Enhancing Temporal Action Localization with Transfer Learning from Action Recognition

Alexander Richard*, Ahsan Iqbal*, Juergen Gall
University of Bonn, Germany
{richard,iqbalm,gall}@iai.uni-bonn.de

Abstract

Temporal localization of actions in videos has been of increasing interest in recent years. However, most existing approaches rely on complex architectures that are either expensive to train, inefficient at inference time, or require thorough and careful architecture engineering. Classical action recognition on pre-segmented clips, on the other hand, benefits from sophisticated deep architectures that paved the way for highly reliable video clip classifiers. In this paper, we propose to use transfer learning to leverage the good results from action recognition for temporal localization. We apply a network that is inspired by the classical bag-of-words model for transfer learning and show that the resulting framewise class posteriors already provide good results without explicit temporal modeling. Further, we show that combining these features with a deep but simple convolutional network achieves state of the art results on two challenging action localization datasets.

1. Introduction

Temporal action localization in videos is of increasing importance for various practical applications such as content based video search, surveillance, and automatic highlight extraction *e.g.* in sports broadcasts. In classical action recognition, where the task is to classify a pre-segmented video clip as an instance of an action class, recent research greatly helped to improve the performance [29, 34, 8, 7]. With the availability of large scale datasets that comprise several hundred thousand clips, video clip classification reaches accuracies of far above 90% [3]. Unfortunately, the assumption that video clips are already pre-segmented and only contain a single action instance does not hold for practical applications.

Localization of actions in temporally untrimmed videos – albeit being of a greater practical importance – is still lacking behind. The reasons are twofold. First, the problem is inherently more difficult since besides finding the correct action label, also accurate action boundaries have to be determined. Second, and even more critically, obtaining large-scale datasets to train temporal localization models is difficult and expensive. Most existing works approach the problem of temporal action localization using complex end-to-end architectures that frequently rely on proposal and classification modules [21, 38, 4, 22] or time-consuming grammar based decoding schemes [23]. These systems typically require thorough architecture engineering and are expensive to train.

In this paper, we address the problem by proposing an efficient transfer learning strategy that leverages the good performance of classical action recognition while still generating generic representations that generalize to unseen videos and can be used for temporal action localization. In order to make the transfer as efficient as possible, we do not finetune an expensive model [3], but map the features from the pre-trained network into a latent probability space that simulates a soft feature quantization as usually done with kMeans in the classical bag-of-words setup. As non-linear function, we use an approximation of the feature map of the χ^2 kernel. The transferred features can then be combined with any temporal model. Since we aim for an efficient model that is fast to train and does not require any post-processing, we use a vanilla temporal convolutional network.

In our experimental evaluation, we show that this efficient transfer learning strategy already performs on par with some state of the art methods on the challenging Thumos benchmark even without any need for temporal modeling. Furthermore, we show that training a vanilla temporal convolutional network on top of the transferred features leads to further improvements even though the model is extremely simple. Since our approach is highly efficient and much simpler than most related approaches, it serves as a very strong baseline for more complex models.

*indicates equal contribution

2. Related Work

Action Recognition. Classical action recognition, *i.e*. recognition of pre-segmented video clips, has been widely studied. From classical feature based methods such as dense trajectories with bag-of-words encodings [32] or Fisher vectors [33], current research mainly focuses on deep architectures that are based on two-stream networks [29, 34] which process an appearance stream and an optical flow stream in parallel. Many variants exist that explore how to pass information from one stream to another [7] or how to incorporate temporal context [8]. In [12] it has been shown that pre-training a deep network on a large scale action dataset and transferring the knowledge embedded in the deep features can be beneficial to improve on smaller action recognition datasets. The most successful architecture today is the I3D network [3] that inflates 2D convolutions of two-stream networks to 3D and processes small spatio-temporal volumes of a video clip. Being trained on huge collections of videos, the approach achieves outstanding accuracies on challenging datasets. In [5], a more general approach for transferring knowledge from a network with 2D convolutions to a network with 3D convolutions was proposed.

