A Coarse-to-Fine Boundary Localization method for Naturalistic Driving Action Recognition

Guanchen Ding[1,*], Wenwei Han[2,*], Chenglong Wang[3,*], Mingpeng Cui[1], Lin Zhou[1], Dianbo Pan[1], Jiayi Wang[1], Junxi Zhang[1], Zhenzhong Chen[1,2,†]
[1]School of Remote Sensing and Information Engineering, Wuhan University, China
[2]School of Computer Science, Wuhan University, China
[3]School of Resource and Environmental Sciences, Wuhan University, China

Abstract

Naturalistic driving action recognition plays an important role in understanding drivers' distracted behaviors in the traffic environment. The main challenge of this task is the accurate localization of the temporal boundary for each distracted driving behavior in the video. Although many temporal action localization methods can identify action categories, it is difficult to predict accurate temporal boundaries for this task since the driving actions of the same category usually present large intra-class variation. In this paper, we introduce a Coarse-to-Fine Boundary Localization method called CFBL, which obtains fine-grained temporal boundaries progressively through three stages. Concretely, in the first coarse boundary generation stage, we adopt a modified anchor-free model Anchor-Free Saliency-based Detector (AFSD) to make an interval estimation of the temporal boundaries of distracted behaviors. In the second boundary refinement stage, we use the Dense Boundary Generation (DBG) model to adjust the estimated interval of the temporal boundaries. In the final boundary decision stage, we build a Localization Boundary Refinement Module to determine the final boundaries of different actions. Besides, we adopt a voting strategy to combine the results of different camera views to enhance the model's distracted driving action classification ability. The experiments conducted on the Track 3 validation set of the 2022 AI City Challenge demonstrate competitive performance of the proposed method.

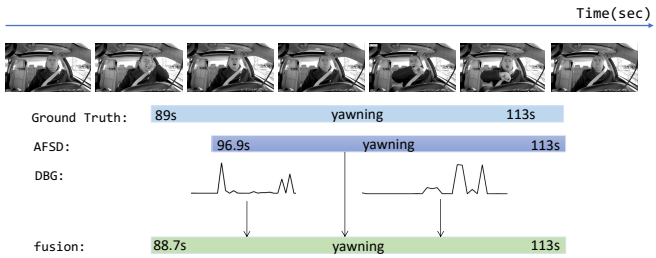

Figure 1. An example of the distracted behavior in the data set. We adapt the label predicted by AFSD and the boundary adjusted by DBG output signals.

1. Introduction

Distracted driving, such as phone call, eating and reaching behind is the main cause of fatal road traffic injuries. Therefore, Naturalistic driving action recognition, which aims at identifying the distracted behaviors of the driver in the traffic environment, has attracted a lot of attention in recent years.

Naturalistic driving action recognition can be regarded as a fine-grained temporal action localization task, which aims to classify action instances in each video and localize the accurate temporal boundaries of them. Currently, Temporal Action Localization (TAL) researches can be roughly divided into anchor-based approaches and anchor-free approaches. The anchor-based approaches usually adopt a two-stage strategy, which first generates candidate video segments as action proposals and then classifies and refines temporal boundaries. However, the anchor-based two-stage approaches may produce a bunch of redundant proposals, which severely influences the efficiency of computation. Recently, popular anchor-free methods assemble both boundary regression and classification in an end-to-end model so as to significantly improve the inference efficiency.

Compared to the general temporal action localization

*Co-first Authors.

†Corresponding author: Zhenzhong Chen (email: zzchen@whu.edu.cn). This work was supported in part by the National Natural Science Foundation of China under Grants 62036005.

task, there are two major challenges in naturalistic driving action recognition of AI City 2022. First, the temporal boundary localization is hard for the driving video, since driving actions of the same category present large intra-class variation. For example, within one video segment which is labeled as *eating*, there are both eating and eating gap. This intra-class variation may puzzle the model to divide one action segment into different parts especially for the anchor-based approaches which produce a bunch of redundant proposals. Second, driver action records from multiple camera views in the vehicle are provided in this task. However, current TAL approaches are mainly based on single camera. It is also challenging to effectively combine the information of multiple camera views to improve the accuracy of action classification and boundary localization.

