Self-Attention Temporal Convolutional Network for Long-Term Daily Living Activity Detection

Rui Dai[1], Luca Minciullo[2], Lorenzo Garattoni[2], Gianpiero Francesca[2], and François Bremond[1]

[1]INRIA Sophia Antipolis, 2004 Route des Lucioles, 06902, Valbonne, France
[2]Toyota Motor Europe, Hoge Wei 33, B - 1930 Zaventem

Abstract

In this paper, we address the detection of daily living activities in long-term untrimmed videos. The detection of daily living activities is challenging due to their long temporal components, low inter-class variation and high intra-class variation. To tackle these challenges, recent approaches based on Temporal Convolutional Networks (TCNs) have been proposed. Such methods can capture long-term temporal patterns using a hierarchy of temporal convolutional filters, pooling and up sampling steps. However, as one of the important features of convolutional networks, TCNs process a local neighborhood across time which leads to inefficiency in modeling the long-range dependencies between these temporal patterns of the video. In this paper, we propose Self-Attention - Temporal Convolutional Network (SA-TCN), which is able to capture both complex activity patterns and their dependencies within long-term untrimmed videos. We evaluate our proposed model on DAily Home LIfe Activity Dataset (DAHLIA) and Breakfast datasets. Our proposed method achieves state-of-the-art performance on both DAHLIA and Breakfast dataset.

1. Introduction

Detecting activities in untrimmed videos has been a long-studied task in computer vision. Its performance impacts domains such as health-care, assistive robotics, and video surveillance. With the impressive success of Convolutional Neural Networks (CNNs), activity recognition techniques have achieved high performance on many trimmed video datasets [28, 19]. However, most videos in real world scenarios are untrimmed. They may contain multiple activities at the same time and the range of duration is essentially limitless. In addition, often a large number of frames are background and consequently hardly usable to detect target activities. Therefore, the task of finding both *where* and *what* are the activities in untrimmed videos is extremely challenging. In our work, we focus on activities of daily living (ADL). Due to their low inter-class variation, high intra-class variation and long temporal duration, activity detection techniques for ADL video datasets need a higher capacity of capturing complex long-term temporal patterns.

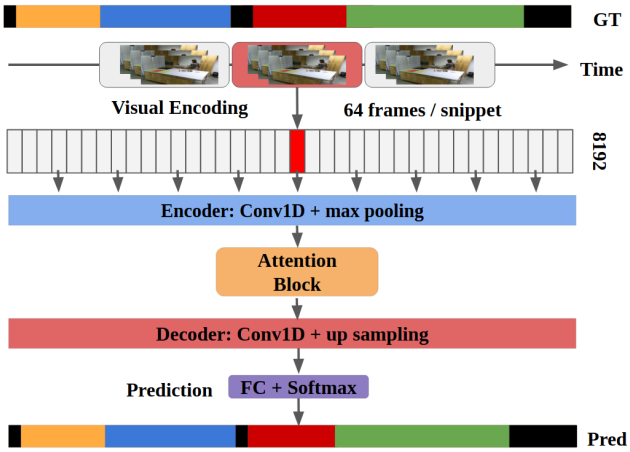

Figure 1. **Overview of SA-TCN framework**. The system contains 5 main parts: (1) visual encoder, (2) temporal encoder, (3) self-attention block, (4) temporal decoder and (5) prediction classifier.

Some existing state-of-the-art methods address this task by using sliding window approaches [31, 16]. Generally, these methods use a classifier trained on trimmed videos. At test time though, a sliding window extracts video segments that are then labelled by the classifier. Some approaches implement further post-processing filters to improve performance. Due to the balance between window size and computational cost, these methods can achieve high performance on short-term videos but tend to struggle on longer

978-1-5386-9294-3/18/$31.00

video sequences.

Following the recent advances of Recurrent Neural Networks (RNNs) in processing sequence data, numerous approaches are using RNN-based model to do activity detection [2, 22]. Memory cells help RNNs capture temporal information from video sequences [10], while forgetting cells drop information that is irrelevant for the long-term encoding. Therefore, RNNs can only capture a limited amount of temporal context in video, which is not suitable to process long-term data.

