Exploring Temporal Context and Human Movement Dynamics for Online Action Detection in Videos

Vasiliki I. Vasileiou, Nikolaos Kardaris and Petros Maragos
School of E.C.E., National Technical University of Athens, Athens 15773, Greece
silavassiliou2@gmail.com, nkardaris@mail.ntua.gr, maragos@cs.ntua.gr

***Abstract*—Nowadays, the interaction between humans and robots is constantly expanding, requiring more and more human motion recognition applications to operate in real time. However, most works on temporal action detection and recognition perform these tasks in offline manner, i.e. temporally segmented videos are classified as a whole. In this paper, based on the recently proposed framework of Temporal Recurrent Networks, we explore how temporal context and human movement dynamics can be effectively employed for online action detection. Our approach uses various state-of-the-art architectures and appropriately combines the extracted features in order to improve action detection. We evaluate our method on a challenging but widely used dataset for temporal action localization, THUMOS'14. Our experiments show significant improvement over the baseline method, achieving state-of-the art results on THUMOS'14.**

***Index Terms*—Action Detection, Action Anticipation, Online Action Detection, Skeleton, THUMOS'14**

I. Introduction

Human Action Recognition (HAR) is one of the most prominent tasks in the field of computer vision, with various applications in robotics [35], [36], [37], data retrieval [42], [43], healthcare [38], [39] etc. Most works deal with action recognition in an offline setting, i.e. the temporal boundaries of an action in a video are known. However, online applications such as on autonomous cars or assistive robotics require recognition capabilities in a continuous video stream, where the starting and ending points of an action have to be estimated on-the-fly, e.g. in order to avoid a car crash [40] or even a human fall [41].

Therefore, the difference between the two approaches lies in the time of the decision. Specifically, in the case of offline recognition the action will be observed in its entirety, whereas in online recognition, the decision will have to be taken before the action is completed. Many methods have been developed for the recognition of human action with remarkable results, however they operate under the condition that recognition is made once the action is completed and the network has been fed with all the necessary information. This condition cannot be however satisfied in an online setting, where only present and past information can be used, making the generalisation of offline methods problematic.

The recently proposed Temporal Recurrent Networks (TRN) [1] introduced a way to bypass the lack of future information by using recurrent networks to predict features that correspond to future frames. TRN processes videos sequentially and for each frame it combines past, present and predicted future information to extract action class probabilities. Inspired by TRN, we explore different ways to enhance temporal action detection. Our contribution is two-fold:

- We employ 3D convolutional networks (I3D [5] & C3D [4]) both in action anticipation and prediction. In this way, information about the temporal context of each frame is encoded both by the recurrent network and in terms of visual features.
- We incorporate human pose in our framework, postulating that the dynamics of the human movement provide valuable information about the temporal boundaries of an action.

Two-stream features are introduced as an input to our model which are interpreted as appearance and motion features. Some combinations of the above mentioned extracted features have therefore been selected in order to increase the efficiency of our model. Experiments on the THUMOS14 [8] public dataset, show significant improvement over the baseline both in action recognition and anticipation.

The rest of this paper is organized as follows: Section II provides a review of the literature work in the field. The methodology applied throughout our experiments is analyzed in section III. Section IV deals with the experimental setup, the dataset and the evaluation methods we utilize, whereas in Section V we discuss the results of our experiments. Finally, in Section VI we present our conclusions and propose directions for further research.

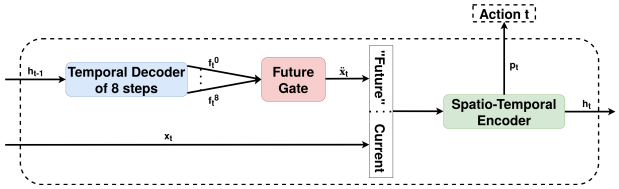

Fig. 1. The TRN Cell Architecture as described in [1].