Temporal Action Localization and Segmentation. Moving from classical action recognition to localization and segmentation of actions in videos, there has been a strong focus on temporally untrimmed videos that contain multiple action instances. There are two main research directions in this area. One is focusing on long-range temporal modeling of actions in videos, particularly for datasets where there are clear semantical dependencies between succeeding actions such as in cooking videos [14, 27, 30]. Richard and Gall [23], for instance, use a statistical language model to capture dependencies between different actions and a length model together with a framewise action classifier to obtain accurate segment boundaries. Various works on weakly supervised action segmentation make use of hidden Markov models and context free grammars to capture long term dependencies and use shallow neural networks or recurrent networks for frame-level action modeling [15, 25, 26]. Lea *et al*. [18] use spatio-temporal convolutions to capture mid-range dependencies and a semi-Markov model for transitions between action segments. In [17], a purely temporal convolutional network (TCN) consisting of an encoder that downsamples the input sequence in the temporal domain and a decoder that upsamples again to full resolution has shown good results on various datasets. The idea of encoder-decoder TCNs has been adapted and improved in various other works [6, 20]. A second major direction is mainly inspired by object detection. Assuming that videos can typically contain large background portions between two action instances, successful temporal modeling of consecutive actions is hard even if there are causal dependencies between them. Therefore, many works neglect these contextual dependencies and treat the localization of actions in videos similar to an object detection task. Zhao *et al*. [38], for instance, use a proposal network that divides segments into start, mid, and end parts before a classificaton network and a completeness network decide about the class label and if the boundaries are correct. Overall, many architectures follow the idea of proposal generation followed by segment classification and propose different architectural variants, losses, or processing stages [28, 9, 10]. Leveraging the success of Faster-RCNN for object detection, [4] develop a variant of this architecture for action localization. Xu *et al*. [35] combine convolutional 3D networks for action recognition with RCNNs to temporally localize actions. Overall, these architectures typically require expensive training of two-stream networks, thorough architecture enginieering, costly proposal generation steps, and post-processing to clean the output from overlapping or unreliable segments. Our approach, on the contrary, is much simpler and achieves state of the art results with a vanilla TCN network without the need for segment proposals or post-processing.

3. Technical Details

In this section, we define the task of temporal action localization and describe the technical details of our proposed method.

3.1. Task Definition and Notation

Temporal action localization is the task of finding all occurrences of action instances in a video. More formally, given a video with T frames $\mathbf{f}_1^T = (f_1, \dots, f_T)$, the task is to assign a class label $c \in \mathcal{C}$ to each frame, where $\mathcal{C}$ is a pre-defined set of classes. We assume that there is a background class that is assigned to frames not being part of an action instance. The training data are videos in which each occurring action instance is annotated with its start and end frame. This way, for each frame it is known if it belongs to the background class or an action class. We denote a training pair as $(\mathbf{f}_1^T, \mathbf{c}_1^T)$ where $\mathbf{f}_1^T$ is a video with T frames as defined above and $\mathbf{c}_1^T = (c_1, \dots, c_T)$ are the corresponding class labels.

3.2. Robust Transfer Learning for Action Localization

Transfer learning is a commonly used technique to exploit deep neural networks that have been trained on a huge amount of data from one domain in order to train a model on a new domain. In our case, the pre-trained model is an I3D network [3] for action recognition. The new domain are temporally untrimmed videos where the goal is to localize all action instances.