Temporal Convolutional Networks (TCNs) use 1D convolutions and are another way to compute features encoded across time. Contrary to RNN-based methods, TCN computations are performed layer-wise: that means that at every time-step its weights are updated simultaneously, which allows TCN to process long-term sequences. There have been already several applications of TCN in activity detection [13, 6]. However, recent studies focus on short-term activity datasets as [24, 7], where the mean activity duration is less than 30 second. This cannot be straightforwardly generalized to ADL datasets, where activities can last several minutes [26]. Because of the limited receptive field of CNN kernels, TCNs still have limitations when dealing with dependencies between long-range patterns in video. As a result, we introduce Self-Attention - Temporal Convolutional Network(SA-TCN), which is a TCN-based model embedded with temporal self-attention block. This block extracts a global temporal attention mask from the hidden representation laying between encoder and decoder. Fig. 1 shows an overview of our proposed network. Thanks to TCN structure and self-attention block, our proposed attention mechanism can better focus on long temporal patterns and their dependencies.

In this work, we used DAHLIA [26] as the main dataset to evaluate our proposed method, along with a medium-term dataset, Breakfast [12], to show the robustness of the framework. Our proposed method achieves state-of-start performance on both datasets.

2. Related Work

In this section, we summarize the recent works in activity detection.

The literature on activity recognition on short trimmed video clips is vast [3, 23, 1]. As a consequence, several approaches apply the lessons learned from the work on trimmed sequences to deal with the untrimmed activity detection problem. Shou *et al.* [20] train a classifier on trimmed videos, using sliding windows with multi-scale window size, along with a post-processing filter to improve performance. Zhu *et al.* [32] employ a multi-task classifier, to detect the boundaries and activity classes from a fixed-size sliding window. However, the fixed window size and the high computational cost limit its applicability to long sequences.

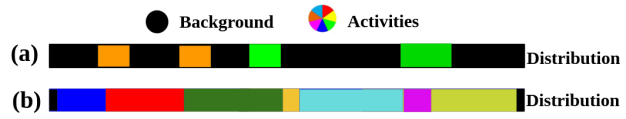

Figure 2. **Distribution of activities.** A typical activity distribution in (a) THUMOS'14 (val#161: Cliff Diving), (b) a long term ADL dataset: DAHLIA (S01A2K1).

Methods like [17, 31] are two-steps methods. In the first step, they generate initial temporal action proposals using an actioness detector. In the second step, the action proposals are classified and the initial coarse activity boundaries are refined. Actioness was introduced in [5] to indicate the likelihood that a generic action instance can be localized at a specific temporal location. This measure is useful for generating accurate action proposals in datasets as THUMOS'14 [8], that include many background frames. As shown in Fig. 2, contrary to THUMOS'14, most ADL datasets do not have as much background since long activities are often connected by short background intervals. Thus, this kind of actioness-based method cannot effectively localize the activities in ADL datasets.

After encoding the video, activity detection can be seen as a sequence-to-sequence problem. The following are approaches that use RNNs to process sequence data. Singh *et al.* [22] feed per-frame CNN features into a bi-directional LSTM model and apply a method analogous to non-maximal suppression to the LSTM output. Yeung *et al.* [30] propose an attention LSTM network to model the dependencies between the input frame features within fixed length window. Nonetheless, RNNs can only capture a limited amount of temporal information and have short-term dependencies because they suffer from the *vanishing gradients* problem.

TCN is a class of time-series models [25]. Different from RNNs, TCN can capture long-range patterns by using a hierarchy of temporal convolutional filters, pooling, and up sampling steps. Lea *et al.* [13] propose Encoder Decoder-TCN (ED-TCN), which has an encoder-decoder structure to capture long-range temporal pattern. ED-TCN achieves state-of-the-art performance in 50 Salad [24] and GTEA [7] datasets. This work proves that TCNs are capable of capturing complex patterns such as activity compositions and activity duration. Ding *et al.* propose TCFPN, an extension of ED-TCN, which has a pyramid structure with lateral connections to reduce computation cost. However, temporal convolution processes the information within local neighborhood, thus using convolutional layers alone is computationally inefficient for modeling long-range dependencies in videos.

