Triplet Temporal-based Video Recognition with Multiview for Temporal Action Localization

Huy Duong Le, Minh Quan Vu, Manh Tung Tran, Nguyen Van Phuc
Viettel Cyberspace Center, Viettel Group
Hanoi, Vietnam
{duonglh9, quanvm4, tungtm6, phucnv37}@viettel.com.vn

Abstract

Temporal action localization (TAL) in untrimmed videos recently emerged as a crucial research topic, which has been applied in various applications such as surveillance, crowd monitoring, and driver distraction recognition. Most modern approaches in TAL divide this problem into two parts: i) feature extraction for action recognition; and ii) temporal boundary for action localization. In this study, we focus on improving the performance of the TAL task by exploiting the feature extraction effectively. Specifically, we present a temporal triplet algorithm in order to enhance temporal density-dependence information for the input video clips. Moreover, the multiview fusion framework is taken into account for enriching action representation. For the evaluation, we conduct the proposed method on the 2023 AI City Challenge Dataset. Accordingly, our method achieves competitive results and belongs to the top public leaderboard in Track 3 of the Challenge.

1. Introduction

Distracted driving refers to any activities which distract the driver's attention away from the road. The distraction can jeopardize the safety of not only the driver but also pedestrians and people in other vehicles. As a result, studying and analyzing the driver's behavior can eliminate driver's distraction and lower the risk of road accidents. The problem of recognizing driver action has received increasing attention in the computer vision community. However, a lack of labels and poor quality of data draw the challenge to apply this study in the actual world. The naturalistic driving study serves as an important platform for investigating driver behavior in real-time, which aims to capture the driver's actions in the traffic environment.

In the early stage, the studies on driver action recognition mainly use 2D Convolutional Neural Network (CNN) to classify driver's actions [2,21]. The study [11] uses a segmentation algorithm to segment the driver, then the output is fed into CNNs. The studies above are designed to classify the driver's action from the image and not utilize the temporal information, then fail to capture the long-term actions in the untrimmed videos. Meanwhile, the approaches that follow the problem of temporal action localization are promising to handle the untrimmed video in naturalistic driving action recognition by identifying the action boundaries and classifying their categories.

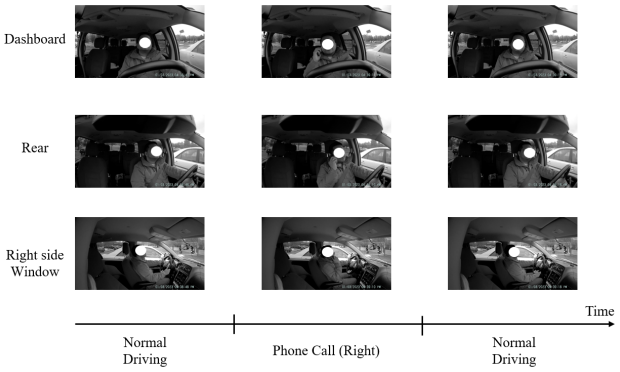

Figure 1. An example of the synthetic naturalistic data of the AI City Challenge 2023. The data is collected from three cameras inside the vehicle.

To develop and go further in the Driving action recognition field, the AI City Challenge 2023 provides the naturalistic driving action dataset and hosts a challenge of naturalistic driving action recognition. Accordingly, the synthetic naturalistic data has been collected from multiple cameras localized inside the vehicle. The objective is to identify the distracted behavior actions by the driver in a given video, an example of the dataset is shown in Figure 1. Technically, there are some critical challenges in the competition: (i) models need to recognize the different actions from multi-views; (ii) the output includes the temporal boundaries, i.e, the start time and the end time of the action and

(iii) the videos have a diversity of lengths within classes due to the untrimmed nature of the videos. To tackle the aforementioned challenges, in this study, we adopt an effective framework, including feature extraction for learning the video representation as well as action recognition and the temporal action detector for localizing the action boundary. Generally, the main contribution of this study is two-fold as follows:

- We propose a temporal triplet algorithm to provide successor state information for a single frame, then it can enhance the temporal information for a clip.
- A two-level feature fusion method is adopted for improving the performance. Specifically, we ensemble features with different backbones (i.e., X3D and MViT) and multi-views (i.e., dashboard view, rear view, and right side view) for improving the final results of the challenge. Our source code is available for further exploitation [1].

