wikipedia and 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) similarity [15] 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 the Hausdorff distance. Consequently, a score is computed for each pair of ELM and CLM. The 14525 considered concepts are ordered according to these scores (in descending order) and the K-top concepts along with their scores constitute our event detector.

In contrast with last year's submission where the K was a fixed number, this year it is computed in a different way. First, we check if the event title is semantically close to any of the available concepts from the concept pool. If so, these concepts are used as the event detector. If this is not the case, we determine the value of K by ordering the scores in descending order, constructing an exponential curve, and then we select the first K concepts so that corresponding area under the curve is at the 10% of the total area under the curve.

Subsequently, the histogram intersection distance is computed between the event detector and the corresponding concept vector of each video of the MED16–EvalSub collection.

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.

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

In our 010Ex and 100Ex submissions, for building our event detectors, firstly, we utilized an extended and speeded-up version of our Kernel Subclass Discriminant Analysis [21, 22] for dimensionality reduction and after that we used a fast linear SVM (KSDA+LSVM). The GPU-accelerated implementation of this method [23] was not used in our MED 2016 experiments due to the limited number of training samples.

Specifically, two types of visual information have been used for training the event detectors: motion features and DCNN-based features. We briefly describe the different visual modalities in the following:

- Each video is decoded into a set of keyframes at fixed temporal intervals (approximately 2 keyframes per second).We annotated the video frames based on 12988 ImageNet [7] concepts, 345 TRECVID SIN [8] concepts, 500 event-related concepts [18], 487 sport-related concepts [19] and 205 place-related concepts [20]. To obtain scores regarding the 12988 ImageNet concepts we used the pre-trained GoogLeNet provided by [10]. We also experimented with a subset of the 12988 concepts; in order to do that we self-trained a GoogLeNet network [24] on 5055 ImageNet concepts (gnet5k). To obtain the scores regarding the 345 TRECVID SIN concepts and the 487 sport-related concepts we fine-tuned (FT) the gnet5k network on the TRECVID AVS development dataset and on the YouTube Sports-1M dataset [19], respectively. We also used the EventNet [18] that consists of 500 events and the Places205-GoogLeNet, which was trained on 205 scene categories of Places Database [20]. All the above networks were also used as feature generators. I.e., the output of one or more hidden layers was used as a global frame representation.
- 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 (motion, model vectors), yielding a new feature vector in $\mathbb{R}^{153781}$.

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 MED16–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 concept pools as well as an extension of the pipeline with an extra online training step based on the top retrieved videos per event. All the submitted runs are fully automatic. We submitted 4 different runs in the 000Ex task, one primary and three contrastive:

- **p-1DCNN13K_1**: In our primary run we use the annotation from 2 different DCNNs: i) The pre-trained GoogLeNet provided by [24] trained on 12988 ImageNet concepts [7] and ii) the EventNet [18] consisting of 500 events.
- **c-1DCNN14K_1**: In this run, we use the annotation from 5 different DCNNs: i) The pre-trained GoogLeNet provided by [24] trained on 12988 ImageNet concepts [7], ii) the GoogLeNet [10] self-trained on 5055 ImageNet concepts (gnet5k) and subsequently fine-tuned for 345 TRECVID SIN [8] concepts, iii) the gnet5k network fine-tuned for 487 sport-related [19] concepts, iv) the EventNet [18] consisting of 500 events and v) the Places205-GoogLeNet, trained on 205 scene categories [20].
- **c-1DCNN05K_1**: In the third run, we used the annotation from 2 different DCNNs: i) The GoogLeNet [10] self-trained on 5055 ImageNet concepts (gnet5k) and ii) the EventNet [18] consisting of 500 events.
- **c-3Train_1**: In the last run an online training stage is utilized using the top-10 retrieved videos from the primary run as positive samples, and the learning procedure from 3.2.2.

