Action Probability Calibration for Efficient Naturalistic Driving Action Localization

Rongchang Li[1], Cong Wu[1,5], Linze Li[1], Zhongwei Shen[2], Tianyang Xu[1],
Xiao-jun Wu[1], Xi Li[3], Jiwen Lu[4], Josef Kittler[5]
[1]Jiangnan University. [2]Suzhou University of Science and Technology.
[3]Zhejiang University. [4]Tsinghua University. [5]University of Surrey.
{li_rongchang, congwu, linze.li}@stu.jiangnan.edu.cn, shenzw@usts.edu.cn
{tianyang.xu, wu_xiaojun}@jiangnan.edu.cn
xilizju@zju.edu.cn, lujiwen@tsinghua.edu.cn, j.kittler@surrey.ac.uk

Abstract

The task of naturalistic driving action localization carries significant safety implications, as it involves detecting and identifying possible distracting driving behaviors in untrimmed videos. Previous studies have demonstrated that action localization using a local snippet followed by probability-based post-processing, without any training cost or redundant structure, can outperform existing learning-based paradigms. However, the action probability is computed at the snippet-level, the input information near the boundaries is attenuated, and the snippet size is limited, which does not support the generation of more precise action boundaries. To tackle these challenges, we introduce an action probability calibration module that expands snippet-level action probability to the frame-level, based on a preset snippet position reliability, without incurring additional costs for probability prediction. The frame-level action probability and reliability enable the use of various snippet sizes and equal treatment for information of different temporal points. Additionally, based on the calibrated probability, we further design a category-customized filtering mechanism to eliminate the redundant action candidates. Our method ranks 2nd on the public leaderboard, and the code is available at https://github.com/RongchangLi/AICity2023_DrivingAction.

1. Introduction

Abnormal driving conditions pose serious safety hazards, involving facial movements such as yawning, or full-body movements such as texting. The naturalistic driving action localization requires the detection of distracted driving actions of the driver, which has the potential to make driving safer by alerting drivers or autonomous vehicles to potential hazards and reducing the number of accidents caused by human error. The AI City Challenge dataset [24, 25] collects driving data using three cameras from different perspectives inside the vehicle, with the objective of accurately identifying distracted actions and localizing their start and end times. This task has significant applications in enhancing traffic safety and developing smart cities.

The previous top-performing solutions [15, 28] for naturalistic driving action localization involved a multi-stage process. First, the video was segmented into smaller snippets, and then an action recognition model was used to predict the action class probability. Finally, a training-free post-processing strategy was employed to convert the snippet-level class probability sequence into the localization results. However, the method of generating localization results based on snippet-level class probabilities has the following issues: (1) The finest temporal resolution is limited by the snippet size, which may decrease the reliability of the predicted class probabilities; (2) Assigning the same class probability score to every temporal position within a snippet is not reasonable for positions near the snippet boundaries.

To address these issues, we propose an action probability calibration method. Our calibration method generates frame-level action probability scores from various snippet-level results with preset reliability, without introducing additional computations. Furthermore, we utilize prior knowledge from the multiple camera views to calibrate the action scores. With the proposed action probability calibration, overlapping snippets of different sizes can be used to parse videos without losing information at each temporal point. Based on calibrated action probabilities, we then propose the class-customized filter mechanism that gradually extends and filters local action recognition results to long-term action regions.