2. Related Work

2.1. Action Recognition

Action recognition is the fundamental task for video understanding that involves recognizing human actions in videos. The common approaches for action recognition include the CNN-based methods and the Transformer-based methods. In the CNN-based approach, the early work [18] uses two attentional streams, one for spatial and the other for temporal information. To capture both spatial and temporal information within a single network, some works [7, 8] use 3D CNN and achieve superior performance compared to the traditional 2D CNN. Unlike the above approach, the Transformer-based methods are inspired by the self-attention mechanism of Transformer [20], recent works [1, 6, 12, 15] have been proposed, which divides an image into several patches to extract feature and achieve competitive performance.

2.2. Temporal Action Localization

Temporal action localization is a video analysis task, where the goal is to localize and classify all actions in a video. The existing methods are technically divided into two categories, including two-stage methods and one-stage methods. Specifically, *the two-stage methods* first generate action proposals and then classify the category of these proposals and refine their temporal boundaries. Most of the recent works focus on the action proposal generation phase, including some works [4, 14, 25] put emphasize on detecting action boundary and other methods classify actions from anchor windows [3, 13]. Nonetheless, the high complexity problem of two-stage methods prevents them from being trained in an end-to-end manner. On the other hand, *the one-stage methods* for TAL, similar to the one-stage methods in object detection, localize and classify actions concurrently without action proposals. Some previous works [3, 22, 23] build the single-stage detector with the convolutional network. Meanwhile, ActionFormer [24] and TriDet [17] are the Transformer-based approaches that use Transformer [20] as the encoder, then regress the action boundaries and determine the categories without anchors.

[1] https://github.com/vtccdivedeeper/2023AICityChallenge-Track3

3. Methodology

The overall network architecture is illustrated in Figure 2. Our architecture includes four main components: (1) Temporal Triplet Algorithm for enhancing temporal information of the video clips, (2) Feature Extraction Backbone for extracting the clip's features, (3) Two-level Feature Fusion for combining the features from different views and different models, and (4) Temporal Action Detector with ActionFormer for localizing actions.

3.1. Temporal Triplet Algorithm

Each input video is split into clips, where the number of clips depends on the length of the video. The duration of clips is calculated based on the number of frames and sampling rate (the distance between two frames) [5]. The primitive split procedure above has some drawbacks: (i) a single frame of a clip contains only spatial information, (ii) the temporal information of a clip is lower density when the sampling rate increases, and (iii) not utilize the channel dimension of each frame in the clip when the video is grayscale. To overcome these drawbacks, we propose the temporal triplet algorithm, which is to retrieve temporal information to single frames, then boosts the temporal information density of the clip. To be specific, at each frame I_1^i, this technique stacks the current frame I_1^i and the next two consecutive equidistant frames I_2^i and I_3^i to obtain a new frame I_{new}^i with three channels. Subsequently, the new frame I_{new}^i is provided with temporal information about the subsequent moments. Therefore, the temporal information density of the clip is higher. Finally, the temporal triplet technique is described in Algorithm 1. Observing an example frame in Figure 3, there are temporal patterns at the positions where there is a motion. Besides the spatial information in the frame, information about the time ahead is added.

