NII_Hitachi_UIT at TRECVID 2019

Martin Klinkigt[1]*, Duy-Dinh Le[2]*,
Atsushi Hiroike[1], Hung-Quoc Vo[2],
Mohit Chabra[1], Vu-Minh-Hieu Dang[2],
Quan Kong[1], Vinh-Tiep Nguyen[2],
Tomokazu Murakami[1], Tien-Van Do[2],
Tomoaki Yoshinaga[1], Duy-Nhat Nguyen[2],
Sinha Saptarshi[1], Thanh-Duc Ngo[2],
Charles Limasanches[1], Tushar Agrawal[1], Jian Manish Vora[1],
Manikandan Ravikiran[1], Zheng Wang[3], Shin'ichi Satoh[3]

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

1 TRECVID 2019 ActEV: Activities in Extended Video

Abstract. We present in this paper the results and system developed for Activities in Extended Video (ActEv) task [1]. ActEV is uses a large collection of multi-camera video data, both of simple and complex activities. Hitachi system consisted of a typical pipeline with object detection, tracking and classification. More specifically Cascade RCNN detector [3] and discriminative correlation filter for tracking were used. The activities were split into three categories: Vehicle only, Vehicle plus person and person only activities. For each of those categories specialized classifiers have been developed.

1.1 ActEV System Overview

Figure 1 below shows an overview of our ActEV system.

Hitachi system for ActEV is composed of a pipeline style, similar to other teams. Compared to previous years system, we focus on improving the performance of the individual models in the system. The first step in this pipeline is detection. In the previous year we used Mask-RCNN [7] for object detection,

*Equal contributions

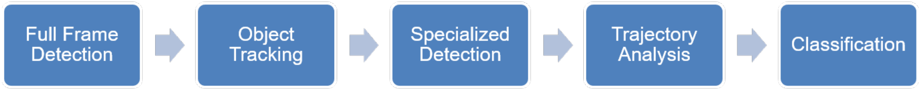

Figure 1: Hitachi ActEv system.

while for this year Cascade RCNN has been used. This switch allowed more and smaller objects to be detected. After detection, tracking is applied to generate trajectories. The previous year a simple SORT [2] algorithm has been used, which seeks to optimize the IoU of the detected objects. This year a combination of SORT with discriminative correlation filters has been implemented to better recover from lost and missed detections. The ID switches have been improved significantly, provided better tracks.

1.2 Activity detection and classification

Depending on the activity to be detected, different specialized classifiers have been implemented.

1.2.1 Vehicle only activities

The vehicle only activities include left turn, right turn and U-turn. For those activities the trajectory of the vehicle box is analyzed and the curvature of the track gives start, end and direction of the turn. At each frame up to a threshold of frames into the future a line of sight is drawn from the trajectory point of the current frame to the trajectory point of the future frame. If the orthogonal distance of any trajectory point and the line of sight exceeds a certain threshold, a turn will be predicted. The direction of the line of sight identifies the left or right turns. U-Turns are seen as a sequence of left and right turns or vice-versa.

1.2.2 Vehicle with Person activities

The eight vehicle with person activities included: Opening, Closing, Entering Exiting, Open Trunk, Closing Trunk, Loading and Unloading.

As training of any multi-class temporal network, either for temporal localization or activity classification is error prune due to the limited amount of training data, Hitachi did not deploy such an approach. Instead all those events have been trimmed to specialized classifiers for single tasks.

By using object detection of vehicles and persons from the pre-processing steps, specialized classifiers have been trained. Those specialized detectors include open doors and trunks of cars as well as carried objects. Individual events are detected and accuracy is improved by upper logic:

- Opening: open door highest confidence frame with fixed interval of event prediction

- Closing: no open door frame, after door was open with fixed interval of event prediction
- Entering: Person disappears close-by a vehicle that has an open door with fixed interval of event prediction (requires Opening detected before hand)
- Exiting: Person appears close-by a vehicle that has an open door with fixed interval of event prediction (requires Opening detected before hand)
- Open Trunk: open trunk highest confidence frame with fixed interval of event prediction
- Closing Trunk: no open trunk frame, after door was open with fixed interval of event prediction
- Loading: Person carried object before coming close to an open door or trunk but not after (requires Open Trunk detected before hand)
- Unloading: Person carried object after coming close to an open door or trunk but not before (requires Open Trunk detected before hand)

1.2.3 Person only activities

Person only activities are: Talking, Carry, Heavy Carry, Pull, Riding and Specialized talking and texting phone.

