Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

TLS-RWKV: Real-time Online Action Detection with Temporal Label Smoothing

Ziqi Zhu
National Computer System Engineering Research Institute of China
Wuchang Shao
National Computer System Engineering Research Institute of China
Dongdong Jiao
jiaodong1108@gmail.com

National Computer System Engineering Research Institute of China

Research Article

Keywords: Online action detection, Online detection of action start, Label smoothing, Computer vision

Posted Date: September 1st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3305376/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Neural Processing Letters on February 19th, 2024. See the published version at https://doi.org/10.1007/s11063-024-11540-0.

TLS-RWKV: Real-time Online Action Detection with Temporal Label Smoothing

Ziqi Zhu[1], Wuchang Shao[1], Dongdong Jiao[1*]

[1*]National Computer System Engineering Research Institute of China, No.25 Tsing Hua East Road, Haidian, 100083, Beijing, China.

*Corresponding author(s). E-mail(s): jiaodong1108@gmail.com;
Contributing authors: lqjszzq@gmail.com; shaowc5@gmail.com;

Abstract

Online action detection (OAD) is a challenging task that involves predicting the ongoing action class in real-time from streaming videos. In this article, we propose a novel approach for OAD based on the RWKV model and introducing temporal label smoothing. The RWKV model effectively captures temporal dependencies while maintaining computational efficiency, making it well-suited for real-time applications. The temporal label smoothing technique enhances the model's robustness in handling ambiguous boundaries and feature similarities near action boundaries. We conducted experiments on two widely used datasets, THUMOS'14 and TVSeries, to evaluate the performance of our proposed approach. Our model achieves state-of-the-art performance with 71.8% mAP on THUMOS'14 and 89.7% cAP on TVSeries. Furthermore, our proposed approach demonstrates impressive efficiency, running at over 600 FPS and maintaining a competitive mAP of 59.9% on THUMOS'14 with RGB features alone. This high efficiency makes our model suitable for real-time deployment, even on resource-constrained devices. These results showcase the effectiveness and competitiveness of our proposed approach in OAD.

Keywords: Online action detection, Online detection of action start, Label smoothing, Computer vision

1 Introduction

In recent years, the widespread use of digital devices has led to an exponential growth in video data, creating a need for efficient methods to automatically analyze and

understand actions in these videos. Online action detection (OAD) [1] aims to classify ongoing actions in real-time from streaming videos, which is useful for applications like autonomous driving[2] and video surveillance[3].

A typical OAD model first extracts features at the snip-level and then predicts the action class over the feature sequence. Early works [4–9] in online action detection utilized Recurrent Neural Networks (RNNs) to model history information. However, RNN-based models suffer from non-parallelism and gradient vanishing [10, 11], resulting in training difficulties and sub-optimal performance.

Recent works [12–15] have introduced the Transformer architecture [16] to address these issues. Despite its benefits, the real-time limitation in OAD restricts the Transformer's ability to handle long context lengths, potentially missing critical long temporal dependencies in streaming video. Additionally, previous works often assign the label of a specific time point to the entire snippet, which is arbitrary and leaves room for improvement.

To address these issues, we propose a online action detection model that leverages the emerging long sequence model RWKV[17] to enhance both performance and efficiency. Additionally, we introduce a temporal label smoothing technique for online action detection.

We conducted experiments on two widely used datasets: THUMOS'14[18] and TVSeries[1]. Our model achieves state-of-the-art performance of 71.8% mAP on THUMOS'14 and 89.7% mcAP on TVSeries. In terms of efficiency, our model can run at 600+FPS alone, making it applicable for real-time online action prediction. Notably, when using only RGB features, the overall system can run at 200+FPS even on CPU while still retaining 59.9% mAP on THUMOS'14, which makes it possible to deploy on edge computing devices.

2 Related Works

Online Action Detection

In the field of online action detection, several notable works have contributed significant advancements. RED[6] introduces a reinforcement loss function to promote early action detection. TRN [7] performs online action detection and anticipation simultaneously, utilizing the predicted future information to improve the performance. IDN [8] incorporates an information discrimination unit to selectively accumulate relevant information for the present action. PKD [9] utilizes curriculum knowledge distillation to transfer knowledge from offline models. Colar [19] employs an exemplar-consultation mechanism to compare similarities and aggregate relevant information. LSTR[13] proposes a long short-term memory mechanism to efficiently modeling video sequence using transformer. GateHub[14] designs a gated history unit to enhance the relevant history information.