Attention mechanisms focus on the salient part of the scene relative to the target task. Employing attention mech-

anisms has gained popularity for the activity-recognition task [14, 23]. Self-attention mechanism is firstly proposed by Non-local Neural Networks [29]. Inspired by non-local image processing, self-attention mechanisms force networks to establish one-to-one temporal relations to capture long-range time dependencies.

3. Proposed Method

In this section we propose our new model: the Self-Attention - Temporal Convolutional Network (SA-TCN), which retains the encoder-decoder architecture of ED-TCN to capture long-range patterns and embeds a self-attention mechanism to capture the long range dependencies between those patterns. The overview of this architecture is shown in Fig. 3 and consists of 3 main components: visual encoding, encoder-decoder TCN, and self-attention block.

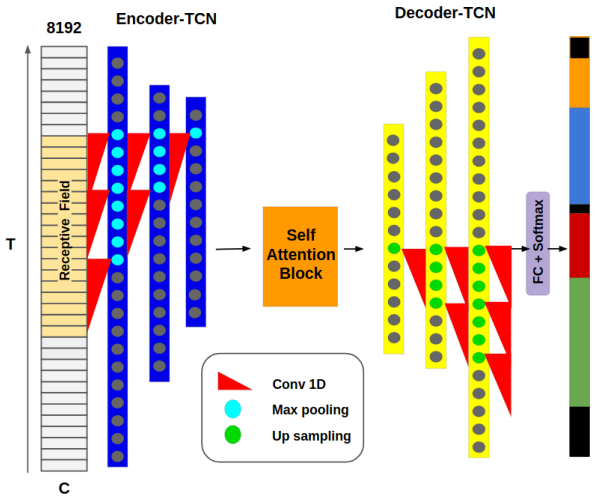

Figure 3. **SA-TCN model.** Given an untrimmed video, we represent each non-overlapping snippet by a visual encoding over 64 frames. This visual encoding is the input to the encoder-TCN, which is the combination of the following operations: 1D temporal convolution, batch normalization, ReLu, and max pooling. Next, we send the output of the encoder-TCN into the self-attention block to capture long-range dependencies. After that, the decoder-TCN applies the 1D convolution and up sampling to recover a feature map of the same dimension as visual encoding. Finally, the output will be sent to a fully connected layer with softmax activation to get the prediction.

3.1. Visual Encoding

The first step in our architecture is the extraction of a visual encoding. As opposed to the other TCN-based methods [13, 6] that use multi-modal inputs (i.e. RGB+flow), we attempted to use RGB only. To reduce the redundancy coming from extracting background features, we apply SSD [15] to detect the subjects and crop patches based on those detections. The patches are then resized to 224×224 and fed into an Imagenet pre-trained Resnet-152. We extract features from the penultimate layer of Resnet-152. We group 64 contiguous extracted feature sets per snippet. The temporal context of the video is handled by the aggregation operator using max and min pooling across the snippets. This pooling mechanism helps to choose salient values from the feature map. The visual encoding that we obtain from this step will be the input of encoder-TCN.

3.2. Improved Encoder-Decoder TCN

SA-TCN retains the encoder-decoder architecture of [13], with the addition of some points of improvement.

As shown in Fig. 4, we have k layers for both the encoder and the decoder. In the encoder part, each layer consists of temporal convolutions, batch normalization, ReLU activation, and a temporal max pooling. We set a fixed convolution kernel size for all the layers. First, we applied temporal convolution (Conv-1D) to extract high-level features. Second, differently from ED-TCN, we applied batch normalization to avoid vanishing or exploding gradients. Third, we added a spatial dropout layer along with a ReLU nonlinearity to help controlling over-fitting and to speed up convergence. Finally, we max pool the feature map across time to halve the temporal dimension. Pooling enables us to efficiently compute activations over long temporal windows.

