PAND: Precise Action Recognition on Naturalistic Driving

Hangyue Zhao[1*], Yuchao Xiao[1*], Yanyun Zhao[1,2†]
[1]Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, China
{zhaohy21315, ycxiao, zyy}@bupt.edu.cn

Abstract

Temporal action localization for untrimmed videos is a difficult problem in computer vision. It is challenge to infer the start and end of activity instances on small-scale datasets covering multi-view information accurately. In this paper, we propose an effective activity temporal localization and classification method to localize the temporal boundaries and predict the class label of activities for naturalistic driving. Our approach includes (i) a distraction behavior recognition and localization method in naturalistic driving videos on small-scale data sets, (ii) a strategy that uses multi-branch network to make full use of information from different channels, (iii)a post-processing method for selecting and correcting temporal range to ensure that our system finds accurate boundaries. In addition, the frame-level object detection information is also utilized. Extensive experiments prove the effectiveness of our method and we rank the 6th on the Test-A2 of the 6th AI City Challenge track 3.

1. Introduction

The AI City Challenge Workshop at CVPR 2022 will specifically focus on problems in two domains where there is tremendous unlocked potential at the intersection of computer vision and artificial intelligence - The Intelligent Traffic Systems (ITS), and the brick and mortar retail business [4]. We mainly discuss Naturalistic Driving Action Recognition issue on track 3.

Distracted driving is extremely dangerous, and it is estimated that 8 people are killed every day in the United States as a result of it [4]. Naturalistic driving research and computer vision methods are now providing a much-needed answer for identifying and removing distracted driving behavior on the road. However, a lack of labels, as well as low data quality and resolution, have posed challenges in extracting insights from data about the driver in the actual world. Naturalistic driving studies serve as an important platform for investigating real-time driver behavior. They detect all of the driver's actions in the traffic environment, including those concerning tiredness or distracted driving. In track 3, which provides synthetic naturalistic data of the driver collected from multiple camera localizations inside the vehicle. The aim is to detect the temporal segments of the distracted behavior executed by the driver, as shown in Figure 1. This task is a Temporal Action Localization (TAL) and activity classification problem.

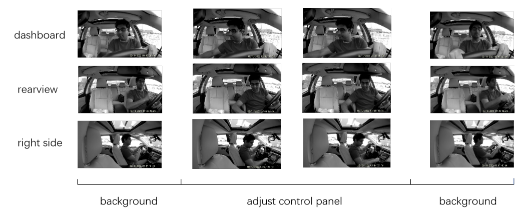


Figure 1. An action instance of adjusting control panel. Note that dataset provided synthetic naturalistic data of the driver collected from multiple camera localizations inside the vehicle.

The dataset contains 90 videos (about 14 hours in total) captured from 15 drivers performing each of 18 different tasks, such as making a phone call, eating, and returning, in random order. When working with datasets, the following challenges were encountered: (1) The scale of the dataset was too small, while there are too many behavior categories, resulting in insufficient diversity of behavior samples; (2) Some behaviors have a large pause during the occurrence, causing slight interference in recognition; (3) Water bottles, mobile phones, food and other objects related to behavior recognition are not labeled.

According to the characteristics of this dataset and evaluation requirements (such as the prohibition of using external data to train models, *etc.*), we propose a naturalistic driving behavior recognition method based on multi-view natural driving videos. In order to solve the above problems, we adopted the following strategies:(1) To expand

*Equal contribution
†Corresponding authors

the number of training samples, the video is cropped into overlapping segments. Making predictions by the action recognition network at the video clip level can improve the accuracy of the model. (2) the accuracy of action recognition networks is not high enough, and misclassification often occur, which results in some proposals being inaccurate or filtered out. We use a temporal localization network to generate a series of candidate segments to complement the temporal proposals caused by missing detections. (3) frame-level object detections of detection network, such as bottles, mobile phones, *etc*., can also provide some useful suggestions for activity prediction. We use the pre-training model on the COCO dataset [18] to detect objects without further fine tuning. By fusing the multi-branch proposals, through a series of post-processing methods, the required accuracy can be achieved. Finally, our result achieves an F1 score of 29.05% on the test set, which is less than 5 percentage points away from the best result in track 3 of this year [29].