3.4.2 010Ex & 100Ex

For each of the training conditions of 010Ex and 100Ex, we submitted 1 primary run:

- **c-1KDALSVM**: In the primary run, the KSDA+LSVM method is used to build the event detectors and perform the event search in the MED16–EvalSub set using motion and DCNN-based features, as discussed in Section 3.2.2.

3.5 Multimedia Event Detection Results

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

Table 3: MAP and InfAP@200 for all submitted runs of the MED task

(a) 000Ex

Run ID	MAP%	mInfAP@200%
p-1DCNN13K_1	14.6	12.2
c-1DCNN14K_1	14.5	11.9
c-2DCNN05K_1	2.4	1.4
c-3Train_1	**16.2**	**14.2**

(b) 010Ex & 100Ex

Run ID	MAP%	mInfAP@200%
010Ex p-1KDALSVM	31.8	34.2
100Ex p-1KDALSVM	46.2	47.5

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 visual concepts, gives a boost to our results compared to our last year's submission. Furthermore, adding the training step by using the top retrieved videos as positive samples has a significant impact to our performance (the relative improvement is 16.39%).
- Regarding the c-1DCNN05K_1 run a bug has been detected, so it is not safe to reach a conclusion for this run.
- Concerning the 010Ex and 100Ex training conditions, we observe that the increase of the number of DCNN based-features makes our KSDA+LSVM method (Section 3.2.2) performing better compared to our previous submission. Specifically, mInfAP@200=0.475 which is the fourth-best result among all participants validated on the MED16–EvalSub set.

4 Instance Search

4.1 Objective of the Submission

According to the TRECVID guidelines, the instance search (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. In order to begin with the search, the user is provided with visual examples of the specific query object. The collection of videos used in the INS task are provided by BBC and they are part of the EastEnders TV series (Programme material BBC).

It should be noted that this year's INS task has focused exclusively on the retrieval of specific persons in specific locations. Thus, given the narrow scope of the INS task, the general object retrieval approaches used in the previous years, can't be applied. Therefore, ITI-CERTH participated in the TRECVID 2016 INS task by submitting a single run that incorporated an algorithm for face detection and retrieval and an algorithm for landscape retrieval. The system and algorithms developed are integrated in VERGE[1] interactive video search engine.

4.2 System Overview

The system employed for the INS was VERGE (2), an interactive retrieval application that combines various search functionalities, considering visual information. The existence of a friendly and efficient graphical user interface (GUI) plays a vital role in the procedure of searching, so VERGE is designed according to these specifications, while integrates the following search modalities:

- Face detection and Face Retrieval Module;
- High Level Visual Concept Retrieval;
- Visual Similarity Search module

Describing the GUI from the top, there is a toolbar that offers a multitude of useful options. In detail, from left to right, a burger icon opens a toggle menu that contains three different search capabilities, namely the Concept- and Topic-based search, and the Clustering. The menu also includes the users selected shots and the complete set of video shots. Next to the application's logo, the user can find a text input field that searches in natural language descriptions of shots, if provided by the dataset, and a slider that modifies the amount of results in the viewport by adjusting the size of the shots. The last toolbar component applies only to the contest and shows the remaining time for the submission, accompanied by an animated red line on the top of the screen. The central component of the interface includes a shot-based representation of the video results in a grid-like view. Clicking on a shot allows the user to navigate through the whole scene where this frame belongs, displaying the related shots in a chronological order. Moreover, each shot supports tools to run the Visual Similarity and the Face Similarity search. Finally, all selected images are saved in a deposit that can be quickly accessed for further searching or just for the submission. It should be noted that the VERGE application is built on open-source Web technologies, such as PHP, HTML5, JavaScript, and the MongoDB database program.

To illustrate the functionality of VERGE, Figures 2 and 3 can serve as two alternative search examples; the first screenshot depicts the results based on Face Similarity, while the second shows the shots that are relative to the "restaurant" landscape, using the Concept-based search. In both scenarios, the user can continue with other retrieval modules, e.g. the Visual Similarity, or the scene navigation.