3.2. Feature Extraction Backbone

After dividing the video into clips, feature extraction backbones are designed for learning the clips' representation. In this study, we adopt two well-known backbone models for action recognition such as X3D (CNN-based

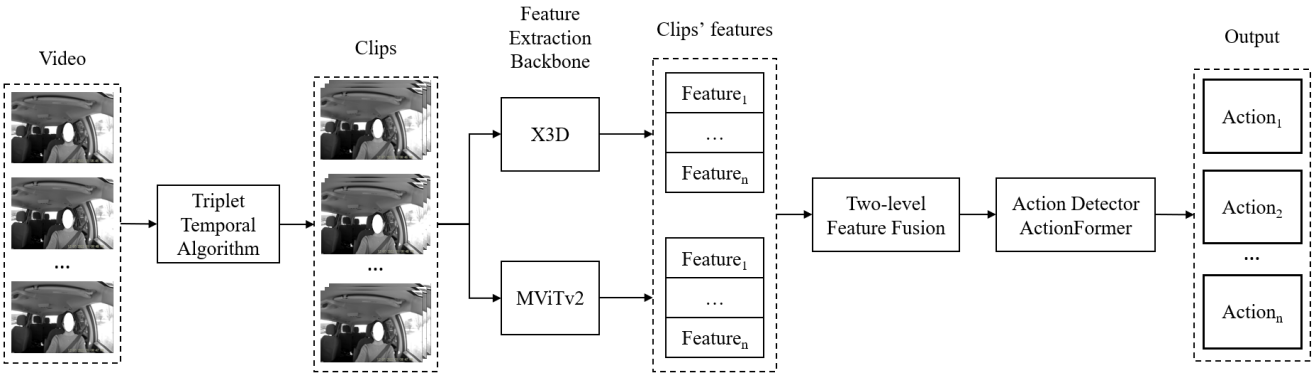

Figure 2. The overall architecture of our method. The video is fed into the temporal triplet algorithm to obtain the clips with enriched temporal information. Subsequently, the clips' features are extracted by the X3D network and MViTv2 network. Then, the clip's features are passed through Two-level feature fusion. Finally, the processed features are fed into ActionFormer to obtain the action outputs.

Algorithm 1: Temporal triplet algorithm

Input:

$X \in R^{t \times n}$: an input video.
t: the number of frames.
n: the feature dimension of a clip.
k: the number of frames per clip.
s: the distance between two consecutive frames in a clip (sampling rate).

Output:

Y: A set of clips.

$d \leftarrow \lfloor s/3 \rfloor$
$clip \leftarrow \{\}$
for $i = 1$ **to** $\lfloor t/s \rfloor + 1$ **do**
  $I_1^i = X[i \times s]$
  $I_2^i = X[i \times s + d]$
  $I_3^i = X[i \times s + 2 \times d]$
  $I_{new}^i = stack(I_1^i, I_2^i, I_3^i)$
  $clip.insert(I_{new}^i)$
  if $length(clip) == k$ **then**
    $Y.insert(clip)$
    $clip \leftarrow \{\}$
  end
end

model) [7] and MViTv2 (transformer-based) [12]. In particular, the two backbone models are sequentially described as follows:

Expand 3D (X3D): X3D network is a CNN-based approach for action recognition. Its strength is the ability to process spatiotemporal features effectively by using a 3D CNN architecture. X3D also has a relatively small number of parameters compared to other 3D CNN networks, making it more computationally efficient. Due to its great performance, we choose X3D as one of our feature extractors.

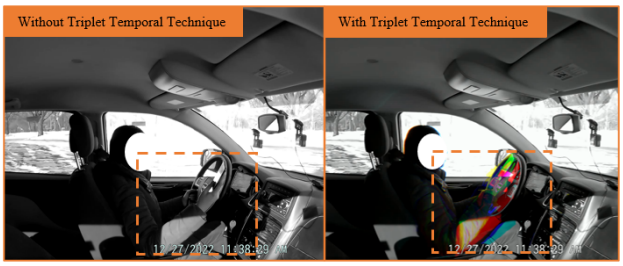

Figure 3. The comparison of not applying (left) and applying (right) temporal triplet algorithm at a frame. Observing the figure, the difference between the left image and the right image is emphasized by the dashed orange rectangle, where the temporal action is expressed.

