NII_Hitachi_UIT at TRECVID 2018

Sang Phan [1], Zheng Wang [1], Aimin Su [2],
Martin Klinkigt [3], Mohit Chabbra [3], Manikandan R [3],
Duy-Dinh Le [4], and Shin'ichi Satoh [1]

[1] National Institute of Informatics, Japan
[2] University Grenoble Alpes, France
[3] Hitachi, Ltd., Japan
[4] University of Information Technology, VNU-HCMC, Vietnam

1 TRECVID 2018 Instance Search: Searching Specific Persons in Specific Locations

Abstract. This report presents the proposed system of our team for TRECVID Instance Search task. In this year system, we extend the focus from time point (single video shot) to time slice (multiple consecutive video shots), because directly aggregating the results shot by shot will not obtain satisfactory results. Different from the systems in previous years, 1) we first obtain person retrieval and location retrieval results by FaceNet, DIR and BOW. 2) we propose a Progressive Extension and Intersection Pushing (PEIP) method to obtain the combination results. Based on these improvements, our team acquires promising results in TRECVID INS 2018.

1.1 Introduction

Fig. 1: Examples for general INS task (top) and person-location pair INS task (bottom). For the general INS task, the system only needs to search out corresponding shots with certain objects, such as *this washing machine* and *the painting*. If the certain object appears in the shot, the shot is the positive one. For person-location pair INS task, the system requires to search out the target person in the right location at the same time, such as "*Dot in Kitchen1*" and "*Shirley in Market*". If target person or target location does not appear simultaneously, the shot will be considered as a negative one. The examples are selected from the TV series Eastenders. Programme material copyrighted by BBC.

Since 2016, the INstance Search (INS) task of TRECVID has started to ask participants to search out shots, which contain a certain person appearing in a certain place [1]. This year, TRECVID INS

kept the format of compound queries. This type of query has many applications in practice such as: surveillance systems, personal video archive management. In detail, given a probe topic, which includes a set of known location example videos and a few images with indicated person, the task is to search out 1000 video shots most likely to contain a recognizable instance of the person in the known location [2,3,4]. Although the general INS task, focusing on a single target independently, has already been well studied, the person-location pair INS task aiming at doing the retrieval job based on two different kinds of instances simultaneously, is challenging and just catching up (Fig. 1 shows the difference with some examples).

As we know, in TV series, person may appear in any angle or any corner of the location. Person's appearance always varies, and scene's viewpoint always changes. It makes the content of person-location pair instance ever-changing. Hence, a person-location pair instance must not be taken as an integral whole. Most of existing methods utilize person INS and location INS modules apart, search for target persons and locations respectively, and finally combine the results together to generate ranking lists. However, person INS module and location INS module are not always effective, in particular at the same time. Fig. 2 gives some examples.

Fig. 2: Difficult samples. In (a) and (b), person faces are non-front or occluded. In (c) and (d), locations are with low light or blur. In (e) and (f), although it is a wide-angle view-scene, the person faces are very small. In (g) and (h), locations are blocked by persons.

- Person INS module may be not effective because faces are not detected when they are non-front or occluded (as shown in Fig. 2(a) and Fig. 2(b)). It should be mentioned that most person INS modules in TV series applications rely on face clues.
- Location INS module may be also not effective because of blur and low light (Fig. 2(c) and Fig. 2(d)).
- In a wide-angle view, location INS module performs well, but person INS module is constrained because persons (in particular faces) in this kind of locations are very small (Fig. 2(e) and Fig. 2(f)).

– Locations may be blocked by persons, which makes location INS module suppressed (Fig. 2(g) and Fig. 2(h)).

Consequently, directly aggregating the results shot by shot will not obtain satisfactory results, if one of the INS modules is suppressed at some shots. Luckily, for the TV series, video shots are arranged in chronological order. If one target person/location is captured in a single shot, it is very likely that this person/location will also appear in the neighbor shots in the time-line. Inspired by the video consecutiveness, we extend the focus from time point (single video shot) to time slice (multiple consecutive video shots). To deal with this type of INS task, 1) we first obtain person retrieval and location retrieval results by FaceNet [5], DIR [6] and BOW. 2) we propose a Progressive Extension and Intersection Pushing (PEIP) method to obtain the combination results. 3) Finally, we filter the irrelevant shots, for example, *outdoor scene shots*, and *shots in video #0*.

