ITI-CERTH participation in TRECVID 2017

Foteini Markatopoulou[1,2], Anastasia Moumtzidou[1], Damianos Galanopoulos[1], Konstantinos Avgerinakis[1], Stelios Andreadis[1], Ilias Gialampoukidis[1], Stavros Tachos[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, moumtzid, dgalanop, koafgeri, andreadisst, heliasgj, staxos, 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 2017 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 with concept-based and semantic-embedding representations of video fragments. Regarding the MED task, this year we participate on Pre-Specified and Ah-Hoc sub-tasks exploiting both motion-based as well as DCNN-based features. The INS task is performed by employing VERGE, which is an interactive retrieval application that integrates retrieval functionalities that consider mainly visual information. For the SED task, we deploy a novel activity detection algorithm that is based on human detection in video frames, goal descriptors, dense trajectories, Fisher vectors and a discriminative action segmentation scheme.

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, in 2012, 2013, 2014 and 2015 in the INS, SIN, MED and MER tasks ([2], [3], [4], [5]) and in 2016 in the AVS, MED, INS and SED tasks ([6]) 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.

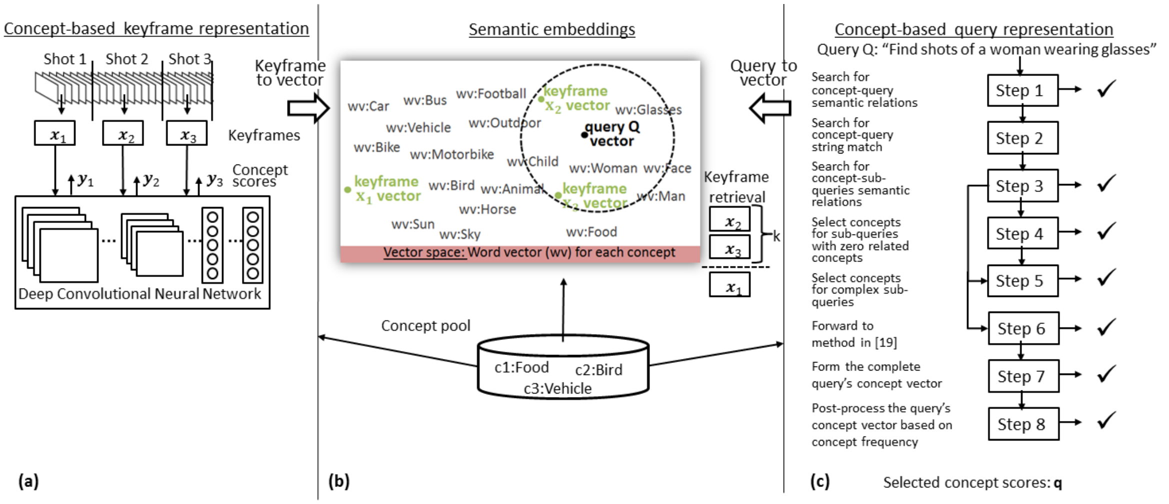
[1] Information Technologies Institute - Centre for Research and Technology Hellas

2 Ad-hoc Video Search

2.1 Objective of the Submission

The goal in the TRECVID 2017 AVS task [7] is the development of techniques for retrieving 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 2016 task was based on representing each query as a vector of related concepts (concept-based representation). This year we extend this approach towards two directions. Firstly, we combine such concept-based representations with semantic-embedding representations, derived from the former, for matching the queries and videos [8]. Secondly, we extend query's linguistic analysis with some extra steps and our aim was to investigate the way that each of these steps affects the final retrieval accuracy.

2.2 System Overview



Concept-based keyframe representation
Shot 1
Shot 2
Shot 3
x_1
x_2
x_3
Keyframes
y_1
y_2
y_3
Concept scores
Deep Convolutional Neural Network
(a)
Keyframe to vector
Semantic embeddings
wv:Car
wv:Bus
wv:Football
keyframe x_2 vector
wv:Glasses
wv:Vehicle
wv:Outdoor
query Q vector
wv:Bike
wv:Motorbike
wv:Child
wv:Woman
wv:Face
keyframe x_1 vector
wv:Bird
wv:Animal
keyframe x_3 vector
wv:Man
wv:Horse
wv:Sun
wv:Sky
wv:Food
Vector space: Word vector (wv) for each concept
Keyframe retrieval
x_2
x_3
k
x_1
Concept pool
c1:Food
c2:Bird
c3:Vehicle
(b)
Query to vector
Concept-based query representation
Query Q: "Find shots of a woman wearing glasses"
Search for concept-query semantic relations
Step 1
✓
Search for concept-query string match
Step 2
Search for concept-sub-queries semantic relations
Step 3
✓
Select concepts for sub-queries with zero related concepts
Step 4
✓
Select concepts for complex sub-queries
Step 5
✓
Forward to method in [19]
Step 6
✓
Form the complete query's concept vector
Step 7
✓
Post-process the query's concept vector based on concept frequency
Step 8
✓
(c)
Selected concept scores: q

Figure 1: Developed AVS system (modified from [8]).