Like for the vehicle with person activities, specialized classifiers have been trained or the trajectory of the person has be analyzed. Specialized objects are the heavy or pulled objects as similar for the loading and unloading activities. If those detections can be associated with a track beyond a certain number of times, the whole track is classified with this activity.

For riding the trajectory of the person itself is analyzed. If the relative movement of the person compared to its box size exceeds a certain threshold riding is associated with this part of the track.

Talking requires two close-by persons for a certain period of time. This applies for standing persons as well as persons having matching trajectories.

Specialized talking and texting phone were tackled by training CNN classifiers and extraction of skeletons via OpenPose and pifpaf. For Skeletons a simple MLP has been trained.

1.3 Score Optimizations

Using the validation set the best start and end frame for each activity based on the AUDC measure is searched. Furthermore, as the confidence score of the activity defines the order and, therefor, the false alert rate, we optimized this score by finding the best approach to combine the individual object scores involved in the activity. Optimizing for either p_miss@0.15tfa or w_p_miss@0.15rfa did not reflect on better scores on AUDC.

Table 1: Results of submitted runs for TRECVID ActEV 2019

Partial AUDC	**mean-p_miss@0.15tfa**	**mean-w_p_miss@0.15rfa**
0.59889	0.50995	0.82406

1.4 Results and conclusion

At the final leaderboard from 1st of October, Hitachi has been ranked at the 4th place.

2 TRECVID 2019 INS: Instance Search

Abstract. TRECVID INS 2019 [1] gives us a new task which is to find a specific person doing a specific action as denoted in query examples. In this work, we split the problem into two separate tasks: finding a person and finding action. To find person, we first use VGGFace2 for face representation, then face is matched and reranked using cosine similarity. For action, two different approaches are used for audio type (e.g.: laughing, shouting, crying) and visual type (e.g.: holding glass, carrying bag), respectively. With audio-type action, we use VGGish for audio representation while for visual-type action, C3D and Semantic feature extracted from VGG-1K are used. Similar to person task, matching and ranking are then applied using cosine similarity. In the end, we fused two computed similarity scores of person and action for the final rank list. Our team achieves 3rd rank in TRECVID INS 2019.

2.1 Introduction

This year, TRECVID Instance Search task is to find a specific person doing a specific action in a large video dataset. As in previous years, target persons are provided with 4 image examples followed by 4 shot examples from which those images are sampled. For the remaining part, different kinds of challenged actions are given such as holding phone, shouting, open door enter, open door leave, etc., each is provided with from 4-6 shot examples. Obviously, there are two main problems we have to tackle including person search and action search. In particular, for person search, we entirely ignore body appearance but focus on the face only.

In this work, the same approach is used to deal with these problems separately. First, different feature descriptors are chosen for representation of either face or action. Next, we use cosine similarity for feature matching. The two rank lists r_{face} and r_{action}, which are the results from face and action search, respectively, now can be achieved by selecting top shots with the highest similarity scores computed in the previous step. Eventually, fusion is applied to attain the final rank list. Furthermore, we observe that there are many bad faces in query (fig. 2), so we propose a new classifier-based method for removing such

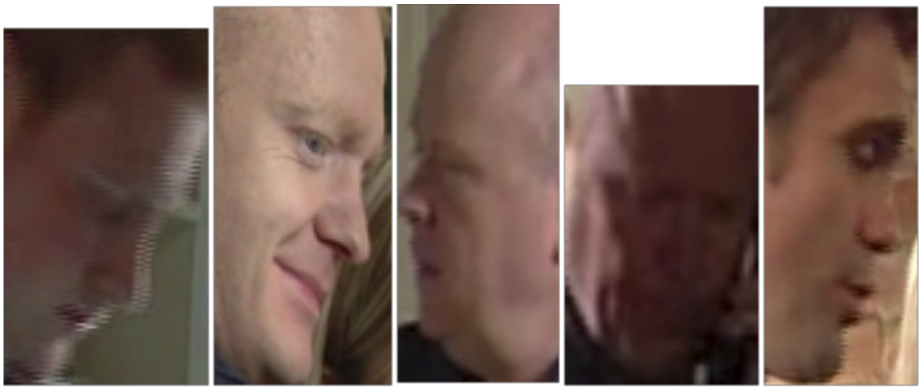

Figure 2: Some bad faces appear in query.

face whose aim is to improve the retrieval performance. Our team achieves 3rd best performance in automatic search.