In summary, our paper has the following contributions:

1. Based on the naturalistic driving action recognition task of AI City Challenge, we propose a distraction behavior recognition and localization method in naturalistic driving videos, which well solves the problem of accurate temporal localization on small-scale data sets. In this evaluation, we win the 6th place on the on the Test-A2 of track 3.

2. We adopted a two-stage training recognition network, which is a strategy to identify different behaviors from different perspectives, based only on small-scale datasets.

3. We apply a temporal localization network to obtain more candidate segments, which solves the problem of missing proposals due to misclassification.

4. In order to accurately correct the localization boundaries, we try to integrate the information of object and body. This information is fused through a series of post-processing strategies, so as to further improve the performance of our method.

2. Related Work

Action recognition networks. Two-stream ConvNet [9, 27] is a classical architecture which combines two modalities of optical flow information and RGB information. SlowFast [8] designs two pathways to operate at different framerates. The slow-fast-pathway enables the model to capture semantic information and rapidly changing motion with high time efficiency and accuracy. TSM [15], which only utilizes 2D convolution and moves the feature channels along the temporal dimension, achieves strong temporal modeling ability. These methods are based on convolutional neural network, which easily lead to over fitting when there are too few training samples. Another series of methods are based on the skeleton modality, which mainly use graph to model the human body and classify actions by graph convolution network. ST-GCN [32] is a successful application of spatial-temporal graph convolution network for skeleton-based action recognition. But all the GCN methods [5,30,32] have problems of robustness to input noise and data generalization. To solve these problems, PoseC3D [7] is proposed which achieves good performance by using a small 3D convolution network with the pseudo image generated according to the skeleton information as input. However, PoseC3D is also affected by the noise in skeleton information. When the human body in the video is occluded, the skeleton detection result will contain a lot of noise, which is not conducive to action recognition. Different from the above methods, Transformer [1, 6, 34], a new paradigm based on self-attention mechanism, has attracted more and more attention. Inspired by Transformer, Ze Liu *et al*. proposed Video Swin Transformer [20], which extends the attention of spatial domain to spatial-temporal domain on the basis of Swin Transformer [21] and achieves the state-of-the-art performance. In practice, Video Swin Transformer can achieve better results with a small amount of training data than other methods.

Temporal action localization. As one of the most important contents of video understanding, the current works of Temporal Action Localization (TAL) mainly focus on the extraction of temporal proposals. R-C3D [31] draws lessons from the idea of Fast-RCNN [11] in object detection, and uses a whole set of pipelines of proposal generation, proposal-wise pooling and final prediction. Similarly, TURN [10] also adopts anchor mechanism, aggregates clip-level features from short video units, and carries out temporal coordinate regression at the unit level. BSN [17] is carried out from bottom to top. It first locates the boundary of the action, then combines the boundary nodes into proposals, and finally evaluates its confidence based on the characteristics of proposal level. BMN [16] and DBG [13] are two improved versions of BSN. They mainly solve the shortcomings of BSN, such as low efficiency, insufficient semantic information and multi-stage architecture, but they are not completely end-to-end. AFSD [14] is a new end-to-end anchor free model. It draws lessons from anchor-free methods in object detection, takes I3D as the backbone, and introduces boundary pooling and boundary consistency learning, which further improves the efficiency of the algorithm. On the premise of scene consistency, AFSD algorithm can achieve good results in action detection in long videos.

Object detection and keypoint detection. In the video action recognition task, there exist actions that usually include two or more objects. At this time, using object detection algorithm to detect objects to assist action recognition is an effective means. At present, object detection algorithms have made great progress. One-stage and multi-