An overview of the system we developed for the AVS task is presented in Fig. 1. Specifically, a sequence of steps is followed, starting with the ad-hoc query and transforming it to a vector of concepts (Fig. 1 (c)). In addition, each video shot is annotated with semantic concepts using deep learning, which results to another vector representation that corresponds to the concepts that are depicted in the video shot (Fig. 1 (a)). Then, query and video shot concept-based representations are transformed to semantic-embedding representations (Fig. 1 (b)) as described in [8]. Finally, given a test query, after the concept-based keyframe representations have been calculated, our system measures their distance from the concept-based query representation, e.g. by calculating the euclidean distance. Similarly, the distance between the semantic embedding keyframe representations and the semantic embedding query representation is calculated and the two distance vectors are combined in terms of arithmetic mean. The 1000 keyframes with the smallest distance are then retrieved. The main components of the above process are further explained below.

2.2.1 Concept-based Keyframe Representation

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 [9], 345 TRECVID SIN [10] concepts (i.e., all the available TRECVID SIN concepts, except for one which was discarded because only 5 positive samples are provided for it), 500 event-related concepts, and 205 place-related concepts.

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 [11], ii) GoogLeNet [12], iii) ResNet [13], iv) VGG Net [14] 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) the ResNet pre-trained ImageNet network on the 345 concepts using the TRECVID AVS development dataset and the extension strategy proposed in [15]. We applied the fine-tuned network on the 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 Support Vector Machine (SVM) per concept. Subsequently, we applied this FT network on the AVS test keyframes to extract features, and used them as input to the trained SVM classifiers in order to gather scores for each of the 345 concepts. Finally, to obtain scores for the event- and place-related concepts we applied the publicly available DCNNs that have been fine-tuned on the EventNet [16] and Places [17] dataset, respectively. The scores obtained from the four pools of concepts (1000 ImageNet, 345 TRECVID SIN, 500 events and 205 places) were concatenated in a single vector. Consequently, a 2050-element concept vector was created for each test keyframe. Each element of this vector corresponds to one concept, from the 2050 available concepts, and indicates the probability that this concept appears in the video shot.

2.2.2 Concept-based Query Representation

The second component of our system represents each query as a vector of related concepts. Given the above pool of 2050 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. It should be noted that our previous year's AVS system has been extended with steps five and eight.

- 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 [18]. 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 we proceed to step seven; 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 breaks 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. For example, given the query "three people or more walking or bicycling on a bridge during daytime" the concept "three or more people" will be chosen in this step. 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 described and we proceed to step five. 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 six.

- 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 five.
- Step five: In this step we further enrich the pool of the selected concepts for the particular query by processing the complex sub-queries as follows: We define as complex sub-queries those that consist of more than one words (e.g., "train station platform"). For each word of each complex sub-query we pick the most relevant concepts in terms of ESA measure. Then we proceed to step seven.
- Step six: For some queries step three is not able to select any concepts. In this case, the original query and the sub-queries are served as input to the zero-example event detection pipeline of [19], 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 step seven.
- Step seven: 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, 5 or 6 the corresponding vector's element is assigned with the relatedness score (calculated using the ESA measure), whereas if it has been selected in step 2 it is set equal to 1. In all the above steps, whenever we mention that a threshold is used in order to take a decision regarding selecting a concept or not, the value of this threshold is set to 0.8. After that we proceed to step eight.
- Step eight: As a final step we recalculate the scores of the selected concepts based on their frequency on the training set. More specifically, selected concept scores are increased/decreased by a factor of $f = [0.8 - 1.2]$. I.e., the score of frequently appearing concepts (e.g. person, indoor, overlaid_text etc.) is decreased while the score of rarely appearing concepts is increased (e.g. diving, bridge, egyptian_pyramids etc.). Our rationale here is that very frequent concepts are not very characteristic of the query because they provide very general information; on the other hand, rare concepts could provide more distinctive information.

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. Similarly, the distance between the semantic embedding keyframe representations and the semantic embedding query representation is calculated and the two distance vectors are combined in terms of arithmetic mean.

2.3 Description of Runs

Four AVS runs were submitted in order to evaluate the potential of the aforementioned approaches on the TRECVID 2017 AVS dataset [7]. The submitted runs are briefly described below:

- ITI-CERTH 1: The combination (late fusion by arithmetic mean) of runs 2, 3 and 4 (explained below).

- ITI-CERTH 2: Complete pipeline using: a) Concept-based keyframe representation: five pre-trained ImageNet networks for annotating the test keyframes with 1000 ImageNet concepts; SVM-based concept detectors for the 345 TRECVID SIN concepts; two pre-trained DCNNs for 500 event-related and 205 place-related concepts, respectively. b) Concept-based query representation using the eight-step process presented in Section 2.2.2. c) Combination of concept-based representations and semantic-embedding representations, derived from the former for matching the queries and videos [8]. d) Euclidean distance for matching the keyframe's vectors (concept-based or semantic embeddings) with the query's vector (concept-based or semantic embeddings).