2.2 Our Approach

2.2.1 Person Search

2.2.1.1 Query Preprocessing Faces in query are first detected using MTCNN [12]. Super-Resolution is then applied to each detected face to improve the quality of blur faces. Moreover, we observe that there are faces that are too skew or too blur, which can cause significantly decreasing in performance. For that reason, we propose a classifier-based approach to remove such faces away from the query. More specific, we train a SVM model to classify given face image into three categories: Good Faces, Skew Faces, and Blur Faces. Figure 3 shows some examples in each class of the dataset we use to train SVM. We use feature extracted from Pool5 layer of VGGFace as face representation to train SVM in the hope that choosing Pool5 would help us getting low-level features which is beneficial for our task. Face is considered bad if it is classified into skew or blur class and is thus being removed. Beside of using only image examples, we also consider using extra faces from shot examples. However, shot examples usually contain extra people other than the target person and even worse, they are not provided with binary masks for specifying where the target person is in the given query image as image examples are. As these shot examples are where image examples sampled from, one way to overcome this drawback is by applying face tracking. Here, we first use the method proposed in [9] to detect all face tracks in shot. Then, we use topic offset (determined by calculating the minimum frame difference between given example image and each frame in example shot) and given mask image to specify corresponding face track of subject in shot examples.

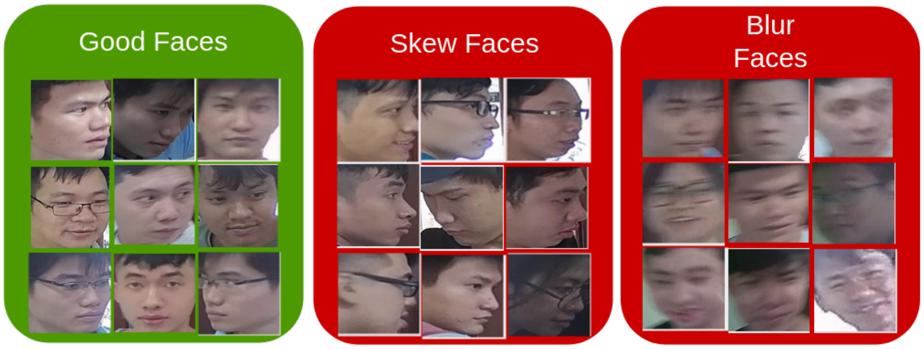

Figure 3: Example faces in each classes of dataset we usedfor training SVM classifier.

2.2.1.2 Deep Face Recognition We use feature extracted from VGGFace2 [4], for face representation, which a vector of shape [256]. Faces in query and shots are then matched using mean-max cosine similarity as follow:

$$sim(query, shot_i) = \frac{1}{N}\sum_{k=1}^{N}(max_{j=1,2,..,M}(cos(desc_k^{query}, desc_j^{shot_i})) \tag{1}$$

where $cos(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$ with A and B are the feature vectors of face in query and face in shot respectively. Here, N is the number of examples in the query set and M is the number of faces in the current shot. The variable $desc_k^{query}$ is the descriptor of the k-th face in query while $desc_j^{shot_i}$ is the descriptor of j-th face in i-th shot. After getting all the similarity score between the query and each video shot for all shot, we finally sort these shots in decreasing order by their score.

2.2.2 Action Search

2.2.2.1 Audio-type Action For actions overlapped with labels set of the AudioSet [6] such as shouting, laughing, crying, those are considered as audio actions. To obtain a feature descriptor for audio action, we first split audio away from shot videos both in examples and search gallery. These audios, which have been saved as WAV files, will be fed into VGGish [8] to obtain its representation. Each descriptor is a vector of shape $[n, 128]$ where n is the audio length in seconds. Matching similarity score between example shots and each shot in storage is computed using maximum pairwise cosine similarity as follow:

$$sim(query, shot_i) = \max_{\substack{k=1..N \\ s_1=1..L1 \\ s_2=1..L2}} cosine(desc_{s_1}^{query_k}, desc_{s_2}^{shot_i}) \tag{2}$$

where N is the numbers of audio examples. $shot_i$ is the current shot in storage that we need to calculate similarity. $L1$, $L2$ is the duration in seconds of query audio and $shot_i$ audio, respectively. The variable $desc_{s_1}^{query_k}$ is the descriptor of s_1-th second of the k-th audio examples while $desc_{s_2}^{shot_i}$ is the descriptor of s_2-th second of the $shot_i$ audio. Lastly, the same arrangement step is carried out as in person search.

