ITI-CERTH participation in TRECVID 2020

Konstantinos Gkountakos, Damianos Galanopoulos, Marios Mpakratsas, Despoina Touska, Anastasia Moumtzidou, Konstantinos Ioannidis, Ilias Gialampoukidis, Stefanos Vrochidis, Vasileios Mezaris, Ioannis Kompatsiaris

Information Technologies Institute, Centre for Research and Technology Hellas,
6th Km. Charilaou - Thermi Road, 57001 Thermi-Thessaloniki, Greece
{gountakos, dgalanop, mbakratsas, destousok, moumtzid, kioannid, heliasgj, stefanos, bmezaris, ikom}@iti.gr

Abstract

This paper provides an overview of the runs submitted to TRECVID 2020 by ITI-CERTH. ITI-CERTH participated in the Ad-hoc Video Search (AVS), Disaster Scene Description and Indexing (DSDI) and Activities in Extended Video (ActEV) tasks. Our AVS task participation is based on an attention-based cross-modal deep network method for retrieving video shots relevant to ad-hoc textual queries. The DSDI task is performed by implementing a multi-label image classification model, trained on all humanly annotated images and estimating the final classes on averaging the predictions on the keyframes of the video shots. For the ActEV task, we deploy an object detection algorithm and then convert the individual detected objects to activities by following an object tracking technique in order to detect human and vehicle-related activities.

1 Introduction

This paper describes the recent work of ITI-CERTH[1] in the area of video analysis and retrieval. TRECVID [1] has always been a target initiative for ITI-CERTH given that is one of the major evaluation activities in the domain of video. 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]), in 2016 and 2017 in the AVS, MED, INS and SED tasks ([6], [7]) of TRECVID, in 2018 in the AVS, INS and ActEV [8] and in 2019 in the ActEV task [9]. 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 three tasks: AVS, DSDI and ActEV. In the following sections we will present in detail the employed algorithms and the evaluation for the runs we performed in the aforementioned tasks.

2 Ad-hoc Video Search

Our goal in the TRECVID 2020 [10] Ad-hoc Video Search (AVS) task is to develop a system for retrieving a ranked list of 1000 video shots for each ad-hoc textual query, ranked from the most relevant to the least relevant shot for the query. For the 2020 AVS task, we proposed an attention-based cross-modal deep network method. Our approach is developed based on an attention-based

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

method [11] inspired by the dual encoding network presented in [12]. To this end, we create a network that encodes video-caption pairs into a common feature subspace, utilizing attention mechanisms for more efficient textual and visual representation, and exploits the benefits of richer textual embeddings.

2.1 Approach

Let $\mathbf{V}$ be a video shot and $\mathbf{S}$ the corresponding caption of $\mathbf{V}$. Our network translates both $\mathbf{V}$ and $\mathbf{S}$ into a new common feature space $\Phi(\cdot)$, resulting in two new representations $\Phi(\mathbf{V})$ and $\Phi(\mathbf{S})$ that are directly comparable. For this, two similar modules, consisting of multiple encoding levels, are utilized for the visual and textual content, respectively.

Our method utilizes an attention-based dual encoding neural network that uses two similar modules [12], each consisting of multi-level encoding for the video shot as well as for the natural language sentence, in parallel. For the initial video shot representation, each video shot is sampled into a fixed number of 16 frames and a pre-trained Resnet-152 model (trained on the ImageNet-11k dataset) is used for representing every shot's keyframe. Regarding the textual module, each sentence is encoded by averaging the individual one-hot-vectors of its words. Then, for every word, two different word embeddings are utilized: i) the Word2Vec model [13] trained on the English tags of 30K Flickr images, provided by [14]; and, ii) the pre-trained language representation BERT [15], trained on Wikipedia content.

Then, both the word and the keyframe representations go through three different encoders (i.e. mean-pooling, attention-based bi-GRU sequential model [11], and biGRU-CNN [16]). This multilevel encoding is used in order to project both text and video instances into a common feature space following the approach of [17] where the improved marginal ranking loss is used to train the entire network.

Our network is trained using the combination of two large-scale video datasets: MSR-VTT [18] and TGIF [19]. Our AVS 2020 system was evaluated on the V3C1 dataset, consisting of 7.475 videos and 1.082.659 video shots.

2.2 Submission

We submitted one run to the main AVS 2020 task and one run to the progress subtask, in order to evaluate our approach on 30 ad-hoc queries in total. These runs combine multiple training configurations in a late fusion scheme. Table 1 summarizes the evaluation results of our runs across all queries using the Mean Extended Inferred Average Precision (MXinfAP) measure.

Figures 1 and 2 illustrate the performance of our submissions compared to the runs of all other teams for the main and the progress task, respectively.