1.2 Our method

The proposed framework of person-location pair INS task is shown in Fig. 3. In the following, the modules of the framework will be illustrated in detail.

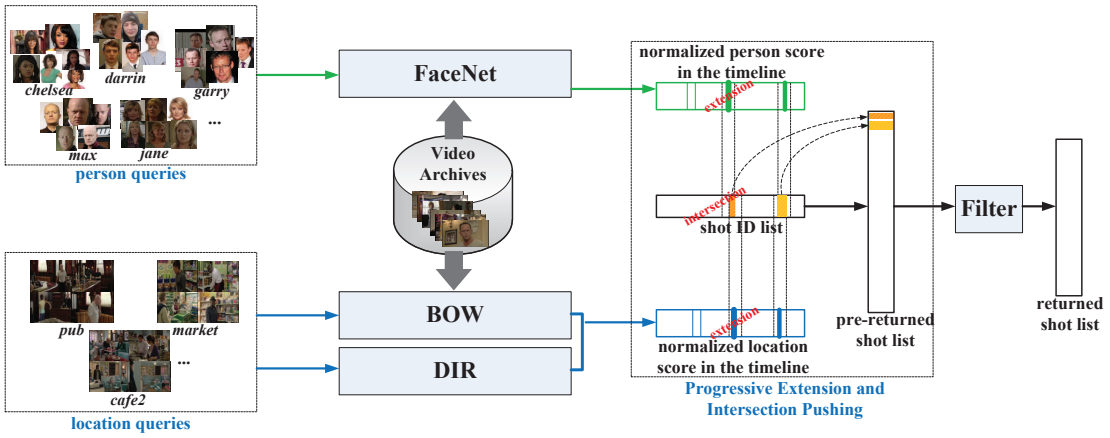

Fig. 3: The framework of our method. It includes four parts. 1) FaceNet stands for the person INS module, and its output is the similarity score for each shot based on the query person. 2) BOW and DIR are used for the location INS. BOW is our traditional retrieval part with hand crafted feature. DIR is the new retrieval part based on deep learning feature. 3) The proposed PEIP module pushes the pre-returned results progressively. 4) After filtering irrelevant shots, the final returned results are obtained.

Person INS Module We use FaceNet to obtain the similarity score between each shot and each person query topic. For each shot, it includes a sequence of key frame images. For each person query topic, it includes ten face images with different views, which were additionally collected off-line. We use MTCNN [7] to detect and align face image for each query and gallery image. Then, the FaceNet extracts face feature for each face image, where the FaceNet was pre-trained on the

VGGFace2 dataset. We use the max-pooling strategy to achieve the similarity between one shot and one query topic. Finally, we normalize all the similarity scores of all gallery shots based on the query topic.

Location INS Module Hand crafted based part: similar to last year system, we retrieve shots containing the query location. For the local feature based approach, we use Bag-of-Visual-Word (BOW) model for location retrieval.

Deep learning based part: we also use Deep Image Retrieval (DIR) feature to search for the shots with certain location. Query Extension Technique is used to combine the features of the sequence of given query images together. Then, we also use the max-pooling strategy to achieve the similarity between one shot and one query topic. Finally, we combine the results of hand crafted based part and deep learning based part together, and normalize all the similarity scores of all gallery shots based on the query topic.

Progressive Extension and Intersection Pushing The propose a Progressive Extension and Intersection Pushing (PEIP) method to obtain a suggestive returned shots. As we know, since person INS module and location INS module are not always effective, in particular at the same time, we cannot aggregate the similarity score together directly. However, the video has its consecutiveness. If one target person/location is captured in a single shot, it is very likely that this person/location will also appear in the neighbor shots in the time-line. We extend the focus from time point (single video shot) to time slice (multiple consecutive video shots).

The PEIP method consists of multiple iteration processes. For each iteration, we select the shots with high similarity score, and extend the shots with neighbor shots. We consider that these shots are highly possible to be the target shots which contains the certain person/location. Then, we get the intersection of the shots respectively from person results and location results. The intersection shots are put to the top of pre-returned shot list. For the following iterations, new intersection shots are pushed after that.