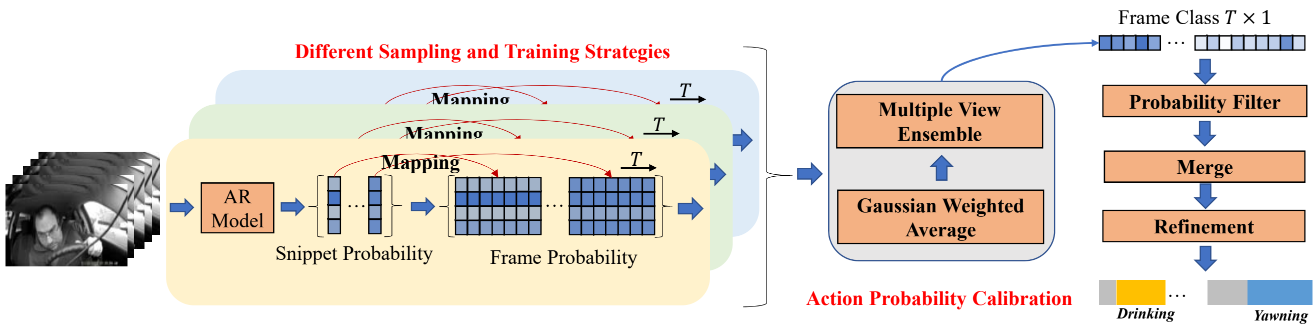

Figure 1. Overall pipeline. The input video is divided into overlapping snippets of various sizes, which are then processed by action recognition (AR) models to predict snippet-level action probabilities. These probabilities are then mapped to frame-level and calibrated using the proposed calibration method. The calibrated probabilities are then fed into the action localization component to generate action regions.

2. Related Work

2.1. Action Recognition

Action recognition is the fundamental research of video analysis, which involves the classification of short-trimmed videos using end-to-end deep learning methods. There are two typical architectures of deep learning based action recognition models: CNN-based and Transformer-based.

The CNN-based model adopts convolution blocks as the basic network units. 2D-based methods first extract spatial features and then fuse the spatial features through temporal modeling [13, 16, 30, 31]. Independent temporal modeling augments temporal features with flexible motion information. Video inherent motion information, *e.g*. spatial difference [14, 20], short-term difference [29] and learnable adaptive motion information [12, 22], have an impact on the 2D-based methods. 3D-based methods process the spatial-temporal information of videos as 3D objects directly [3, 27, 34]. SlowFast [10] and X3D [9] explore the influence on performance of different model dimensions, and propose high-performance network structures according to different task scales. Recently proposed transformer-based methods split videos into 3D tokens and utilize transformer layer to extract the spatial-temporal features [1, 2, 8, 21].

2.2. Temporal Action Localization

Video temporal action localization aims to recognize the class of an action and accurately localize its temporal boundaries. There are two main categories of methods for this task: two-stage and one-stage methods.

Two-stage methods typically generate temporal candidate proposals and then classify and refine the temporal boundaries of these proposals. Previous works have employed methods such as sliding windows aggregation [7] or boundary detection refinement [4, 18]. Recent researchers have proposed approaches that model action contexts using attention mechanisms [26, 32] and graph structures [35, 36]. DCAN [5] improved the quality of the generated action proposals by aggregating contextual semantics at the boundary level and proposal level to improve the localization performance. Despite the more complex nature of the two-stage approach, it has higher localization accuracy.

One-stage action localization does not generate proposals and intends to localize the action directly without generating a proposal in a single shot [17, 19, 33, 37, 38]. Actionformer [37] utilizes the FPN of the Transformer to perform one-stage action localization. However, previous approach [28] proved that the above method is not applicable to the AI City Challenge Track3 dataset [24, 25] due to the limited number of training samples. The previous methods adopt post-processing techniques to obtain action localization results, without requiring additional training. But in their solutions, the temporal resolution is related to the size of the snippet, and the classification results at the snippet level are directly assigned to all frames, neglecting the difference between the boundary frames and the central frames in the snippet. Our proposed method also adopts the training-free post-processing solution. But we design a novel probability calibration method to overcome the mentioned problems without introducing extra inference costs.

3. Method

3.1. Overall pipeline

As shown in Fig. 1, the main components of our proposed process are as follows:

- **Snippet-level Action Recognition**: The input video is divided into snippets of varying sizes, then recognition models are used to predict the action probability of each snippet.