Transformer-based models[12–15] have demonstrated promising performances in online action detection. However, the square compute complexity of self-attention brings significant computational cost, especially when dealing with long video streams. OadTR[12] only keeps a few recent frames to reduce computational demands. LSTR improves efficiency by performing compression on long-term memories using

Perceiver[20], and TesTra[15] further improves efficiency by replacing the first layer of LSTR encoder with linear attention[21]. But LSTR/TesTra still have fixed context length and excessive token reduction could result in the loss of critical information.

Online Detection of Action Start

Online Detection of Action Start (ODAS) is a task closely linked to online action detection, focusing on identifying the precise starting point of an action instance and minimizing the time gap between the actual start time and prediction. Few works specifically focus on this task. Shou et al.[22] first introduced this task and proposed three methods to train an ODAS model. StartNet[23] decompose ODAS into two stage: action classification and start point localization, addressing the challenges of subtle appearance near action starts and limited training data. While WOAD[24] and SCOAD[25] also conduct experiment on this task, their original designs were intended for weakly supervised online action detection.

Efficient Long Sequence Modeling

While the Transformer model has demonstrated remarkable capability in handling long-distance dependencies, it is hindered by the square compute complexity of cross self-attention. To address this challenge, a variety of approaches have been proposed. Some focus on optimizing the attention mechanism, employing techniques such as sparse self-attention[26], kernelization[21], low-rank approximations[27], and other methods[28, 29]. Other researches explore alternative modules to replace attention. MLP-Mixer[30] replaces attention with Multilayer Perceptrons(MLPs), while the Attention Free Transformer(AFT)[31] introduces a computationally efficient alternative to the traditional dot-product self-attention mechanism. Inspired by AFT, RWKV[17] simplifies interaction weights to enable a RNN-style implementation for inference. For a comprehensive overview of more efficient Transformer variants, a survey by Tay et al.[32] can be referenced. Additionally, some approaches modify recurrent neural networks(RNN) to increase context length, such as the Recurrent Memory Transformer[33], Linear Recurrent Unit[34], and state space models(SSM)[35–38]. These techniques offer alternative strategies to enhance the context modeling capabilities of sequence-based models.

3 Approach

OAD aims to recognize actions in a video stream with only current and historical information. Mathematically, an OAD model provides an action class probability vector $\hat{\boldsymbol{y}}_t \in \mathbb{R}^k$ for the current frame $\boldsymbol{f}_t \in \mathbb{R}^s$ at time t, based on the sequence of current and historical frames $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_t\}$. Here, k represents the number of action classes, including the background class, and s denotes the size of the input frame. To achieve this, we employ a pretrained feature extractor[39] to process each video frame $\boldsymbol{f}_t$ into a feature vector $\boldsymbol{x}_t \in \mathbb{R}^d$ of d dimensions[1]. These feature vectors are then fed into our model. The overall architecture of our model is illustrated in Fig. 1.

[1]For simplicity, we use the single frame notation, although many works use feature vectors extracted from small video snippets in practice.