Fig. 2. The Pipeline of the One-Stream model which is fed directly by the extracted C3D features.

The work of N. Kardaris & P. Maragos has been co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH CREATE INNOVATE (iWalk, T1EDK- 01248).

II. Related Work

A. Human Action Recognition

Action recognition in videos has stimulated the interest of the research community for years. Traditional methods, utilizing feature descriptors paved the way to more complex neural networks, being proposed nowadays.

Early works on action recognition extracted hand-crafted features, such as Histogram of Oriented Gradients (HOG) / Histogram of Optical Flow [9], extended Speeded-Up Robust Feature [11], Dense Trajectories [12] and encoded each video using Bag-Of-Words, Fisher vector and other orderless representations. Support Vector Machines were typically used to classify those features.

Although such methods have shown promising results [44], the abundance of visual data and advances in hardware design led the research community to embrace neural network models. Convolutional Neural Networks (CNN) in particular, have provided significant improvements in image recognition tasks [18], something that intuitively bolstered their application to videos and action recognition. Initially, Donahue et al. in [13] proposed to add a recurrent layer to the CNN to encode state and capture temporal sequence and long-term dependencies. Due to 2D features limitation of 2D CNNs, Ji et al. developed a 3D CNN model [15] that extracts features from both spatial and temporal dimensions through 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. Based on this work, Tran et al. created an efficient descriptor -C3D [16]- which can be used as a pre-trained feature extractor for other video analysis tasks. In [14], a two-stream network, introduced by Simonyan and Zisserman, analyzes spatiotemporal features via RGB images and optical flow. The 3D-fused extension [19] of the previous model introduces a better performance by fusing spatial and flow streams after the last convolutional layer. Finally, Carreira et al. combined the above models into a new one -I3D [5]- aiming to very deep, naturally spatio-temporal classifiers.

On the other hand Noori et al. based on the fact that skeleton based action recognition can avoid explicitly model the dynamics of actions, propose in [20] the use of OpenPose [21] and Recurrent Neural Networks (RNNs) [22], [23] to recognize the activities.

B. Offline Action Detection

Regarding the variant of offline recognition, the sample videos are entirely known a-priori, so the task is to estimate the starting and ending timestamp of each action. This type of problem has the advantages of the whole frame sequence knowledge and the lack of processing time limitation. Early works [24], [25], [26] used sliding windows, where each window is considered as an action candidate, subject to classification. Escorcia et al. [27] exploit Long Short-Term Memory (LSTM) cells to encode a video sequence as a set of discrete states in order to demonstrate proposal scores, while Gao et al. proposed the TURN TAP [28] where a long untrimmed video is decomposed into video units, which are then reused as basic building blocks of temporal proposals. Furthermore, Shou et al. [29] introduced the CDC network, which makes dense per-frame predictions through downsampling in space and upsampling to localize the temporal boundaries. Finally, Long et al. presented GTANs [30], which, in contrast to the previous methods, leverage the temporal structure in an one-stage action localization framework.

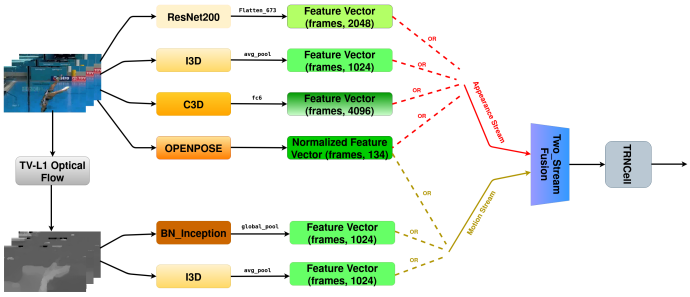

Fig. 3. The pipelines of the eleven possible two-stream models, of which the six most effective combinations have been selected. In particular, the selected are: i)ResNet-200 - BN-Inception, ii)I3D (RGB) - I3D (Flow), iii)C3D - OpenPose, iv)ResNet-200 - OpenPose, v)I3D(RGB) - OpenPose, vi)OpenPose - BN-Inception