- **Action Probability Calibration**: We propose the reliability score for each frame to guide aggregating

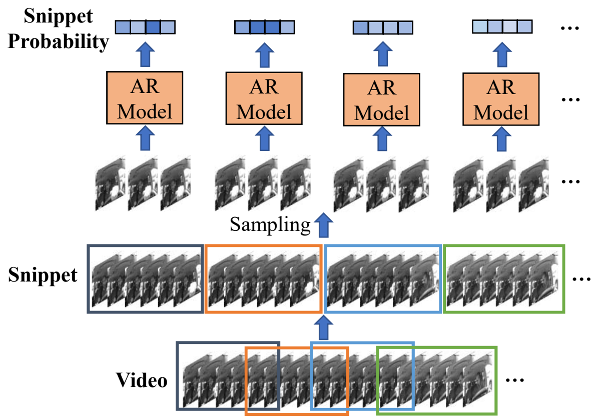

Figure 2. Snippet-level action recognition. We design different sampling strategies to partition the snippets and obtain the corresponding action probability predictions.

the action probabilities of different snippets. This allows us to calibrate snippet-level action probabilities to frame-level probabilities while maintaining inference cost. Subsequently, the frame-level probabilities are integrated and calibrated from multiple viewpoints using prior knowledge to obtain the final probability score.

- **Action Localization**: Based on the frame-level action probabilities, we gradually expand the action boundaries to obtain the final localization results. Additionally, we propose a category-customized filtering mechanism to filter the redundant candidates.

3.2. Snippet-level Action Recognition

Given a video, as shown in Fig. 2, we divide it into n overlapped snippets $\{S_1, S_2, \ldots, S_n\}$. Then F frames are sampled from each snippet with an interval of R. These sampled frames are then fed into an action recognition (AR) model to obtain the action probability scores $P \in \mathbb{R}^C$ for each snippet. We split the training videos into meaningful segments according to the annotations and use these segments to train the action recognition network. Though we can choose arbitrary action recognition models to predict action probabilities, we chose the lightweight X3D model [9] due to the characteristics and scale of the naturalistic driving action dataset.

3.3. Action Probability Calibration

The above-mentioned procedure yields action probabilities for each video snippet. [15, 28] determine the probability of each frame position as the average score of various snippets containing the frame, thus making the action recognition results owe a snippet-level temporal resolution.

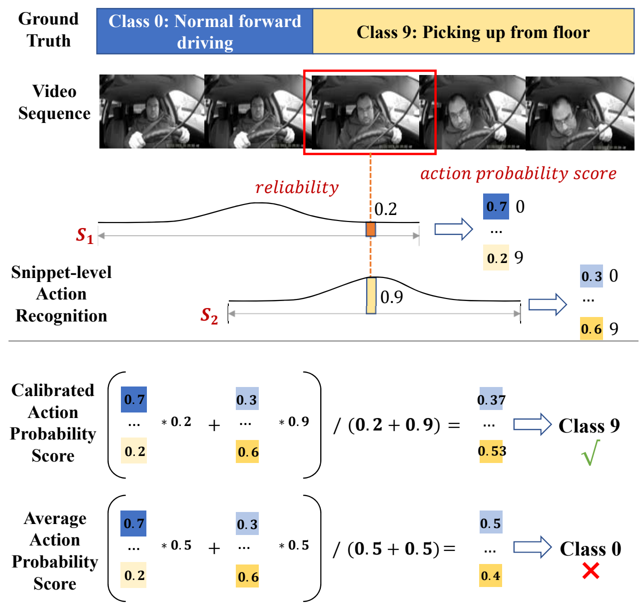

Figure 3. Reliability-based action probability calibration. S_1 and S_2 are snippets containing the target frame. The proposed calibration method relies on the reliability score to fuse the results of multiple snippets.