To address above challenges, we propose a Coarse-to-Fine Boundary Localization method called CFBL to predict the action segment boundary accurately. Specifically, we first give a coarse estimation of the action segments which cover the approximate time range of each action and then refines the start time and end time of each segment to get more accurate temporal boundaries. For the coarse boundary generation stage, we adopt a modified anchor-free model called AFSD [8], which consists of an I3D [2] feature extraction module, a boundary regression module and an action classification module. Note that the modified AFSD model does not make a point estimation of the starting time and ending time of an action in this stage, it makes an interval estimation of the boundary instead. For the fine-grained boundary refine stage, we use a Dense Boundary Generation model called DBG [7] to adjust the time range of the temporal boundary. For the last boundary decision stage, we build a Localization Boundary Refinement module to determine the exact boundaries of different action instances. Through this three-stage coarse-to-fine boundary location strategy, our model could find more accurate start time and end time of each driving action. Moreover, to effectively combine the information of multiple view cameras, we introduce a Multi-View Filter Module to ensemble models with different camera videos as input. To be specific, for one action instance, we use three different camera views to predict the temporal boundary and adopt a voting strategy to determine the final boundary.

2. Related work

2.1. Anchor-based Temporal Action Localization

Current TAL models of anchor-based methods mainly get results through learning the adjustment of pre-defined anchors. Existing anchor-based methods can be divided into two categories: one-stage approaches and two-stage approaches. SSAD [10] is a classic one-stage TAL network, which is a temporal convolutional network on multi-granularity feature sequences. SSAD network directly predicts boundaries and confidence scores for multiple action categories skipping the proposal generation step. GTAN [13] novelly applies a temporal structure into a one-stage action localization framework and exploits Gaussian kernels to optimize temporal scale of each action proposal dynamically. Meanwhile, with the help of the video visual features and position embedding information, MGG [12] performs the temporal action proposal from different granularities perspectives.

Compared to one-stage TAL approaches, a two-stage approach for TAL first generates candidate video segments as proposals, and further classifies these proposals in order to get the action categories and the corresponding and refined temporal boundaries. R-C3D [18] improves the Faster R-CNN [15] pipeline and get the temporal localization based on 1-D sequence. In an end-to-end learning manner, it uses a 3-D fully convolutional network to encode the video streams. After encoding the video streams, R-C3D generates candidate temporal regions containing actions and finally classifies candidate regions into definite actions. TURN [4] predicts action proposals and generates the temporal boundaries by temporal coordinate regression. CBR [5] also uses temporal coordinate regression to generates the temporal boundaries of the sliding windows. Subsequently, TAL-Net [3] uses a multi-scale architecture to improve receptive field alignment and better exploits the temporal context of actions for both proposal generation and action classification.

2.2. Anchor-free Temporal Action Localization

Although anchor-based TAL methods have achieved remarkable results on benchmark data sets, such methods are still limited to some points. For example, anchor-based methods have to produce a bunch of redundant proposals, which severely influences the efficiency of computation. Furthermore, anchor-based methods are sensitive to some hyper-parameters, such as the size of pre-defined anchors. Instead, an optional approach for TAL is to resort to the anchor-free method, which assembles both boundary regression and classification in one model, thus being more efficient while having less parameters. A2Net [19] combines the anchor-free module with a conventional anchor-based module. To be more specific, in the anchor-free module, an action instance can be represented as a point and its distances to the starting boundary and ending boundary. In this way, the pre-defined anchor restriction is alleviated in terms of action localization and duration. A novel purely anchor-free TAL framework called AFSD is proposed in [8]. Considering the impact of boundary features, AFSD adopts a novel boundary pooling method to generate fine-grained predictions. Recently, self-attention based Transformer models have achieved promising results