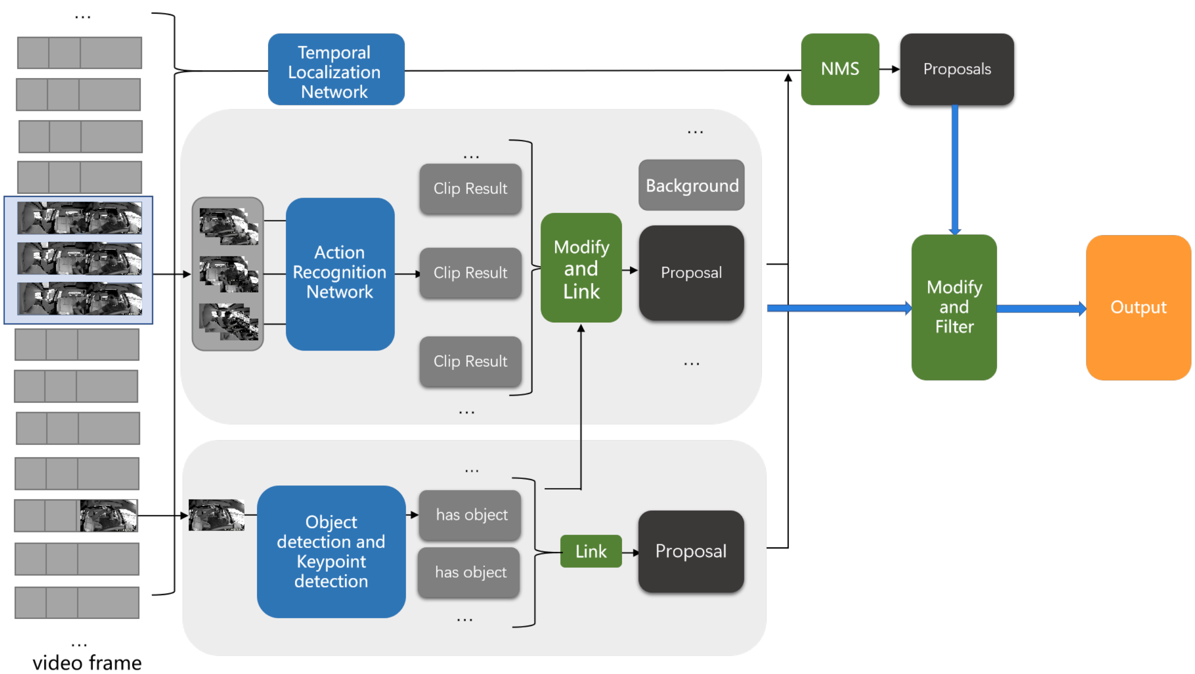


Figure 2. The overview of our approach. The whole architecture contains three branches, namely detection branch, action recognition branch and temporal localization branch. The green blocks together constitute the post-processing module.

stage methods both have their own advantages and disadvantages. The one-stage methods [19, 25] are more efficient and faster, but usually has poor accuracy. While the two-stage or multi-stage methods [2, 11, 23, 26] are inefficient, the detection accuracy is high. In order to obtain higher accuracy, this paper uses DetectoRS [23] as the object detector to detect people and objects in distracted driving behavior. For the mirror action, such as the behavior of left-hand and right-hand calling, we introduce skeleton recognition information as an aid to recognize the mirror action. HRNet [28] is used as the keypoint detector to extract the position information of head joint points, and compare the relative position relationship of objects, so as to distinguish the mirror behaviors.

At present, a lot of progress has been made in the research of temporal action localization. Most existing TAL models rely on anchor-base [31], anchor-free or actionness-guided [17] localization methods. They generate anchor windows or action proposals, which are capable of roughly delineating the behavior in the video. The current best result [34] achieves 65.6% mAP at $tIoU = 0.5$ on THUMOS14 [33], but it still cannot meet the requirements of precise temporal boundaries and accurate classification.

3. Method

3.1. Overall Architecture

According to the requirements of track 3, as well as the small size of the training data set, the many types of behavior categories, the lack of key objects and regional annotations, in order to better identify and locate the distracted behavior of drivers in natural driving videos, this paper proposes a method for precise temporal localization on small-scale data set in naturalistic driving videos, as shown in Figure 2. The whole architecture contains 4 modules namely detection stage, action recognition stage, temporal localization stage and post-processing.

In the action recognition module, video clips are input to the module to be classified into predefined categories. We used a series of post-processing methods to connect and filter out these discontinuous clips.

In the temporal localization module, we adopt the off-the-shelf methods AFSD [14] to obtain the temporal boundaries proposals of the actions.

