Multi-View Spatial-Temporal Learning for Understanding Unusual Behaviors in Untrimmed Naturalistic Driving Videos

Huy-Hung Nguyen*, Chi Dai Tran*, Long Hoang Pham, Duong Nguyen-Ngoc Tran,
Tai Huu-Phuong Tran, Duong Khac Vu, Quoc Pham-Nam Ho, Ngoc Doan-Minh Huynh,
Hyung-Min Jeon, Hyung-Joon Jeon, Jae Wook Jeon†
Department of Electrical and Computer Engineering,
Sungkyunkwan University, Suwon, South Korea
{huyhung91, tdc2000, phlong, duongtran, taithp, vukhacduong,
hpnquoc, ngochdm, hmjeon, joonjeon, jwjeon}@skku.edu

Abstract

The task of Naturalistic Driving Action Recognition aims to detect and temporally localize distracting driving behavior in untrimmed videos. In this paper, we introduce our framework for Track 3 of the 8th AI City Challenge in 2024. The approach is primarily based on large model fine-tuning and ensemble techniques to train a set of action recognition models on a small-scale dataset. Starting with raw videos, we segment them into individual action sequences based on their annotation. We then fine-tune four different action recognition models, with K-fold cross-validation applied to the segmented data. Following this, we execute a multi-view ensemble, selecting the most visible camera views for each action class to generate clip-level classification results for each video. Finally, a multi-step post-processing algorithm, which is designed for the AI City Challenge dataset's specific features, is employed to perform temporal action localization and produce temporal segments for the actions. Our solution achieves a final mOS score of 0.7798 and attains the 5th rank on the public leaderboard for the test set A2 of the challenge. The source code will be publicly available at `https://github.com/SKKUAutoLab/AIC24-Track03`.

1. Introduction

Distracted driving behavior refers to different activities that divert a driver's attention away from the primary task of navigating the vehicle, such as texting, making phone calls, or drinking. This significantly increases the risk of accidents and compromises road safety. In 2022, distracted driving was a factor in the loss of 3,308 lives and injury of nearly 290,000 people in the United States. Notably, almost 20 percent of the fatalities were pedestrians, cyclists, and others outside the vehicle, according to the report by the National Highway Traffic Safety Administration (NHTSA) [6]. Therefore, naturalistic driving studies, leveraged by computer vision techniques, are crucial in identifying and eliminating distracted driving behavior on the road. They capture all driver actions in the traffic, including those related to drowsiness or distracted behavior.

In recent years, many effective solutions [1, 4, 15, 31, 33] have been developed to detect driving behavior on the road. However, challenges such as insufficient labeling, subpar data quality, and low resolution continue to hinder the extraction of meaningful insights from real-world driver data. To address this, AI City Challenge [28] has established a Naturalistic Driving Action Recognition challenge track to analyze the distracted behavior of the driver and introduced synthetic distracted driving SynDD1 [21], SynDD2 [22] datasets to be used in the challenge. The 2024 iteration of the dataset was collected inside a stationary vehicle under three camera angles: on the dashboard, near the rear-view mirror, and on the top right-side window corner. For each driver, each activity is divided into two groups: with and without appearance blocks (e.g., wearing a hat or sunglasses). There are 16 distracted actions (such as phone calls, eating, and texting) for each participant that happen at random times and in order. The training annotations of the dataset are labeled manually for each activity, including start and end times (the start time may be annotated early 10 seconds before the action starts). The objective is to accurately detect distracted actions as well as their start and

*These authors contributed equally to this work.

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korean government(MSIT) (No. 2021-0-01364, An intelligent system for 24/7 real-time traffic surveillance on edge devices).

end times in a given untrimmed video. Thus, this task can be considered a temporal action localization (TAL) problem [27].

Compared to TAL, the Naturalistic Driving Action Recognition track presents distinct challenges. Firstly, the collected dataset size is relatively limited, yet it comprises 16 action categories that require classification. Secondly, certain actions pose difficulties in differentiation. For instance, ”Singing or dancing with music” and ”Talking to passengers at backseat” actions share some similarities. A driver may glance at the rear-view mirror while conversing with the backseat passenger, a gesture that could be mistaken as singing. Thirdly, the task permits the utilization of multiple camera views for action prediction. ”Eating” action can be identified easily from any camera view, but ”Adjusting control panel” action may only be discernible in the Right-side view. Finally, some actions occur in a mere second, such as ”Drinking” or ”Yawning”, making it difficult for action recognition systems to distinguish between these actions.

In this paper, we present a solution to the driving action recognition challenge. Our approach begins with a preprocessing phase, where we segment the raw videos from different camera angles in the dataset into class-specific and temporally annotated action sequences. Then, we perform fine-turning of three pre-trained action recognition models, including VideoMAE, UniformerV2 and X3D, with five-fold cross-validation employed on the segmented data across all 16 classes. We also fine-tune an additional UniformerV2 model on two particularly challenging ”Talking to passenger at the right”, ”Talking to passenger at backseat” classes alongside ”Normal forward driving” background class. Next, we ensemble the trained models to generate clip-level classification results, tailoring different camera view selections corresponding to each model and each action class. Finally, a multi-step post-processing algorithm is applied to perform temporal action localization task, which merges together the clips within the same action into a temporal segment and discards any inaccuracies with low confidence scores.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the proposed method. The experiments are examined in Section 4, and the conclusion and future work are in Section 5.