Filtering As we know, all the given topics are indoor scenes, so we can filter outdoor scenes. We filter shots with the vehicle categories of the ImageNet 1000 categories. Following [8], from the results of ImageNet classfication, we select 37 categories about vehicles, such as *ambulance*, *minibus* and *police van*. When the score of classification result of any category is more than 0.3, the image is judged to include vehicles. If all key frames in the shot have retrieved vehicles, the shot is considered as an outdoor scene and filtered out. Analogously, shots with other 52 categories (such as *hippopotamus*, *Indian elephant* and *castle*) only appear outdoor, and should be filtered as well.

As the locations of shots are mutually exclusive, we can filter the shots that must be in the other locations. For example, if the target retrieval location is '*Cafe2*', we can filter the shots that their location is with a high probability to be '*Market*', '*Pub*', '*Laundrette*'. For each location, we select the top 3000 of ranking results of location INS module as the certain shots for filtering.

We also filter the shots in video #0, *i.e.*, the shots with prefix-ID '*shot0_*'.

1.3 Results

We submitted 4 automatic runs and 1 interactive run, and results of our submissions on Instance Search task of TRECVID 2018 are shown in Tab.1. In the table, 'Extension' stands for the number of neighbor shots extended for each high similarity score shot. 'Iteration' stands for the times of intersection shots pushing. 'Shots before Intersection' stands for the number of shots selected before intersection both in the person score timeline and location score timeline.

The code of our method: https://github.com/wangzwhu/INS2018

RUN-ID	MAP	Method
F_NII_Hitachi_UIT_1	0.369	Extension = 6, Iteration = 50, Shots before Intersection = 100
F_NII_Hitachi_UIT_2	0.362	Extension = 12, Iteration = 69, Shots before Intersection = 100
F_NII_Hitachi_UIT_3	0.317	Extension = 10, Iteration = 5, Shots before Intersection = 1000
F_NII_Hitachi_UIT_4	0.287	Extension = 10, Iteration = 6, Shots before Intersection = 1000
I_NII_Hitachi_UIT_1	0.367	Delete Negative Samples from F_NII_Hitachi_UIT_1

Table 1: Results of our submitted 5 runs on Instance Search task of TRECVID 2018.

2 TRECVID 2018 Ad-hoc Video Search: Combining Concept Features and Dependency Features

Abstract. Ad-hoc Video Search is a challenging problem in TRECVID evaluation [2]. This is due to the high semantic gap between the text query and the video content. A rich source of semantic information is video metadata e.g. title, summary, or textual transcript provided by video owners. However, such amount of semantic information is still far from enough to fully describe video content as it can be observed by human being. Hence, it causes low accuracy in searching videos with complex query. Our approach towards enriching semantic description and presentation is combining concept-based representation and dependency-based representation. Experimental results show that dependency features are complementary to concept features for this task. However, using only dependency features is not reliable because of its sparsity in the text query as well as in the video representation.

2.1 Introduction

With the rapid growth of video data from many sources such as social sites, broadcast TVs, films, one of the most fundamental demand is to search a particular video in huge video databases. In some cases, users did not see any target video shots before. No visual example is provided. The input query could be a text string with ad-hoc description about the content they want to search. Fig 1. gives an example of this query type, "finding shots of a man lying on a tree near a beach".

To deal with AVS query type, high-level features (i.e. semantic based features) are usually extracted to match againts the text description. To this end, we leverage high level feature using deep convolutional neural network (CNN). Because the query topics given by users are unpredictable, we combine multiple concepts from multiple datasets including ImageNet [9] and Places database [10] to cover popular topics that users may be interested in.

To further capture the semantic information from the video, we propose to use the dependency matching method. Dependencies are syntactic relations like subject and object that represents a relationship between concepts. Therefore, dependency representation can convey a richer level of semantic information which can not be found from encoding individual concepts. This idea is related to our previous work [11], in which we utilized the dependencies obtained from image captions for video event detection. Different from last year, in this year we experimented with more concepts in our concept collection.