Accordingly, using the larger snippet size will cause coarser recognition results, which will further deteriorate the subsequent localization performance. Moreover, it seems unreasonable to assign the snippet-level action category probability to all frames within the snippet.

To address these problems, we propose a training-free probability calibration method to generate frame-level action probability scores from snippet-level results, as shown in Fig. 3. The method includes allocating the snippet-level action probability scores to each frame within the snippet and pre-defining the reliability of the probability allocation for different frames, which determines whether to trust the allocated action category score. Specifically, we assume that the reliability of the allocated probability follows a Gaussian distribution at different positions within the snippet. The distribution makes sure that the frames closer to the middle have higher reliability than the frames closer to the edges, which is in line with common sense. The reliability weight for frame f that is l_f distance away from the center of snippet s is formulated as:

$$w_{f,s} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{l_f^2}{2\sigma^2}} \tag{1}$$

where σ is the standard deviation. According to this formula, the reliability weights at the edges decrease as the size of the video snippet increases. Based on the reliability weight of probability, we define the probability score for

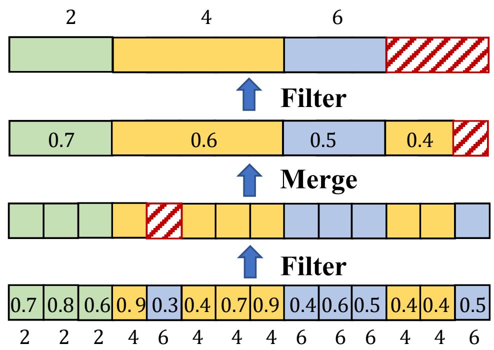

Figure 4. Temporal action localization. We propose an efficient post-processing strategy to merge local candidates and discard potentially incorrect candidates.

frame s as:

$$\hat{P}(f) = \frac{\sum_{s\in\Omega} w_{f,s}P_s}{\sum_{s\in\Omega} w_{f,s}} \tag{2}$$

where Ω is the snippet set containing frame f, P_s is the snippet-level action probability. In this way, we refine the temporal resolution to the frame-level without increasing the inference cost compared to [15]. Moreover, we can fully utilize the results of snippets of any size and any overlap ratio in a flexible manner.

Finally, we incorporate prior knowledge of viewpoints to calibrate the probability scores during the multi-view ensemble step. Intuitively, the reliability of recognition results from different viewpoints may vary for certain action categories. For instance, neither the Dashboard nor the Rear View can fully capture the entire range action of 'Adjusting control panel', while the Right Side may not be able to provide sufficient visibility of the driver's left-hand movements. Here we leverage this prior knowledge to adjust the score ratios of corresponding action categories from the three viewpoints. The final calibrated probability score of class c is formulated as:

$$P(c) = \frac{\sum_{v\in V} w_{v,c}\hat{P}_{v,c}}{\sum_{v\in V} w_{v,c}} \tag{3}$$

Here V is the set of views, $w_{v,c}$ means the preset class weight for view v and $\hat{P}_{v,c}$ means the frame-level probability score of class c from view v.

3.4. Temporal Action Localization

After calibrating the action probability, we choose the class with the highest score as the classification result, and its corresponding probability score is assigned as the confidence score. As shown in Fig. 4, for each category, we determine a threshold to filter out frames with low confidence scores. After the filtering process, we obtained some rough candidate regions of interest for the foreground, i.e. distracted driving actions. As these candidate regions represent local results, we next merge regions belonging to the same action category and with intervals not exceeding λ_1 to obtain candidate regions of longer duration. Then, we again filter the candidates that are too short (i.e. duration is smaller than λ_2). We further define the confidence of candidate regions as the average confidence score of the frames within the region. Then, for each category, we filter out candidate regions with low confidence scores to obtain the final localization results. As candidate regions can be redundant, we set the filter threshold for class c as $\mathrm{Max}(\mathrm{P}(c)) * ratio$.