Task	Main	Progress
MXinfAP	0.202	0.159

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

3 Disaster Scene Description and Indexing

In the DSDI task of TRECVID2020 [10], the aim is to detect and provide probabilities for the 32 concepts that are related to disaster scenes. The task provides as input, videos that are coming from the LADI dataset; however, the annotation is done on image level.

In the following, we describe the dataset used for training and testing our models, the approach we followed and the results of our submission.

3.1 Dataset

The training dataset consists of 31.146 multi-labeled human annotated images. The annotation is done on an image level with one or more labels/classes, which are overall 32. The available classes

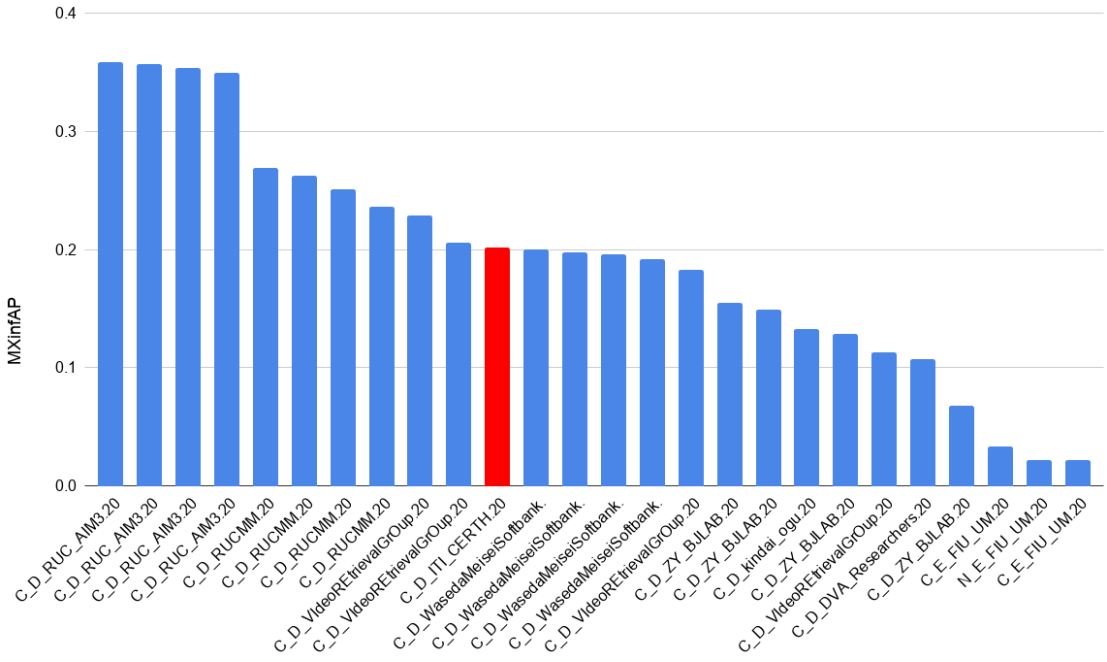

Figure 1: The performance of our AVS 2020 run compared with the runs of other teams for the main task (Fully automatic runs only).