2.2 Concept Extraction

In this section, we propose to extract semantic features to match with ad-hoc query given by users.

Since the number of concepts is unlimited and the query of the user is unpredictable, to increase the recall of the system, we propose to extract as much semantic description and presentation of a video at frame-level as possible. Inspired by recent success of deep learning techniques, we also leverage the powerful of deep features in semantic search task. In this system, semantic concepts includes:

- Objects: We mainly rely on object detection neural networks that was pretrained on ImageNet. Specicially, we use VGG16 [12] to extract 1K concepts. Futhermore, ImageNet Shuffle [13] concepts leverage the complete ImageNet hierarchy with more than 21K classes from 14M images. We took different variants of ImageNet Shuffle as pre-trained deep networks for concept extraction.

Fig. 4: Example result of finding shots of a man lying on a tree near a beach.

Table 2: List of concept banks and its number of concepts

Concept bank	Number of concepts
TRECVID SIN	345
ImageNet 1000	1,000
ImageNet + Places	1,365
ImageNet Shuffle 4000	4,000
ImageNet Shuffle 4437	4,437
ImageNet Shuffle 8201	8,201
ImageNet Shuffle 12988	12,988
ImageNet Shuffle 21841	21,841
All	54,177

– Scene Attributes: includes indoor/outdoor labels, building, park, kitchen etc.. In our system, the attributes are extracted from the state-of-the-art models trained on MIT scene and SUN attribute dataset [10].
– TRECVID SIN345 concepts [14]. We use the concept detection scores for the IACC.3 dataset that are shared by the ITI-CERTH team [15].
In total, our concept banks contain more than 54K concepts. The detail number of concepts per collection can be found in Table 2. Different from last year, features from all frames are aggregated using the max pooling approach.

2.3 Dependency Extraction

We propose to select the co-occurrence concepts in a systematic way based on the syntactic dependencies. The motivation behind using dependency matching is simple. For instance, consider this AVS query: "Find shots of a policeman where a police car is visible". In this query, the dependency "police car" is crucial for searching. If we only use concept-based representation, we might able to search videos that contains both "car" and "police" but might not be "police car". Dependency representation can resolve this ambiguity.

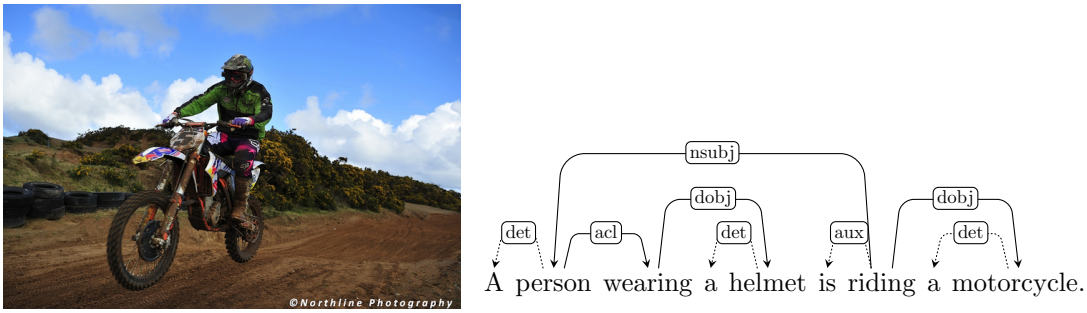

Fig. 5: Example of dependencies extracted from a text description.

Figure 5 shows examples of possible dependencies that can be extracted from the text description. The dependency tree of the caption is obtained by applying Stanford Parser [16]. In practice, we do not always have access to the full sentence description of an image. For examples, for dataset like ImageNet, Places or SUN, we only have the category labels of each image, which can be a word, a phrase, or several phrases. In this work, we directly apply the Stanford Parser [16] on those category labels to extract the dependencies, though for some classes, the dependency is not available.

2.4 Concept/Dependency Matching

After extracting semantic features including the concept-based features and dependency features, the searching task is now equivalent to text based retrieval task. This stage is to index semantic text returned from the previous stage. A standard *TF-IDF* scheme is used to calculate weight of each word. The pipeline of our system can be described in the following five steps.