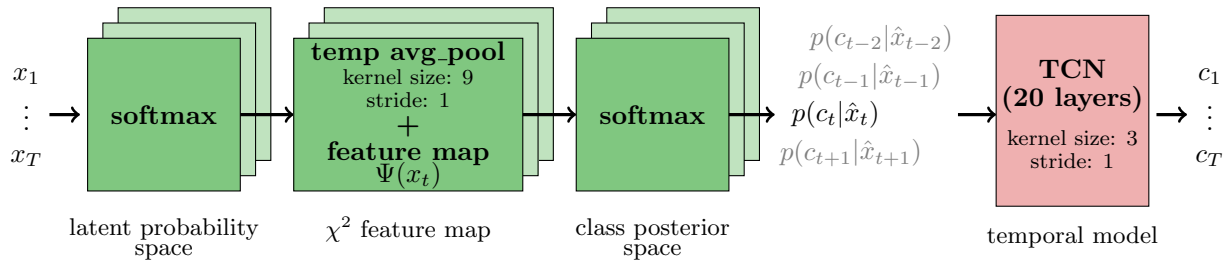

Figure 1. Overview of the BoW-Network (green). For temporal modeling, the class posteriors of the BoW-Network at each frame are used as an input into a 20-layer vanilla TCN.

3.2.1 Output Layer Retraining

We start with a discussion of a commonly used transfer learning strategy, *output layer retraining*, that does not only find application in various computer vision tasks [13, 12] but also in other fields such as speech recognition [16].

The simplest way of transfer learning is to fix all layers of a pre-trained network but the output layer. Let $\mathcal{N}(\cdot)$ denote the function realized by a network up to its penultimate layer. Given an input video with frames $\mathbf{f}_1^T = (f_1, \dots, f_T)$, we denote the sequence of framewise features that results from application of $\mathcal{N}$ to the input frames as $\mathbf{x}_1^T$.

In action recognition, the class label of a video is usually determined by averaging over multiple snippets within the video. Since this is not possible for an untrimmed video where frames have different labels, we apply a temporal smoothing over a window of $2\delta + 1$ frames centered at each x_t, *i.e.*

$$\hat{x}_t = \frac{1}{2\delta + 1} \sum_{\tau=t-\delta}^{t+\delta} x_\tau \tag{1}$$

is a smoothed feature vector for frame t. After smoothing the sequence of I3D output features $\mathbf{x}_1^T$, we retrain the output layer by optimizing the cross-entropy loss over posterior probabilities of the ground truth class c_t given the features in the window $(t - \delta, t + \delta)$,

$$p(c_t|\mathbf{x}_{t-\delta}^{t+\delta}) = p(c_t|\hat{x}_t) = \text{softmax}(\mathbf{W}^T \hat{x}_t + b). \tag{2}$$

The class posteriors for each time t are thus computed by sliding a window of size $2\delta + 1$ over the feature sequence $\mathbf{x}_1^T$ and applying the retrained output layer to the windowed frames at each temporal position.

3.2.2 BoW-Network Transfer Learning

A major drawback of output layer retraining is that it only applies a linear transformation to map from the features of the pre-trained network to the output posteriors of the new task. Given that most datasets for action localization have a rather small number of videos, just adding more layers can quickly lead to overfitting as a huge amount of additional parameters is introduced. In the following, we address this issue by replacing output layer retraining with a bag-of-words inspired neural network. A BoW-Network has first been proposed in [24] and simulates a feature quantization step as performed in the classical bag-of-words model followed by an approximation of a χ^2 kernel using explicit feature maps as proposed in [31]. The kernel approximation provides a non-linear mapping of the features without the need to introduce multiple additional network layers.

The network implements three steps which are illustrated in Figure 1 (green part). First, the features from the pre-trained network are mapped into a latent probability space that simulates a soft feature quantization as usually done with kMeans in the classical bag-of-words setup. We use $4,000$ components for this latent probability space, as this corresponds to a commonly chosen number of visual words in classical bag-of-words methods [32] and is also used in [24]. Second, the soft-quantized features are pooled along a temporal window and transformed with the non-linear function

$$\Psi(x_t) = \begin{bmatrix} \psi_{-2}(x_t) \\ \vdots \\ \psi_2(x_t) \end{bmatrix},$$

$$\psi_j(x_t) = \begin{cases} \sqrt{0.5 \cdot \kappa(0) \cdot x_t} & \text{if } j = 0, \\ \sqrt{\kappa\left(\frac{j+1}{4}\right) \cdot x_t} \cdot \cos\left(\frac{j+1}{4} \log x_t\right) & \text{if } j \text{ odd}, \\ \sqrt{\kappa\left(\frac{j}{4}\right) \cdot x_t} \cdot \sin\left(\frac{j}{4} \log x_t\right) & \text{if } j \text{ even}, \end{cases}$$