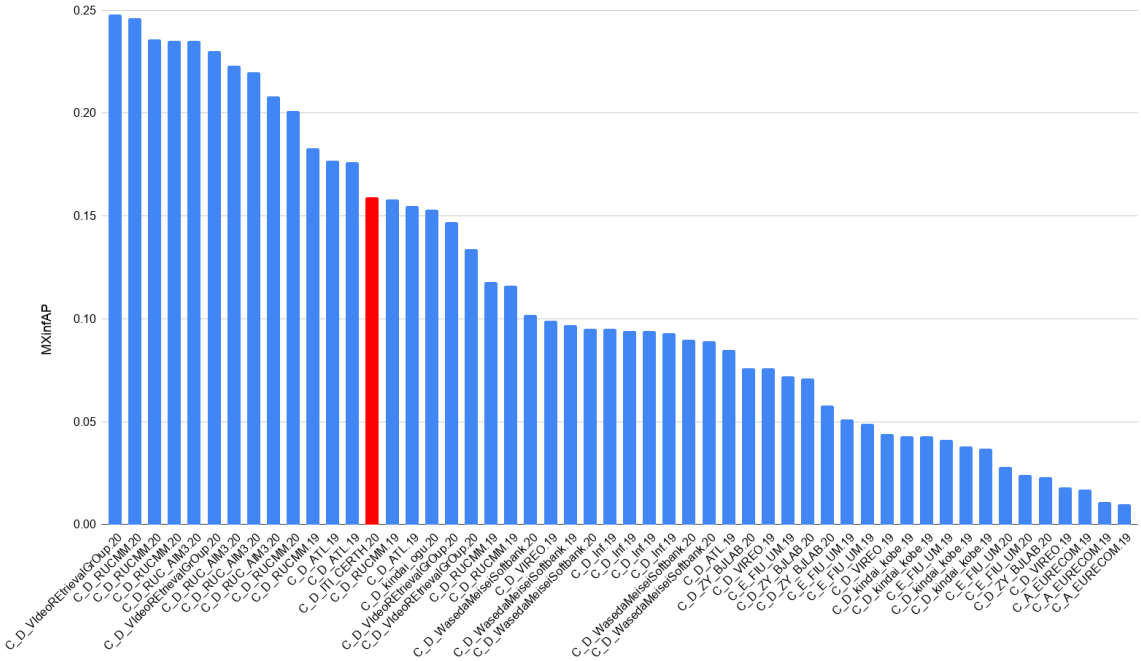

Figure 2: The performance of our AVS 2020 run compared with the runs of other teams for the progress task (Fully automatic runs only).

Damage	Environment	Infrastructure	Vehicles	Water
damage (misc) – 12,825	dirt – 6,627	bridge – 1,982	aircraft – 142	flooding – 2,493
flooding / water damage – 14,671	grass – 5,592	building – 6,294	boat – 1,597	lake / pod – 3,248
landslide – 831	lava – 30	dam / levee – 139	car – 5,577	ocean – 1,380
road washout - 674	rocks- 167	pipes – 101	Truck – 409	paddle – 1,036
rubble / debris – 4,180	sand - 243	utility of power lines / electric towers – 192		river / stream – 2,597
smoke / fire – 1,248	shrubs – 704	railway – 108		
	snow / ice - 56	road – 1,612		
	Trees – 1,480	water tower – 76		
		wireless / radio communication towers - 65		

Table 2: LADI Dataset images per label.

belong to one of the following categories: damage, environment, infrastructure, vehicles and water. The full overview of the categories and the labels inside each category can be found in Table 2. These images were used for training and validating our model. One major issue that was noticed in the provided dataset, is that the classes were greatly unbalanced. Table 2 depicts apart from the name of classes, the exact number of images for each class.

Regarding the testing dataset, it consists of 1825 video-shots of various orientation, resolution and duration. Given the large number of video frames, we considered a video segmentation service [2] called "The Video Shot and Scene Segmentation" for extracting keyframes of each of the provided videos.

3.2 Approach

The approache we followed is based on a deep neural network (DNN) that resembles VGG [20] in order to calculate the probability of each class. The model consists of three blocks where each one includes two consecutive Convolutional layers with 3×3 filters, a max pooling and a dropout layer. Then, a flatten layer, a dense and a dropout layer. Eventually, a Dense layer with a sigmoid activation provides the probability for each of the 32 classes. The binary cross entropy loss function applied. A difficulty that we encountered regarding the dataset, was that the sets of images had multiple resolutions; while as far as their orientation is concerned most of them had landscape orientation and a few had portrait. However, we need a unique resolution for all of them, in order to use them as input in our VGG [20] like architecture. Thus, we resized the images to the same resolution, preserving the aspect ratio of all images to prevent distortions. Thus, the most seen aspect ratio of 3:2 was used. In order to preserve the target resolution of 350×350, we added zero padding.

Then, we split the dataset to 70% for training and 30% for testing. The best results were observed when Adam optimizer and a learning rate of $0,001$ were used (figure 3). Also, the batch size was set to 10 and the model was trained for 100 epochs. The metric used for evaluating the model's performance was the F-score, and the best value that was achieved was $0,417$ while the loss reached $0,409$.

3.3 Submission

For DSDI, we submitted a single run that considered only the manually annotated dataset. Also, given, that the DSDI' task output is a file with the top-1000 videos for each class that were picked from the 1825 video-shots of the final test set, an extra step was added that provides a summarization of the concepts found on image level and then transforms them to video level. This is achieved by