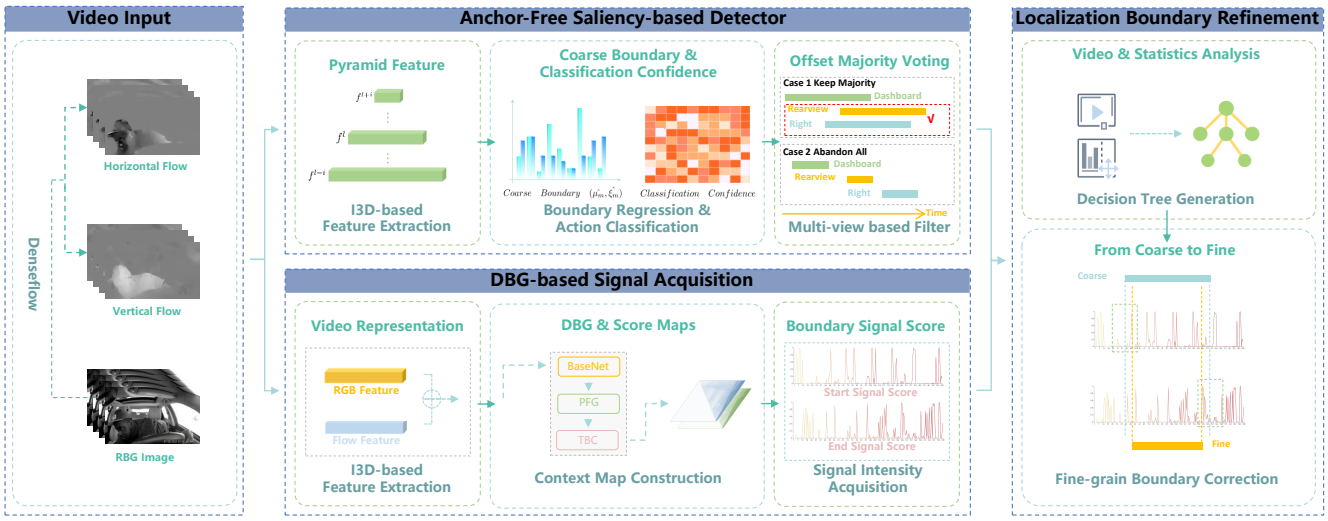

Figure 2. The pipeline of our proposed Coarse-to-Fine Boundary Localization model(CFBL). Given a video, our model first utilizes DenseFlow [17] to extract optical flows, whose results, together with original RGB frames, are regarded as the inputs of the network. Then, an Anchor-Free Saliency-based Detector (AFSD) [8] is applied to obtain the classification result and coarse boundary prediction. What's more, a DBG-based [7] Signal Acquisition Module is designed to model starting and ending signals. Finally, combining the two results and using the Localization Boundary Refinement Module as an auxiliary, the fine boundary is obtained.

in image classification and object detection. Inspired by these success, some algorithms based on Transformer models have emerged in the field of video understanding. ActionFormer [20] extracts multi-scale feature representations through local self-attention modules and uses a lighter decoder to efficiently classify the moment and estimate the corresponding action boundaries.

2.3. Actionness-guided Temporal Action Localization

Unlike above two categories of TAL algorithms, actionness-guided localization methods mainly focus on evaluating frame-level actionness which indicates the score of a potential action. CDC [16] places CDC filters on top of 3D ConvNets. The CDC filter predicts actions at the frame-level granularity by performing temporal upsampling and spatial down-sampling operations simultaneously. BSN [11] uses a temporal evaluation module to evaluate actionness score, starting probability and ending probability. From local to global, BSN locates temporal boundaries and evaluates the confidence score of whether a proposal contains an action. BSN is further extended by BMN [9]. In BMN, the boundaries matching confidence map are densely predicted and the confidence map is used to select action proposals. A mechanism called Boundary-Matching(BM) mechanism is introduced in BMN which is used to evaluate confidence scores of densely distributed proposals. Based on BM mechanism, BMN can generate proposals with precise temporal boundaries and more reliable confidence scores simultaneously. Inspired by boundary-sensitive methods, DBG [7] implements boundary classification and action completeness regression for densely distributed proposals. Specifically, DBG consists of two modules, named Temporal boundary classification (TBC) and Action-aware completeness regression (ACR), which aim to provide two temporary boundary confidence maps and generate an action completeness score map. To sum up, these actionness-guided methods adopt a bottom-up pipeline and usually localize action instances via multiple separate procedures.