C. Online Action Detection

Online Action Detection was initially proposed by De Geest et al. [31], who also created the TVSeries dataset for the same purpose. The same research group later proposed a two-stream LSTM model [32], focusing both on the interpretation of the frames and on the temporal dependencies between actions. The RED network [33], created for action anticipation, can also be used for action detection, if the anticipation time is set to zero. The main idea of this work is the prediction of feature actions using a CNN for feature extraction in combination with a LSTM and a reinforcement loss. Generative Adversarial Networks (GANs) have been also proposed by Shou et al. [34] to predict starting time precisely.

Temporal Recurrent Network (TRN) [1] proposed by Xu et al. is a novel method which uses the predicted future information to enrich the online action detection accuracy. Based on this work we explore different feature extraction and fusion methods to better estimate information about the temporal dynamics of actions and therefore to achieve higher scores both in anticipation and recognition task.

III. Methodology

To address the problem of online action recognition we propose a framework consisting of two main components, one that explores the temporal context of videos, and one that corresponds to the TRN cell proposed in [1].

A. TRN Cell

We first provide a brief description of the TRN cell, which is crucial for the intuition of our method. In particular, the central idea behind TRN is to anticipate future frames' features and aggregate it with past and present information to properly categorize the action. As shown in Fig. 1, it consists of

the temporal decoder, the future gate and the spatiotemporal encoder. Both the temporal decoder and the spatial encoder are LSTM units, where the former accepts serial input vectors and exports the predicted future information and the corresponding hidden states. The second unit sequentially receives the concatenated input and future vectors as well as the hidden state and estimates a probability distribution for each action.

B. Exploiting Contextual Information

Inspired by the two-stream model used in [1], we experimented by extracting I3D features, which are low-level spatial features. Those are computed using an I3D network that is pre-trained on Kinetics [5], to improve the ability of the model to generalize and avoid overfitting. We extract the $2\times$(1024-dimensional) frame-level features from the last global average polling layer. The two-stream features - appearance and motion - are concatenated and fed to a Linear layer with a ReLU activation. Then, the fused information enters the TRN cell, as shown in Fig. 3.

In addition we experimented with C3D features, being a very generic video feature representation [4]. This is because 3D convolutional modules can extract both spatial and temporal components, as opposed to the ResNet that is utilized by the original method [1] but limited to appearance representation. The aforementioned one-stream features are computed using C3D network pre-trained on Sports 1M, something that reduces the need to fine-tune. We extract 4096-dimensional features per frame from the fc6-layer and we insert them directly to the TRN cell, as shown in Fig. 2.

Skeleton joint coordinates are of high precision and can accurately represent the temporal dynamics of actions [45], so we experimented with 2D skeletons extracted from OpenPose [6], over the baseline RBG and Optical Flow features. Thus, 134-dimensional vectors per frame were created. The human pose consists of 25 keypoints for pose/foot estimation and $2\times$ 21 keypoints for hand estimation. Since 2D models are used, each keypoint consists of two spatial variables, its coordinates and a confidence parameter. We first normalize the features we get from OpenPose to address different camera setups. Specifically, we define the middle of the pelvis as the center of our coordinates and normalize with respect to the distance between the pelvis and the shoulders (average height). In the case of multiple actors in a frame the one whose coordinates have the highest confidence score is used.

Since the extracted skeleton features are primarily motion features [45], we added another stream to the C3D model. Specifically, we arranged the C3D features in the appearance stream and the pose features in the motion stream. This framework shows indeed an improvement in performance, compared to the corresponding one-stream model. Motivated by these results we apply the same framework to the I3D model: we arrange the I3D RGB data in the appearance stream and the OpenPose data in the motion stream.