- ITI-CERTH 3: A modified version of ITI-CERTH 2, where query analysis does not stop in step one even if a concept that describes very well the query is detected (Section 2.2.2); that is, any such concept is returned but the process of Section 2.2.2 continues with step 2 etc.

- ITI-CERTH 4: A modified version of ITI-CERTH 2, where steps five and eight of the query analysis pipeline are excluded (Section 2.2.2).

2.4 Ad-hoc Video Search Task Results

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

Submitted run:	ITI-CERTH 1	ITI-CERTH 2	ITI-CERTH 3	ITI-CERTH 4
MXinfAP	0.093	0.086	**0.095**	0.093

Table 1 summarizes the evaluation results of the aforementioned runs in terms of the Mean Extended Inferred Average Precision (MXinfAP). Our team submitted only fully-automatic runs. According to Table 1 we conclude as follows:

- The additional query analysis steps (five and eight), as presented in Section 2.2.2 are not required; excluding them i.e. using the query analysis steps of our 2016 AVS system presents better accuracy (ITI-CERTH 4 run outperforms ITI-CERTH 2 run).

- Slightly modifying the first step of our query analysis process seems to improve the overall system with MXInfAP increasing from 0.086 to 0.095.

- Combining the results of individual runs, as in run ITI-CERTH 1, did not improve the performance; this is likely due to the runs being combined having only minor algorithmic differences, thus also resulting in very similar lists of retrieved video shots.

- Overall, our fully-automatic runs performed satisfying in this challenging task. We will further investigate the parts of our system that can be improved in order to further increase its accuracy.

3 Multimedia Event Detection

3.1 Objective of the Submission

This year in the MED task only the 10Ex evaluation condition exists, which means that only 10 positive and 5 related video exemplars are available per event. Our team participates in both the Pre-Specified (PS) and Ad-Hoc (AH) sub-tasks.

3.2 System Overview

3.2.1 10Ex: Learning video event detectors from positive and related video examples

In our PS and AH 10Ex submissions, for building our event detectors, firstly, we utilized an extended and speeded-up version of our Kernel Subclass Discriminant Analysis [20, 21] for dimensionality reduction, and after that we used a fast linear SVM (KSDA+LSVM). The GPU-accelerated implementation of this method [22] was not used in our MED 2017 experiments due to the limited number of training samples, which did not necessitate the use of GPU computing resources.

Visual Features: Two types of visual information were 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 (2 keyframes per second). We annotated the video frames based on 5055 ImageNet [9] concepts, 345 TRECVID SIN [10] concepts, 500 event-related concepts [23], 487 sport-related concepts [24] and 205 place-related concepts [17]. To obtain scores regarding the 5055 ImageNet concepts we self-trained a GoogLeNet network [25] 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 [24], respectively. We also used the EventNet [23] that consists of 500 events and the Places205-GoogLeNet, which was trained on 205 scene categories of Places Database [17]. The output of one or more hidden layers of the above networks was used as a global frame representation.
- For encoding motion information we use improved dense trajectories (DT) [26]. 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 $\mathbb{R}^{101376}$.

All the above video feature vectors for a video are then concatenated to a single high-dimensional feature vector in $\mathbb{R}^{130648}$. These feature vectors are then used to build one detector for each event.

Event Detection: A nonlinear discriminant analysis (DA) method is used to derive a lower dimensional embedding of the original data, and then fast Linear SVMs are employed in the resulting subspace to learn the events. Particularly, for dimensionality reduction we utilize a new very fast kernel subclass-based method [20, 21], which is shown to outperform other DA approaches. Then we apply each one of the trained event detectors to every video of the MED17-EvalFull set and order their scores in descending order.

Online Training: Finally, an online training step based on the top retrieved videos per event is applied (i.e. a pseudo-positive feedback mechanish). More specifically the top-20 retrieved videos from the above procedure along with the top-10 retrieved videos from the zero-example event detection system [19] are used as new positive videos. These videos along with the provided positive training samples form a new training set with 40 positive samples, and then the same training procedure as described above is followed in order to generate the final event detectors that are applied to the test collection.

3.3 Dataset Description

For training our Pre-Specified (PS) and Ad-Hoc event detectors we used the PS-Training video set consisting of 200 positive (or near-miss) videos and the AH-Training video set which also consists of 200 positive (or near-miss), along with the Event-BG video set containing 5000 (200 hours) of background videos. The 10 PS event classes and 10 AH event classes are shown in Table 2 for the shake of completeness.

MED 2017 PS events	MED 2017 AH events
E051 - Camping	E071 - Fencing
E052 - Crossing a barrier	E072 - Reading a book
E053 - Opening a package	E073 - Graduation ceremony
E054 - Making a sand sculpture	E074 - Dancing to music
E055 - Missing a shot on a net	E075 - Bowling
E056 - Operating a remote controlled vehicle	E076 - Scuba diving
E057 - Playing a board game	E077 - People use a trapeze
E058 - Snow-sculpting activity	E078 - People performing plane tricks
E059 - Making a beverage	E079 - Using a computer
E060 - Cheerleading	E080 - Attempting the clean and jerk

Table 2: MED 2017 Pre-Specified (PS) and Ad-Hoc (AH)events.