– Step 1: Detect concepts and dependencies from text queries
– Step 2: Build the concept and dependency from the concept banks
– Step 3: Detect concepts/dependencies using the pretrained concept models
– Step 4: Calculate the dot product scores, weighted by TF-IDF
– Step 5: Late fusion to combine concepts and dependencies

2.5 Results

Table 4 shows the performance of concept and dependency matching. In all cases, concept-based matching performs better than dependency-based matching. This is reasonable because we observed that the dependency representation can only be obtained in a few queries. For most queries, we could not extract any dependency from the text queries that is also appeared in the dependency vocabulary (obtained from all the concept category labels). Therefore, performance of dependency features is zero in these queries. Combining both concept and dependency can provide a small performance gain, as shown in Table 5. Example queries where dependency matching performs better than concept matching are shown in Tabel 3.

Table 3: Example queries where dependency matching performs better than concept matching

Query	Concept results	Dependency results
533: a man sitting down on a couch in a room	0.0142 - down, room, sit, couch, man	**0.1272** - (man, sit), (sit, room), (sit, down), (sit, couch)
537: one or more people swimming in a swimming pool	0.3436 - one, swim, people, pool, more	**0.6038** - (pool, swimming), (people, more), (people, swim), (people, one)

Table 4: Results of using concept and dependency matching on AVS 2017

Concept bank	Concept results	Dependency results
TRECVID SIN	0.0539	0.0254
ImageNet 1000	0.0474	0.0000
ImageNet + Places	0.0601	0.0102
ImageNet Shuffle 4000	0.0835	0.0124
ImageNet Shuffle 4437	0.0808	0.0144
ImageNet Shuffle 8201	0.0785	0.0138
ImageNet Shuffle 12988	0.0759	0.0126
ImageNet Shuffle 21841	0.0846	0.0145
All	0.1316	0.0411

Our best runs on previous year's test sets are the runs that combining both concept and dependency matching. Based on this results, we submitted three automatic runs to this year's Ad-hoc

Table 5: Results of our submitted runs in three AVS test sets

Our system	Concept results	Dependency results	Results of Concept + Dependency
AVS 2016	0.0620	0.0287	0.0718
AVS 2017	0.1316	0.0411	0.1381
AVS 2018	0.0300	0.0030	0.0310

Video Search task. Performance of these runs in this year's test sets can be found in the last row of Table 5. We observe a significant performance loss in this year. This may be the limitation of the concept-based and dependency-based matching method. In the future, we plan to learn a joint visual-semantic for the retrieval task, which can better bridge the semantic gap between the text query and the video content.

3 TRECVID 2018 ActEV: Activities in Extended Video

Abstract. We present in this paper the results and system developed for Activities in Extended Video (ActEv) task, which is a pilot task in TRECVID 2018. ActEV is an extension of the annual TRECVID Surveillance Event Detection (SED) evaluation by adding a large collection of multi-camera video data, both of simple and complex activities. We participated and submitted for both sub tasks of Activity detection and Activity-Object detection respectively for which we developed multiple systems based using combination of standard detector, tracker and trajectory with CNN's and LSTM. More specifically we divided system for subtask 1 into two parts with a subsystem consisting of Faster-RCNN detector [17], Simple Online Realtime Tracker (SORT) [18] and plotted trajectories with a CNN for detecting activities where notion of reference is key. For rest of the classes, we used an LSTM trained with CNN features of localized activity regions. For subtask 2, we use the localization information obtained from Faster-RCNN and SORT tracker of subtask 1. With these in place, we achieved a result of 0.977 among all of our runs.

3.1 Tasks

In ActEV evaluation, there are two subtasks focusing on temporal localization of activities and spatial localization of its objects from videos of multi-camera environment. Each of the subtasks and its systems are described in brief as follows.

Subtask 1: Activity Detection (AD) In this subtask, given an input video our system had to automatically detect and temporally localize all the instances belonging the proposed 12 target activities for phase 1.A evaluation.

We divide our overall system for subtask-1 into two parts, firstly a system **A** for set of target classes where the notion of reference is to considered. Example classes include 'vehicle turning left', 'vehicle turning right' and 'vehicle u turn'. For rest of the target classes we create a separate system **B**. Both of our system uses common set components namely Faster-RCNN, SORT Tracker that are described in next section along with a schematic representation.