Our decoder is similar to the encoder, except for the fact that we replace the pooling operation with up sampling. This up sampling step is similar to [13]: each entry is repeated twice. After that, another temporal convolution is performed to reduce the aliasing effect of up sampling. Finally, a snippet-wise fully-connected layer with softmax activation is used to generate the class probabilities at each time step.

3.3. Self-Attention Block

In this section, we introduce our temporal self-attention block. We construct this temporal attention mechanism based on the scoring system presented in [27].

The purpose of attention block is to build a one-to-one association between all the temporal moments. We do not rely on any outside information, so it is called self-attention. To implement this, the input I is branched out into three copies $Query$, Key and $Value$. Through the calculation of similarity between $Query$ and each Key, we can get the attention score s, which is the importance of different temporal moments. This attention score is then normalized by softmax to have a mask α. Finally, we multiply the $Value$ by this mask to have the attention-weighted feature, and then, add back the input to have our output result O.

Fig. 5 shows a diagram of the self-attention block, where $I \in \mathbb{R}^{C \times T}$ denotes the input features from the previous hidden layer. I is first transformed into two feature spaces $Query$, Key, where $Query(I) = W_{Query} I$, $Key(I) =$

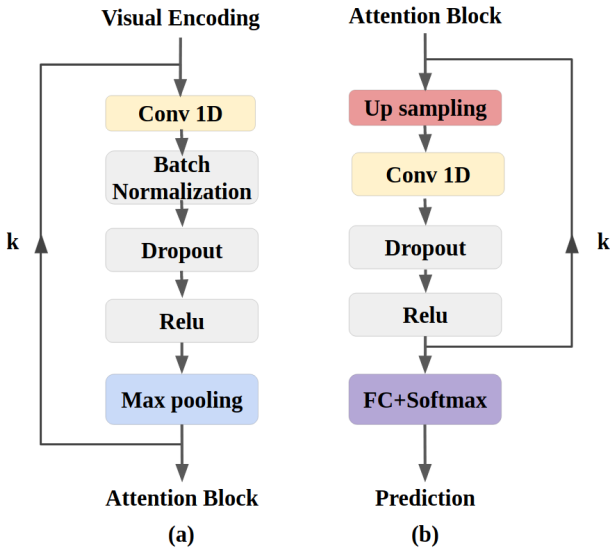

Figure 4. **Encoder-decoder architecture**. This figure represents the network structure of (a) encoder-TCN and (b) decoder-TCN. As the architecture has k layers, it will have k iterations.

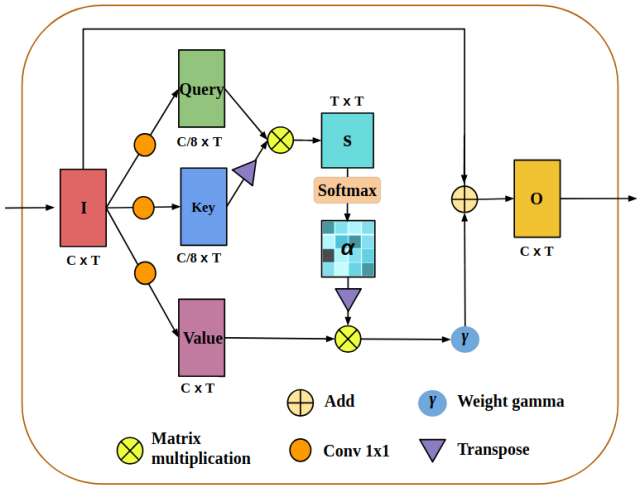

Figure 5. **Structure of self-attention block** between encoder-TCN and decoder-TCN.