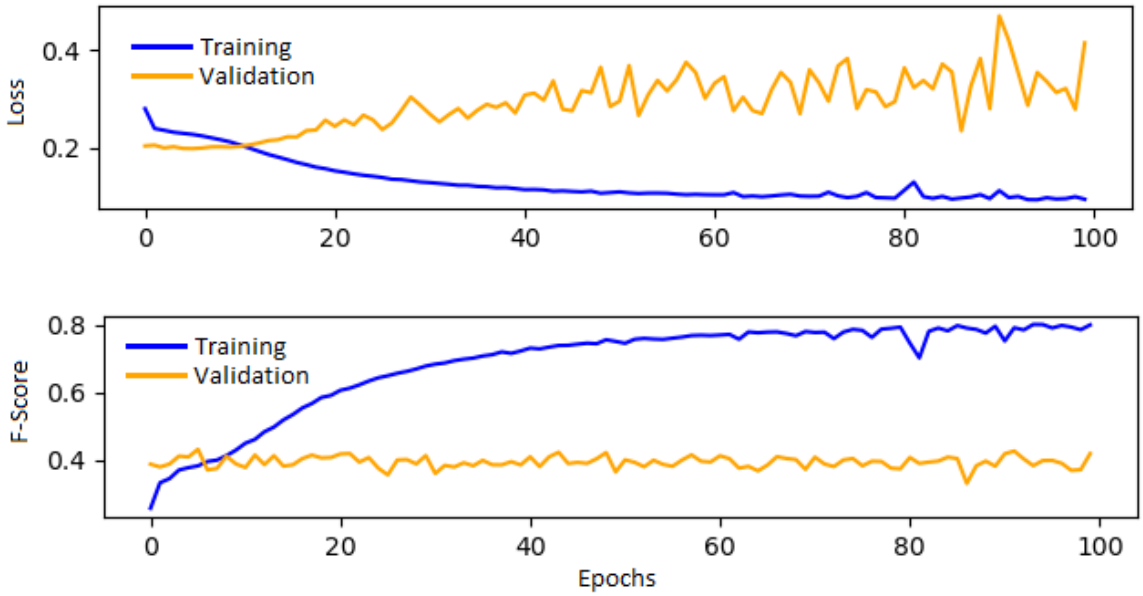

Figure 3: F-score and Loss for our Deep Neural Network model.

calculating the mean of the probability values of the keyframes extracted for each video that was provided as input in the test set.

Therefore, for each video-shot, the keyframes were extracted, they were then passed through the classifier and the mean probability was estimated for each video-shot. Eventually, the top-1000 videos for each class were submitted, resulting to an F-score of $0,076$. We noticed that reduction of the size of the initial high-resolution images resulted to decrease of the detail of the images, which made difficult to discriminate smaller objects, thus favoring classes like miscellaneous damage or flooding.

4 Activities in Extended Video

The task of detecting Activities in Extended Video (ActEV) challenge requires both the accurate recognition and temporal localization of actions that a video includes. To this end, the proposed approach consists of two steps; the first focuses on the temporal activity detection and the second on the classification of the resulting temporal boundaries using deep neural network architectures. For the first step, we utilized a real-time object detector YOLOv4 as proposed by [21], in order to detect and track objects related to target activities, generating temporal boundaries for each detected object. On the second step, we used a classifier to assign an activity label to each temporal boundary. In our experiments, we used 3D-ResNet [22] and Inception I3D [23] models for the classification task.

The rest of the section is organized as follows: The next subsection (4.1) gives an overview for the objective of the submission while subsections 4.2 and 4.3 outline the methods of the action detection and recognition, correspondingly. The section concludes with a report about the submitted systems in subsection 4.4 and a discussion of the results in subsection 4.5.

4.1 Objective of the Submission

The ActEV evaluation is accomplished using data collected from the VIRAT [24] video dataset. Specifically, the VIRAT dataset is a benchmark dataset for computer vision tasks, as it is designed to be realistic, natural and challenging for video surveillance domains in terms of its resolution, time, diversity in scenes, human activity etc. in contrast to other action recognition datasets that are generally describe trimmed activities. More specifically, it is divided into two sets, VIRAT-V1 and VIRAT-V2, from which only 214 and 150 videos respectively were used for the challenge. A further

[2]https://mklab.iti.gr/results/video-shot-and-scene-segmentation/

Target activities		
person_closes_facility _or_vehicle_door	person_closes_trunk	vehicle_drops_off_person
person_enters_facility_or_vehicle	person_exits_facility_or_vehicle	person_interacts_object
person_loads_vehicle	person_opens_trunk	person_opens_facility _or_vehicle_door
person_person_interaction	person_pickups_object	vehicle_picks_up_person
person_pulls_object	person_pushs_object	person_rides_bicycle
person_sets_down_object	person_talks_to_person	person_carries_heavy_object
person_unloads_vehicle	person_carries_object	person_crouches
person_gestures	person_runs	person_sits
person_stands	person_walks	person_talks_on_phone
person_texts_on_phone	person_uses_tool	vehicle_moves
vehicle_starts	vehicle_stops	vehicle_turns_left
vehicle_turns_right	vehicle_makes_u_turn	

Table 3: Target activities in ActEV challenge 2020.