3. Methodology

Denote our video data set as $D = \{D^{Train}, D^{Test}\}$. For any video instance in D^{Train}, it can be depicted as $V = \{X, \Phi_X\}$, where $X = \{x_t\}_{t=1}^{T}$ represents that this video contains T RGB frames or optical flows, and $\Phi_X = \{\phi_m\}_{m=1}^{M_X}$ is the corresponding annotations. To be specific, M_X is the number of action instances in this video and $\phi_m = (\mu_m, \xi_m, a_m)$ means the starting time, ending time and action category of the m-th action instance. In this task, $a_m \in \{\eta_1, ..., \eta_{17}\}$ is one of the 17 behaviors (such as phone call, drinking, and reaching behind) that could potentially distract people from driving.Our goal is to train a robust model, which can give accurate boundary prediction and action classification for any video instance in D^{Test}.

Overview. As shown in Fig. 2, we propose a Coarse-to-Fine Boundary Localization model dubbed CFBL, which can adjust boundaries from coarse to fine. Concretely, given a video X, we first utilize an Anchor-Free Saliency-based Detector (AFSD) [8] to obtain classification confi-

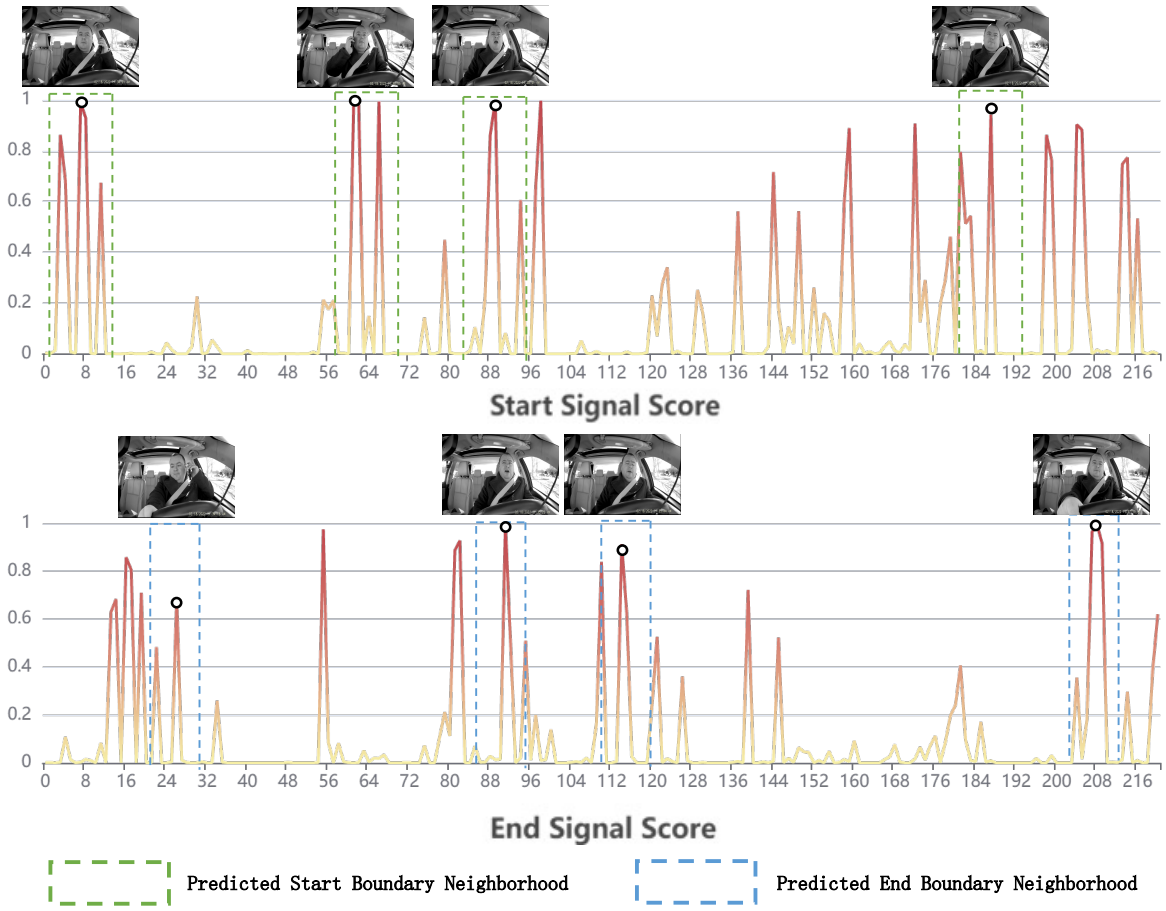

Figure 3. The process of searching for signals to refine our boundary prediction. It shows that the real boundary lies in the neighbor of the previous predicted proposal and achieves a high score.

dence and coarse boundary prediction(Sec. 3.1). These preliminary results are afterward made a close combination with DBG-based starting and ending signals(Sec. 3.2). Finally, by virtue of a Localization Boundary Refinement Module(Sec. 3.3), fine boundary prediction is obtained. In the following sections, we introduce each module in the sequence of processing.