where $\kappa(\lambda) = \text{sech}(\pi\lambda)$. As shown in [31], the function Ψ approximates the feature map of the χ^2 kernel. Finally, the transformed features are projected onto class-posterior probabilities using a second softmax layer. The outputs of this layer are then the class posteriors $p(c|\mathbf{x}_{t-\delta}^{t+\delta})$.

Note that in [24] this kind of network has been used standalone, not being integrated into a larger network architecture and it has only been used for classical action recognition. We are the first to apply this technique for both, transfer learning and a sequence-to-sequence modeling task like temporal action localization.

3.2.3 Temporal Modeling Using Frame Posteriors

Both transfer learning strategies, output layer retraining and BoW-Networks, predict class posterior probabilities for the input frames. Apart from a local windowing, no temporal context is included in these posteriors. Recently, temporal convolutional networks have been a popular tool to model temporal context in video sequences [17, 6]. Recent architectures feature ecoder-decoder TCNs [17], WaveNet style networks with huge receptive fields [1], or multi-resolution networks [19]. In general, TCN architectures can be arbitrarily complex.

We show that using a deep vanilla TCN on top of the posteriors obtained from transfer learning already suffices to achieve state of the art performance. Our temporal model is a TCN with 20 temporal convolutions each with kernel size three. The network architecture does not contain any pooling operations or complex elements like gated convolutions, attention layers, or multi-scale convolutions. It can therefore be considered as a light-weight model that serves as a strong baseline for more complex networks. The source code is available online.[1]

4. Experiments

In this section, we analyze the proposed model and compare our approach to several state of the art methods on two challenging action localization benchmarks, Thumos and Hollywood Extended.

The **Thumos** dataset [11] is the most widely used dataset for temporal action localization. It features a training set with 200 videos and a test set with 212 videos. The train and test set comprise about 1.2 and 1.3 million frames, respectively. In total, there are $5,902$ action instances of 20 different classes and $6,105$ background segments. Action instances are rather sparsely distributed through the videos and about 70% of all frames are labeled as background. We follow the official evaluation protocol and report mean average precision (mAP) based on a segment-level intersection over union. A detection is considered correct if the overlap with the ground truth is larger than a specified threshold. Results are usually reported for the thresholds $0.1, 0.2, 0.3, 0.4$, and 0.5.

Hollywood Extended [2] is a dataset collected from 69 Hollywood movies. It comprises a total of 937 video clips which have been annotated with 16 different action classes. On average, there are 5.9 action instances per video. With 61%, the background ratio is similarly high as in Thumos. The overall number of frames is $780,000$. For evaluation, we use a ten-fold cross-validation where the test data for the i-th split are all videos that end with the digit $i-1$. We report the frame accuracy and average intersection over union (IoU) per class.

[1] https://github.com/alexanderrichard/coview2019

	segment classification accuracy (%)
I3D (output layer retraining)	82.2
I3D + BoW-Network	85.8

Table 1. Segment accuracy on pre-segmented action clips of the 212 videos from the Thumos test set. For training, pre-segmented action clips have been extracted from the 200 videos of the Thumos validation set.

4.1. Transfer Learning: The Use of BoW-Networks

In this section, we show that transfer learning with BoW-Networks is beneficial compared to output layer retraining. Using an I3D network pre-trained on the large scale Kinetics action recognition dataset as starting point, our results suggest that transfer learning is an effective and efficient way to improve temporal action localization. Even without the use of an explicit temporal model, results close to state of the art can be obtained.