Subtask 2: Activity and Object Detection (AOD) For the Activity and Object Detection task, our system had to output spatial localization of objects associated with activities identified in subtask 1. For predicting localization results of this task we use the bounding boxes produced by Faster-RCNN and SORT Tracker from subtask 1

3.2 ActEV System Overview

Figure 6 below shows an overview of our ActEV system. Our ActEV system consists of the following steps:

- **Preprocessing:** During training, we select frames corresponding to a given activity and during inference we select all the frames with block size of 64 and temporal stride size 1 leading to overlap of 16 frames in order of $1, 64 \rightarrow 16, 80 \rightarrow 32, 96$ and so on.
- **Person/Vehicle detection:** For this step, we use the Faster-RCNN method proposed by [17]. We used the publicly available pre-trained models without any fine tuning.

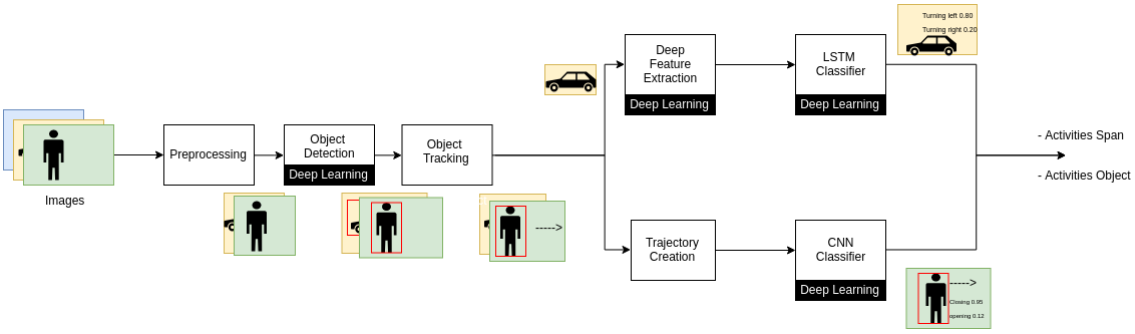

Fig. 6: NII_Hitachi_UIT ActEv system.

– **Object tracking:** After Person/Vehicle detection detection, the system associates detection regions across multiple frames by using SORT object tracker proposed by [18] which results in temporal coordinates of detected objects. This step also preserves identities of detected objects across consecutive frames if the IoU threshold of successive detection is greater than the predefined threshold used in our work.
– **Trajectory Generation/Alignment:** For the system **A** of subtask 1, we use consecutive tracking from previous step to create object trajectories, which are smoothed to remove noisy detections and plotted on an X-Y plane. The trajectories are then aligned such that the end point of the trajectories always lie on the Y-axis. Example trajectories are as shown in 7

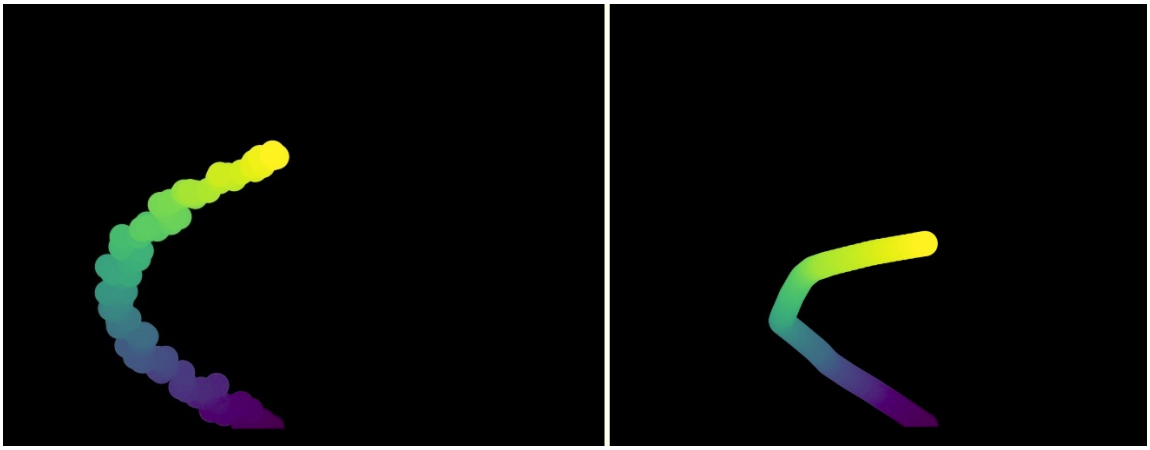

Fig. 7: Example trajectory plots, artificially generated (left) and generated by tracked id (right).