4.2.1 Face Detection and Face Retrieval Module

This module involves the following two sub-modules: 1) the face detection sub-module that identifies human faces in images, and 2) the face retrieval sub-module that captures the face features from the faces recognized in the first step.

Regarding the face detection module that involves the retrieval of video frames depicting human faces, several algorithms were tested and evaluated on limited dataset that contained 10000 images

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

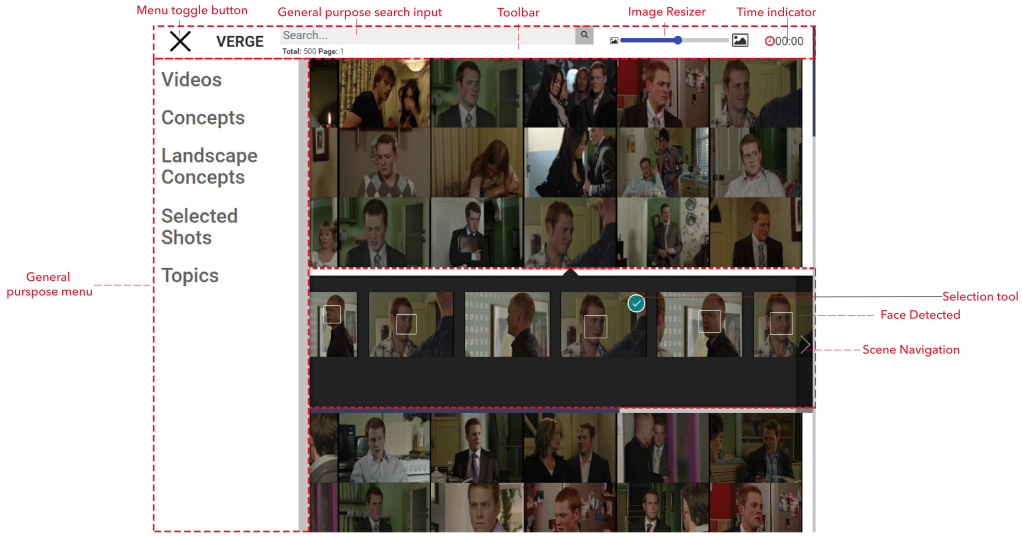

Figure 2: VERGE, a video retrieval application.

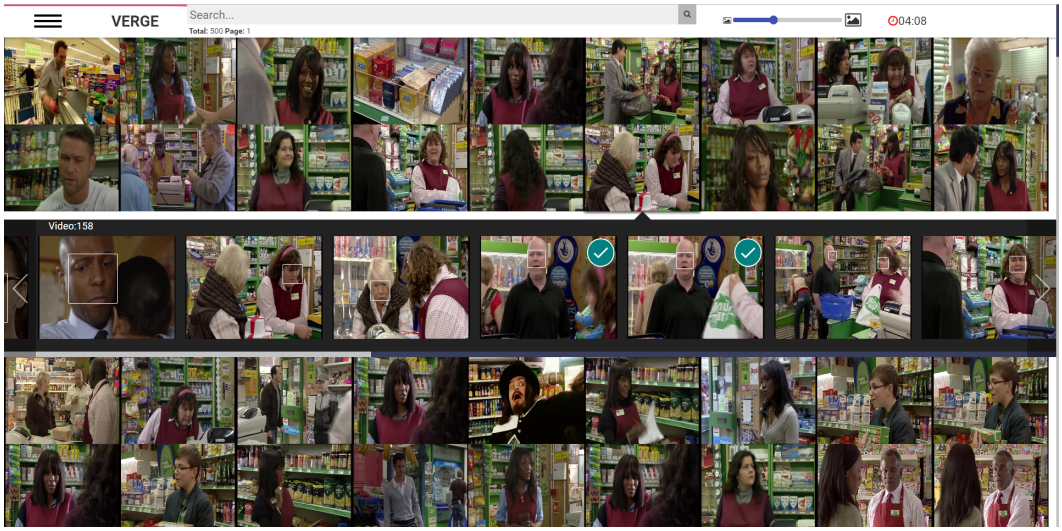

Figure 3: Screenshot of VERGE, with results of Concept-based search.