4. Experiment and Results

In this section, we present the experiments results of our proposed method on the AI City Challenge 2023 Track3 Dataset. Additionally, we provide a detailed description of the dataset, evaluation metric, implementation details, and extensive ablation experiments.

Datasets. The AI City Challenge 2023 Track3 dataset consists of 210 videos performed by 35 drivers, totaling approximately 34 hours. The dataset is divided into three subsets: training set A1, validation set A2, and testing set B, which contains videos performed by 25, 5, and 5 drivers, respectively. The dataset was collected by having each of the 35 drivers complete two data collection tasks under different appearance blocks. Each task consisted of 16 different activities, such as talking on the phone, eating, and reaching back. The activities were completed in approximately eight minutes each. Videos were recorded synchronously at 30 fps from three perspectives: Dashboard, Rear View, and Right Side. Hence, A1, A2 and B contain 150, 30, 30 videos respectively.

Evaluation metric. Evaluation for AI City Challenge 2023 track 3 is based on activity identification performance, measured by the average activity overlap score, which is defined as follows.

$$os(p,g) = \frac{\max(\min(ge,pe) - \max(gs,ps), 0)}{\max(ge,pe) - \min(gs,ps)}, \tag{4}$$

where gs and ge are the start time and end time of ground-truth activity g, respectively. p is the best predicted activity match of the same category to g, os means highest overlap. With the additional condition that the start time ps and end time pe of the predicted activity must fall within a temporal range of [gs - 10s, gs + 10s] and [ge - 10s, ge + 10s], respectively. The overlap between g and p is defined as the ratio between the time intersection and the time union of the two activities. After matching each ground truth activity in order of their start times, any unmatched ground truth activities or unmatched predicted activities will receive an overlap score of 0. The final score is calculated as the average overlap score among all matched and unmatched activities.

4.1. Implementation Details.

Training action recognition models. We utilize the Kinetics-400 [3] pre-trained X3D-L [9] to predict action probabilities. To train the action recognition model, we trim the long videos in the training set (A1) to meaningful action segments according to the annotations. To better utilize the information from the background class, we also trim the unlabeled video (belonging to the background class) regions to form the enlarged training set. We call the enlarged training set A1_expand. For each input video, two options are available for frames sampling: $F = 8$ or $F = 16$ frames, with sample rate R setting as 4, 8, 12 or 2, 4, 6. So, the snippet lengths are 32, 64, or 96. For each fixed-length snippet, we then vary the overlap ratio to 0%, 25%, 50%, and 75%, respectively. The initial learning rate is 5e-4. We utilize the Adam [11] optimizer with cosine annealing [23] as the learning rate schedule. The batch size is set as 48 and the number of total train epochs is 35. The input is first resized to 512×512 and then cropped to 448×448. We adopt scale jitter, rand augment [6] and mixup [39] for data augmentation. All the models are trained on 2 NVIDIA GeForce RTX 3090 GPUs.

Action probability inference. For Action probability inference, we maintain the same video pre-processing settings. Specifically, we resize each frame of the input video to 512×512, and use a snippet of size 32, 64, and 96 (corresponding to sampled frames×sample rate). The standard deviation σ in Eq. (1) used for calibrating the action probability is set to 30, which we find can give reasonable reliability weights to frames near the snippet boundary. Finally, we ensemble the results on A1 and A1_expand by different sampling methods, different snippet overlap rates, and different views.

Temporal action localization. When filtering out frames of low confidence, the specific threshold of class c is first defined as the average confidence scores of frames recognized as class c minus a margin (0.1). Then, we compress the thresholds greater than 0.5 to 0.5, and raise the thresholds less than 0.1 to 0.1, in order to adjust the thresholds and make them more reasonable. For firstly filtering candidate regions, λ_1 is set as = 8s, and λ_2 is set as 1s. For second filtering candidate regions, the $ratio$ is set as 0.95 for filtering as many redundant regions as possible.