2. Related work

2.1. Video action recognition

Video action recognition is a fundamental task of computer vision that focuses on identifying and categorizing human actions from video data. Advances in this field have been significantly influenced by the integration of Convolutional Neural Networks (CNNs) and Transformers.

Initially, 2D CNN-based methods extract spatial features before combining them via temporal modeling [14, 16]. These methods are computationally efficient and suitable for large-scale video datasets. However, since temporal information is fused after spatial feature extraction, there is a risk of losing fine-grained temporal nuances essential for accurate action recognition. Moreover, the effectiveness of 2D CNN-based approaches heavily depends on the quality of spatial feature extraction, making them sensitive to variations in pose, viewpoint, and object occlusion. In contrast, 3D-CNN-based methods treat video objects as 3D entities, employing convolutions across both spatial and temporal dimensions to capture intricate motion patterns over time [3, 25]. By simultaneously capturing spatial and temporal information, these models preserve crucial temporal context for accurate action recognition. Moreover, 3D CNN-based models exhibit greater robustness than their 2D counterparts when dealing with spatial variations such as pose changes, different viewpoints, and object occlusions. However, processing spatiotemporal volumes demands more computational resources, resulting in longer inference time. Additionally, due to their model complexity, 3D CNN-based methods may require larger datasets for effective training, which can be challenging to obtain in certain domains.

Transformer-based methods, inspired by the success of Transformers in the image domain, use Vision Transformers (ViT) as backbones and achieve outstanding results on various video understanding benchmarks [2, 18]. Furthermore, recent works take advantage of video foundation models and multi-model approaches to further enhance performance in the video domain [12, 23]. Utilizing Transformer-based models allow capturing long-range dependencies, thereby enhancing action recognition accuracy. In addition, Transformer-based architectures are highly scalable, which enables processing both short and long video sequences without computational inefficiency. However, Transformer-based methods often require substantial labeled data for pre-training and fine-tuning, which can be challenging in domains with limited access to labeled datasets.”

2.2. Temporal action localization

Temporal action localization aims to detect activities in an untrimmed video and output time boundaries for each action. Typically, it is divided into three categories, multi-stage, two-stage, and one-stage methods.

In the multi-stage approach, the process begins with frame-level classification to identify potential action instances. Subsequently, post-processing techniques are employed to refine this classification and determine the precise start and end timestamps for each action [17, 30].

Conversely, the two-stage approach operates by initially generating candidate proposals for potential actions within a video sequence. These proposals are then subjected to classification and refinement stages to accurately determine the temporal boundaries of each action [29, 32]. Despite the model complexity, both multi-stage and two-stage approaches yield high temporal localization accuracy owing to their multi-step refinement process.

Lastly, the single-stage approach integrates action classification and temporal localization steps into a single stage without the need for explicit proposal generation. This approach aims to simplify the process by directly predicting action labels and their temporal boundaries. Although this approach is simple, it has been demonstrated ineffective when applied to the AI City Challenge dataset because of the limited amount of the data [26].

3. Proposed method

Overall, our framework is based on fine-tuning large models on a small-scale dataset to build a strong set of video action recognition models. Then, we use a multi-view, multi-fold ensemble to obtain clip-level classification probabilities. Finally, a multi-step post-processing algorithm is applied to localize temporal segments corresponding to abnormal actions. The overview architecture of our model is illustrated in Fig 1.

3.1. Video action recognition model fine-tuning

Due to the constrained size of the provided dataset, relying solely on a single large model could lead to overfitting. To mitigate this, we employ a variety of action recognition models in an ensemble, capitalizing on the unique visual features each model can learn.

In the image domain, Masked Autoencoders (MAE) [9] has been one of the state-of-the-art self-supervised models that use mask modeling to perform diverse computer vision tasks such as image classification, object detection, and segmentation. In the video domain, VideoMAE [24] extends MAE to 3D space by using the potential of a vanilla Vision Transformer for video action recognition. It can also be considered the first masked video pre-training framework that adopts plain ViT backbones and has an excellent performance on various video understanding benchmark datasets such as Kinetics-400 [10], Something-Something V2 [7], and AVA v2.2 [8]. Therefore, inspired by the success of this model and the winning solution last year [33], we utilize VideoMAE as one of our base models for distracted driving action classification. Furthermore, to learn more representative features, we also initialized VideoMAE on Kinetics-710 [11].