which were manually annotated. The first algorithm refers to a state-of-the-art method that includes two steps: i) the extraction of HOG features from an image following a sliding window approach in multiple scales, and ii) classification using a linear SVM. This scheme was originally proposed for pedestrian detection [26], but later was also applied for face detection with great success. Another approach that was tested is the Viola-Jones object detection algorithm, which is a machine learning framework proposed in 2001 for fast discovery of objects [27]. It involves training a cascade of classifiers using Haar-like features. Even though it can be used to detect any type of object, the main motivation for developing it was to detect faces. We tested the pre-trained classifiers for frontal and profile face detection provided by OpenCV, using OpenCV-Python library. Finally, we tested the face detector provided by Sun et al. [28], which is part of their system for detecting facial landmarks using a cascade of deep convolutional neural networks (CNNs). The tool accepts as input images (or video frames), processes them sequentially and returns the coordinates of all bounding boxes containing the detected face(s) per image. The evaluation results of face detection algorithms are presented in Table 4. Based on the results, it is evident that the CNN algorithm outperforms the other two, although it is slightly slower. However, the processing time is not considered important for the evaluation of face detection algorithms given that the procedure is realized off line.

Regarding the face retrieval module, two descriptors were tested. The first is the Local Binary Patterns (LBP), which is a texture-based visual descriptor proposed by [29]. LBP is a gray scale invariant feature vector and it depends on local representations of texture. To calculate the LBP vector, we first calculate a value for each pixel based on the intensity of its neighboring pixels. After concentrating the values for all the pixels of the image, we compute the histogram of them to form the final vector. In this work, we extracted vectors by applying an extension of LBPs proposed by Ojala [30]. The major differences of this variant are that the final vector mainly takes into account the uniform patterns of an image, and that the vector is not only gray scale invariant but also rotation invariant. The library used for calculating these vectors were the scikit-image Python library. The other approach that was tested were the VGG-Face CNN descriptors which were computed using the VGG-Very-Deep-16 CNN architecture described in [31]. The difference between "very deep" networks and deep networks is that they comprise a long sequence of convolutional layers. Specifically, the CNN architecture comprises 11 blocks, each containing a linear operator followed by one or more non-linearities such as ReLU and max pooling. The first eight such blocks are said to be convolutional, while the last three blocks are called Fully Connected (FC). All the convolution layers are followed by a rectification layer. As feature vector, we considered the last FC layer with size 2622. The face retrieval algorithms were evaluated visually by submitting query faces to the VERGE system and observing the retrieved results. The results showed that the CNN-based algorithm outperforms significantly the LBP algorithm and thus it was selected over the other.

Eventually, the face features were used for constructing an IVFADC index similar to the one created in 4.2.2 that allows fast face retrieval.

4.2.2 Visual Similarity Search Module

The visual similarity search module performs content-based retrieval using deep convolutional neural networks (DCNNs). Specifically, we have trained GoogleNet [10] on 5055 ImageNet concepts. Then, the output of the last pooling layer, with dimension 1024, was used as a global keyframe representation. In order to achieve fast retrieval of similar images, we constructed an IVFADC index for database vectors and then computed K-Nearest Neighbours from the query file. Search is realized by combining an inverted file system with the Asymmetric Distance Computation [32].

4.2.3 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 2015 SIN task using the techniques and the algorithms described in detail in [5] Section 2 (Semantic Indexing).

Apart from the 346 TRECVID concepts, a set of 205 scene categories using the GoogLeNet CNN network was used for scene/ landscape recognition [20].