Type of dataset	Number of videos	Number of activities
Train	64	4311
Validate	54	3521
Evaluate	246	-

Table 4: ActEV challenge 2020 dataset sets.

division happens due to the need for more reliable evaluation, to training, validation and evaluation sets. The numbers of the videos for each set are depicted in Table 4. Finally, the 35 target activities as they are provided from the annotations, are presented in Table 3.

4.2 Activity detection module

In this subsection, further details about the generation of temporal boundaries are described giving the pipeline that we followed. At first there is an overview about the object detector that was used and after more analysis, about the steps of the post-processing method, is presented.

4.2.1 Object detection

To tackle the problem of activity detection in untrimmed videos, we begin approaching it as a problem of activity localization in space at first. To this end, we deploy an object detector to localize objects in every frame for each video. Inspired by the fast and accurate performance of YOLOv4 [21] in real-time object detection, we adopt it as a means to capture this spatial information. YOLOv4 [21] composes an improved version of YOLOv3 [25] as it combines a fast operating speed of an object detector in production systems and optimizations for parallel computations building a state-of-the-art detector. For our experiments, we utilized the pre-trained model of YOLOv4 [21] using Microsoft COCO [26] dataset to extract the objects of every video in the validation and evaluation sets. We have selected this dataset as it consists of the objects that participate in the activities of the challenge's dataset, including but not limited to "person", "car" and "truck". The detected objects for each frame are described by a bounding box and the corresponding confidence score.

As a natural component of an object detector, we employed an object tracker to link the objects in time (over the frames) and calculate their trajectories. The algorithm of the object tracker calculates the Euclidean distance between the current object on frame t and the proposal objects on frame $t+1$. The closest object is selected as the future position of the current object. The Euclidean distance is calculated between the centroids of the objects' bounding boxes.

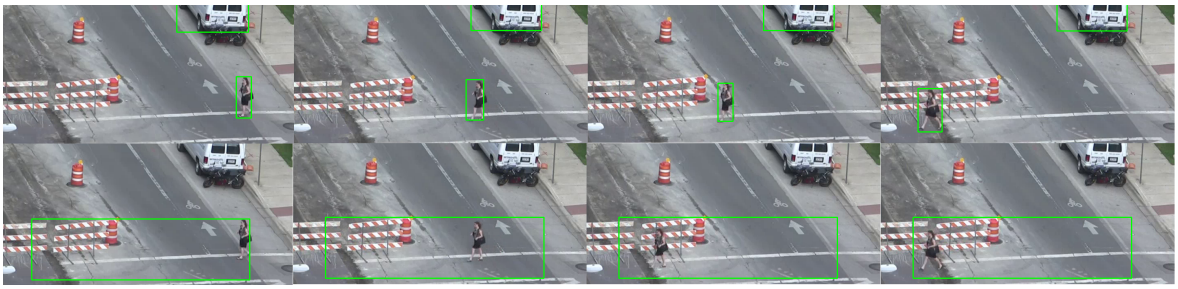

Figure 4: First row: frames sequence with detected objects of YOLOv4 [21]. Second row: frames sequence with the EABBox of the selected objects after post-processing.

4.2.2 Post-processing

In order to have ready the tracked objects for the final step of activity classification, it's required an additional procedure. This briefly includes the selection of the objects of interest and the creation of the Extended Activity Bounding Box (EABBox) for every detected object. The last involves the bounding box generated by the union of the separated bounding boxes, for a specific object, during its tracking as illustrated in figure 4. It noted that we are able to calculate EABBoxes as the videos comprising the dataset are totally captured using static cameras. During the experiments, lots of strategies were examined in order to select the best post-processing method suited for the task of ActEV. The strategies that reported better performance on the validation set are described below.

4.2.2.1 Post-processing version 1

At the first stage, the objects of every video were processed with the steps that follow:

1. Process only the selected frames according to the annotations of the dataset.
2. Select the objects with class name belong to one of the following categories: "person", "car", "bus", "truck" and "bicycle".
3. Get the predictions that have confidence score above a certain threshold. The value of the threshold was set at 25% empirically to deal with the wrong detections of YOLOv4 [21].
4. Select only one class for every object. In the case that YOLOv4 [21] gives for a specific object two or more labels ("car" and "truck") we keep the one class with the biggest confidence score. This is tested only in the classes "car", "bus", "truck" and "bicycle" not in the "person" class. In this way, we avoid including the same object twice in the procedure. This is tested using the Intersection over the Union (IoU) with a certain threshold as a metric for overlapping. The threshold was set at 70% experimentally.
5. Select only the moving objects. The detection of moving objects happens checking the start and the end position, so in the case that these positions are close enough the object is excluded as considered static. The class "person" is an exception in this step as the motion of one person is not comparable with a motion of vehicles from the aspect of distance and velocity.
6. Merge objects. Considering that there are activities which describe the interaction of two objects, we implement a step that merges two objects according to three criteria:
   (a) A pair that is under investigation has to include the class "person" with any other vehicle class ("car", "bus", "truck" or "bicycle").
   (b) Temporal overlapping with less than 30 frames difference.
   (c) Spatial overlapping with IoU bigger than 70%.
7. Calculate the EABBox for each object as illustrated in figure 4.