Although the sequences of skeleton features sufficiently represent the temporal dynamics, the appearance and scene information is still missing. Based on the previous claim and

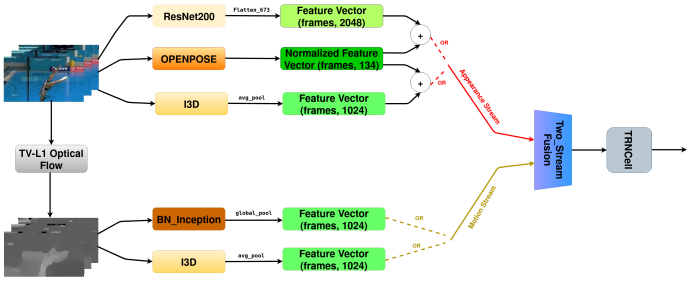

Fig. 4. The pipelines of the four possible two-stream fused models, of which the selected are: i) ResNet-200 concatenated with OpenPose - BN-Inception, ii) I3D (RGB) concatenated with Openpose - I3D (Flow).

on the feedback from our experiments so far, we attempted to combine each of our two-stream models - baseline and I3D - with the information from the skeleton. Specifically, we fused the RGB data with the OpenPose data and created a fused two-stream model as shown in Fig. 4.

IV. Experimental Setup

For the evaluation of our model we used the THUMOS'14 dataset [46] as it contains long and untrimmed videos from various sports events, which are annotated with 20 actions. Its training set however contains only trimmed videos that cannot be used for the task of temporal localization. As a result, based on the previous work [33], we train our model on the validation set (200 untrimmed videos) and validate it on the test set (213 untrimmed videos).

All experiments were preformed on Nvidia GeForce RTX 2080 Ti GPUs. Adam optimizer was used for the training session [47] with learning rate and weight decay parameters set to 5×10^{-4}. Due to GPU memory limitations, the batch size was set to 2 and the input sequence length was set to 64, whereas we included 8 decoder steps. To permit fair comparisons against the original method [1], we performed in-house testing for the baseline TRN with the previous settings.

As for video preprocessing, we extracted video frames at 30 fps and experimented with video chunk sizes of 6 and 16, in line with the examined set of experiments. The TV-L1 Optical Flow [48] algorithm was used to extract the optical flow frames through the Dense-Flow tool. Finally, we report per-frame Mean Average Precision (mAP) for evaluation. To provide more insight, we also report mAP for anticipation times ranging from 0.25 to 2 sec.

V. Results

The utilized models that we mentioned in Sec. 3 are shown in Fig. 2,3,4, while the results we obtain from our experiments are depicted in Tables I, II, III. These are organized based on the methodology used to extract the features. Specifically in Table I, ResNet-200 and BN-Inception were used for RGB and Flow features extraction respectively, while, in Table II and III, we used C3D and I3D respectively. In all three tables, we denote the use of OpenPose features, either by replacing or complementing flow and RGB features, thus creating motion features and appearance features respectively.