Improved Multiscale Vision Transformers (MViTv2): Owing to the powerful ability of Transformer [20] not only in natural language processing tasks but also in computer vision tasks, we adopt a transformer-based model named Improved Multiscale Vision Transformers (MViTv2) as one of our backbones. MViTv2 is the improved version of MViT [6], which learns a hierarchy from dense (in space) and simple (in channels) to coarse and complex features. This network shows competitive performance compared to other vision transformers in video tasks.

3.3. Two-level Feature Fusion

The 2023 AI City Challenge Dataset is recorded from three camera views, including the dashboard, rear, and right-side window. Observing the dataset, we find out that some categories are difficult to identify in the dashboard view, but seem to be easy in the other views. Therefore, the fuse of three views is taken into account in order to further improve the accuracy of prediction. Accordingly, we adopt a two-level feature fusion that fuses the feature from differ-

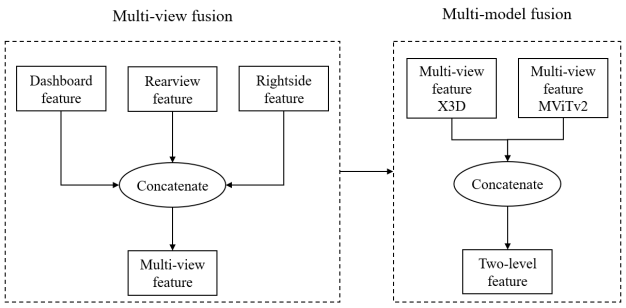

Figure 4. Two-level feature fusion.

ent perspectives. There are two steps: (1) Multi-view fusion for fusing three view features and (2) Multi-model fusion for fusing features from two networks X3D and MViTv2. Subsequently, the corresponding feature $feat \in R^D$ is obtained. The first step Multi-view fusion takes three features $feat_{v1}, feat_{v2}$ and $feat_{v3}$ from the dashboard view, rear view, and right-side window view respectively, and outputs the multi-view feature of the model $feat_{mv}$, as

$$feat_{mv} = concat(feat_{v1}, feat_{v2}, feat_{v3}) \quad (1)$$

The multi-view feature $feat_{mv}$ generally contains more information than the single view which helps the action detector can get to know the context better. Subsequently, two multi-view features of the X3D network and MViTv2 network pass through the Multi-model fusion step, which is represented as

$$feat_{tl} = concat(feat_{mv1}, feat_{mv2}) \quad (2)$$

The overall of this technique is illustrated in Figure 4.

3.4. Temporal Action Detector with ActionFormer

For the action localization, we apply ActionFormer [24] model, which identifies actions and classifies their categories within a single shot without using action proposals or relying on pre-defined anchor windows. Technically, the model consists of two parts: an encoder and a decoder. The encoder takes clip features $X \in R^{T \times D}$ as an input, where T is the temporal dimension (the number of clips) and D is the feature dimension. The encoder is a Transformer network with a temporal feature pyramid and outputs the feature pyramid of different temporal resolutions. The decoder is a lightweight convolutional network with a classification and a regression head, which classify the action category and regress the temporal boundary.

4. Experiments

4.1. Datasets

Track 3 of Naturalistic Driving Action Recognition, the AI City Challenge 2023 provides synthetic naturalistic data of the driver while driving [16]. The dataset is collected from three camera locations inside the vehicle at the dashboard, rear, and right-side window views. The whole dataset contains 210 video clips (about 34 hours in total) captured from 35 drivers, which include a total of 16 different actions (labels). Each driver performs the data collection tasks twice. In one instance, the driver completes the tasks without any obstruction to their appearance, while in the other instance, they perform the tasks while wearing an appearance block (e.g., sunglasses, hat). In this regard, there are 6 collected videos per driver, which include 3 videos in sync with no appearance block and 3 videos in sync with some appearance block. The dataset is split into three sub-datasets such as A1, A2, and B, in which the number of drivers in each sub-datasets is 25, 5, and 5, respectively.