4.2. Main results

Tab. 1 displays the Top-10 methods on the Track3 public leaderboard, where our proposed method achieved the 2nd place with an average overlap score of 0.7041.

Rank	TeamID	Score
1	209	0.7416
2	**60(Ours)**	**0.7041**
3	49	0.6723
4	118	0.6245
5	8	0.5921
6	48	0.5907
7	83	0.5881
8	217	0.5426
9	152	0.5424
10	11	0.5409

Table 1. Comparison to other submissions methods on AI City Challenge 2023 Track3 A2 validation dataset.

Frames	Rate		Score
	$F = 8$	$F = 16$	
8	4		0.5697
8	(4,8)		0.5791
8	(4,8,12)		0.5805
(8,16)	(4,8,12)	(2,4,6)	**0.5955**

(a) Study on different snippet size.

Snippet overlap ratio	Score
0	0.5783
25%	0.5761
50%	0.5855
75%	**0.5955**

(b) Study on different overlap ratio.

Table 2. Study on different sample strategy.

4.3. Ablation studies

All ablation experiments are conducted using our self-defined training and test sets. Specifically, we adopt 25% samples of the given training set as the validation set (user id: 85870, 86356, 86952, 96269, 99882), and the remaining 75% samples for action recognition model training. The ablation experiments are based on the same training and validation sets as described in Sec. 4.1.

Study on different sample strategy. We experimented with different sampling strategies to evaluate the effectiveness of various snippet sizes in probability correction, as provided in Tab. 2 (a). The baseline achieved a score of 0.5697 when the sampling frame was $F = 8$ and the sampling rate was $R = 4$ (with a snippet size of 32). By integrating model with a sampling rate of $R = 8$ (with a snippet size of 64), the score improved to 0.5791. The performance was further improved by continuing to integrate the model with $R = 12$. Finally, we integrated all the models with snippet sizes of 32, 64, and 96 for sampling

Dashboard	Rear View	Right Side	Score
✓			0.5784
	✓		0.5665
		✓	0.5465
✓	✓	✓	**0.5955**

(a) Study on different views.

Views fusion type		Category	Score
mean	gaussian	Adjustment	
✓			0.5774
✓		✓	0.5923
	✓		0.5821
	✓	✓	**0.5955**

(b) Study on calibrations.

Table 3. Study on action probability calibration settings.

frames of 8 and 16, achieving the best result with a score of 0.5955. Meanwhile, we conducted detailed experiments on different snippet overlap ratios, as shown in Tab. 2 (b). When the snippets are not overlapped, the final localization score achieved is 0.5783. Adopting 25% overlap results in a degradation of performance. However, when the overlap rate is increased to 75%, the problem of weakened edge information is greatly alleviated by combining our sampling and fusion strategies to further refine the temporal resolution, leading to the best performance.

Study on action probability calibration settings. Tab. 3 (a) showcases our investigations of the probabilistic fusion of different views. It shows that single view probabilities are not sufficient for effective localization post-processing, hence multi-view fusion is necessary. Tab. 3 (b) presents the view fusion settings and category adjustment strategy. As mentioned in Sec. 3.3, it is not reasonable to assign the probability of the snippet level directly to all frames within the snippet, as the reliability of different frames within a snippet can vary. Similarly, the reliability of certain categories may differ across views. Experimental results support these ideas, showing that using mean fusion without adjusting view categories results in average performance. However, using Gaussian fusion and category adjustment operations significantly improves performance.

Study on different Temporal Action Localization settings. Similarly, we conducted detailed experiments on the filtering parameters for post-processing of localization, as shown in Tab. 4. We compare different thresholds for region merging and region filtering. The results indicate that $\lambda_1 = 8$ and $\lambda_2 = 1$ are the most appropriate thresholds.