$W_{Key}I$. Both W_{Query} and $W_{Key} \epsilon \mathbb{R}^{C \times \frac{C}{8}}$. In this work, $Value$ is computed from I with a 1×1 convolution layer. Thus we have $Value(I_i) = W_{Value}I_i$, where $W_{Value} \epsilon \mathbb{R}^{C \times C}$. The number of filters of $Value$ is same as the channel size of I. $Query$ and Key are similar to $Value$, except for the fact that the number of filters is one-eighth of $Value$. If $\alpha_{j,i}$ indicates the extent to which the model attends to the i^{th} location when synthesizing the j^{th} region, we have:

$$\alpha_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{T} exp(s_{ij})}, \tag{1}$$

where $s_{ij} = Query(I_j)Key(I_i)^T$.
Then the output of the weighted attention map is $Att = (Att_1, Att_2, ..., Att_j, ..., Att_T) \epsilon \mathbb{R}^{C \times T}$, where,

$$Att_j = \sum_{i=1}^{T} \alpha_{j,i} Value(I_i) \tag{2}$$

Finally, we add back the input feature map to assign weight to non-local evidence. Therefore the output O_i is given by:

$$O_i = \gamma \times Att_i + I_i \tag{3}$$

where γ represents a learnable parameter. The output O will be fed into decoder-TCN.

4. Experiments and Results

In this section, we describe the datasets and the baseline methods used in our study. We provide a comparative analysis of our method against other activity detection architectures. In all experiments, frame-wise accuracy(FA1), F-score, Intersection over Union(IoU) and mean Average Precision(mAP) are reported.

4.1. Datasets

In this work, we performed experiments on two datasets: DAHLIA and Breakfast.

DAHLIA [26] is one of the biggest public ADL dataset for detection. Contrary to some widely used datasets, in which labeled activities are very short and with low-semantic level, DAHLIA focuses on high-semantic level longer activities. It contains 8 ADL activity classes performed by 51 subjects on 3 camera views. The duration of videos ranges from 24 *mins* to 64 *mins*. In each video, an average of 6.7 activities are performed. The mean duration of activities is 6 *mins*. We performed experiments using the cross-subject protocol described in [16]. The final result is obtained as the average of the results on the 3 camera views.

Breakfast [12] features over $1.7k$ video sequences of cooking in a kitchen environment. The overall duration is $66.7h$. The dataset contains 48 activity classes. In each video, an average of 4.9 activities are performed. The mean duration of activities is about $30s$. Activities are thus shorter than those in DAHLIA, but they are more diverse. We performed our experiments using the protocol described in [6].

4.2. Implementation Details

We implemented our model in Keras 2.0.8 with Tensorflow as back-end. The experiments were performed on a GTX 1080 Ti GPU with 11 GB memory. For the visual encoding, we performed experiments using both Resnet-152 [9] and I3D [3] as the feature extractor. With Resnet, we extracted the features as described in detail in section *3.1* leading to 8192 features per snippet. With I3D, we chose the Kinetics pre-trained I3D. First, we added a fully

connected layer with 1024 units before the classification layer. Secondly, we fine-tuned the architecture on the NTU-dataset[19] and extracted features from the new fully connected layer (1024 features per snippet). We ran experiments with both Resnet-152 and I3D on DAHLIA. The results obtained with the two feature extractors are similar. On the Breakfast dataset, we use the features provided on the dataset's website. The length of these features is 64/snippet.

In our model, the attention operation does not change the dimension of the feature map. Besides, we assign the parameters of the encoder-decoder TCN so that the size of the feature map before the first encoder layer is the same as the output of the last decoder layer: we set the pooling and up sampling rate to 2, the number of filters in the three layers to {48, 64, 96} and {96, 64, 48} for encoder and decoder respectively. Finally, we compared several kernel sizes for the 1D convolution, and found that a size of 25 for every layer gives the best results.

The training was conducted with RMSprop with a learning rate of 0.001 and batch size 8 for both DAHLIA and Breakfast datasets. On DAHLIA, we split the train and validation set with 15% validation rate. We trained the model for 100 epochs and measured detection performance on the test set.

4.3. State-of-the-Art Methods

Several activity detection methods have been applied to DAHLIA. For our evaluation, we chose the two better-performing ones: **DOHT** [4] utilizes both skeleton and dense trajectory modalities along with a voting-based system. Each frame codeword has a certain weight in the vote for assigning the label of neighboring frames, and the weighting function is learned using a novel optimization method. **Negin *et al.*** [16] utilize a sliding window approach. This method obtains features from Resnet-152 [9] for each frame to form a code-book and then trains a SVM classifier.