2.2.2.2 Visual-type Action The remaining actions including holding glass, carrying bag, go up down stairs, etc. are visual actions. Two types of feature are used, which are C3D [11] and VGG-1K [10].

C3D, which is able to capture temporal features that are beneficial for the action recognition task. To obtain descriptor from C3D, we first sample uniformly 16 frames from each shot in both examples and storage. These 16 frames is then fed into the C3D network to attain one descriptor of shape [4096]. Here, we use mean cosine similarity for matching:

$$sim(query, shot_i) = \frac{1}{N}\sum_{k=1}^{N} cosine(desc_k^{query}, desc^{shot_i}) \tag{3}$$

given that N is the number of action shot examples and $shot_i$ is the current shot in storage that we need to calculate similarity. $desc_k^{query}$ and $desc^{shot_i}$ are descriptor of k-th shot example and i-th storage shot, respectively.

For VGG-1K, because it is pretrained on ImageNet [5] whose classes is highly varied, features from this network is useful for actions where a person interacts with an object such as holding glass, holding phone, etc. We first feed all keyframes from each shot into VGG-1K network, features extracted from logit layer are then used for matching. To obtain similarity score, we use mean max cosine which is similar to that of person search.

2.2.3 Fusion

After having the three rank lists from person search (VGGFace2) and action search (C3D and VGG-1k), we finally need to fuse these three to obtain the final result. Top 5000 shots from each rank list are chosen for fusion. The similarity of each shot is first normalized using either *zscore* or *tanh*. The final similarity score is computed using the following formula:

$$score_{final} = w_1 * score_{vggface2} + w2 * score_{c3d} + w3 * score_{vgg1k} \tag{4}$$

where w_1, w_2 and w_3 is weight for VGGFace2, C3D and VGG-1K respectively.

2.3 Evaluation

We use four different fusion settings for submissions. Each setting consists of two submitted runs, which are for type A and type E, respectively. Table 2 shows all the official evaluation results of our submissions.

Table 2: Result of our submitted 8 runs on Instance Search task of TRECVID 2019.

RUN	mAP	Fusion Description			
		Normalization	Combination		
			Face weight	Action weight	Semantic weight
NII_Hitachi_UIT_R1_A	0.0226	z-score	0.1	0.3	0.6
NII_Hitachi_UIT_R1_E	0.0234				
NII_Hitachi_UIT_R2_A	0.0236	tanh	0.1	0.3	0.6
NII_Hitachi_UIT_R2_E	0.0243				
NII_Hitachi_UIT_R3_A	0.0243	z-score	0	0.5	0.5
NII_Hitachi_UIT_R3_E	0.0211				
NII_Hitachi_UIT_R4_A	0.0184	z-score	0.1	0.6	0.3
NII_Hitachi_UIT_R4_E	0.0191				

2.4 Conclusion

We propose a simple search pipeline which includes person retrieval and action retrieval. These works are performed separately before we apply fusion to attain the final rank list. We also propose a method for removing bad faces away from a query using SVM classifier which can make independent decision without relying on the other examples in the query. Evaluation results show that the two best runs of our system are: NII_Hitachi_UIT_R2 and NII_Hitachi_UIT_R3. We notice that there are some cases where action type is correct but the final result turns out to be wrong as the action is not made by the target but another person. This problem should be focused more on future works to improve retrieval performance. A simple approach is to track person and action jointly.

References

[1] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, and Georges Quénot. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID 2019*. NIST, USA, 2019.

[2] 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)*, pages 3464–3468, 2016.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988, 2017.

[8] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[9] Thanh Duc Ngo, Duy-Dinh Le, Shin'ichi Satoh, and Duc Anh Duong. Robust face track finding in video using tracked points. In *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pages 59–64. IEEE, 2008.

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

[11] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[12] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.