$\lambda_1(s)$	$\lambda_2(s)$	Score
8	1	**0.5955**
12	1	0.5883
16	1	0.5924
20	1	0.5868
8	2	0.5947
8	4	0.5501
8	6	0.4978

Table 4. Study on different Temporal Action Localization settings.

5. Conclusion

We propose an action localization method that utilizes a calibration mechanism to improve action probability scores. Our method employs prior knowledge to assess the reliability of action results, obtaining reasonable action probability scores for each frame without increasing inference costs. Additionally, we propose the category-customized filtering mechanism to extend the frame-level classification results to action regions, achieving competitive results in Track 3 of the AI City Challenge 2023.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No.U1836218, 62020106012, 62106089, 61672265), the 111 Project of Ministry of Education of China (Grant No.B12018), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No.KYCX22_2307, KYCX22_2299), and in part by the EPSRC Grants: MVSE (EP/V002856/1), JADE2 (EP/T022205/1), and the EPSRC/dstl/MURI project EP/R018456/1.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 2

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 5

[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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1130–1139, 2018. 2

[5] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. Dcan: Improving temporal action detection via dual context aggregation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 248–257, 2022. 2

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 5

[7] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 768–784. Springer, 2016. 2

[8] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2

[9] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 2, 3, 5

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 345–362. Springer, 2020. 2

[13] Rongchang Li, Xiao-Jun Wu, and Tianyang Xu. Video is graph: Structured graph module for video action recognition. *arXiv preprint arXiv:2110.05904*, 2021. 2

[14] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. 2

[15] Junwei Liang, He Zhu, Enwei Zhang, and Jun Zhang. Stargazer: A transformer-based driver action detection system for intelligent transportation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3160–3167, 2022. 1, 3, 4

[16] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[17] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996, 2017. 2

[18] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[19] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11612–11619, 2020. 2

[20] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, pages 11669–11676, 2020. 2

[21] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 2

[22] Fuchen Long, Zhaofan Qiu, Yingwei Pan, Ting Yao, Jiebo Luo, and Tao Mei. Stand-alone inter-frame attention in video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3192–3201, 2022. 2

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[24] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023. 1, 2

[25] Mohammed Shaiqur Rahman, Jiyang Wang, Senem Velipasalar Gursoy, David Anastasiu, Shuo Wang, and Anuj Sharma. Synthetic Distracted Driving (SynDD2) dataset for analyzing distracted behaviors and various gaze zones of a driver, 2022. arXiv:2204.08096. 1, 2

[26] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13526–13535, 2021. 2

[27] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017. 2

[28] Manh Tung Tran, Minh Quan Vu, Ngoc Duong Hoang, and Khac-Hoai Nam Bui. An effective temporal localization method with multi-view 3d action recognition for untrimmed naturalistic driving videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3168–3173, 2022. 1, 2, 3

[29] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1895–1904, June 2021. 2

[30] 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 conference on computer vision*, pages 20–36. Springer, 2016. 2

[31] 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 Conference on Computer Vision*, pages 20–36, 2016. 2

[32] Lining Wang, Haosen Yang, Wenhao Wu, Hongxun Yao, and Hujie Huang. Temporal action proposal generation with transformers. *arXiv preprint arXiv:2105.12043*, 2021. 2

[33] Xiang Wang, Changxin Gao, Shiwei Zhang, and Nong Sang. Multi-level temporal pyramid network for action detection. In *Pattern Recognition and Computer Vision: Third Chinese Conference, PRCV 2020, Nanjing, China, October 16–18, 2020, Proceedings, Part II*, pages 41–54. Springer, 2020. 2

[34] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 2

[35] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 2

[36] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7094–7103, 2019. 2

[37] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 492–510. Springer, 2022. 2

[38] Da Zhang, Xiyang Dai, Xin Wang, and Yuan-Fang Wang. S3d: single shot multi-span detector via fully 3d convolutional networks. *arXiv preprint arXiv:1807.08069*, 2018. 2

[39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 5