– **Deep feature extraction:** For the system **B** of subtask 1, We cropped the localized activity regions and fed them to a pretrained CNN to extract corresponding deep features.
– **Activity Classification:** We use two classifiers each for the system **A** and system **B** of subtask 1. System **A** uses a 3d-resnet101 convolutional neural network classifier which is trained using

plotted trajectories from and system **B** uses a single cell LSTM which is trained using deep features of localized objects.

3.3 Results and conclusion

Table 6: Results of submitted runs for TRECVID ActEV 2018 - Subtask 1

Run-ID	**Description**	**mean-p_miss@0.15rfa**
baseline		0.977662907268
trial		0.977662907268
init		0.987186716792

Table 7: Results of submitted runs for TRECVID ActEV 2018 - Subtask 2

Run-ID	**Description**	**mean-p_miss@0.15rfa**
baseline		0.988009085213
init		0.989653822055

We submitted a total of 3 and 2 runs using all systems described previously for subtasks 1 and 2 respectively. Table 6 and 7 shows run IDs, descriptions and performances in ***mean-p_miss@0.15rfa*** of each runs where their priority is sorted from the highest to lowest for both the subtasks.
The final result shows that separating the classes that are sensitive to notion of reference improved the performance a little bit. Since our ActEv system pipeline is heavily dependent on tracking, in our future work we would like to improve the tracker performance.

References

1. George Awad, Wessel Kraaij, Paul Over, and Shin'ichi Satoh, "Instance search retrospective with focus on trecvid," *International journal of multimedia information retrieval*, vol. 6, no. 1, pp. 1–29, 2017.
2. George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, David Joy, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Joao Magalhaes, David Semedo, and Saverio Blasi, "Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search," in *Proceedings of TRECVID 2018*. NIST, USA, 2018.
3. George Awad, Asad Butt, Jonathan Fiscus, David Joy, Andrew Delgado, Martial Michel, Alan F Smeaton, Yvette Graham, Wessel Kraaij, et al., "Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking," .
4. George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F Smeaton, Georges Quénot, Maria Eskevich, Robin Aly, Gareth JF Jones, et al., "Trecvid 2016. evaluating video search, video event detection, localization and hyperlinking," 2016.
5. Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
6. Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus, "Deep image retrieval: Learning global representations for image search," in *European Conference on Computer Vision*. Springer, 2016, pp. 241–257.
7. Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
8. Wang Zheng, Yang Yang, Guan Shuosen, Han Chenxia, Lan Jiamei, Shao Rui, Wang Jinqiao, and Liang Chao, "WHU-NERCMS at trecvid2016: Instance search task," 2016.
9. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
10. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 487–495. Curran Associates, Inc., 2014.
11. Sang Phan, Yusuke Miyao, Duy-Dinh Le, and Shin'ichi Satoh, "Video event detection by exploiting word dependencies from image captions," in *26th International Conference on Computational Linguistics (COLING)*, 2016, pp. 3318–3327.
12. Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
13. Pascal Mettes, Dennis C Koelma, and Cees GM Snoek, "The imagenet shuffle: Reorganized pre-training for video event detection," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, 2016, pp. 175–182.
14. George Awad, Cees G. M. Snoek, Alan F. Smeaton, and Georges Quénot, "Trecvid semantic indexing of video: A 6-year retrospective," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 187–208, 2016, Invited paper.
15. Nikiforos Pittaras, Foteini Markatopoulou, Vasileios Mezaris, and Ioannis Patras, "Comparison of fine-tuning and extension strategies for deep convolutional neural networks," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 102–114.
16. Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *ACL*, 2014, pp. 55–60.

17. Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.
18. Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.