3.1. Anchor-Free Saliency-based Detector

To obtain coarse boundaries for further processing, we adopt a modified anchor-free temporal localization method called Anchor-Free Saliency-based Detector(AFSD) [8]. Compared with anchor-based temporal action localization models and actionness-guided temporal action localization models, this modified model could not only avoid being bothered with a large number of outputs and heavy tuning of localizations and sizes corresponding to different anchors, which is of great significance in naturalistic driving action recognition, but also achieve competitive results. As shown in Fig. 2, it contains three modules, including I3D-based Feature Extraction, Boundary Regression & Action Classification, and Multi-view based Filter. We are going to give a brief introduction to each module below.

I3D-based Feature Extraction. In the same way as [8] does, we first make use of a Kinetics pre-trained I3D model [2] to extract both spatial and temporal features. Secondly, by virtue of global convolution and flattening operation, we obtain a 1-D feature sequence. Finally, a feature pyramid network is utilized to further merge spatial and temporal information, which contains several temporal convolutions and can model actions on different time scales. However, different from the original network, in order to meet the needs of high-resolution, we modify it appropriately by changing the kernel size and fusion process. On the whole, assuming that the input video is X, I3D will first offer us a 4-D feature $F \in R^{T\times C\times H\times W}$, where T, C, H, W denotes the time step, channel, height, and width. After the last two steps, six pyramid features $f_l \in R^{T_l\times C}$, where $l \in \{1, 2, ..., 6\}$,will be acquired, based on which coarse boundary regression and action classification will be done

subsequently.

Boundary Regression & Action Classification. Considering the limited receptive field of temporal convolutions, we utilize both the basic prediction module and the saliency-based refinement module of Anchor-Free Saliency-based Detector (AFSD) [8] to gain coarse boundaries. For instance, for the $l\text{-}th$ pyramid feature, we first process it with shallow convolutions to get coarse starting and ending boundary distances $(\hat{d}_i^s, \hat{d}_i^e)^{coarse}$ and classification confidence $\hat{y_i}^{coarse}$ for each location i. Secondly, we decode it to gain the predicted temporal region $(\hat{\mu}_i, \hat{\xi}_i)$. Thirdly, by virtue of Boundary Pooling proposed by [8], we carefully construct small neighbors, select the largest activated cell,$i.e.$,the most salient moment,and concatenate it with the original feature. Finally, taking advantage of concatenated feature, we get offsets' prediction after a new convolution, combined with which our refined preliminary results are obtained.

Multi-view based Filter. The particularity and complexity of naturalistic driving action recognition require that our model must act as an effective reminder, in which case it cannot frequently output invalid results, nor can it miss any distractions. In order to solve the above problems, we introduce a novel Multi-view based Filter Module. Specifically, the Multi-view based Filter Module determines the final output by combining the voting results from the three views. For each view, the voting result $\mathcal{V}_m^i$ is calculated as follows:

$$\mathcal{V}_m^i = \sum_{\substack{j\in\{Da,Re,Ri\}\\ i\neq j}} \mathbb{1}((|\hat{\mu}_m^i - \hat{\mu}_m^j| \leq \delta) \wedge (|\hat{\xi}_m^i - \hat{\xi}_m^j| \leq \delta)) \tag{1}$$

where the superscript i and j represent different views, $\hat{\mu}_m$ and $\hat{\xi}_m$ represent the predicted starting time and ending time of the $m\text{-}th$ action instance, δ is a small threshold and $\mathbb{1}$ is an indicator function.

The number of votes received for each action instance is calculated as follows:

$$\hat{\mathcal{V}}_m = \sum_{i\in\{Da,Re,Ri\}} \mathcal{V}_m^i \tag{2}$$

Since there are three views for each action instance, the module will select the instance if the votes are greater than or equal to 2/3 of the maximum number of votes empirically.