In the detection module, every video frame is input to the detection network to obtain a key point for person and other objects, such as bottle, cellphone, *etc.* , which the person is interacting with.

In the post-processing module, all of the results from former modules are considered and generate the final results.

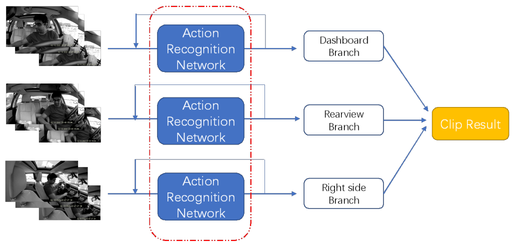


Figure 3. Action recognition module: Fine-tune separately after the information fusion of three perspectives.

Proposals are processed using Temporal Non-Maximum Suppression (TNMS) [22] to suppress misclassification. Finally, 18 proposal over the video temporal segment are obtained.

3.2. Action recognition module

Recognition of distracted behavior of drivers in natural driving videos requires temporal localization of behaviors and identification of their categories in untrimmed videos containing background and multiple behaviors. A action recognition network model and a temporal localization network model are required. This section introduces the action recognition network model in our method.

Convolutional Neural Network (CNN) [12] has great advantages in extracting low-level features and visual structures, but it has limitations in modeling low-level features with a wide range of dependencies. Different from the CNN network, Transformer [6] is very powerful in focusing on global information and long-range dependencies modeling.

Swin Transformer [20] is an improved model recently proposed on the basis of Vision Transformer [1]. The structure of Swin Tranformer is shown in Figure 4 which has 4 stages in total. It not only has the ability of Transformer to focus on global information modeling, but also uses the method of moving windows to connect across windows so that the model can focus on information related to adjacent other windows. Interacting with features across windows extends the field of perception to a certain extent, resulting in higher efficiency. Due to a series of advantages of the Swin Transformer, we use it as the backbone network for behavior recognition. Swin Transformer has different types according to the size of the models and we selected Swin-B(ase) for the task.

Due to the small size of the data set and the limited number of behavioral samples, the training model is first trained on all perspectives, as shown in Figure 3. We pass the data of the three perspectives through a parameter-sharing recognition network and a model with a top1 accuracy rate of 0.5322 was obtained. Obviously, this kind of accuracy does not meet the precision requirements of this task. Considering that every action contains three different perspectives

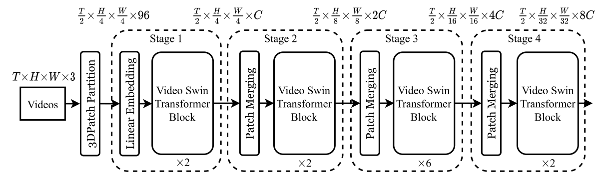


Figure 4. Overview of swin transformer [20].

at the same time, taking full advantage of the information from the three perspectives is essential for this task. Different perspectives have different effects on various behavior recognition, and different behaviors have different characteristics under different perspectives. For example, some behaviors are difficult to distinguish from the dashboard view, but very clear from the rearview view. In order to allow the model to learn the features of different perspectives in a more targeted manner on the basis of learning all video features, we adopt a strategy of training three perspectives separately to further improve the accuracy of the model.

3.3. Temporal Localization module

To get more accurate action temporal boundaries in the video, which is essential in this task, we adopt a temporal localization module. In this module, we utilize an open-source temporal localization model [14] to obtain the temporal boundaries proposals of the actions in video clips. For a video $X = \{x_t\}_{t=1}^{T}$ with T frames, the temporal boundary locations of its corresponding actions can be expressed as $\{(\phi_m, y_m)\}^{N_X}$ where N_X signifies the number of action instances in X, $\phi_m = (\psi_m, \xi_m)$ denotes the start time, end time and y_m indicates the action category.

The proposals obtained by TAL module are ambiguous, and those obtained by the detection branch are generally precise. To prevent them from being filtered out while assisting in the screening of other more likely proposals, we mixed it with a high confidence score into the output of the TAL model. The output of the TAL model covers a wide range of proposals, and TNMS [22] is the most commonly used method to filter out the most appropriate proposals based on confidence scores. We employ targeted strategies based on the particularity of the dataset: (1) The interval between different behaviors of the dataset is usually 5 to 10 seconds, we suppress proposals within 5 seconds of maximum score proposals, (2) A video contains only 18 behaviors, and TNMS only takes the first 18 maximum score proposals.