4.2. Evaluation Metrics

The evaluation metrics are measured by the average activity overlap score, which is represented as

$$os(p, g) = \frac{\max(\min(ge, pe) - \max(gs, ps), 0)}{\max(ge, pe) - \min(gs, ps)} \quad (3)$$

where g is a ground-truth activity with start time gs and end time ge, p is a predicted activity with start time ps and end time pe. The evaluation will find the closest predicted activity for each ground-truth activity with a constraint that starts time ps and end time pe are in the range $[gs-10s, gs+10s]$ and $[ge - 10s, pe - 10s]$, respectively. After matching each ground truth activity in order of their start times, all unmatched ground truth activities and all unmatched predicted activities will receive an overlap score of 0. The final score is the average overlap score among all matched and unmatched activities.

4.3. Implementation Details

The proposed method is implemented based on the PySlowFast [5] for feature extraction, and ActionFormer [24] for Temporal Action Localization. All training and inference processes are executed on 08 NVIDIA A100 GPUs. More details of the implementation are described below.

Feature Extraction Backbone Models: We employ the X3D-L model and MViTv2-B model, pre-trained on the Kinetics dataset [9]. We use Adam optimizer [10] with the cosine annealing schedule for both two models. Specifically, the parameters for training are shown in Table 1. For data augmentation, we do not apply the horizontal flip like the default setting, due to the left-right discrimination of the dataset, e.g, the class Phone Call (left) and Phone Call (right). Moreover, we set all video segments without labels to label 0 (Normal Forward Driving) followed by the study [19]. Subsequently, when extracting clips' features, the number of frames and the sampling rate are set to 6 and 5 for both X3D-L and MVitV2-B. Therefore, with the FPS

is 30, the feature is extracted from corresponding 1-second segments.

Parameter	X3D-L	MViTv2
Number of epochs	18	200
Warmup epoch	1	35
Learning rate	0.0005	0.0005
Batch size	32	32
Training crop size	448×448	384×384
Test crop size	448×448	384×384
Number of frames	6	8
Sampling rate	5	9

Table 1. The training parameters of feature extraction backbones

Temporal Action Detector: We train ActionFormer by using Adam optimizer [10] with learning rate is set to $1e{-}4$. The model is trained on 50 epochs with the linear warm-up in the first 5 epochs, using a batch size equal to 1. The input feature dimension is specifically based on the level of applying Two-level feature fusion: equal to 3×2048 when applying the multi-view feature fusion, equal to $2 \times 3 \times 2048$ when applying the whole two-level feature fusion, and equal to 2048 for otherwise.

Data Split: We use K-Fold for cross-validation in both two parts: feature extraction and action detector. Particularly, we randomly select 20 and 5 drivers for the training and validation, respectively.

4.4. Experimental Results

Main Results: We evaluate the proposed method on set A2 dataset of the AI City Challenge 2023 Track 3. Table 2 shows the top results from the leaderboard of the challenge, which are evaluated on the challenge metric mentioned in Section 4.2. Our best submission applies both the temporal triplet algorithm and the two-level feature fusion and achieves the score of $58,81\%$ in the challenge evaluation.

Rank	Team ID	Team Name	LB Score
1	209	Meituan-IoTCV	0.7416
2	60	JNU_boat	0.7041
3	49	ctc-AI	0.6723
4	118	RW	0.6245
5	8	Purdue Digital Twin Lab	0.5921
6	48	BUPT-MCPRL	0.5907
7	**83 (Ours)**	**DiveDeeper**	**0.5881**
8	217	INTELLI_LAB	0.5426
9	152	AILAB	0.5424
10	11	AIMIZ	0.5409

Table 2. Top 10 Leaderboard of NVIDIA AI City Challenge 2023 Track 3 Naturalistic Driving Action Recognition.