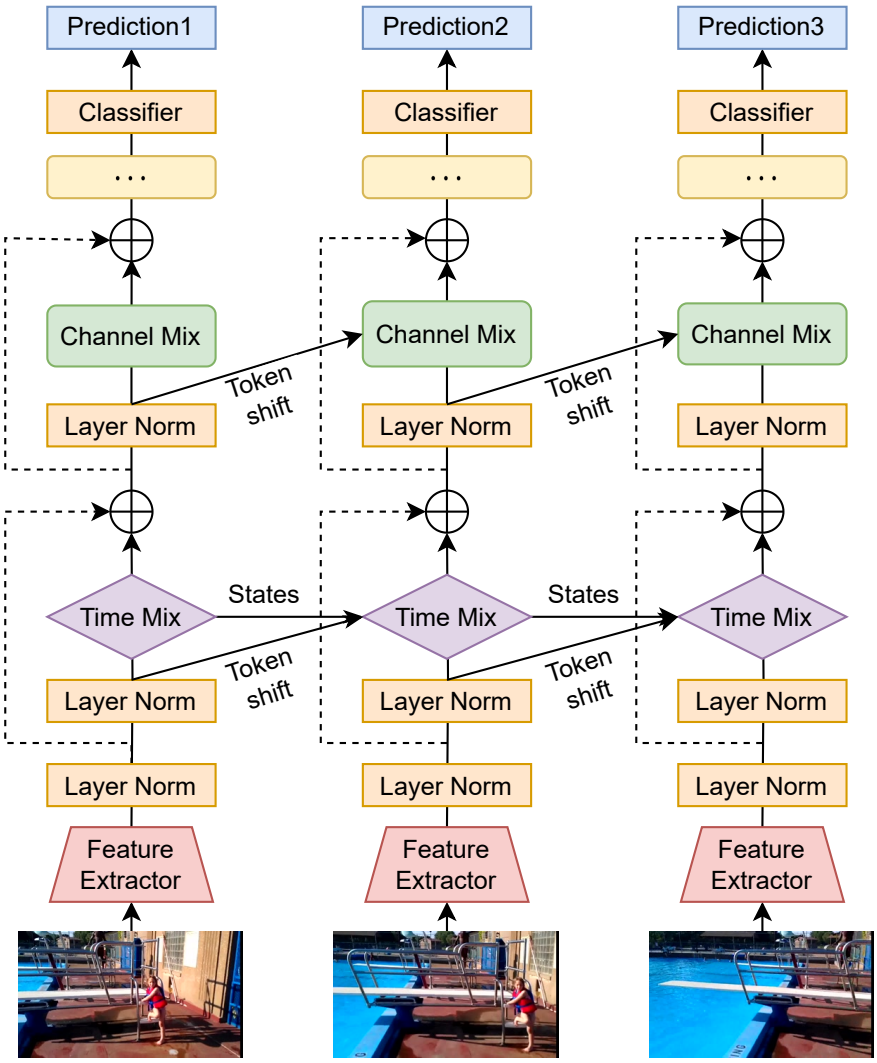


Fig. 1 Overview of our model. Raw video frames are initially processed using a pretrained feature extractor to obtain a feature sequence. The feature sequence is then fed into the RWKV model to capture temporal dependencies. Finally, a fully connected layer(classifier) generates the action class prediction.

3.1 Base Model

We utilize the RWKV model to capture the temporal dependencies, which offers a balance between performance and efficiency. A RWKV layer contains a time-mixing block and a channel-mixing block. In the time-mixing block, we employ the original formulas

$$r_t = W_r \cdot (\mu_r x_t + (1 - \mu_r) x_{t-1}), \tag{1}$$

$$k_t = W_k \cdot (\mu_k x_t + (1 - \mu_k) x_{t-1}), \tag{2}$$
$$v_t = W_v \cdot (\mu_v x_t + (1 - \mu_v) x_{t-1}), \tag{3}$$
$$wkv_t = \frac{\sum_{i=1}^{t-1} e^{-(t-1-i)w+k_i} v_i + e^{u+k_t} v_t}{\sum_{i=1}^{t-1} e^{-(t-1-i)w+k_i} + e^{u+k_t}}, \tag{4}$$
$$o_t = W_o \cdot (\sigma(r_t) \odot wkv_t), \tag{5}$$

where equations (1)-(3) represent the token shift operation, and equation (4) plays the role of cross-attention in standard Transformers. Equation (4) can be efficiently rewritten into recursion-style with linear computational cost, potentially enabling the model to handle infinite context length and effectively capture long-term dependencies in streaming video.

The channel-mixing block works as the FFN layer in regular Transformers and it is given by

$$r_t = W_r \cdot (\mu_r x_t + (1 - \mu_r) x_{t-1}), \tag{6}$$
$$k_t = W_k \cdot (\mu_k x_t + (1 - \mu_k) x_{t-1}), \tag{7}$$
$$o_t = \sigma(r_t) \odot (W_v \cdot f_{\text{laplace}(k_t)}), \tag{8}$$

where we replace the original squared ReLU activation[40] with the Laplace activation[28]:

$$f_{\text{laplace}} = 0.5 \times \left[1 + \text{erf}(\frac{x - \mu}{\sigma\sqrt{2}})\right], \mu = 1/\sqrt{2}, \sigma = 1/\sqrt{4\pi}. \tag{9}$$

The Laplace activation is an approximation of the squared ReLU activation with both a bounded range and gradient, which can address the gradient explosion issue.[28] We find this modification increases the stability of the model.