4.2.2.2 Post-processing version 2

In a way to reduce the steps of post-processing and check what really affects the performance of the algorithm we implement a more abstract method which is described below:

1. Process only the selected frames according to the annotations of the dataset.
2. Select the objects with class name belong to one of the following categories: "person", "car", "bus", "truck" and "bicycle".
3. Select only the moving objects, similarly to post-processing version 1 (4.2.2.1).
4. Examine the immobility of an object at short intervals of time, which experimentally set at 10 seconds. Considering that there are no activities for static vehicles we make a more detailed attempt to exclude all the time slots where a vehicle does not move in its total trajectory. To this end, after this step we keep only the slots of time that an object moves, as separate activities, splitting long events that maybe was misleading for the classification task.
5. Calculate the EABBox for each object as shown in figure 4.
6. Sequences with less than 20 frames were rejected.

4.3 Activity recognition module

Given the temporal boundaries of the tracked objects, the next step is to recognize their classes. For this task, we trained two activity classifiers a 3D-ResNet [22] and an Inception I3D [23]. The selection of these networks was highly correlated with their ability to process the data in a 3D state exploiting also the temporal information that they include in contrast to 2D Convolutional Neural Networks (CNNs) that can learn only spatial correlations. For the training process, we utilized the provided videos of the training and validation set, extracting the EABBox for each activity of the annotation files. As previously described EABBox takes into account the bounding boxes of the objects inside the activity, creating the total region of object's motion as referred in the post-processing steps. The training was treated like a multi-class classification task.

4.3.1 3D-ResNet

3D-ResNet is a deep neural network model which comprises a 3D-convolutional based architecture achieving faster processing appropriate for activity recognition in real-time state. Also, it simultaneously uses (batch) frame processing. In particular, the architecture with 50 layers [22] has been implemented. Specifically, the architecture consists of bottleneck blocks, where each block is consisted of three 3D-convolution layers followed by batch normalization and ReLU activation layers, with the convolution kernels being 1x1x1 for the first and the third convolution layers while the intermediate layer a size of 3x3x3 is applied. It should be noted that the weights of the Kinetics dataset [27] were pre-loaded. The Kinetics dataset was selected for the application as it covers a large number of human activity classes (400 classes). About the training process, 3D-ResNet samples 16 continuous frames per activity and also the total number of epochs were 750 from which was selected the best epoch based on the validation loss.

As it was observed the class labels of the dataset are highly imbalanced, which makes a classifier focus more on the most frequent data classes. One way to deal with this problem was to adapt a weighted cross-entropy loss [28]. The last was used as an improvement in the training process of the 3D-ResNet.

4.3.2 Inception I3D

The second classifier, that was tested, uses an architecture of a Two-Stream Inflated 3D ConvNet (I3D) [23] which is based on a 2D ConvNet inflation. This means that filters and pooling kernels of deep image classification convolutional networks are expanded from 2D into 3D. As a result, it becomes possible for a network to learn spatial and also temporal information that a video includes. An important clue is that with the inflation it can be leveraged successful ImageNet architecture