3.2. DBG-based Signal Acquisition

In practice, drivers' distracted behaviors may cause serious traffic accidents and massive property damage. So it is a pressing and vital issue to introduce a module that can help effectively limit both starting and ending boundary errors into a smaller range. Due to the parallel optimization of the action classification task and boundary regression task, the boundary proposals we got from the previous model are trade-off products that can still be optimized. As for the classification task, a slight error will not cause a significant impact on the inference result. Its results have already satisfied our needs and will be selected to help determine the final category to which each video fragment belongs. Now, the target of optimization becomes to reduce the gap of the temporal boundaries between predictions and ground truths. To better adjust the time range of starting and ending points, we adapt the dense boundary generator model dubbed DBG [7] which focuses on the video context information and dense boundary generation.

Context Map Construction. Taking the RGB and flow features extracted by I3D as input, DBG re-conduct frame information into matrix form after mapping and integration. In order to make better use of the time series information, DBG proposed a module called PFG to concatenate frame information, including frames near starting points, frames near ending points, and frames near the middle points. Then the network can learn knowledge with the help of sharing context and generate three score maps standing for the confidence of start, end, and tIoU, respectively.

Signal Intensity Acquisition. According to the analysis of the training data sets, we find that not each behavior maintains the state of being distracted. Some cyclical behaviors, such as yawning, can be divided into several small parts, separated by normal driving behaviors, which cause a profound distortion to boundary loss. Besides, the duration of a distracted behavior may last for a long time, resulting in useless information sampling, which will also lead to a bad result. So we try to re-consider the starting and ending signals that can still keep satisfactory performance under the circumstances mentioned above. The start confidence map and end confidence map can capture most of the useful information and time series information. It is easier and more efficient to utilize these significant signals that contain enough information to help determine the final result.

3.3. Localization Boundary Refinement

With the help of DBG-based Signal Acquisition module, we can easily acquire the starting and ending signals, which are helpful for further boundary refinement. Specifically, as shown in Fig. 3, after obtaining the coarse boundary prediction, we search the pre-defined neighbors in order to get the strongest signal that indicates the starting or ending point. The strongest signals can be calculated as follows:

$$\hat{\mu}_m^{fine} = \underset{i\in\mathcal{B}(\hat{\mu}_m^{coarse})}{\operatorname{argmax}} S(i) \tag{3}$$

$$\hat{\xi}_m^{fine} = \underset{i\in\mathcal{B}(\hat{\xi}_m^{coarse})}{\operatorname{argmax}} S(i) \tag{4}$$

where $\mathcal{B}$ denotes the pre-defined neighbor and S is the meaningful criteria to select the signal.

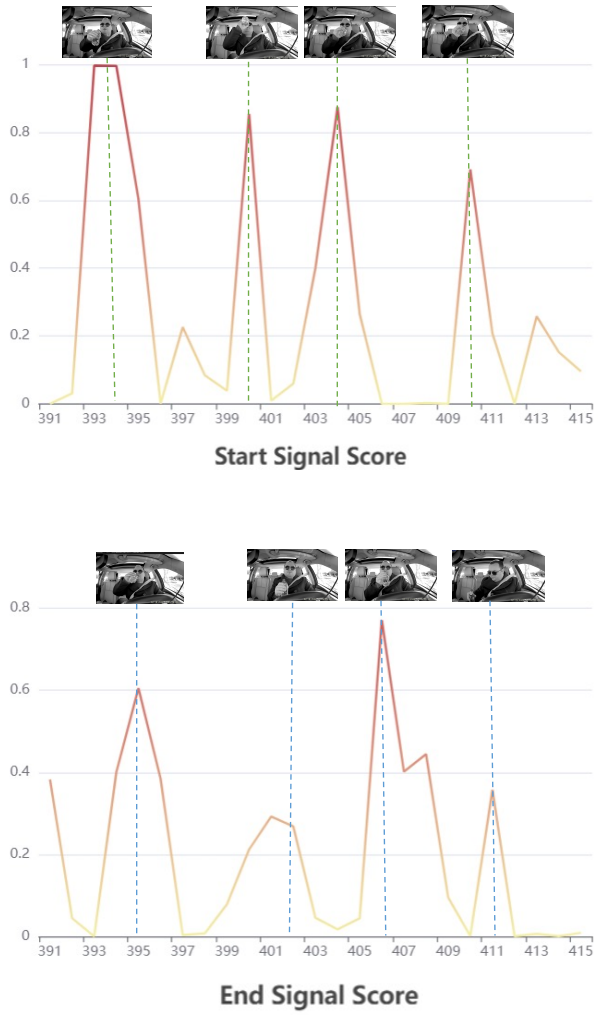

Figure 4. Problems of refining boundaries only depending on the strongest signal. The starting and ending salient action features arise many times, which disturb our model's performance on localization task.

