Neural Processing Letters (2024) 56:57
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]

Accepted: 9 January 2024 / Published online: 19 February 2024

Abstract
Online action detection (OAD)is a challenging task that involves predicting the ongoing action class in real-time streaming videos, which is essential in the field of autonomous driving and video surveillance. In this article, we propose an approach for OAD based on the Receptance Weighted Key Value (RWKV) model with temporal label smooth. The RWKV model captures temporal dependencies and computes efficiently at the same time, which makes it well-suited for real-time applications. Our TLS-RWKV model demonstrates advancements in two aspects. First, we conducted experiments on two widely used datasets, THUMOS'14 and TVSeries. Our proposed approach demonstrates state-of-the-art performance with 71.8% mAP on THUMOS'14 and 89.7% cAP on TVSeries. Second, 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. Notably, this efficiency surpasses the prior state-of-the-art model, TesTra, by more than two times. Even when executed on a CPU, our model maintains a commendable speed, exceeding 200 FPS. 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 led to an exponential growth in video data, creating a need for efficient methods to automatically analyze and understand actions

✉ Dongdong Jiao
jiaodong1108@gmail.com

Ziqi Zhu
lqjszzq@gmail.com

Wuchang Shao
shaowc5@gmail.com

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

in these videos. Conventional video analysis methods, while effective, often lack real-time responsiveness. Online Action Detection (OAD) [1] aims to classify ongoing actions in real-time streaming videos, which plays a pivotal role in real-time video analysis for applications such as autonomous driving [2] and video surveillance [3].

The OAD task poses a significant challenge, primarily due to the real-time constraint. The crux of the challenge lies in striking a delicate balance between achieving optimal performance and maintaining computational efficiency.

A typical OAD model first extracts features at the snippet 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 an online action detection model that leverages the emerging long sequence model RWKV [17] to enhance both performance and efficiency. RWKV stands out as an innovative and promising model, with both Transformer-like and RNN-style formulations. This unique characteristic enables training as a Transformer while inferencing as an RNN model, thus achieving an optimal balance between performance and efficiency. In adapting RWKV for the OAD task, we employ the Laplace activation to tailor it specifically for OAD requirements. Additionally, we introduce a temporal label smoothing technique for online action detection.

In summary, our primary contribution involves the adaptation of the RWKV architecture for the OAD task. Additionally, we introduce the temporal label smoothing technique to further enhance robustness.

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 model video sequences 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 computational 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 has a 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] decomposes ODAS into two stages: 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 experiments 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 computational 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 researchers 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 an 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.