3.2 Temporal Label Smoothing

Due to the less explicit boundaries of actions and the higher similarity of features near the action boundaries compared to objects which may confuse the model, we propose a temporal smoothing technique to refine the ground-truth labels. Given a ground-truth label $G(t)$, a general temporal smoothing operation is defined as

$$G^*(t) = \int_{-\infty}^{+\infty} \phi(\tau) \cdot G(t - \tau) \mathrm{d}\tau. \tag{10}$$

Here, we choose Gaussian function as the kernel $\phi(\tau)$

$$\phi(\tau) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\tau - \mu)^2}{2\sigma^2}\right]. \tag{11}$$

which assumes that the ground-truth labels have errors following a normal distribution. By applying the Gaussian smoothing, nearby time points contribute to the refined

label $G^*(t)$ in a continuous and gradual manner, effectively reducing the impact of label uncertainties and ambiguities near action boundaries. For a video snippet, the calibrated ground-truth label is obtained by integrating over time. We utilize $G^*(t)$ to train our model, which makes it more robust to handle the ambiguous boundaries and feature similarities near action boundaries.

3.3 Training Model

For data augmentation, we utilize a technique similar to the temporal VideoMix[41] and randomly stack an average of 3 video clips as one input sample during training. To train our model, we employ the cross-entropy loss between the predicted probability vector $\hat{\boldsymbol{y}}_t$ and the ground-truth label $\boldsymbol{y}_t \in \mathbb{R}^k$ at each frame. The per-frame loss $\mathcal{L}_t$ is calculated as follows:

$$\mathcal{L}_t = -\sum_{i=0}^{k} y_t^i \log \hat{y}_t^i, \tag{12}$$

where y_t^i and $\hat{y}t^i$ represent the i-th element of the ground-truth and predicted vectors, respectively, at time t. The total loss is obtained by summing over each frame:

$$\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}_t, \tag{13}$$

where T represents the total number of frames in the video.

4 Experiments

4.1 Datasets and Evaluation Metrics

THUMOS'14

The THUMOS'14 dataset includes 413 untrimmed videos with 20 action classes for online action detection. The dataset is split into 200 training videos and 213 testing videos according to the public data split. In line with previous studies[12–15, 23–25], we evaluate OAD using per-frame mean Average Precision (mAP) and evaluate ODAS using point-level mean Average Precision (pAP)[22].

Most existing works follow the two-stream paradigm[42], utilizing both optical flow and RGB features to achieve optimal performance. However, this leads to high computational costs due to optical flow computation. Hence, we also evaluate the performance of OAD using mAP solely with RGB features. This evaluation allows us to specifically focus on scenarios where computational efficiency is a critical factor to consider.

TVSeries

The TVSeries dataset includes 16 hours of 27 untrimmed videos with 30 everyday action classes, collected from 6 TV series. This dataset is originally designed for OAD and introduces the calibrated Average Precision (cAP)[1] as the evaluation metric. Therefore, we employ cAP as the evaluation metric for the TVSeries dataset.

4.2 Implement Details

We implemented our model using PyTorch[43] and conducted all experiments on a system equipped with an Intel i9-12900 CPU and an NVIDIA RTX 3090 graphics card. We followed the approach outlined in [13–15] to preprocess the videos, where we converted all videos to 24 frames per second (FPS) and then extracted features at 4 FPS. For feature extraction, we utilized the TSN models pretrained on the Kinetics-400 dataset[44], which were implemented in the MMAction2 framework[45]. The TSN model is a two-stream network, where we employed Resnet-50[46] for handling RGB features and BN-Inception[47] for optical flow features.

Regarding the RWKV model, we set the hidden dimension size to 512 and used 4 RWKV layers. Instead of the initialization proposed in RWKV, we employed Kaiming Initialization[48] for our model.