However, as shown in Fig. 4, some points that do not represent starting or ending points are highly similar to actual starting or ending points in cyclical behaviors' signals. Due to this complexity, additional judgment conditions, except for the strongest signal, are introduced as appropriate. Specifically, for cyclical behaviors, when we search for signals in the neighbor, we not only consider the intensity of signals, but also make judgments in combination with the number of times the signal appears. In this way, a hierarchical decision tree supporting our judgments is constructed and our boundary prediction is finally refined.

Table 1. Our results on Track 3 validation set

F1-Score	Precision	Recall
0.2902	0.4868	0.2067

4. Experiments

4.1. Track 3 data set

The Track 3 data set [14] in 2022 AI City Challenge has 90 videos lasting about 14 hours in total, captured from 15 drivers through three different camera views. The difference between videos performed by the same driver is whether they wear appearance blocks or not. In each video, the driver does the 17 different distracted behaviors once without order. Each video has an approximate length of 10 minutes, a frame rate of 30 fps and a resolution of 1920 × 1080. This data set is equally partitioned into A1, A2, and B1 parts, each containing five drivers. A1 and A2 parts are provided for participants to train and evaluate, while B1 part is reserved for later testing, based on which the final rank will be determined. The main target of the challenge is to identify and localize distracted behaviors in test videos, which requires us to return the action category, starting time, and ending time of the distracted behavior.

4.2. Implementation details

For the modified AFSD module, we slide the window on raw videos with $stride$ = 30 and $length$ = 256 after downsampling its size to 224×224. We use optical flows extracted by DenseFlow [17] with its settings the same as AFSD [8] does. When training, we use Adam [6] for optimization. The batch size is set to 1. The learning rate is set to 10^{-5} for 50 epochs. During testing, on the basis of using Soft-NMS [1] to pick out the top-5000 proposals, we reserve the action instances with the highest score in each category.

For the DBG-based Signal Acquisition Module, to provide high quality input to it, we utilize a two-stream I3D model pre-trained on Kinetics [2] to extract video features. To be specific, we first feed 16 consecutive frames as the input to I3D, using a sliding window with stride 8 and extract a 1024-D feature before the last fully connected layer. Then a concatenation operation is further executed to get a 2048-D feature, which acts as the role of DBG's input.

For the Track 3 data set, after obtaining the feature extracted by the two-stream I3D model mentioned before, we slide the window on video features with $stride$ = 30 and $length$ = 160. When training, we use Adam [6] for optimization. The batch size is set to 32. We train 100 epochs in total. The learning rate is set to 10^{-3} for the first 40 epochs, and we decay it to 10^{-4} for the rest epochs.