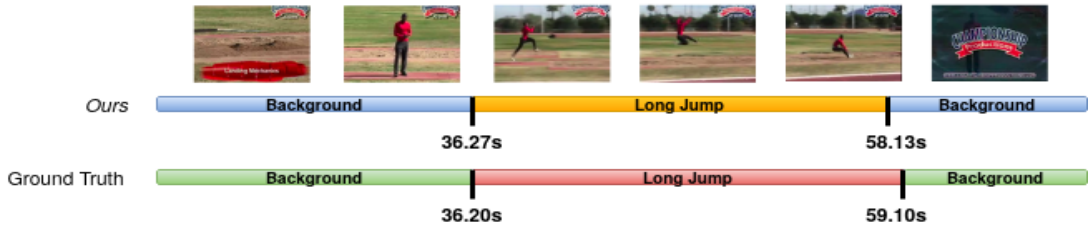

Fig. 5. Visualization of our best method - I3D

TABLE I
CHUNK SIZE 6 EXPERIMENTS, USING RESNET-200, BN-INCEPTION AND OPENPOSE

Method	Features Chunk size = 6 frames	Encoder	Decoder - Time predicted into the future (seconds)								
			0.25s	*0.50s*	*0.75s*	*1.00s*	*1.25s*	*1.50s*	*1.75s*	*2.00s*	*Avg*
Baseline	RGB – Flow	25.93	**26.15**	25.89	25.79	25.73	25.66	**25.68**	**25.66**	**25.57**	**25.77**
Ours	{RGB + OpenPose} – Flow	24.25	23.11	25.63	**26.72**	26.18	25.57	24.94	24.40	23.94	25.06
Ours	RGB – OpenPose	**37.57**	25.54	**25.93**	26.44	**26.60**	**26.28**	25.57	24.75	24.00	25.64
Ours	OpenPose – Flow	36.30	21.77	22.59	23.57	23.19	22.28	21.30	20.49	19.83	21.88

TABLE II
CHUNK SIZE 16 EXPERIMENTS, USING C3D AND OPENPOSE

Method	Features Chunk size = 16 frames	Encoder	Decoder - Time predicted into the future (seconds)								
			0.25s	*0.50s*	*0.75s*	*1.00s*	*1.25s*	*1.50s*	*1.75s*	*2.00s*	*Avg*
Ours	C3D (One-Stream)	35.43	**34.34**	**31.05**	28.22	26.46	25.37	**24.75**	**24.39**	**24.22**	**27.35**
Ours	{C3D (RBG)} – OpenPose	**36.44**	32.98	30.56	**28.37**	**26.61**	**25.38**	24.54	23.78	23.22	26.93

TABLE III
CHUNK SIZE 16 EXPERIMENTS, USING I3D AND OPENPOSE

Method	Features Chunk size = 16 frames	Encoder	Decoder - Time predicted into the future (seconds)								
			0.25s	*0.50s*	*0.75s*	*1.00s*	*1.25s*	*1.50s*	*1.75s*	*2.00s*	*Avg*
Ours	**I3D**	**55.25**	**52.57**	**46.69**	**41.94**	**38.39**	**35.90**	**34.22**	**33.00**	**32.08**	**39.35**
Ours	{I3D (RGB) + OpenPose} – {I3D (Flow)}	49.21	46.65	40.78	36.42	33.19	30.90	29.42	28.43	27.71	34.19
Ours	{I3D (RGB)} – OpenPose	47.43	44.59	40.08	36.77	34.24	32.37	31.29	30.56	30.06	35.00
Ours	{I3D (RGB)} – {I3D (Flow) + OpenPose}	44.47	29.55	31.92	29.62	27.21	25.63	24.78	24.20	23.68	27.07

By inspecting Table I, where chunk size has been set to 6 frames, we observe that the baseline method, using ResNet-200 for RGB information and BN-Inception for flow information, displays the highest accuracy of 25.77% for the precision task, the prediction accuracy for the classification task however is only 25.93%. By replacing flow information with OpenPose features we notice that, while the average decoder accuracy decreases slightly to 25.67%, the encoder accuracy is significantly improved and reaches 37.57%. However, it is true that the use of OpenPose features to enhance or replace RGB information does not provide any further improvement, either in the anticipation phase or in the process of detection. It is worth mentioning though that, in both cases that we used OpenPose, we observe better performance for the period 0.5s - 1.25s, but also much smaller than the baseline in longer-term predictions, something that leads to a decline of the average accuracy in these cases.

Table II shows the results for C3D, giving one-stream features, where the chunk size has been set to 16. We highlight that the use of human pose as motion features, by introducing a second stream in our model, gives a boost compared to the simple C3D in the phase of action detection, from 35.43% for the former to 36.44% for the latter, as opposed to action anticipation, for which the accuracy drops to 26.93%. Additionally, we should note the large discrepancy between the performance of short-term and long-term anticipation, reaching as much as 10%. By comparing this table to the previous one, we observe that OpenPose shows, as motion information, better anticipation performance, compared to the corresponding simple model, in the interval 0.75s - 1.25s. Moreover, although in C3D models we observe larger anticipation accuracy, the action detection accuracy does not exceed that of the models of Table I.