During the training process, we used a batch size of 1 since multiple video clips were already stacked together. The model was trained for 30 epochs using the Adam optimizer[49] with a weight decay of 9e-8 and a base learning rate of 6e-4. The learning rate was linearly increased from zero to 6e-4 during the first 10 epochs and then reduced to zero following a cosine function.

4.3 Main Results

Online Action Detection

We present a comparison of our method with recent works on OAD task in Table 1. All the compared models utilize TSN features pretrained on the Kinetics-400 dataset as input. Notably, our model achieves 71.8% mAP on THUMOS'14 and 89.7% cAP on TVSeries, surpassing the performance of all previous method. Moreover, in the RGB-only setup, our model achieves a remarkable mAP of 59.9%, effectively narrowing the performance gap between the two-stream method and the RGB-only method. These results highlight the efficacy and competitiveness of our proposed approach in online action detection.

Table 1 Online action detection results on THUMOS'14 and TVSeries using TSN features pretrained on Kinetics-400.

	THUMOS'14 mAP(%)		TVSeries mcAP(%)
	OF+RGB	RGB	OF+RGB
IDN[8]	60.3	-	86.1
TRN[7]	62.1	-	86.2
OadTR[12]	65.2	51.2	87.2
Colar[19]	66.9	52.1	88.1
LSTR[13]	69.5	-	89.1
GateHUB[14]	70.7	-	89.6
TesTra[15]	71.2	56.4*	-
Ours	**71.8**	**59.9**	**89.7**

*This result was obtained using the official code base and run by ourselves, as it was not reported in the paper.

Online Detection of Action Start

We present a comparison of our method with recent works on ODAS task in Table 2. Both StartNet and our model utilize the TSN feature extractor, while WOAD and SCOAD employ the I3D[50] feature extractor. It is worth noting that both feature extractors, TSN and I3D, are pretrained on the Kinetics-400 dataset and demonstrate similar performance characteristics, ensuring a fair comparison.

Our model consistently achieves the best results across various time thresholds, demonstrating superior performance in most cases. Even in scenarios where our model does not secure the top position, it still achieves the second-best results. These outcomes highlight the effectiveness and competitiveness of our proposed approach compared to state-of-the-art methods for the ODAS.

Table 2 Online detection of action start results on THUMOS'14 using feature extractor pretrained on Kinetics-400. StartNet and our model utilize the TSN feature extractor, while WOAD and SCOAD employ the I3D feature extractor. The bold numbers indicate the best results, while the underlined numbers denote the second best results.

	pAP@Time Threshold(Seconds)									
	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
StartNet[23]	21.9	33.5	39.6	42.5	46.2	46.6	47.7	48.3	48.6	49.0
WOAD[24]	28.0	40.6	45.7	48.0	50.1	51.0	51.9	52.4	53.0	53.1
SCOAD[25]	**30.6**	**42.3**	**48.2**	<u>51.9</u>	**54.5**	<u>55.4</u>	<u>56.0</u>	<u>56.5</u>	<u>56.9</u>	<u>57.0</u>
Ours	<u>28.7</u>	<u>41.6</u>	<u>48.1</u>	**52.2**	<u>54.3</u>	**55.5**	**57.9**	**58.9**	**59.8**	**60.4**

4.4 Ablation Study

We conducted ablation experiments on the THUMOS'14 dataset to analyze the performance of our model. We evaluated four different setups, as follows:

Baseline

This setup represents the model with raw RWKV layers and serves as the baseline for comparison.

Baseline+LA

In this setup, we replaced the original squared ReLU activation in RWKV with the Laplace activation to examine the impact of this modification.

Baseline+TLS

We used the proposed temporal smoothed ground-truth labels to train the baseline model to assess the effectiveness of temporal label smoothing.

Baseline+LA+TLS

This setup represents our final proposed method, where we combined the Laplace activation and temporal label smoothing.

Table 3 presents the performance comparison of these different methods on the THUMOS'14 dataset, demonstrating the effectiveness of each component in improving the model's performance.

Table 3 Ablation experiments on THUMOS'14 evaluating the impact of Laplace Activation (LA) and Temporal Label Smoothing (TLS).