Additionally, we applied the following methods as our baseline on both DAHLIA and Breakfast: **GRU**, implemented following the modifications described in [18]. This model enables temporal alignment and inference over long sequences. We selected GRU to measure the performance of a RNN-based method in long-term activity detection. **ED-TCN** is the original activity detection model proposed in [13]. **TCFPN** [6] is an extension of ED-TCN which features a pyramid structure with lateral connections to reduce computation cost. We selected these two TCN baselines to compare our method against TCN-based models.

4.4. Results Analysis

In this section, we analyze the results of our method and of the other state-of-the-art baselines.

Table 1. Activity detection results on DAHLIA dataset with the average of view 1, 2 and 3. *marked methods have not been tested on DAHLIA in their original paper.

Model	FA1	F-score	IoU	mAP
DOHT [4]	0.803	0.777	0.650	-
GRU* [18]	0.759	0.484	0.428	0.654
ED-TCN* [13]	0.851	0.695	0.625	0.826
Negin *et al.* [16]	0.847	0.797	0.723	-
TCFPN* [6]	0.910	**0.799**	0.738	**0.879**
SA-TCN	**0.921**	0.788	**0.740**	0.862

Table 2. Activity detection results on Breakfast dataset.

Model	FA1	F-Score	IoU	mAP
GRU [18]	0.368	0.295	0.198	0.380
ED-TCN [13]	0.461	0.462	0.348	0.478
TCFPN [6]	**0.519**	0.453	0.362	0.466
SA-TCN	0.497	**0.494**	**0.385**	**0.480**

Table 3. Average precision of ED-TCN on DAHLIA.

Activities	Background	House work	Working	Cooking
AP	0.36	0.65	0.95	0.96
Activities	**Laying table**	**Eating**	**Clearing table**	**Wash dishes**
AP	0.90	0.97	0.80	0.97

Table 4. Combination of attention block with other TCN-based model: TCFPN. (Evaluated on DAHLIA dataset)

Model	FA1	F-score	IoU	mAP
TCFPN [6]	0.910	**0.799**	0.738	0.879
SA-TCFPN	**0.917**	**0.799**	**0.748**	**0.894**

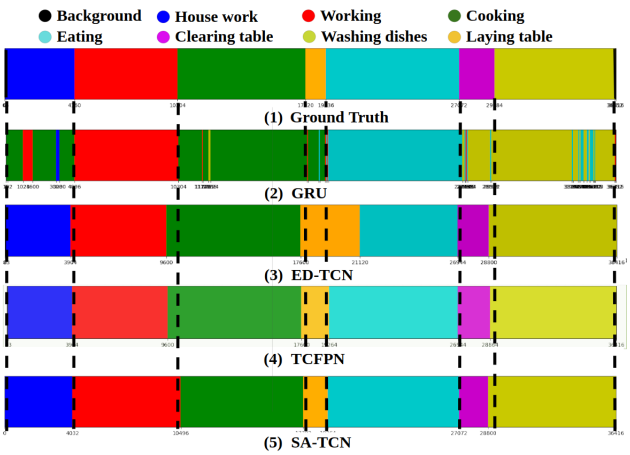

Figure 6. **Detection visualization**. The detection visualization of video 'S01A2K1' in DAHLIA: (1) ground truth, (2) GRU [18], (3) ED-TCN [13], (4) TCFPN [16] and (5) SA-TCN.

Table 1 and 2 show the results of all the methods considered on DAHLIA and Breakfast datasets, respectively. Our method achieves state-of-the-art performance on both datasets.

DOHT and Negin *et al.*'s method, which train a SVM with deep or hand-crafted feature encoding, do not perform

well on DAHLIA. This is because approaches based on a sliding window can only capture window-size patterns. Although a post-processing step is used to filter noise, these approaches still fail at capturing long temporal information.