4.1.1 Output Layer Retraining vs. BoW-Networks

While our final goal is to leverage a network trained for action recognition in order to improve temporal action localization, we first consider the task of action recognition and show that BoW-Networks result in higher classification accuracies than simple output layer retraining. Therefore, we interpret the Thumos dataset as a classical action recognition task, *i.e.* instead of using untrimmed videos, we cut each video in the dataset into its action instances and treat them as single clips that need to be classified as one of the 20 action classes or background, respectively. Since this task is a typical video classification task, a single class posterior for each video clip is required, *i.e.* for a segment $\mathbf{x}_{t_s}^{t_e}$ ranging from t_s to t_e, the posterior probability $p(c|\mathbf{x}_{t_s}^{t_e})$ needs to be modeled. Therefore, for output layer retraining, first average pooling is applied over the range (t_s, t_e) and then the softmax output layer is retrained to predict the segment classes. Similarly, for transfer learning using BoW-Networks, the average pooling step is not performed over a fixed size window but over all frames of the action segment.

In Table 1, the segment classification accuracy on Thumos is shown for both, output layer retraining and BoW-Networks. Compared to a simple retraining of the output layer, the BoW-Network achieves a 3.6% higher accuracy and is therefore a promising candidate to be applied to untrimmed videos in the context of temporal action localization. Note that both transfer learning steps are extremely fast: after extraction of the I3D features, the output layer retraining takes 12 minutes and the BoW-Network trains for 27 minutes only for the 11 hours of Thumos training data.

	mAP@				
	0.1	0.2	0.3	0.4	0.5
Current Best Systems					
Structured Segment Networks [38]	66.0	59.4	51.9	41.0	29.8
Re-thinking Faster-RCNN [4]	59.8	57.1	53.2	**48.5**	**42.8**
GTAN [22]	**69.1**	**63.7**	**57.8**	47.2	38.8
Sliding Window Baseline					
I3D (output layer retraining)	61.2	56.6	48.0	35.1	25.1
I3D + BoW-Networks	65.2	60.0	52.7	42.1	29.7

Table 2. With transfer learning from a pre-trained I3D network, even a simple sliding window baseline almost reaches state of the art performance.

Figure 2. Qualitative comparison of three approaches from Table 3 for a video from the Thumos test set containing instances of `long_jump` (green). While the background is gray, red indicates the prediction of another foreground class. First row: I3D features from penultimate layer with moving average + TCN; second row: output layer retraining + TCN; third row: BoW-Network + TCN; fourth row: ground truth.

4.1.2 Application to Temporal Action Localization

Both transfer learning approaches – retraining the output layer as well as BoW-Network based transfer learning – effectively are trained to predict class posteriors $p(c|\mathbf{x}_{t_1}^{t_2})$ for a short video snippet ranging from time t_1 to t_2. While this approach is sufficient for the classification of pre-segmented short video clips, the task of temporal action localization is more involved. Particularly, in an untrimmed video, multiple action instances and their boundaries have to be localized reliably, which usually requires a certain temporal context. While most existing approaches rely on complex and oftentimes expensive proposal and classification networks inspired by object detection, we show that with well tuned features, simple temporal models already achieve state of the art results.

As a first naive baseline, we compute the class posterior probabilities for each frame using a sliding window of width nine. In more detail, in order to obtain the posterior probabilities $p(c_t|\mathbf{x}_{t-\delta}^{t+\delta})$ for frame t, we consider the I3D features in the range $(t-\delta, t+\delta)$ with $\delta = 4$. For the approach with the retrained output layer, the features in this window are averaged and then classified using the retrained layer. In the BoW-Network, the features are processed as illustrated in Figure 1 (green part). Note that we do not perform any post-processing but evaluate the system directly on the obtained frame posteriors.

The results are shown in Table 2. The model based on frame posteriors from the pre-trained I3D network with BoW-Network transfer learning clearly outperforms the model in which the output layer of I3D has been retrained. Additionally to these simple baselines, the table contains the three currently best systems on Thumos [4, 38, 22]. Remarkably, the sliding window approach with transfer learning features is already close to the current state of the art and even outperforms two of the currently best systems in mAP at 0.2 overlap ratio.