Baseline	LA	TLS	mAP(%)
✓	×	×	71.4
✓	✓	×	71.6
✓	×	✓	71.5
✓	✓	✓	71.8

4.5 Efficiency Analysis

We compare the efficiency of our method with TesTra in terms of online inference. To ensure a fair comparison, we re-run TesTra on our platform to control variables. The results are presented in Table 4. Our model achieves a speed that is over two times faster than TesTra, with a slight 0.6% improvement in performance. Additionally, we observe that our method experiences minimal drop in speed when running on a pure CPU platform, achieving an impressive 206.2 FPS for end-to-end online inference with RGB features only. This suggests that our model is suitable for deployment on low-end platforms, such as edge computing devices.

Table 4 Running speed in frames per second (FPS) comparison. The asterisk (*) denotes using RGB features only. The double asterisk (**) indicates that the experiment runs on a CPU with RGB features only.

	OF Comp.	OF Feat.	RGB Feat.	Model	Overall	mAP(%)
TesTra	11.2	97.5	754.4	263.5	11.2	71.2
TesTra*	-	-	754.4	263.5	263.5	56.4
TesTra**	-	-	206.2	110.9	110.9	56.4
Ours	11.2	97.5	754.4	624.3	11.2	71.8
Ours*	-	-	754.4	624.3	624.3	59.9
Ours**	-	-	206.2	599.7	206.2	59.9

5 Conclusion and Future Work

In this study, we proposed a novel model for online action detection, which combines the RWKV architecture with temporal label smoothing. Our model achieved significant improvements in both performance and efficiency. Experimental results on the

THUMOS'14 and TVSeries datasets showcased the superiority of our approach, surpassing state-of-the-art methods in terms of mAP and cAP. Moreover, our model demonstrated impressive inference speed, outperforming the TesTra model by more than two times while maintaining competitive performance. Notably, our model exhibited remarkable efficiency even on CPU platforms, enabling deployment on resource-constrained devices. In future work, it would be valuable to explore alternatives to optical flow computation or ways to reduce its reliance, as it is the running speed bottleneck for the overall system.

Statements and Declarations

Conflict of Interests The authors have no relevant financial or nonfinancial interests to disclose.The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

[1] De Geest R, Gavves E, Ghodrati A, et al (2016) Online action detection. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14, Springer, pp 269–284, https://doi.org/10.1007/978-3-319-46454-1_17

[2] Kim J, Misu T, Chen YT, et al (2019) Grounding human-to-vehicle advice for self-driving vehicles. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 10583–10591, https://doi.org/10.1109/CVPR.2019.01084

[3] Shu T, Xie D, Rothrock B, et al (2015) Joint inference of groups, events and human roles in aerial videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4576–4584, https://doi.org/10.1109/CVPR.2015.7299088

[4] De Geest R, Tuytelaars T (2018) Modeling temporal structure with lstm for online action detection. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1549–1557, https://doi.org/10.1109/WACV.2018.00173

[5] Li Y, Lan C, Xing J, et al (2016) Online human action detection using joint classification-regression recurrent neural networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14, Springer, pp 203–220, https://doi.org/10.1007/978-3-319-46478-7_13

[6] Gao J, Yang Z, Nevatia R (2017) Red: Reinforced encoder-decoder networks for action anticipation. In: BMVC, https://doi.org/10.5244/c.31.92

[7] Xu M, Gao M, Chen YT, et al (2019) Temporal recurrent networks for online action detection. In: Proceedings of the IEEE/CVF international conference on

computer vision, pp 5532–5541, https://doi.org/10.1109/ICCV.2019.00563

[8] Eun H, Moon J, Park J, et al (2020) Learning to discriminate information for online action detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 806–815, https://doi.org/10.1109/CVPR42600.2020.00089

[9] Zhao P, Xie L, Wang J, et al (2022) Progressive privileged knowledge distillation for online action detection. Pattern Recognition 129:108741. https://doi.org/10.1016/j.patcog.2022.108741, URL https://www.sciencedirect.com/science/article/pii/S0031320322002229