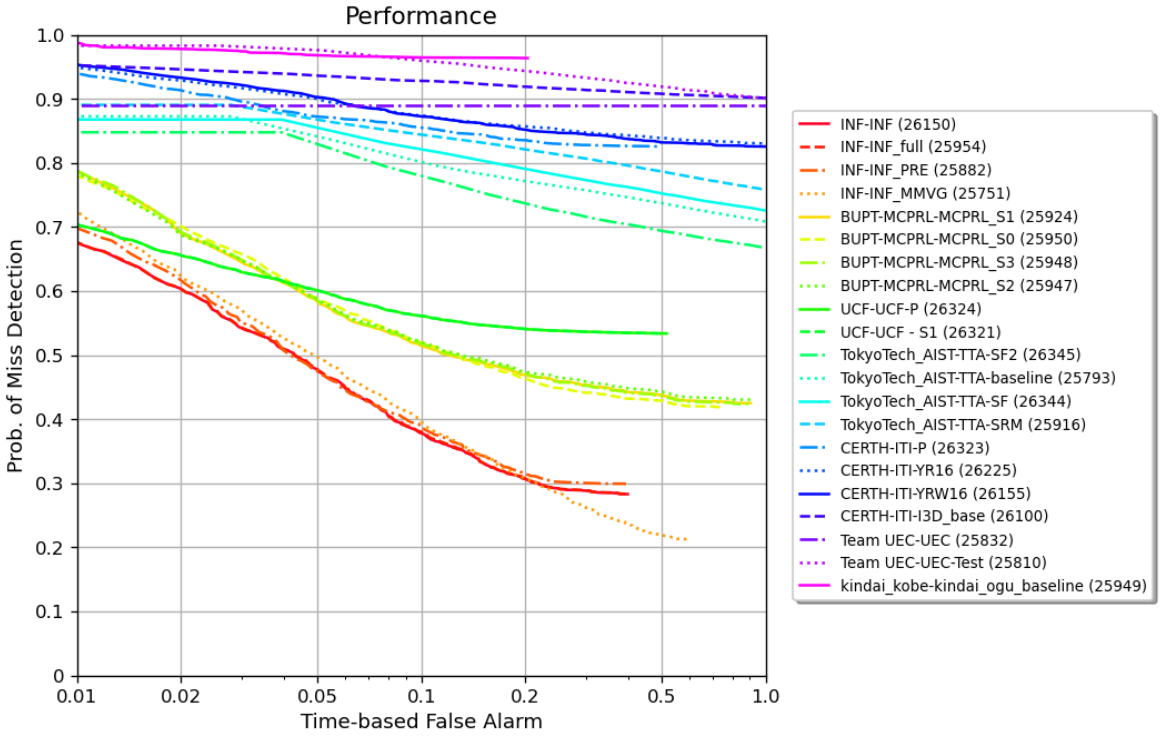

Figure 5: ActEV 2020 Leaderboard.

designs and even their parameters. In the experiments, we exploited this new architecture using Inception [29] model trained on the Kinetics dataset [27]. About the training process, Inception I3D samples 32 continuous frames per activity which means activities with less than 32 frames in total are rejected. Furthermore the epochs of training were 1000 from which was selected the best epoch based on the validation loss.

4.4 Submitted Systems

In this section, we present the four systems that we submitted and the evaluation results for TRECVID 2020 on figure 5 and on table 5.

- CERTH-ITI-I3D_base: It is the baseline of our method and combines YOLOv4 [21] and post-processing version 1 for the activity detection task, with Inception I3D [23] model fine-tuned using Actev20 dataset for the activity classification task.
- CERTH-ITI-YRW16: This system contains YOLOv4 [21] and post-processing version 1 for the activity detection task and 3D-ResNet [22] fine-tuned using Actev20 dataset for the activity classification task. We added the weighted cross-entropy loss during training of the 3D-ResNet [22].
- CERTH-ITI-YR16: This system contains YOLOv4 [21] and post-processing version 1 for the activity detection task and 3D-ResNet [22] fine-tuned using Actev20 dataset for the activity classification task. An update of the previous version considering only activities more than 20 frames.
- CERTH-ITI-P: Our last system includes YOLOv4 [21] and post-processing version 2 for the activity detection task and 3D-ResNet [22] model fine-tuned using Actev20 dataset with the weighted cross-entropy loss for the activity classification task.

System Name	PARTIAL AUDC	MEAN-P MISS@0.15TFA	MEAN-W_P MISS@ 0.15RFA
CERTH-ITI-I3D_base	0.93125	0.92318	0.92850
CERTH-ITI-YRW16	0.88530	0.86136	0.91187
CERTH-ITI-YR16	0.88511	0.86165	0.89439
CERTH-ITI-P	**0.86576**	**0.84454**	**0.88237**

Table 5: Our ActEV challenge 2020 results, ranked using PARTIAL AUDC as primary metric.

4.5 Experimental results

In this section, further discussion about the performance of every submitted system is reported. Firstly, regarding the results of our baseline method, CERTH-ITI-I3D_base, where Inception I3D [23] was used, we can obviously underling that the constrain of using activities with more than 32 frames, maybe stands ambiguous. From one aspect, losing small activities (with less than 32 frames) affects the training of the network which now focuses more on bigger activities reducing its reliability. Despite this, it was observed in validation set that very small activities are false alarms produced from wrong detections of YOLOv4 [21]. From the above, we can conclude that a constrain of rejecting small activities can be added to our pipeline but with a further examination of the right threshold value. After experiments, we keep the activities with more than 20 frames as it was added at the last step of the post-processing version 2.