For the evaluation of our systems we processed the MED17EvalFull set consisting of 200, 000 videos. We submitted runs for the 10Ex evaluation conditions (i.e. 10 positive exemplars, respectively, are used for learning the specified event detector) for both the PS and AH task.

3.4 Description of Runs

3.4.1 10Ex

For both tasks (PS and AH), we submitted 1 primary and 1 additional run:

- **p-1KDALSVM**: In the primary run, the KSDA+LSVM method is used to build the event detectors and perform the event search in the MED17EvalFull set using motion and DCNN-based features, as discussed in Section 3.2.1.
- **c-1KDALSVMRF**: In this run, we use the online training step to find more positive samples, and then the event search is performed as in our primary run.

3.5 Multimedia Event Detection Results

In Table 3, the evaluation results of our PS 10Ex and AH 10Ex systems for the MED task are shown in terms of InfAP@200.

PS			AH		
Event ID	c-1KDALSVMRF	p-1KDALSVM	Event ID	c-1KDALSVMRF	p-1KDALSVM
E051	28.5	25.9	E071	73.7	77.5
E052	0.5	0.5	E072	7.7	7.8
E053	6.0	5.3	E073	32.6	49.3
E054	12.5	8.1	E074	6.8	7.0
E055	28.1	37.3	E075	93.5	93.4
E056	3.2	5.7	E076	70.1	69.4
E057	20.6	16.7	E077	65.5	68.7
E058	12.3	11.9	E078	49.7	37.9
E059	20.7	21.2	E079	27.3	28.9
E060	4.2	5.2	E080	39.1	50.1
Mean	13.7	**13.8**	Mean	46.6	**49.0**

Table 3: InfAP@200 for all submitted runs of the MED 17 task

From the analysis of the evaluation results we can conclude that the online training step is very sensitive and depends on the nature of the input events. In the PS task, online training slightly decreases the overall performance. In contrast, at the AH task the performance is significantly reduced. It is worthwhile to mention that the per-event performance can vary considerably between the two approaches. For example, for the event E078 the use of the online training step improves performance from 37.9% to 49.7%, while for event E073 the performance is decreased from 49.3% to 32.6%.

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 inside a video collection for video segments of a specific person in a specific place. The user is provided with two sets of visual examples; the first contains the specific person and the second the specific location. 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).

ITI-CERTH participated in the TRECVID 2017 INS task by submitting a single run that incorporated several algorithms that consider mostly visual information. The system and algorithms developed are integrated in VERGE[1] interactive video search engine.

4.2 System Overview

The general procedure of image search, as well as specific tasks such as the INS, require an interactive, efficient and easy to use application. VERGE (Fig. 2, Fig. 3) is a graphical user interface that serves as a retrieval tool by combining a multitude of search capabilities, mostly based on visual information. The main integrated modalities for retrieving an image or a video include: a) Visual Similarity Search module; b) High Level Visual Concept Retrieval; c) Face Detection and Face Retrieval module; d) Scene Similarity Search module; e) Multimodal Fusion module; and f) Clustering.

As seen in both figures Fig. 2 and Fig. 3, the VERGE interface consists of two basic parts: a toolbar with various useful features and a results panel. Describing the toolbar from left to right, a burger icon toggles a sliding menu that provides some options to initiate the search procedure. Namely, the user is able to navigate to the complete set of video frames or selected shots, to places or miscellaneous concepts, to clusters or topics. Next, there is a one-line text input field that looks for given keywords in natural language metadata of videos, i.e. actors lines spoken during each shot. Moreover, the toolbar includes a slider to adjust the size of the shots and a countdown that shows the remaining time for the submission and applies only to the INS task.

The main component of VERGE displays a shot-based representation of video results as images in a grid view. Hovering on a single shot allows the end user to perform further search alternatives, based on visual, face, scene, and fusion similarity, by clicking the respective icon. Also, a check button on the upper right corner can be used to save images, in order to submit them later to the contest. Finally, clicking on a shot presents the scene that this frame belongs to and the user is able to navigate through the whole video.

The system has been built on common Web technologies, e.g. HTML5, CSS, JavaScript, jQuery, PHP, and open-source libraries, e.g. Bootstrap and Kendo UI.

To demonstrate the core functionality of VERGE, Fig. 2 displays the outcome of the scene similarity search module (a coffee shop) and the scene navigation, while Fig. 3 depicts the results based on face similarity. Using these shots, the user can continue with other approaches, for example the visual similarity or a combination of face and scene similarity, e.g. the fusion module.

4.2.1 Visual Similarity Search Module

The visual similarity search module performs content-based retrieval using deep convolutional neural networks (DCNNs). Specifically, we have trained GoogleNet [12] 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 [27].