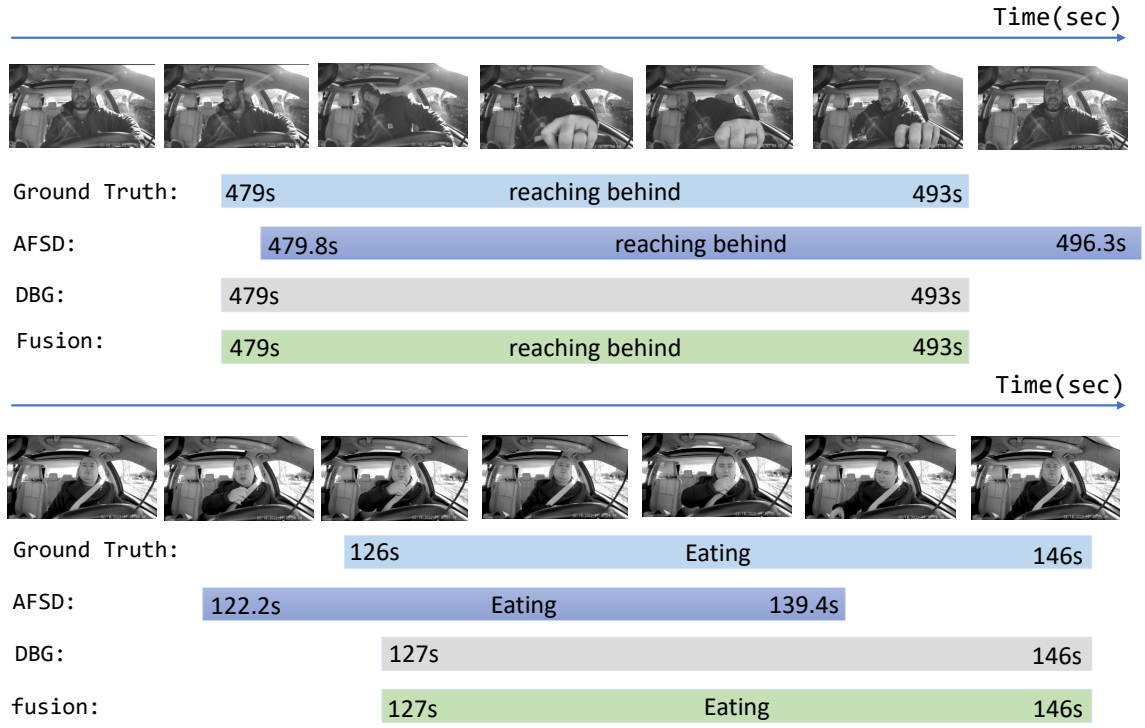

Figure 5. The visualization results of naturalistic driving action recognition. With DBG-based Signal Acquisition module, predicted boundaries become closer to the ground truth.

4.3. Evaluation metrics and experimental results

For Track 3, the evaluation index for algorithm performance is $F1 - Score$, representing the identification accuracy. Specifically, the Track 3 score will be computed as:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{5}$$

where $F1$ determines the harmonic mean of precision and recall. A true-positive(TP) action identification will be considered when the action was correctly identified as starting time within one second and ending time within one second of the action. It is noticed that each action will only be analyzed once.

As shown in Table 1, we evaluate our methodology on the Track 3 validation data and obtain F1-Score at 0.2902, with a precision of 0.4868 and recall of 0.2067. From the examples presented in Fig. 5, we can see that our model can accurately localize the action boundary due to the introduced of DBG-based Signal Acquisition module.

5. Conclusion

In this paper, we introduce a Coarse-to-Fine Boundary Localization (CFBL) method for naturalistic driving action recognition. Our method obtains fine-grained temporal boundaries progressively. Specially, it first provides a coarse estimation of the approximate interval of each distracted action and makes the classification result. Then it refines the temporal boundary of each segment to get a more accurate interval that contains distracted actions. Furthermore, we adopt a voting strategy to combine the results of different camera views to enhance the model's classification ability. The experiments conducted on the Track 3 validation set of the 2022 AI City Challenge demonstrate competitive performance of the proposed method.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-NMS - Improving Object Detection with One Line of Code. In *IEEE International Conference on Computer Vision, 2017*, 2017. 6

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 4, 6

[3] 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 Com-*

puter Vision and Pattern Recognition, pages 1130–1139, 2018. 2

[4] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3628–3636, 2017. 2

[5] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*, 2017. 2

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, 2015*, 2015. 6

[7] 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, 3, 5

[8] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2021. 2, 3, 4, 5, 6

[9] 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. 3

[10] 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

[11] 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*, pages 3–19, 2018. 3

[12] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3604–3613, 2019. 2

[13] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019. 2

[14] Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Jiyang Wang, Senem Velipasalar Gursoy, David Anastasiu, and Shuo Wang. Synthetic distracted driving (syndd1) dataset for analyzing distracted behaviors and various gaze zones of a driver. *arXiv preprint arXiv:2204.08096*, 2022. 6

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 2

[16] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017. 3

[17] Shiguang Wang, Zhizhong Li, Yue Zhao, Yuanjun Xiong, Limin Wang, and Dahua Lin. denseflow. https://github.com/open-mmlab/denseflow, 2020. 3, 6

[18] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5783–5792, 2017. 2

[19] 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

[20] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *arXiv preprint arXiv:2202.07925*, 2022. 3