Compared to TCN-based networks, GRU does not perform well on DAHLIA. Fig. 6 shows that GRU fails at distinguishing short activities performed between long activities (i.e. laying table and clearing table). Moreover, GRU produces noise while detecting long activities due to the fact that RNN-based networks can not focus on long temporal information.

ED-TCN lacks precision in detecting the activity boundaries. As CNNs have a limited receptive field for each layer, they fail in detecting the dependencies between long-distanced features. The results obtained by ED-TCN on DAHLIA are reported in Table 3. The low precision achieved on the 'Background' activity is due to the shorter duration of this activity compared to the others, which results in a lower number of training samples.

Both TCFPN and our SA-TCN outperform ED-TCN. The pyramid structure with lateral connections helps TCFPN to make use of both low-level and high-level features. The temporal attention block of our SA-TCN enables a better understanding of the dependencies between the different activities performed in the video.

To understand if our solution can be integrated with other temporal models, we embedded our temporal self-attention block in TCFPN to obtain SA-TCFPN. As reported in Table 4, SA-TCFPN outperforms TCFPN on all the metrics on DAHLIA. This shows that our temporal attention block is general and can be effectively integrated with other temporal models.

5. Conclusions

In this paper, we proposed a novel temporal convolutional framework for long-term activity detection: SA-TCN. We improved the encoder-decoder architecture from the ED-TCN and introduced a temporal self-attention block. On one hand, the TCN structure makes the model capable of learning long-term data with complex spatio-temporal patterns. On the other hand, the temporal attention block can well capture the dependencies between these patterns. Moreover, we have shown that this attention block can be integrated with other temporal models and improve their performance. Our experiments prove that the SA-TCN can be fast-trained and achieve state-of-the-art performance on two ADL datasets: DAHLIA and Breakfast. However, this model still has limitations. For example, the system currently cannot be trained end-to-end and cannot process data online. As future work, we plan to extend our model to online processing. Finally, we want to be able to deal with even more challenging datasets such as those containing overlapping activities[21, 11].

References

[1] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor. Glimpse clouds: Human activity recognition from unstructured feature points. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[2] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. SST: Single-stream temporal action proposals. In *CVPR*, 2017. 2

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017. 2, 4

[4] A. Chan-Hon-Tong, C. Achard, and L. Lucat. Deeply optimized hough transform: Application to action segmentation. In *International Conference on Image Analysis and Processing*, pages 51–60. Springer, 2013. 5

[5] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 748–755, 2014. 2

[6] L. Ding and C. Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 4, 5

[7] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. 2

[8] A. Gorban, H. Idrees, Y.-G. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2015. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. 2

[11] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 6

[12] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 2, 4

[13] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017. 2, 3, 5

[14] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 816–833, Cham, 2016. Springer International Publishing. 3

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector.

In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[16] F. Negin, A. Goel, A. G. Abubakr, F. Bremond, and G. Francesca. Online detection of long-term daily living activities by weakly supervised recognition of sub-activities. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018. 1, 4, 5

[17] H. Qiu, Y. Zheng, H. Ye, Y. Lu, F. Wang, and L. He. Precise temporal action localization by evolving temporal proposals. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 388–396. ACM, 2018. 2

[18] A. Richard, H. Kuehne, and J. Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017. 5

[19] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 5

[20] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016. 2

[21] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 6

[22] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970, 2016. 2

[23] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI Conference on Artificial Intelligence*, pages 4263–4270, 2017. 2, 3

[24] S. Stein and S. J. McKenna. User-adaptive models for recognizing food preparation activities. In *Proceedings of the 5th international workshop on Multimedia for cooking & eating activities*, pages 39–44. ACM, 2013. 2

[25] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 125, 2016. 2

[26] G. Vaquette, A. Orcesi, L. Lucat, and C. Achard. The daily home life activity dataset: a high semantic activity dataset for online recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 497–504. IEEE, 2017. 2, 4

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[28] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1

[29] X. Wang, R. B. Girshick, A. Gupta, and K. He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 3

[30] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018. 2

[31] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 1, 2

[32] Y. Zhu and S. Newsam. Efficient action detection in untrimmed videos via multi-task learning. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 197–206. IEEE, 2017. 2