The results of employing I3D as well as its variations are shown in Table III, where the chunk size is set to 16 frames. We notice that both the simple I3D model and its modifications show much better performance, with the simplest I3D model giving the biggest boost and reaching 39.35% in the anticipation phase and 55.25% in the detection phase. However, the use of OpenPose in this set of experiments, both as an additional cue to RGB and flow information and as a unique motion information did not offer any improvement. On the

contrary, it limited its effectiveness. This divergence is likely due to the substantially strong capacity, offered by I3D flow information.

VI. Conclusion

In this paper, we propose several ways to improve online action detection, building upon Temporal Recurrent Networks. Our results highlight the value of temporal context and human pose as useful cues for localizing action in time. We demonstrate that most of our models outperform the original TRN method [1] by a significant margin, even though our baseline results are lower than the original paper's due to the smaller batch size we used, with the best of them (I3D) achieving state-of-the-art results. Specifically, observing the variations of models' behavior in the analysis and detection phase, we believe that the use of different models for anticipation and recognition could benefit the task of online action detection. We plan to pursue this goal in our future work.

References

[1] M. Xu and M. Gao and Y. Chen and L. S. Davis and D. J. Crandall: Temporal Recurrent Networks for Online Action Detection. In Proc. ICCV, 2019.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. CVPR, 2016.

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.arXiv:1502.03167, 2015.

[4] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, Learning Spatiotemporal Features with 3D Convolutional Networks. In Proc. ICCV, 2015.

[5] J. Carreira, & A. Zisserman (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In Proc. CVPR, 2017

[6] Z. Cao, G. Hidalgo, T. Simon, S. Wei, Y. Sheikh: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In Proc. CVPR, 2017.

[7] A. Shahroudy, J. Liu, T. Ng, G. Wang: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In Proc. CVPR, 2016.

[8] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev,M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

[9] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In Proc. CVPR, 2008.

[10] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In Proc. BMVC, 2008.

[11] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In ECCV, 2008

[12] H. Wang, A. Kläser, C. Schmid et al. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. Int J Comput Vis 103, 60–79 (2013).

[13] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Vebugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In Proc. CVPR, 2015.

[14] K. Simonyan, & A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In NIPS, 2014.

[15] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. IEEE Trans on pattern analysis and machine intelligence, 35(1):221–231, 2013.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri.Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015

[17] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In Proc. CVPR, 2017.

[18] Y. Lecun , L. Bottou , Y. Bengio , P. Haffner , Gradient-based learning applied to document recognition. In IEEE, vol. 86, pp.2278, 1998

[19] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In Proc. CVPR, 2016.

[20] F. M. Noori, B. Wallace, Md. Z. Uddin,and J. Torresen: A Robust Human Activity Recognition Approach Using OpenPose, Motion Features, and Deep Recurrent Neural Network. In Proc. SCIA, 2019.

[21] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using part affinity fields". In CVPR, 2017.

[22] Q. Ke, M. Bennamoun, S. An, F. Sohel, F. Boussaid: A New Representation of Skeleton Sequences for 3D Action Recognition. In Proc. CVPR, 2017

[23] A. Shahroudy, J. Liu, T. Ng, G. Wang: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In Proc. CVPR, 2016

[24] S. Karaman, L. Seidenari, and A. Del Bimbo. 2014. Fast saliency based pooling of fisher encoded dense trajectories. In ECCV THUMOS Workshop.

[25] D. Oneata, J. Verbeek, and C. Schmid. 2014. The LEAR submission at Thumos 2014. In ECCV THUMOS Workshop, 2014.