In the action recognition module, we train 17 classes of clips with actions together with 1 class of background classes. In the inference phase, we are able to obtain the likelihood of action and background class scores for each clip. Then a threshold is set to binarize the background class results, as shown by the yellow line in Figure 5, where the

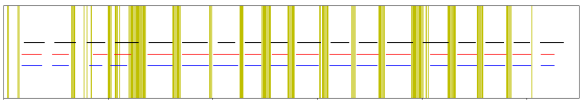


Figure 5. Results of proposal correcting. Black lines represent labels, red lines represent uncorrected proposals, blue lines represent corrected proposals, and yellow lines represent background class.

horizontal axis represents time. Here we take the average of the background class scores in a video as the threshold. It can be observed that background classes are detected between most of the actions.

Hence, we are able to use this information to correct the proposals obtained from the action recognition module by adjusting the temporal boundaries and eliminate the proposals which are too short in temporal domain. The specific algorithm is as follows: if the start and end position is close to the yellow line in Figure 5, move it to the yellow line; if the interval between two proposals is too long, move the start and end position of the proposal with a lower score closer to the higher one. The final proposed result is shown as the blue line in Figure 5.

3.4. Detection module

Objects that appear in a video frame, such as bottles, mobile phones, *etc*., can provide a lot of useful information for the recognition of activities. We input the video frame into the detection network, and then output the bounding box of the objects contained in the frame. This module directly uses the pre training model on the COCO dataset [18] to detect objects without further fine tuning.

In order to distinguish the mirror activity, such as left-hand and right-hand phone calls, we use the public skeleton detection model HRNet [28] to detect the driver's key points in the video. After getting the key points, we justify whether the mirror activity is happened with the right hand or the left hand through comparing the relative position between objects and the center key point, such as the nose point. Through this approach, we improve the action recognition performance of left-hand and right-hand phone calls.

3.5. Post process

The aim of post-processing module of our framework is to connect clips, get proposals of activities, and screen the final results according to the confidence scores of these proposals.

Integrate the scores of three perspectives. In order to make the best use of the results under each perspective, a binary weight matrix $\boldsymbol{M}$ is used to weight the scores of various actions under each perspective, it is one of the hyper-meters that choose by experience:

$$\boldsymbol{M} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \end{pmatrix}_{3\times C} = \begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \end{bmatrix} \tag{1}$$

where the $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, $\boldsymbol{m}_3$ are three binary vectors, representing the class weights of three perspectives respectively, and C is the number of classes. The scores of three perspectives are represented by

$$\boldsymbol{S}_i = \begin{bmatrix} \boldsymbol{s}_{1i} \\ \boldsymbol{s}_{2i} \\ \vdots \\ \boldsymbol{s}_{Ni} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{s}}_{1i} & \widetilde{\boldsymbol{s}}_{2i} & \cdots & \widetilde{\boldsymbol{s}}_{Ci} \end{bmatrix} \tag{2}$$

where $\boldsymbol{s}_{ni}(n = 1, 2, \cdots, N, i = 1, 2, 3)$ is the n-th clip score of the i-th perspective in the category dimension, N is the number of clips, and $\widetilde{\boldsymbol{s}}_{ci}(c = 1, 2, \cdots, C, i = 1, 2, 3)$ is the c-th class score of the i-th perspective in the temporal dimension. We multiply the scores of the three perspectives by the corresponding weight, and then take maximum as the integrated score $\boldsymbol{S}'$. The integration of three perspectives is as follows:

$$\boldsymbol{S}' = \max_{i=1,2,3}(\boldsymbol{S}_i \odot \boldsymbol{m}_i) = \begin{bmatrix} \max\limits_{i=1,2,3}(\boldsymbol{s}_{1i} \odot \boldsymbol{m}_i) \\ \max\limits_{i=1,2,3}(\boldsymbol{s}_{2i} \odot \boldsymbol{m}_i) \\ \vdots \\ \max\limits_{i=1,2,3}(\boldsymbol{s}_{Ni} \odot \boldsymbol{m}_i) \end{bmatrix} \tag{3}$$

where $\odot$ represents the multiplication of elements.