4.3 Instance Search Task Results

We submitted a single run (I_A_ITI_CERTH_1) to the interactive INS task, that utilized the aforementioned algorithms. According to the TRECVID guidelines, the number of topics were 20 and the time duration for the run was set to five minutes. Table 5 contains the mean average precision as well

Table 4: Evaluation of face detection algorithms.

Metrics \ Algorithms	HOG	Viola-Jones	CNN
Precision	0.910	0.977	0.873
Recall	0.471	0.723	0.841
F-score	0.621	0.831	0.856
Execution time (per image)	62ms	85ms	540ms

the recall for the submitted run along with the results from our last year's participation. Compared to last year's results, the results obtained this year are improved, although a direct comparison is not possible due to the different aim of the INS task (object detection vs face detection). However, it should be noted that there is a lot of room for improvement given that the efficiency of our system is still low compared to the other competing systems. We consider as the main reason for this difference the fact the VERGE system is user-driven, meaning that all searches are initiated by the user. Also, we assume that the system would benefit from a fusion between face retrieval and landscape concepts.

Table 5: MAP and Recall for all submitted runs of the INS task.

Run IDs	Mean Average Precision	Recall
I_A_ITI_CERTH_1	0.114	1000/11197
I_A_ITI_CERTH_1 (2015)	0.064	831/8817
I_A_ITI_CERTH_2 (2015)	0.053	651/8817
I_A_ITI_CERTH_3 (2015)	0.046	525/8817

5 Surveillance Event Detection

5.1 Objective of the Submission

The Surveillance Event Detection (SED) task aims at developing new technologies able to scale on large surveillance video data collections where specific visual events should be detected. TRECVID SED 2016 consists of approximately 100 camera hours of training data derived from Gatwick airport and a 10-hour subset of the multi-camera airport surveillance domain for the main evaluation, as collected by the Home Office Scientific Development Branch (HOSDB). The desired events to be identified are seven: PersonRuns, CellToEar, ObjectPut, PeopleMeet, PeopleSplitUp, Embrace, and Pointing.

A subset of five events (PersonRuns, PeopleMeet, PeopleSplitUp, Embrace, Pointing) was used for the ITI-CERTH's participation in the SED task this year. Our proposed system extended the algorithm proposed in [5].

5.2 System Overview

An activity detection system that allows the user to detect visual events that would be important for airport security management was developed for the SED task (Fig. 4). In this real case scenario, 5 different cameras were placed in Gatwick airport, from which only 4 were processed from our activity detection system. Several pre-segmented videos were used for training purposes, from which our algorithm extracted dense trajectory feature vectors [33], encoded them using the Fisher vector encoding [34] and fed them to separate SVM models (i.e. number-of-classes = 5 models for each camera input) in order to learn how to separate each activity from the others.

Untrimmed videos were then analyzed as test data in order to find the activities that they contain. Spatio-temporal activity localization was computed by using Motion Boundary Activity Areas (MBAAs) which provide: *when* the activity starts and ends (i.e. the activity boundaries) and *where* it occurs inside the video frame. After the activity boundaries have been detected, an overlapping sliding window process sequentially the candidate video sequence by encoding the action descriptors (i.e. dense trajectories) to Fisher vectors and feeding them as input to the SVM models in order to compute prediction score for each activity [35]. Prediction scores are then sorted and thresholded in order to provide the final prediction list for each activity.