[26] L. Wang, Y. Qiao, and X. Tang. Action recognition and detectionby combining motion and appearance features. In THUMOS14 Action Recognition Challenge, 2014.

[27] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep Action Proposals for Action Understanding. In ECCV, 2016.

[28] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. In ICCV, 2017.

[29] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In Proc. CVPR, 2017.

[30] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo and T. Mei. Gaussian Temporal Awareness Networks for Action Localization. In Proc. CVPR, 2019.

[31] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars. Online action detection. In Proc. ECCV, 2016.

[32] R. De Geest and T. Tuytelaars. Modeling temporal structure with lstm for online action detection. In Proc. WACV, 2018.

[33] J. Gao, Z. Yang, and R. Nevatia. RED: Reinforced encoder-decoder networks for action anticipation. In Proc. BMVC, 2017.

[34] Z. Shou, J. Pan, J. Chan, K. Miyazawa, H. Mansour, A. Vetro, X. Giro-i Nieto, and S.-F. Chang. Online action detection in untrimmed, streaming videos-modeling and evaluation. In Proc. ECCV, 2018.

[35] A. Zlatintsi, A.C. Dometios, N. Kardaris, I. Rodomagoulakis, P. Koutras, X. Papageorgiou, P. Maragos, C.S. Tzafestas, P. Vartholomeos, K. Hauer, C. Werner, R. Annicchiarico, M.G. Lombardi, F. Adriano, T. Asfour, A.M. Sabatini, C. Laschi, M. Cianchetti, A. Güler, I. Kokkinos, B. Klein, and R. López. 2020. I-Support: A robotic platform of an assistive bathing robot for the elderly population. In Robot. Auton. Syst., 2020.

[36] G. Chalvatzaki, Petros Koutras, A. Tsiami, C. Tzafestas and P. Maragos. i-Walk Intelligent Assessment System: Activity, Mobility, Intention, Communication. In Proc. ECCV Workshops, 2020.

[37] J. Hadfield, G. Chalvatzaki, P. Koutras, M. Khamassi, C. S. Tzafestas and P. Maragos. A Deep Learning Approach for Multi-View Engagement Estimation of Children in a Child-Robot Joint Attention Task. In Proc. IROS, 2019.

[38] Y. Gao, X. Xiang, N. Xiong, B. Huang, H. J. Lee, R. Alrifai, X. Jiang, Z. Fang. Human Action Monitoring for Healthcare Based on Deep Learning. In IEEE Access vol. 6, pp. 52277, 2018.

[39] D. Burns, N. Leung, M. Hardisty, C. Whyne, P. Henry, S.tewart McLachlin. Shoulder Physiotherapy Exercise Recognition: Machine Learning the Inertial Signals from a Smartwatch. In Physiol, vol. 39, 2018.

[40] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, E. M. Atkins. Unsupervised Traffic Accident Detection in First-Person Videos. In Proc. IROS, 2019.

[41] G. Serpen and R. H. Khan. Real-time Detection of Human Falls in Progress: Machine Learning Approach. In Proc. CASE, 2018.

[42] M. Ramezani and F. Yaghmaee. A review on human action analysis in videos for retrieval applications. Artif. Intell. Rev. 46, 4, 2016, 485–514.

[43] X. Zhai, Y. Peng, and J. Xiao. 2013. Cross-media retrieval by intra-media and inter-media correlation mining. Multimedia Syst. 19, 5, 2013, 395–406.

[44] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In Proc. ICCV, 2013

[45] Y. Du, Y. Fu and L. Wang. Representation Learning of Temporal Dynamics for Skeleton-Based Action Recognition. In IEEE Trans. on Im. Proc., vol. 25, no. 7, pp. 3010, July 2016.

[46] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. In Proc. ICCV 2013.

[47] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

[48] J. Sánchez & E. Meinhardt-Llopis, & G. Facciolo. TV-L1 optical flow estimation. In Proc. IPOL, 2013.