Then the fused scores are processed along the time dimension according to categories. If the mean value of top-20 scores on the whole time axis of a category is less than th_1, then mean filtering and maximum normalization is adopted to process the category along the time dimension. The mean filtering is as follows:

$$\widetilde{\boldsymbol{s}}'_c(j) = \frac{1}{5}\sum_{k=j-2}^{j+2} \widetilde{\boldsymbol{s}}_c(k) \tag{4}$$

where the $\widetilde{\boldsymbol{s}}_c(k)$ is the k-th value of $\widetilde{\boldsymbol{s}}_c$, and $\widetilde{\boldsymbol{s}}'_c(j)$ is the j-th value of mean filtered score $\widetilde{\boldsymbol{s}}'_c$. The maximum normalization is as follows:

$$\widetilde{\boldsymbol{s}}''_c(j) = \frac{\widetilde{\boldsymbol{s}}'_c(k)}{\max(\widetilde{\boldsymbol{s}}'_c(k))} \tag{5}$$

Then use the results of key points on persons, and the object detection results from the detection module to correct the normalized score $\boldsymbol{S}''$ as follows:

$$\boldsymbol{S}_{modified} = \boldsymbol{Modify}(\boldsymbol{S}'', \boldsymbol{detections}) \tag{6}$$

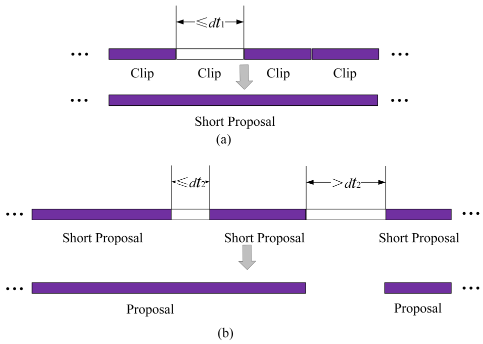


Figure 6. Fragment connection. (a) shows the clip-level linking process; (b) shows the proposal-level linking process.

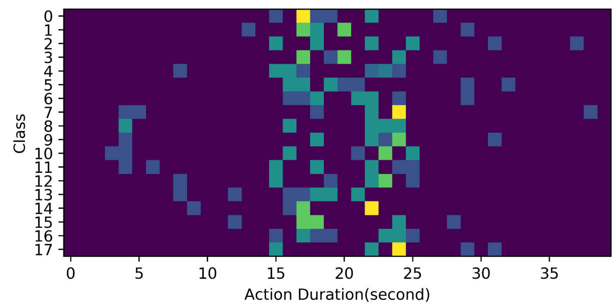


Figure 7. Training set action duration statistics heatmap. The axis is the action duration, from 0 to 40 seconds, and the y axis is the class id, from 0 to 17. The warmer the color of a pixel is, the greater the number of action for the corresponding duration and category is.

So far, the preliminary work has been completed.

Fragment connection. This step aims to associate the clips into long proposals. We use the mean value of top-200 scores of each class as the threshold to judge whether a clip is the class. If the interval between start frames of two same category clips is less than dt_1, the two clips are connected to a short proposal. The empirical value of dt_1 is 4 seconds. After obtaining the compact short proposals, we conduct the post connection to obtain the long proposal. If the interval between two short proposals is less than dt_2, we connect the two short proposals, until there is no short proposals to connect. The empirical value of dt_2 is 8s. Now, the original proposals $\boldsymbol{P}_{original}$ extraction is complete. These two processes are shown in Figure 6.

Temporal action localization correction. Based on the results of the temporal action localization, if the IoU between a proposal P_2 in the TAL results $\boldsymbol{P}_{TAL}$ and one of the original proposals P_1 in $\boldsymbol{P}_{original}$ is greater than th_2, replace P_1 with P_2. In practice, this operation is only carried out for categories with confidence lower than th_3.