In the second submission, CERTH-ITI-YRW16, we changed the activity classifier to 3D-ResNet [22] as a way to see how different classifiers perform to the task of assigning a label to the proposed temporal boundaries. From the table 5, it's clear that this addition improved the results. This improvement may rely on that 3D-ResNet [22] doesn't have any constraints about the length of activities' sequences. As a way to tackle the problem of the imbalanced dataset we also added the weighted cross-entropy loss at the training process of the network 3D-ResNet.

In the third system, CERTH-ITI-YR16, we evaluate the same system to CERTH-ITI-YRW16 but considering only the activities with more than 20 frames because as already discussed small activities are usually false detections generated by YOLOv4. The last observation was also confirmed after a detailed examination of the training and validation sets, which mainly contain activities with more than 20 frames. Despite our expectations, the results got slightly improved showing that this part does not affect so much to the total performance.

Having done experiments at the domain of activity classifier we focused next at the post-processing method. Our last submission CERTH-ITI-P shows that splitting long activities with the criterion of keeping only the slots of time that an object vehicles moves as different activities, improved the performance. Also the new constrain, of drooping detected activities with small length (smaller than 20 frames) may contributes at the outcome.

5 Conclusion

In this paper we reported the ITI-CERTH framework for the TRECVID 2020 evaluation [10]. ITI-CERTH participated in the AVS, DSDI and ActEV tasks in order to evaluate new techniques and algorithms. Regarding the AVS task, we utilized an attention-based cross-modal network to learn a new common feature space for the text and video instances. Moreover, the combination of multiple training configurations of this network leads to good results. For the DSDI, our approach is based on a multi-label classifier that trained using the providing annotated data. During inference, the predictions are generated by averaging the probabilities of keyframes for each video shot. At ActEV task a method that firstly detects target objects of interest and subsequently generates their trajectories and final calculates the corresponding proposal boundaries and assign them to predefined categories. Though the results are not the expected ones some aspects of the process seem promising and we intend to

intensify our effort for finer system tuning and proper model training in the future.

6 Acknowledgements

This work was partially supported by the European Commission under contracts H2020-832876 aqua3S, H2020-786731 CONNEXIONs, H2020-833115 PREVISION, and H2020-780656 ReTV.

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] F. Markatopoulou, A. Moumtzidou, D. Galanopoulos, and K. Avgerinakis et al. ITI-CERTH participation in TRECVID 2017. In *TRECVID 2017 Workshop*. NIST, USA, 2017.

[8] Konstantinos Avgerinakis, Anastasia Moumtzidou, Damianos Galanopoulos, Georgios Orfanidis, Stelios Andreadis, Foteini Markatopoulou, Elissavet Batziou, Konstantinos Ioannidis, Stefanos Vrochidis, Vasileios Mezaris, et al. Iti-certh participation in trecvid 2018. *International Journal of Multimedia Information Retrieval*, 2018.

[9] Konstantinos Gkountakos, Konstantinos Ioannidis, Stefanos Vrochidis, and Ioannis Kompatsiaris. Iti-certh participation in trecvid 2019. 2019.

[10] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA, 2020.

[11] D. Galanopoulos and V. Mezaris. Attention mechanisms, signal encodings and fusion strategies for improved ad-hoc video search with dual encoding networks. In *Proc. of the ACM Int. Conf. on Multimedia Retrieval*, (ICMR '20). ACM, 2020.

[12] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang. Dual encoding for zero-example video retrieval. In *Proceedings of IEEE Conf. CVPR 2019*, pages 9346–9355, 2019.

[13] T. Mikolov, G.s Corrado, K. Chen, and J. Dean. Efficient estimation of word representations in vector space. In *1st Int. Conf. on Learning Representations, Workshop Track Proceedings*, ICLR '13, 2013.

[14] J. Dong, X. Li, and C. G. Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia (TMM)*, 20(12):3377–3388, Dec 2018.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[17] F. Faghri, D. J. Fleet, et al. VSE++: Improving visual-semantic embeddings with hard negatives. In *Proc. of the British Machine Vision Conference (BMVC)*, 2018.

[18] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of IEEE CVPR 2016*, pages 5288–5296, 2016.

[19] Y. Li, Y. Song, L. Cao, J. Tetreault, et al. TGIF: A new dataset and benchmark on animated gif description. In *Proceedings of IEEE CVPR 2016*, 2016.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[22] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[23] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[24] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160. IEEE, 2011.

[25] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[27] Andrew Zisserman, Joao Carreira, Karen Simonyan, Will Kay, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, et al. The kinetics human action video dataset. 2017.

[28] Yuri Sousa Aurelio, Gustavo Matheus de Almeida, Cristiano Leite de Castro, and Antonio Padua Braga. Learning from imbalanced data sets with weighted cross-entropy function. *Neural Processing Letters*, 50(2):1937–1949, 2019.

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.