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

[1] VERGE: http://mklab-services.iti.gr/verge/trec2017/

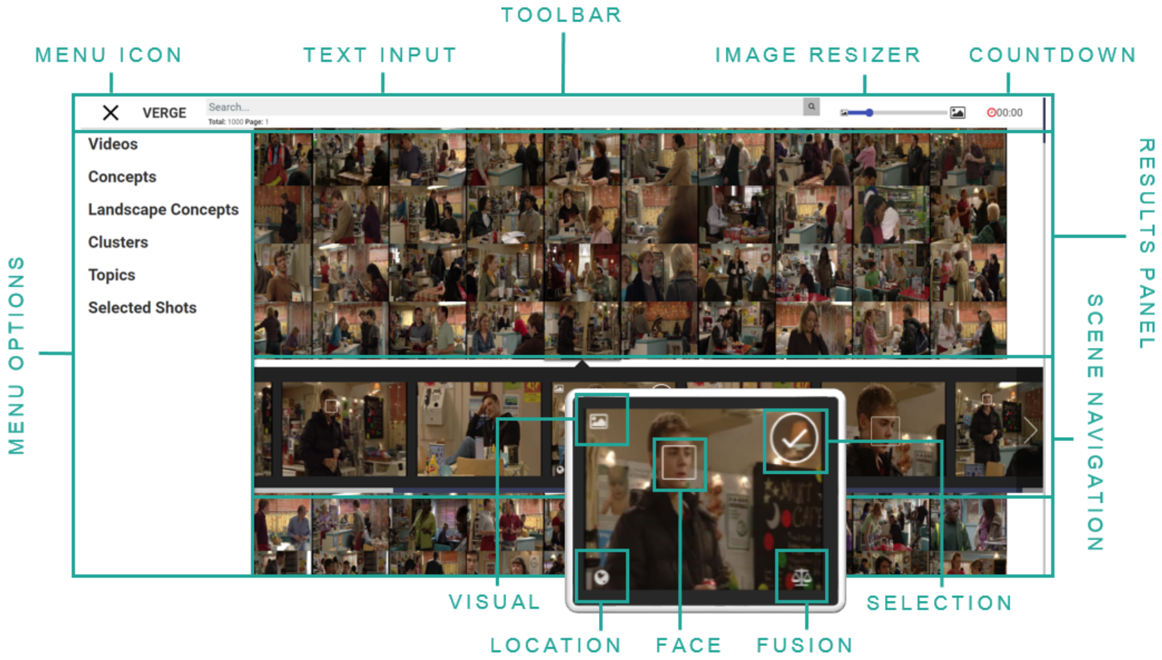


Figure 2: Components of the VERGE application.

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/ place recognition [28].

4.2.3 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 the algorithm that was applied was [29], while the algorithm used for face retrieval module was the VGG-Face CNN descriptors which were computed using the VGG-Very-Deep-16 CNN architecture described in [30]. As feature vector, we considered the last FC layer with size 2622. Eventually, the face features were used for constructing an IVFADC index similar to the one created in 4.2.1 that allows fast face retrieval.

4.2.4 Scene Similarity Search Module

This module uses the output of two inception layers (inception_3a and inception_3b) of the GoogLeNet CNN that is trained for recognizing scenes (Section 4.2.2) as a global keyframe representation. The size of the feature vector is 736. Eventually, the scene features were used for constructing an IVFADC index similar to the one created in 4.2.1 that allows fast scene retrieval.

4.2.5 Multimodal Fusion Module

Given that the aim of INS task is to retrieve a specific person in a specific place, this module fuses the dcnn-based face descriptors and the dcnn-based scene descriptors in a late fusion approach. These two descriptors (or modalities) are fused with a non-linear graph-based fusion approach [31], that is based on the construction of a uniform multimodal contextual similarity matrix and the non-linear combination of relevance scores from query-based similarity vectors.