4.2. Posterior Probabilities for Temporal Modeling

In the previous section, we have already shown that BoW-Networks and transfer learning provide strong results even without any temporal modeling. Here, we analyze how those features can facilitate temporal action localization if additional temporal modeling is applied. Therefore, we feed the learned posterior probabilities into a deep vanilla

	mAP@				
	0.1	0.2	0.3	0.4	0.5
I3D features from penultimate layer + TCN					
no feature averaging	58.1	49.8	39.9	30.5	19.7
moving average over 9-frame window	57.8	50.2	40.8	29.4	20.4
Transfer Learning + TCN					
output layer retraining + TCN	65.6	60.5	52.6	40.3	29.6
BoW-Network + TCN	68.5	63.5	55.7	45.0	31.6

Table 3. Applying a TCN on top of the 2048 dimensional features of the penultimate I3D network layer does not improve the accuracy, cf. Table 2. Temporal modeling on top of the transfer learning posteriors, however, results in much better performance.

	mAP@				
	0.1	0.2	0.3	0.4	0.5
State of the Art Approaches					
Statistical Language Model [23]	39.7	35.7	30.0	23.2	15.2
Frame Glimpses [36]	48.9	44.0	36.0	26.4	17.1
Multistage CNNs [28]	47.7	43.5	36.3	28.7	19.0
Structured Max Sums [37]	51.0	45.2	36.5	27.8	17.8
Cascaded Boundary Regression [10]	60.1	56.7	50.1	41.3	31.0
R-C3D [35]	54.5	51.5	44.8	35.6	28.9
BSN + UNet [21]	–	–	53.5	45.0	36.9
Structured Segment Networks [38]	66.0	59.4	51.9	41.0	29.8
Re-thinking Faster-RCNN [4]	59.8	57.1	53.2	**48.5**	**42.8**
GTAN [22]	**69.1**	**63.7**	**57.8**	47.2	38.8
Ours					
output layer retraining + TCN	65.6	60.5	52.6	40.3	29.6
BoW-Network + TCN	68.5	63.6	55.7	45.0	31.6

Table 4. Comparison to state of the art on Thumos. Our simple approach performs on par with the most recent complex approaches for mAP@$\{0.1, 0.2, 0.3\}$ albeit relying solely on a simple transfer learning strategy and a vanilla temporal convolutional network.

	Frame Accuracy	IoU
HTK [15]	39.5	8.4
ED-TCN [17]	36.7	10.9
TCFPN [6]	54.8	**20.4**
Ours (BoW-Network + TCN)	**61.0**	19.2

Table 5. Comparison to state of the art on Hollywood Extended.

TCN with 20 layers as described in Section 3.2.3. To show the effectiveness of posteriors as temporal features, we compare them to a system in which the 2048 dimensional output of the penultimate I3D layer are directly fed into the TCN without explicit transfer learning. For the latter setup, we provide an additional experiment where the I3D features are averaged over the same temporal window that is also used for output layer retraining and for the average pooling in the BoW-Network. This way, the temporal context is the same as for the transfer learning experiments.

The results in Table 3 show that explicit transfer learning is crucial for good results. Using the 2048 dimensional features from the penultimate I3D layer directly without transfer learning results in a degradation of about ten percent points compared to the models with transfer learning. Again, the BoW-Network shows the best performance, outperforming the output layer retraining constantly by two to five percent. Also note that despite its simplicity, the combination of BoW-Network and vanilla TCN is highly effective. Qualitative results comparing the different models are shown in Figure 2. Using I3D features from the penultimate layer with moving average and the TCN, the segmentation is prone to wrong classifications into both, background frames and frames from other action classes. Using output layer retraining as transfer learning strategy and the TCN on top, the results are already much more accurate but the model suffers from over-segmentation at the segment boundaries. The BoW-Network with TCN, on the contrary, makes stable predictions even at the action boundaries and is less sensitive to over-segmentation.