5.2.1 Surveillance Event Detection System

An action detection system targeting on a subset of 5 events of interest, i.e. PersonRuns, PeopleMeet, PeopleSplitUp, Embrace and Pointing, was designed for TRECVID SED 2016. The events involving

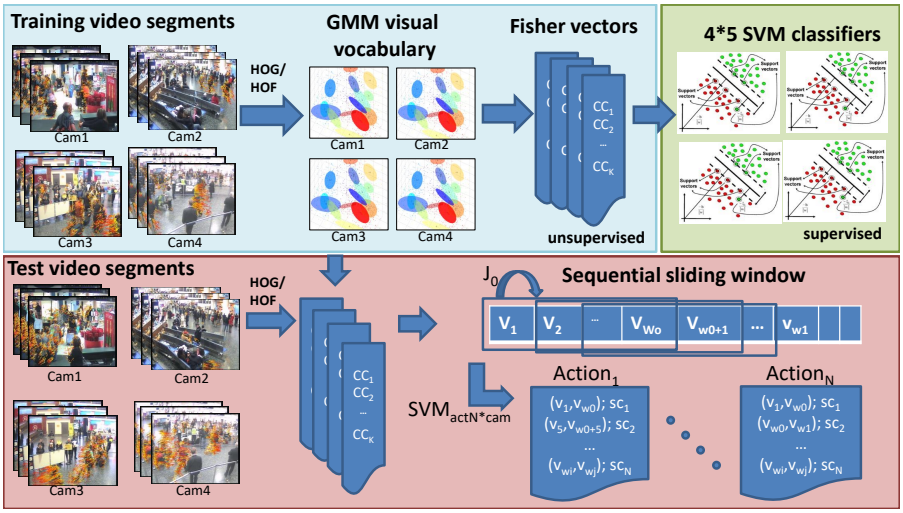

Figure 4: Block diagram of the surveillance event detection system.

a single person plus an object (i.e. CellToEar and ObjectPut) are considered more challenging and were not included in our models.

Our surveillance event detection system consists of a training and a test algorithmic framework [35]. Training framework entails the computation of two unsupervised and one supervised layer in order to compute the appropriate activity models for each camera separately:

1. **Unsupervised** low-level feature extraction, which uses MBAAs in order to sample dense trajectories and represent them with HOG/HOF descriptors.

2. **Unsupervised** intermediate representation level, which constructs a thorough visual vocabulary by using Gaussian Mixture Model (GMM) followed by Fisher vector encoding.

3. **Supervised** learning with the use of linear SVMs for event learning and classification.

More thoroughly, the first offline/training stage, uses MBAAs in the trimmed videos of the training set in order to sample trajectory points and build dense trajectories [25]. HOGHOF [36] descriptors are then computed around the trajectory points in order to capture appearance and motion information and are 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 [37]. A visual vocabulary is subsequently constructed by using a GMM with 64 clusters and Fisher vectors are deployed in order to describe each video segment. Finally, the 5 considered event models are learned by 5 linear SVMs for each of the 4 cameras. Training data from cameras 1,2,3 and 5 are 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).

Test framework entails the activity detection part that is deployed on the untrimmed videos and involves the computation of the dense trajectories inside the MBAAs and their Fisher vector encoding, for a predefined temporal window of 80-frames with a 60 frames temporal step. MBAAs are used to detect spatio-temporal activity boundaries (i.e. the start and the end frame) and provide the video sequence, where the overlapping sliding window defines the regions that will be encoded into a Fisher vector and computes the prediction scores for each activity model.

The overall process is depicted in Fig. 4.

5.3 SED Results

Our team submitted one run to the SED 2016 task. The activity detection task was performed on a 64-bit Windows PC with Intel Core i7 3.50 GHz and 32 GB RAM. The performance of our system is reported in Table 7 and can be compared with the one that we acquired last year in Table 6. According to these results, the performance of our algorithm was not the expected one. This happened because

we omitted the interface system that we used in the last year's submission [5] which was used to threshold our data and return automatically the top-100 or top-200 video sequences with the highest scores for each event. The omission of this post-processing step which functioned as a filter in our data, led in many correct detections, but also in a large amount of false alarms as seen in Table 7.