Filtering. Inspired by the idea of NMS [22], we propose the algorithm of priority filtering. According to the inferred results for every clips, a category order sorted by predicted scores can be obtained, and then we filter the proposals level by level. When filtering, the category with highest score determines the final location, and suppresses the proposals of other categories in this location (multiply the score by a coefficient less than 1). According to the time limit and confidence score, the most confident proposal is obtained as the final result for each action category. Consequently, we finally obtain at most one proposal for each class, which can improve the precision of the algorithm.

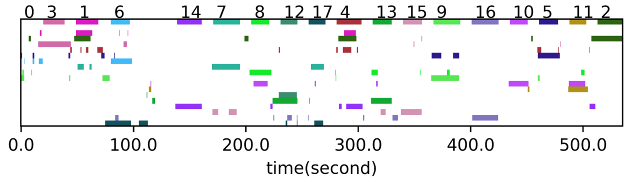
4. Experiments

4.1. Datasets and Settings

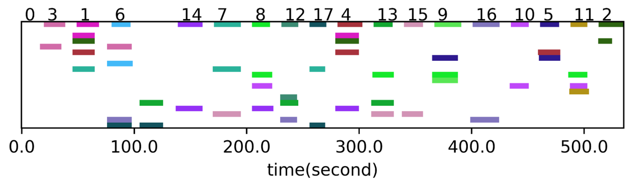
The track 3 dataset [24] in the 6th AI City Challenge contains 30 training videos (A1) and 30 test videos (A2) with a length of approximately 8 minutes, a frame rate of 30 fps and a resolution of 1920×1080, which include 17 distracted action classes and 1 normal driving action class. All the actions in training dataset are within 4 seconds to 38 seconds. The training dataset's statistic heatmap is shown as Figure 7. One action appears only once in the video. Before training and inferring, we first sample frames from the original videos at the $8/30$ sample rate, so as to decline the memory cost. Label annotations are generated at the clip level. Each clip at the action time has a class label with a length of 16 frames. The sliding step during inference is 4 frames, while the training label is generated with a fixed step size of 8 frames to prevent over fitting. Finally, we get 8880 training samples.

For the action recognition, data from different perspectives are trained separately because different behaviors have different characteristics under different perspectives. For example, some behaviors are difficult to distinguish from the dashboard view, but very clear from the rearview view. For the train phase, the training of deep learning network needs a lot of data, but the scale of data set A1 is very small. In order to make up for the problem of too few data sets, we adopt a pretrain-and-finetune manner, that is, putting the data from three perspectives together for pre training, and then fine tuning them separately. Otherwise, training three perspectives separately allows the model to learn the features of different perspectives in a more targeted manner on the basis of learning all the video features, thereby further improving the accuracy of the model. For the pre training process, we use a pre trained Swin Transformer model on Kinetics-400 [3]. All training and inference processes are conducted on four NVIDIA Tesla V100 GPUs.

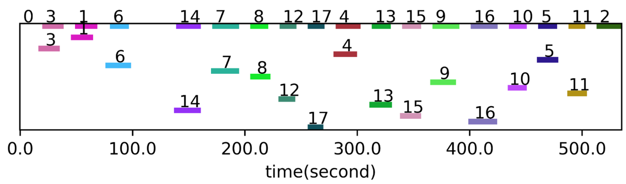
For the temporal action localization, we train and infer under the same hardware conditions. Similarly, we use a pre trained I3D model on Kinetics-400 [3] as the AFSD's backbone and fine tune it on the training data set.



(a) Fragment connection results



(b) TAL correction results



(c) Filtered results

Figure 8. Method results visualization on verification data set. The top line is the ground-truth distribution, and other lines show the method results after processing.