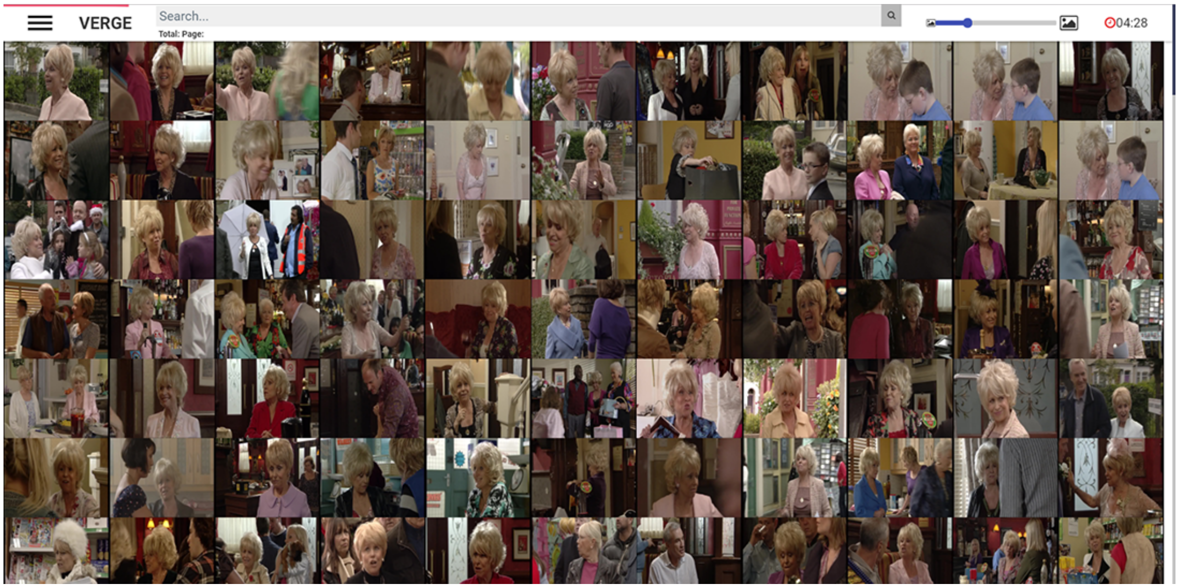


Figure 3: Screenshot of VERGE, presenting results of the Face Retrieval module.

In brief given a query shot, an initial filtering stage is applied that keeps only the $top - i$ relevant images according to the dominant modality and then computes an $i \times i$ similarity matrix and an $i \times 1$ similarity vector per modality. The similarity matrices and vectors are then fused in a non-linear and graph-based way, providing a fused relevance score vector s_q for the retrieved shots.

In the current implementation, two ranked lists with retrieved shots were created after considering both modalities as dominant. Eventually these two lists with top-k retrieved results of the runs are merged in a single list using combMax late fusion.

4.2.6 Clustering

This module clusters images by using the scene features (4.2.4). The algorithm that was applied involved two steps. The first step required a random sample of 6000 features, which was used for estimating the number of clusters using a density-based approach, namely the DBSCAN-Martingale [32]. The second step involved using the estimated number of clusters, and applying k-means clustering to assign each vector (scene feature) to each cluster. In this hybrid approach, we benefit from density-based clustering where the number of clusters is not required as input, but a lot of noise is usually produced, i.e. items are left unassigned.

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 4 contains the mean average precision for the runs submitted the last three years in TRECVID INS task. Although, the results obtained in INS 2017 are better, there is room for improvement given that the efficiency of our system is still low compared to the winning system. A possible explanation for this difference could be the low number of faces recognized. Specifically, the total number of shots of the database is 469,539 while the number of faces recognized is 418,530, which is rather low given that the majority of shots contain more than one faces.

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 10 minutes, in our case. TRECVID iSED 2017 provides approximately 100-hour videos for development (2008 DevSet and EvalSet, Gatwick data) and an 11 hour subset of the multi-camera data for the main evaluation. A new Group Dynamic Subset (SUB17) using only 2 hours of this video and limited to the Embrace, PeopleMeet and PeopleSplitUp events. As a third ITI-CERTH's participation on the TRECVID-SED task, we focused on improving our results both in SED and iSED tasks and build a robust activity detection system that overpower our previous years' performances.

5.2 System Overview

A simple, but efficient interface was developed for both interactive and non-interactive SED tasks and it can be seen in Fig. 4. The main target of the tool is to provide the end users with surveillance video shots, in order to detect visual events that could be critical for the security of an airport. From a drop-down list at the upper side of the page, users can select an Activity of Interest (AoI) and the interface returns image shots from video segments that most probably meet the activity's definition. Then, users are able to determine which shots do not contain the AoI by clicking on the check button in the upper right corner of each image. When the procedure of evaluation is completed, the "Exclude" box above the shots serves as a submission button.