[10] Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: Dasgupta S, McAllester D (eds) Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 28. PMLR, Atlanta, Georgia, USA, pp 1310–1318, URL https://proceedings.mlr.press/v28/pascanu13.html

[11] Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:150600019 https://doi.org/10.48550/arXiv.1506.00019

[12] Wang X, Zhang S, Qing Z, et al (2021) Oadtr: Online action detection with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 7565–7575, https://doi.org/10.1109/ICCV48922.2021.00747

[13] Xu M, Xiong Y, Chen H, et al (2021) Long short-term transformer for online action detection. In: Ranzato M, Beygelzimer A, Dauphin Y, et al (eds) Advances in Neural Information Processing Systems, vol 34. Curran Associates, Inc., pp 1086–1099, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/08b255a5d42b89b0585260b6f2360bdd-Paper.pdf

[14] Chen J, Mittal G, Yu Y, et al (2022) Gatehub: Gated history unit with background suppression for online action detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 19893–19902, https://doi.org/10.1109/CVPR52688.2022.01930

[15] Zhao Y, Krähenbühl P (2022) Real-time online video detection with temporal smoothing transformers. In: Avidan S, Brostow G, Cissé M, et al (eds) Computer Vision – ECCV 2022. Springer Nature Switzerland, Cham, pp 485–502, https://doi.org/10.1007/978-3-031-19830-4_28

[16] Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, et al (eds) Advances in Neural Information Processing Systems, vol 30. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[17] Peng B, Alcaide E, Anthony Q, et al (2023) Rwkv: Reinventing rnns for the transformer era. arXiv preprint arXiv:230513048 https://doi.org/10.48550/arXiv.2305.13048

[18] Jiang YG, Liu J, Roshan Zamir A, et al (2014) THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/

[19] Yang L, Han J, Zhang D (2022) Colar: Effective and efficient online action detection by consulting exemplars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3160–3169, https://doi.org/10.1109/CVPR52688.2022.00316

[20] Jaegle A, Gimeno F, Brock A, et al (2021) Perceiver: General perception with iterative attention. In: Meila M, Zhang T (eds) Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 139. PMLR, pp 4651–4664, URL https://proceedings.mlr.press/v139/jaegle21a.html

[21] Katharopoulos A, Vyas A, Pappas N, et al (2020) Transformers are RNNs: Fast autoregressive transformers with linear attention. In: III HD, Singh A (eds) Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 119. PMLR, pp 5156–5165, URL https://proceedings.mlr.press/v119/katharopoulos20a.html

[22] Shou Z, Pan J, Chan J, et al (2018) Online detection of action start in untrimmed, streaming videos. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 534–551, https://doi.org/10.1007/978-3-030-01219-9_33

[23] Gao M, Xu M, Davis LS, et al (2019) Startnet: Online detection of action start in untrimmed videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 5542–5551, https://doi.org/10.1109/ICCV.2019.00564

[24] Gao M, Zhou Y, Xu R, et al (2021) Woad: Weakly supervised online action detection in untrimmed videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1915–1923, https://doi.org/10.1109/CVPR46437.2021.00195

[25] Ye N, Zhang X, Yan D, et al (2022) Scoad: Single-frame click supervision for online action detection. In: Proceedings of the Asian Conference on Computer Vision, pp 2156–2171, https://doi.org/10.1007/978-3-031-26316-3_14

[26] Child R, Gray S, Radford A, et al (2019) Generating long sequences with sparse transformers. arXiv preprint arXiv:190410509 https://doi.org/10.48550/arXiv.1904.10509

[27] Wang S, Li BZ, Khabsa M, et al (2020) Linformer: Self-attention with linear complexity. arXiv preprint arXiv:200604768 https://doi.org/10.48550/arXiv.