4.2. Evaluation Metric

Evaluation for track 3 is measured by the F1-score as Equation (7). In order to calculate the F1 score, when an action is correctly recognized (matching the action id), that is, it starts within 1 second of the action start time and ends within 1 second of the action end time, it will be considered as true positive (TP) action recognition. A false-positive (FP) is an identified activity that is not a TP activity. A false-negative (FN) activity is a ground-truth activity that is not correctly identified.

$$F1-score = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

4.3. Main results

In order to better simulate the data distribution of the test set A2 and verify the effectiveness of the model, we randomly selected all videos of a tester as the verification set. The ablation on the verification set are shown as Table 1 and Table 2. The former shows the action-level precision, recall and F1-score with different correction strategies. It can be found that the detection module's improvement is not obviously, because the data set is too small. But the TAL module significantly improves the recall from 2.9% to 11.8% and the precision from 6.7% to 28.6%. The priority filtering is also effective to the precision, which is increased from 28.6% to 33.3%. The Table 2 shows the ablation results on clip-level precision and recall with different recognition strategies in three perspectives. Obviously, training the data set respectively in three perspectives can further improve the performance of the recognition module.

Swin	Detection	TAL	Filter	P(%)	R(%)	F1(%)
✓				6.67	2.94	4.08
✓	✓			6.67	2.94	4.08
✓	✓	✓		28.57	11.76	16.67
✓	✓	✓	✓	33.33	11.76	17.39

Table 1. Ablation on PAND with different correction strategies.

Perspective	Dashboard		Rearview		Rightside	
Metric	P(%)	R(%)	P(%)	R(%)	P(%)	R(%)
Together	50.4	86.1	45.9	71.3	58.8	91.3
Respective	51.4	86.6	49.3	74.1	57.3	91.4

Table 2. Ablation on recognition with different strategies.

To more intuitively show the effect of our model, the performance of the model is visualized in Figure 8. It's noticeable that our method performs excellently in classification precision. This is because we make full use of the results of action recognition from three perspectives and exploit lots of correction processes. Especially, the correction of temporal action localization greatly improves the precision and

Metric	Value
Precision(%)	36.75
Recall(%)	24.02
F1-score(%)	29.05

Table 3. The result of our method on test dataset A2 [29].

recall.

The final results on test dataset A2 are shown in Table 3. We achieve 36.75% precision, 24.02% recall and 29.05% F1-score.

5. Conclusion

In this paper, we design an architecture fused object detection, keypoint detection, action recognition and temporal action localization to detect actions in untrimmed videos, which works well in the 6th AI City Challenge track 3 and wins the 6th place with 29.05% F1-score [29]. Different from the past, we use a detailed post-processing pipeline to improve the accuracy of the model, which shows an excellent performance on verification data set. Specifically, we reuse the action recognition results to generate and modify proposals in TAL module, which significantly improves the precision and recall of action recognition. To get a better precision, we filter the proposals according to the category priority, which shows a good performance on the test data set A2.

6. Acknowledgements

This work is supported by Chinese National Natural Science Foundation under Grants (U1931202,62076033).

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, 4

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 3

[3] 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. 6

[4] AI City Challenge. https://www.aicitychallenge.org/, 2022. 1

[5] Wenwen Ding, Xiao Li, Guang Li, and Yuesong Wei. Global relational reasoning with spatial temporal graph interaction networks for skeleton-based action recognition. *Signal Processing: Image Communication*, 83:115776, 2020. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4

[7] Haodong Duan, Yue Zhao, Kai Chen, Dian Shao, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. *arXiv preprint arXiv:2104.13586*, 2021. 2

[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] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 2

[10] 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

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 3

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 4

[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] 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

[15] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 2

[16] 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

[17] 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, 3

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[20] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021. 2, 4

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[22] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006. 3, 4, 6

[23] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10213–10224, 2021. 3

[24] Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Jiyang Wang, Senem Velipasalar Gursoy, D. Anastasiu, and Shuo Wang. Dataset for analyzing various gaze zones and distracted behaviors of a driver. *arXiv preprint arXiv:2204.08096*, 2022. 6

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[26] 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. 3

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

[28] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 3, 5

[29] AI City 2022 Evaluation System. Trecvid 2021 actev: Activities in extended video. https://eval.aicitychallenge.org/aicity2022/submission, 2021. 2, 7, 8

[30] Oytun Ulutan, ASM Iftekhar, and Bangalore S Manjunath. Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13617–13626, 2020. 2

[31] 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, 3

[32] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 2

[33] Rahul Sukthankar Yugang Jiang, Amir R. Zamir. Thumos challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 3

[34] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *arXiv preprint arXiv:2202.07925*, 2022. 2, 3