Table 6: 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

Table 7: The Actual DCR and Minimum DCR of the 2016 interactive result

Event	Rank	ADCR	MDCR	#CorDet	#FA	#Miss
Embrace	7	6.2212	1.0005	161	12321	12
PeopleMeet	7	6.1644	1.0005	297	12185	26
PeopleSplitUp	7	6.1691	0.9650	172	12310	4
PersonRuns	7	6.2335	1.0005	61	12421	2
Pointing	7	6.1024	1.0005	717	11765	212

6 Conclusions

In this paper we reported the ITI-CERTH framework for the TRECVID 2016 evaluation [6]. ITI-CERTH participated in the AVS, MED, INS and SED tasks in order to evaluate new techniques and algorithms. Regarding the AVS task, useful conclusions were reached regarding the different steps of our concept-based video shot annotation and the query linguistic analysis components. Concerning the MED task, our KSDA+LSVM algorithm continues to provide good performance in 010Ex and 100Ex task. Also, our 000Ex method presented improvement in the overall accuracy due to better exploitation of the visual concept pool. As far as INS task is concerned, the results reported were significantly better than last year's results but there is still a lot of room for improvement in order for the system to become competitive against the other systems. The conclusions that were drawn from this year runs was that fusion between different modules should be realized and the system should be less user-driven, thus it should able to provide results more automatically without requiring constantly user's input. Finally, as far as SED task is concerned, while we performed a number of changes in the core of the action detection system, we did not succeed to run correctly our action detection algorithm on the test data and the results were spoiled with a vast amount of false alarms and missed detections. As a future work we plan to enrich our action representation and machine learning system, so that we achieve a better performance.

7 Acknowledgements

This work was partially supported by the European Commission under contracts H2020-693092 MOVING, H2020-687786 InVID, 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, N. Gkalelis, and P. Sidiropoulos et al. ITI-CERTH participation to TRECVID 2012. In *TRECVID 2012 Workshop*, Gaithersburg, MD, USA, 2012.

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

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

[5] F. Markatopoulou, A. Ioannidou, and C. Tzelepis et al. ITI-CERTH participation to TRECVID 2015. In *TRECVID 2015 Workshop*, Gaithersburg, MD, USA, 2015.

[6] G. Awad, J. Fiscus, and M. Michel et al. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *TRECVID 2016 Workshop*. NIST, USA, 2016.

[7] O. Russakovsky, J. Deng, and H. Su et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[8] 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.

[9] 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.

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

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

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

[13] N. Pittaras and et al. Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In *MultiMedia Modeling Conf. (MMM 2017)*, (accepted for publication), 2017.

[14] 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.

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

[16] 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.

[17] 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.

[18] Y. Guangnan, Yitong L., and Hongliang X. et al. Eventnet: A large scale structured concept library for complex event detection in video. In *ACM MM*, 2015.

[19] A. Karpathy, G. Toderici, and S. Shetty et al. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[20] B. Zhou, A. Lapedriza, and J. et al. Xiao. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[21] 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.

[22] 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.

[23] 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.

[24] P. Mettes, D. Koelma, and C. Snoek. The imagenet shuffle: Reorganized pre-training for video event detection. *arXiv preprint arXiv:1602.07119*, 2016.

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

[26] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of the 2001 IEEE Computer Society Conference on CVPR..*, volume 1, pages I–511. IEEE, 2001.

[28] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013.

[29] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proc. of the 12th IAPR International Conference on Pattern Recognition.*, volume 1, pages 582–585 vol.1, Oct 1994.

[30] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[31] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. of the British Machine Vision Conference (BMVC)*, 2015.

[32] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, January 2011.

[33] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[34] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer Berlin Heidelberg, 2010.

[35] K. Avgerinakis, A. Briassouli, and Y. Kompatsiaris. Activity detection using sequential statistical boundary detection (ssbd). *Computer Vision and Image Understanding*, 2015.

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

[37] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris. Activities of daily living recognition using optimal trajectories from motion boundaries. *Journal of Ambient Intelligence and Smart Environments*, 7(6):817–834, 2015.