2006.04768

[28] Ma X, Zhou C, Kong X, et al (2022) Mega: moving average equipped gated attention. arXiv preprint arXiv:220910655 https://doi.org/10.48550/arXiv.2209.10655

[29] Hua W, Dai Z, Liu H, et al (2022) Transformer quality in linear time. In: Chaudhuri K, Jegelka S, Song L, et al (eds) Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 162. PMLR, pp 9099–9117, URL https://proceedings.mlr.press/v162/hua22a.html

[30] Tolstikhin IO, Houlsby N, Kolesnikov A, et al (2021) Mlp-mixer: An all-mlp architecture for vision. In: Ranzato M, Beygelzimer A, Dauphin Y, et al (eds) Advances in Neural Information Processing Systems, vol 34. Curran Associates, Inc., pp 24261–24272, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Paper.pdf

[31] Zhai S, Talbott W, Srivastava N, et al (2021) An attention free transformer. arXiv preprint arXiv:210514103 https://doi.org/10.48550/arXiv.2105.14103

[32] Tay Y, Dehghani M, Bahri D, et al (2022) Efficient transformers: A survey. ACM Comput Surv 55(6). https://doi.org/10.1145/3530811

[33] Bulatov A, Kuratov Y, Burtsev M (2022) Recurrent memory transformer. In: Koyejo S, Mohamed S, Agarwal A, et al (eds) Advances in Neural Information Processing Systems, vol 35. Curran Associates, Inc., pp 11079–11091, URL https://proceedings.neurips.cc/paper_files/paper/2022/file/47e288629a6996a17ce50b90a056a0e1-Paper-Conference.pdf

[34] Orvieto A, Smith SL, Gu A, et al (2023) Resurrecting recurrent neural networks for long sequences. arXiv preprint arXiv:230306349 https://doi.org/10.48550/arXiv.2303.06349

[35] Gu A, Goel K, Re C (2021) Efficiently modeling long sequences with structured state spaces. In: International Conference on Learning Representations, https://doi.org/10.48550/arXiv.2111.00396

[36] Gupta A, Gu A, Berant J (2022) Diagonal state spaces are as effective as structured state spaces. In: Koyejo S, Mohamed S, Agarwal A, et al (eds) Advances in Neural Information Processing Systems, vol 35. Curran Associates, Inc., pp 22982–22994, URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9156b0f6dfa9bbd18c79cc459ef5d61c-Paper-Conference.pdf

[37] Nguyen E, Goel K, Gu A, et al (2022) S4nd: Modeling images and videos as multidimensional signals with state spaces. In: Koyejo S, Mohamed S, Agarwal A, et al (eds) Advances in Neural Information Processing Systems, vol 35. Curran

Associates, Inc., pp 2846–2861, URL https://proceedings.neurips.cc/paper_files/paper/2022/file/13388efc819c09564c66ab2dc8463809-Paper-Conference.pdf

[38] Smith JT, Warrington A, Linderman S (2022) Simplified state space layers for sequence modeling. In: The Eleventh International Conference on Learning Representations, https://doi.org/10.48550/arXiv.2208.04933

[39] Wang L, Xiong Y, Wang Z, et al (2016) Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision, Springer, pp 20–36, https://doi.org/10.1007/978-3-319-46484-8_2

[40] So DR, Mańke W, Liu H, et al (2021) Primer: Searching for efficient transformers for language modeling. arXiv preprint arXiv:210908668 https://doi.org/10.48550/arXiv.2109.08668

[41] Yun S, Oh SJ, Heo B, et al (2020) Videomix: Rethinking data augmentation for video classification. arXiv preprint arXiv:201203457 https://doi.org/10.48550/arXiv.2012.03457

[42] Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Ghahramani Z, Welling M, Cortes C, et al (eds) Advances in Neural Information Processing Systems, vol 27. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf

[43] Paszke A, Gross S, Massa F, et al (2019) Pytorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, et al (eds) Advances in Neural Information Processing Systems, vol 32. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[44] Kay W, Carreira J, Simonyan K, et al (2017) The kinetics human action video dataset. arXiv preprint arXiv:170506950 https://doi.org/10.48550/arXiv.1705.06950

[45] Contributors M (2020) Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2

[46] He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778, https://doi.org/10.1109/CVPR.2016.90

[47] Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach F, Blei D (eds) Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 37. PMLR, Lille, France, pp 448–456, URL https://proceedings.mlr.press/v37/ioffe15.html

[48] He K, Zhang X, Ren S, et al (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034, https://doi.org/10.1109/ICCV.2015.123

[49] Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 https://doi.org/10.48550/arXiv.1412.6980

[50] Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6299–6308, https://doi.org/10.1109/CVPR.2017.502