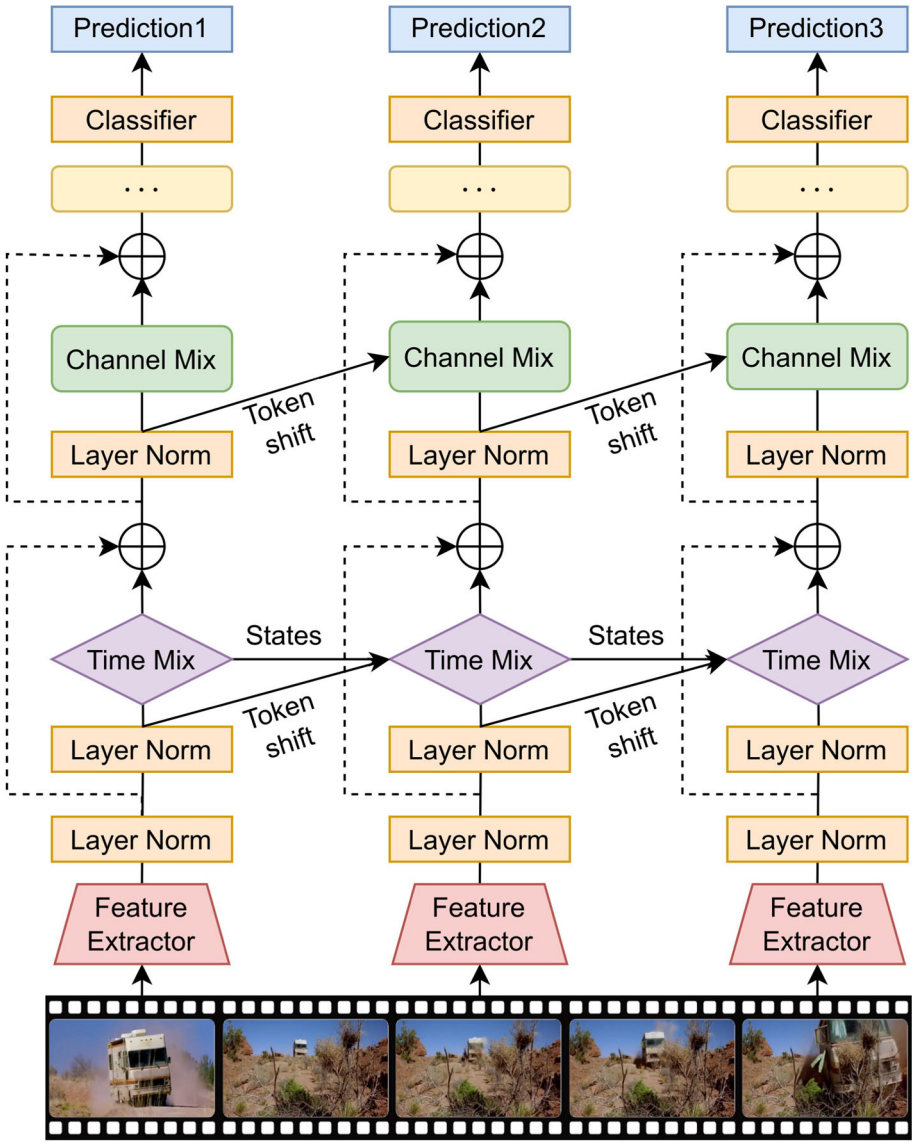

Fig. 1 Overview of our model. Raw video frames are initially processed using a pretrained feature extractor to obtain a feature sequence. The feature sequence is then fed into the RWKV model to capture temporal dependencies. Finally, a fully connected layer (classifier) generates the action class prediction

3.1 Base Model

The overall architecture of our model is illustrated in Fig. 1. To complete the OAD task, 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.

We utilize the RWKV model to capture the temporal dependencies, which offers a balance between performance and efficiency. A RWKV model comprises multiple RWKV layers, each containing a time-mixing block and a channel-mixing block illustrated in Fig. 1. In the time-mixing block, we employ the original RWKV 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, aiding the model in better information propagation. Equation (4) plays the role of cross-attention in standard Transformers. In this formulation, the model can be trained in a parallel manner, similar to Transformers. Equation (4) can also be easily 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]:

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

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. After the RWKV layers, a fully connected layer (classifier) generates the action class prediction for the current time.

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

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

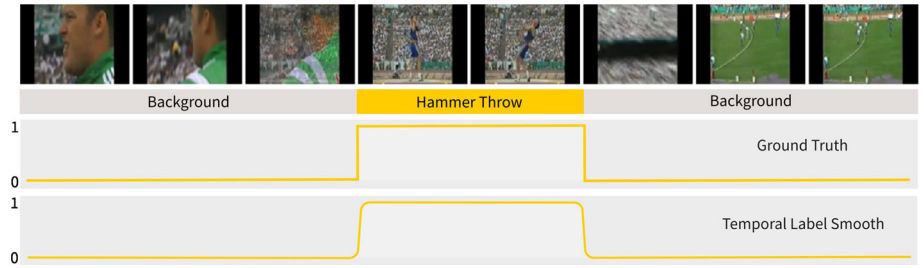

Fig. 2 Illustration of Temporal Label Smoothing (TLS). This diagram illustrates the application of a Gaussian smoothing kernel to refine ground-truth action labels over time. The Gaussian kernel transforms the ground truth from a binary (0-1) function into a gradual and continuous representation

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 the 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 (Fig. 2).

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 h 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 the 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 methods. 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

"OF" denotes using optical flow features, and "RGB" denotes using RGB features
*This result was obtained using the official code base and run by ourselves, as it was not reported in the paper

Table 2 Online detection of action start results on THUMOS'14 using feature extractor pretrained on Kinetics-400

	pAP@time threshold (s)									
	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**	51.9	**54.5**	55.4	56.0	56.5	56.9	57.0
Ours	28.7	41.6	48.1	**52.2**	54.3	**55.5**	**57.9**	**58.9**	**59.8**	**60.4**

The time threshold ranges from 1 s to 10 s. 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

Online Detection of Action Start We present a comparison of our method with recent works on the 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.

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.

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

The checkmark indicates it's enabled in the configuration while the cross-mark indicates that it's not

Table 4 Running speed in frames per second (FPS) comparison

	OF Comp	OF Feat	RGB Feat	Model	Overall	mAP (%)
TRN	11.2	97.5	754.4	188.8	11.2	62.1
LSTR	11.2	97.5	754.4	253.6	11.2	69.5
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

The asterisk (*) denotes using RGB features only. The double asterisk (**) indicates that the experiment runs on a CPU with RGB features only

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.

4.5 Efficiency Analysis

We compare the efficiency of our method with TRN, LSTR and TesTra in terms of online inference. To ensure a fair comparison, we re-run all models on our platform to control variables. The results are presented in Table 4. Our model outperforms all other models in terms of speed, and specifically, our model achieves a speed that is over two times faster than the previous state-of-the-art method, TesTra, with a slight 0.6% improvement in performance. Additionally, we observe that our method experiences a 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.

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. The RWKV architecture seamlessly integrates the advantages of the Transformer's long context length and RNN's efficient inference, creating a model with an optimal balance between performance and efficiency. This characteristic is particularly well-suited for OAD tasks, where the demand for long context length aligns with the necessity for bounded inference time. Moreover, the incorporation of Laplace activation and temporal label smoothing further enhances the model's robustness.

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.

Declarations

Conflict of interest 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 Recognit 129:108741. https://doi.org/10.1016/j.patcog.2022.108741 (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. 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:1506.00019. 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. 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. 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:2305.13048 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. 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. 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) WAOD: 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:1904.10509 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:2006.04768 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:2209.10655 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. 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. 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:2105.14103 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. 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. 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:2303.06349 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. 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. 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:2109.08668 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:2012.03457 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. 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. 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:1705.06950 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. 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:1412.6980 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