4.3. Comparison to State of the Art

We compare our approach to the current state of the art on Thumos and Hollywood Extended. The results on Thumos are shown in Table 4. For an mAP with overlap ratios below 0.4, the BoW-Network with a vanilla TCN achieve comparable results with [22] and outperforms all other approaches. For overlap ratios 0.4 and 0.5, only [4, 21, 22] perform better than our approach. Note that these approaches are complex architectures. Lin *et al*. [21] use a proposal generation module and a proposal evaluation module. Moreover, post-processing of the output is required to suppress redundant proposals. The architecture used in [4] is inspired by the Faster-RCNN for object detection and highly optimized for temporal action localization. Their architecture also requires post-processing to discard bad proposals. The method of Mei *et al*. [22] is also based on proposal generation and post processing to suppress redundant proposals.

On Hollywood Extended, we follow [6] and report frame accuracy and IoU, see Table 5. For the first, our method achieves better results than the current state of the art, for the latter, the results are only slightly lower. Note that the state of the art approaches on Hollywood Extended differ from those used on Thumos. HTK [15] is a speech recognition software that has been applied to temporal action segmentation, ED-TCN [17] is a deep temporal convolutional model that downsamples the temporal dimension in an encoding step and upsamples it in a later decoding step, and TCFPN [6] is a similar architecture with an iterative optimization scheme. Overall, our approach achieves state of the art results on both benchmarks while being highly efficient: once the I3D features of the penultimate layer are extracted, the BoW-Network plus TCN takes only 43 minutes to train on Thumos with a GTX 1080.

5. Conclusion

In this paper, we addressed the problem of temporal action localization in videos. Using transfer learning from a network pre-trained on a large scale action recognition dataset and using a bag-of-words inspired network in combination with a simple vanilla temporal convolutional network, we achieve state of the art results on two action localization benchmarks. Our approach is efficient to train and does not require complex network architectures, proposal methods, or post-processing. Overall, we show comparable performance to the best results on Thumos for several overlap ratios albeit following a rather simple approach.

Acknowledgement: The work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) GA 1927/4-1 (FOR 2535 Anticipating Human Behavior) and the ERC Starting Grant ARCA (677650).

References

[1] Yazan Abu Farha and Juergen Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[2] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conf. on Computer Vision*, 2014.

[3] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[5] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M. Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *European Conf. on Computer Vision*, 2018.

[6] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[7] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal multiplier networks for video action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[8] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Temporal residual networks for dynamic scene recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[9] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Int. Conf. on Computer Vision*, 2017.

[10] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *British Machine Vision Conference*, 2017.

[11] Haroon Idrees, Amir Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[12] Andrej Karpathy, George Toderici, Sachin Shetty, Tommy Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[14] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[15] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 2017.

[16] Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. Transfer learning for speech recognition on a budget. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 168–177, 2017.

[17] Colin Lea, Michael Flynn, René Vidal, Austin Reiter, and Gregory Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[18] Colin Lea, Austin Reiter, René Vidal, and Gregory Hager. Segmental spatiotemporal CNNs for fine-grained action segmentation. In *European Conf. on Computer Vision*, 2016.

[19] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[20] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM Conf. on Multimedia*, 2017.

[21] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *European Conf. on Computer Vision*, 2018.

[22] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[23] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[24] Alexander Richard and Juergen Gall. A bag-of-words equivalent recurrent neural network for action recognition. *Computer Vision and Image Understanding*, 156:79–91, 2017.

[25] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with RNN based fine-to-coarse modeling. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[26] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. NeuralNetwork-Viterbi: A framework for weakly supervised video learning. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[27] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[28] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage CNNs. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014.

[30] Sebastian Stein and Stephen McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, 2013.

[31] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 480–492, 2012.

[32] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal on Computer Vision*, 103(1):60–79, 2013.

[33] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Int. Conf. on Computer Vision*, 2013.

[34] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conf. on Computer Vision*, 2016.

[35] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: region convolutional 3d network for temporal activity detection. In *Int. Conf. on Computer Vision*, 2017.

[36] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[37] Zehuan Yuan, Jonathan Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[38] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Int. Conf. on Computer Vision*, 2017.