Results Analysis of Temporal Triplet: we present our experiments with the temporal triplet algorithm in Table 3. We conduct the result of applying the temporal triplet algorithm in the feature extraction task and measured by the accuracy metric. The results are obtained from the validation process of the X3D model in five folds and three views. Observing the table, results are significantly improved in all folds and all views when we apply the temporal triplet algorithm. Similarly, we also use the temporal triplet for the MVitV2 model, then two models applying the temporal triplet are used for feature extraction. In addition, the results are remarkably different between folds. Notably, the accuracy scores in fold 2 and 3 are lower than in other folds. Intuitively, the difference comes from the annotations in some driver videos that are not reasonable. For instance, in the video of user id 61962, some annotations last longer than reality. Therefore, the ensemble technique for models from folds is referable to improve the performance of the results.

Fold	View	X3D	
		Without TTrA	With TTrA
0	Dashboard	82.79	87.34
	Rear	84.09	88.31
	Right-side	82.47	87.01
1	Dashboard	81.17	86.36
	Rear	83.77	87.99
	Right-side	79.55	85.71
2	Dashboard	70.00	75.33
	Rear	71.00	76.12
	Right-side	71.33	76.20
3	Dashboard	72.33	78.33
	Rear	73.67	84.67
	Right-side	69.33	77.33
4	Dashboard	84.98	89.78
	Rear	80.51	87.54
	Right-side	83.71	89.46

Table 3. The comparison of not applying (Without TTrA) and applying (With TTrA) the temporal triplet algorithm. The results are measured by the accuracy and evaluated in the feature extraction process.

Results Analysis of Two-level Feature Fusion: we conduct the ablation experiments of two-level feature fusion, which is shown in Table 4. We run the experiments in each feature extraction backbone, including two approaches: applying the multi-view feature fusion and ensembling the output of ActionFormer for multi-view by using post-processing mentioned in Section 4.3. Considering the single backbone, the result of using the multi-view feature fusion is better than the result of ensembling the output for multi-view. Finally, we combine the feature of two backbones by Two-level feature fusion and achieve the best

result.

Model	Method	LB score
X3D	NMS ensemble	0.5146
	Multi-view feature fusion	0.5704
MViTv2	NMS ensemle	-
	Multi-view feature fusion	0.5840
X3D & MViTv2	Two level feature fusion	**0.5881**

Table 4. The ablation study of different multi-view fusion methods.

5. Conclusion

In this study, we adopt the temporal action localization framework for naturalistic driving action recognition and put emphasis on the feature extraction process. Specifically, we propose a temporal triplet algorithm for enhancing the temporal density of video clips and the two-level fusion feature for enriching the video representation. The experiments conducted on the Track 3 validation set of the 2023 AI City Challenge achieve a competitive result with $58,81\%$ score on the leaderboard.

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] Bhakti Baheti, Suhas Gajre, and Sanjay Talbar. Detection of distracted driver using convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1032–1038, 2018. 1

[3] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017. 2

[4] 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

[5] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. https://github.com/facebookresearch/slowfast, 2020. 2, 4

[6] 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, 3

[7] 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

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2

[9] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 5

[11] Maitree Leekha, Mononito Goswami, Rajiv Ratn Shah, Yifang Yin, and Roger Zimmermann. Are you paying attention? detecting distracted driving in real-time. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 171–180. IEEE, 2019. 1

[12] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 2, 3

[13] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11499–11506, 2020. 2

[14] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3889–3898, 2019. 2

[15] 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

[16] 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. 4

[17] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. *arXiv preprint arXiv:2303.07347*, 2023. 2

[18] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 2

[19] 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. 4

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[21] Shiyang Yan, Yuxuan Teng, Jeremy S Smith, and Bailing Zhang. Driver behavior recognition based on deep convolutional neural networks. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 636–641. IEEE, 2016. 1

[22] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020. 2

[23] Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. Basictad: an astounding rgb-only baseline for temporal action detection. *arXiv preprint arXiv:2205.02717*, 2022. 2

[24] 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, 4

[25] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 539–555. Springer, 2020. 2