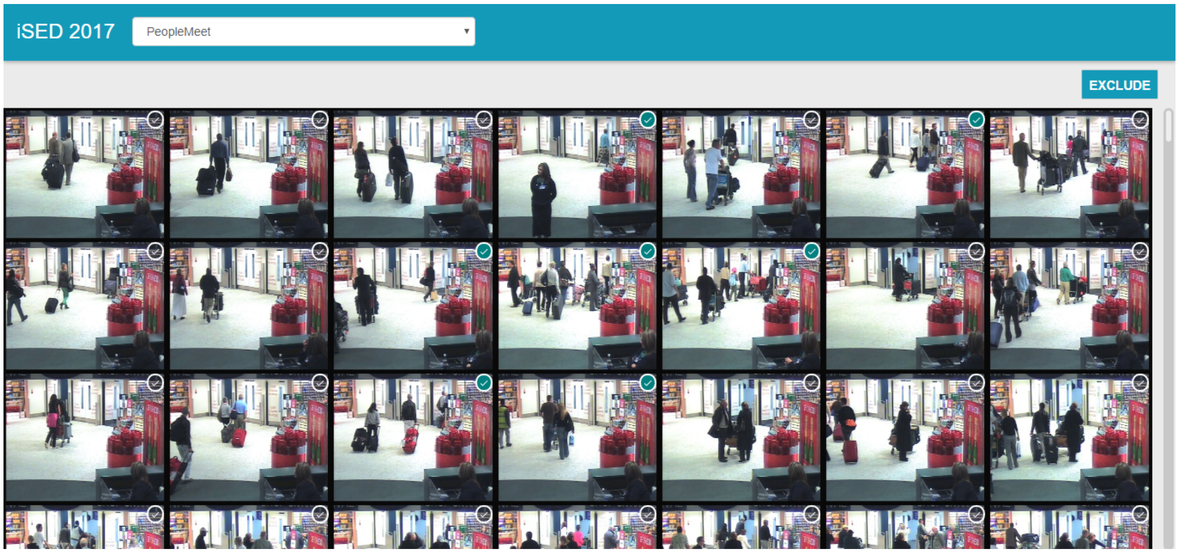


Figure 4: Our interactive Surveillance Event Detection system.

Table 4: MAP of INS task.

Run IDs	Mean Average Precision
I_A_ITI_CERTH_1 (2017)	0.135
I_A_ITI_CERTH_1 (2016)	0.114
I_A_ITI_CERTH_1 (2015)	0.064

5.3 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) were considered more challenging, they possibly require a different approach to be deployed and have not been included in the current version of our system.

Our surveillance event detection system is based on the following four steps [33]:

- Low-level feature extraction, which is performed on human detection bounding boxes to sample dense trajectories and represented with HOG/HOF descriptors.
- 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.
- Finding the most discriminative features and forming their regions of certainty so as to determine the goal descriptors and perform accurate and abrupt temporal activity segmentation.
- High level representation with the use of linear SVMs for activity learning and classification.

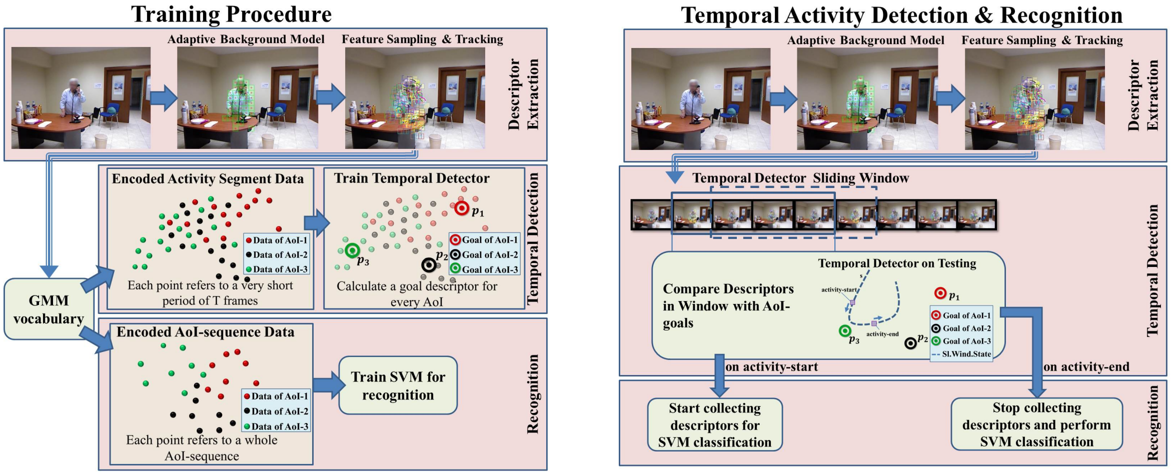

Training Procedure
Adaptive Background Model
Feature Sampling & Tracking
Descriptor Extraction
Encoded Activity Segment Data
Data of AoI-1
Data of AoI-2
Data of AoI-3
Each point refers to a very short period of T frames
Train Temporal Detector
p_1
p_2
p_3
Goal of AoI-1
Goal of AoI-2
Goal of AoI-3
Calculate a goal descriptor for every AoI
Temporal Detection
GMM vocabulary
Encoded AoI-sequence Data
Each point refers to a whole AoI-sequence
Train SVM for recognition
Recognition
Temporal Activity Detection & Recognition
Temporal Detector Sliding Window
Temporal Detector on Testing
Compare Descriptors in Window with AoI-goals
activity-start
activity-end
Sl.Wind.State
on activity-start
on activity-end
Start collecting descriptors for SVM classification
Stop collecting descriptors and perform SVM classification

Figure 5: Training (left figure) and Testing (right figure) of our goal based activity detection system that has been used in TRECVID-SED challenge.

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 YOLO human detector [34] in the videos of the training set in order to sample trajectory points inside them and build dense trajectory descriptors [35]. HOGHOF descriptors are then extracted 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 [36]. A visual vocabulary is then 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. These vectors are then considered so as to determine the most discriminative vectors of each Activity of Interest (AoI), determine the goal descriptors of TRECVID-SED dataset and help on the abrupt and accurate video segmentation on the testing phase of our activity detection algorithm. Additionally, the same vectors are used to train 5 linear SVMs in order to build a recognition model for each activity that is taken into account. Training data from cameras 1,2,3 and 5 were used in this offline stage so as to build the event models (the videos of CAM4 were discarded since they contain a very limited number of events). In the

Table 5: The actual and minimum DCR of our activity detection system.

TrecVid SED 2017 - Goal descs						
Event	Rank	ADCR	MDCR	#CorDet	#FA	#Miss
Embrace	6/6	1.5465	1.1901	51	1685	122
PeopleMeet	3/5	1.6009	1.1376	82	1712	241
PeopleSplitUp	4/5	1.5269	1.0421	60	1738	116
PersonRuns	5/7	1.6737	1.1661	11	1699	52
Pointing	4/6	1.7318	1.1774	105	1692	824

Table 6: The actual and minimum DCR of our activity detection system when the interactive platform was taken into account.

TrecVid SED 2017 - Goal descs + interactive						
Event	Rank	ADCR	MDCR	#CorDet	#FA	#Miss
Embrace	5/6	1.1088	1.1088	33	600	140
PeopleMeet	3/5	1.3370	1.3370	40	923	283
PeopleSplitUp	3/5	1.2477	1.2477	34	883	142
PersonRuns	5/7	1.2451	1.2451	4	618	59
Pointing	4/6	1.2404	1.2404	48	585	881

online/testing stage, goal descriptors are used so as to detect the start and end video frame that an activity occurs inside a large video sample. Human detection bounding boxes, dense trajectories and Fisher vectors are then deployed, so as to collect the activity descriptors that exist inside the detected video frames and predict the activities that might exist inside them. The overall process is depicted in Fig. 5.

5.4 Interactive Surveillance Event Detection (iSED) Results

As already reported, we submitted two runs to the TRECVID-SED 2017 task. Seven 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 and Table 6. As it can be seen from the results, we outperformed two teams almost in all activities that we participated except from one i.e. the embrace. Our interactive system improved our results from 0.3 to 0.5 in actual DCR metric, leading to a more accurate action detection system with fewer false alarms and missing samples.

6 Conclusions

In this paper we reported the ITI-CERTH framework for the TRECVID 2017 evaluation [7]. 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 the good performance, while the performance of the new online training step seems to be sensitive depending on the nature of the input event. 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 we should focus on face detection algorithm in order to increase the faces detected and train the dcnn networks used on our dataset in order to better recognize the existing scenes and faces. As far as SED task is concerned, not only we managed to develop an improved event detection system, but we also deployed an interactive system that improves its results even more. Our future goal is to deploy deep convolutional network techniques so as to build an even more accurate and robust activity localization outcome.

7 Acknowledgements

This work was partially supported by the European Commission under contracts H2020-732665 EMMA, H2020-687786 InVID, H2020-700475 beAWARE and H2020-700024 TENSOR.

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] F. Markatopoulou, A. Moumtzidou, and D. Galanopoulos et al. ITI-CERTH participation in TRECVID 2016. In *TRECVID 2016 Workshop*, Gaithersburg, MD, USA, 2016.

[7] G. Awad, A. Butt, J. Fiscus, D. Joy, A. Delgado, and M. et al. Michel. Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *Proceedings of TRECVID 2017*. NIST, USA, 2017.

[8] F. Markatopoulou, D. Galanopoulos, V. Mezaris, and I. Patras. Query and keyframe representations for ad-hoc video search. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 407–411, New York, NY, USA, 2017. ACM.

[9] 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.

[10] 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.

[11] 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.

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

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

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

[15] 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.

[16] 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.

[17] B. Zhou, A. Lapedriza, and J. et al. Xiao. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014.

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

[19] D. Galanopoulos, F. Markatopoulou, V. Mezaris, and I. Patras. Concept language models and event-based concept number selection for zero-example event detection. In *ICMR*, pages 397–401. ACM, 2017.

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

[21] 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.

[22] 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.

[23] 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.

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

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

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

[27] 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.

[28] 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.

[29] 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.

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

[31] I. Gialampoukidis, A. Moumtzidou, D. Liparas, S. Vrochidis, and I. Kompatsiaris. A hybrid graph-based and non-linear late fusion approach for multimedia retrieval. In *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*, pages 1–6. IEEE, 2016.

[32] I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris. A hybrid framework for news clustering based on the dbscan-martingale and lda. In *Machine Learning and Data Mining in Pattern Recognition*, pages 170–184. Springer, 2016.

[33] S. Tachos, K. Avgerinakis, A. Briassouli, and I. Kompatsiaris. Mining discriminative descriptors for goal-based activity detection. *Computer Vision and Image Understanding*, 160:73–86, 2017.

[34] J. Redmon, S. D. Kumar, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016.

[35] S. Tachos, K. Avgerinakis, A. Briassouli, and I. Kompatsiaris. Appearance and depth for rapid human activity recognition in real applications. In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 38.1–38.12, 2015.

[36] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris. Recognition of activities of daily living for smart home environments. In *The 9th International Conference on Intelligent Environments, Athens, Greece, June 18-19, 2013 (workshops on 16-17 July 2013)*, pages 173–180, 2013.