Owing to the powerful performance of Transformer-based approaches in computer tasks, we also adopt UniFormerV2 [11] as our second base model. UniformerV2 is an improved version of Uniformer [13] that meticulously refined the local and global relation aggregators, blending the strengths of both ViT and Uniformer, resulting in a seamlessly integrated architecture. It is worth noting that UniformerV2 achieved 90% top-1 accuracy on the Kinetics-400 dataset, positioning it among the first models to reach this milestone. Furthermore, the network demonstrates competitive results across eight popular video understanding benchmarks when compared to previous Transformer-based models. Our decision to include UniFormerV2 was also influenced by the overlap in action labels between the Kinetics dataset and the SynDD2 dataset, such as ”yawning” / ”Yawning”, ”talking on cell phone” / ”Phone call”, and ”singing” / ”Singing or dancing with music”.

Upon examining of the inference performance of VideoMAE and UniformerV2, we discover that certain actions, specifically ”Talking to passenger at the right”, ”Talking to passenger at backseat” frequently challenge action recognition models. To address these difficulties, we prepare two versions of UniFormerV2. The first version is trained across all 16 classes, paralleling the training of VideoMAE. For the second version, we redefine the remaining actions as normal driving behavior and focus the training on three specific classes. This approach significantly enhances our models' capability to detect the aforementioned challenging actions.

For the last base model, we employ X3D [5], a lightweight CNN-based network that progressively expands the network axes of a tiny 2D image in terms of space, time, width, and depth, respectively. X3D also requires $4.8\times$ fewer multiply-adds operations and $5.5\times$ fewer parameters compared to previous CNN-based networks, making it more computationally efficient to train on a large-scale dataset. During training, we observed that actions easily visible from the Rear-view and Right-side such as ”Phone call”, ”Texting”, or ”Adjusting control panel” posed challenges for UniformerV2. However, X3D performs well on these actions. Consequently, X3D is adopted alongside VideoMAE and UniformerV2 to learn more diverse features.

3.2. Multi-view multi-fold model ensemble

First, to enhance the generalization of the video models, we employ a K-fold cross-validation strategy with $(\mathrm{K} = 5)$ in the training pipeline, drawing inspiration from [26].

Secondly, because the dataset provides calibrated videos from three camera views, it is advantageous to incorporate information from multiple camera views in detecting driver actions. For instance, the Rear-view camera excels in identifying hand-related actions such as “Drinking”, “Phone call (left)” and “Hand on head”, while the Dashboard camera is particularly good at capturing instances of “Drinking”. Conversely, the Right-side camera is invaluable for actions that may otherwise be obscured, including “Phone call (right)”, “Texting (right)”, “Texting (left)”, “Adjusting control panel” and “Singing or dancing to music”. Based on

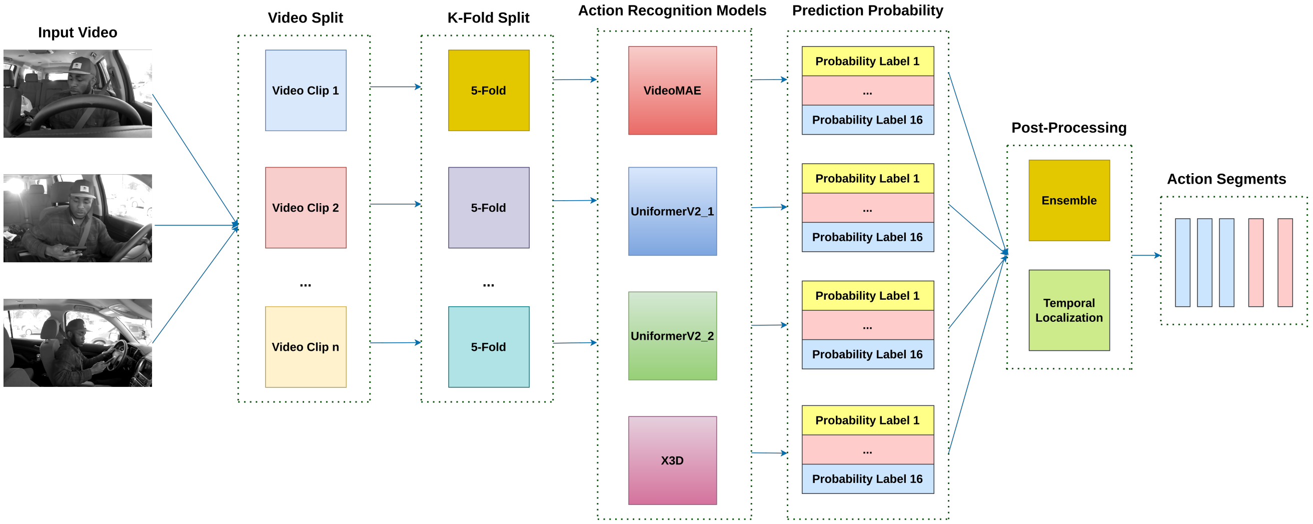

Figure 1. The overview architecture of the proposed framework. Initially, each input video is divided into multiple segments corresponding to 16 action classes from different camera views. Then, they are split into five folds before being fed into four action recognition models for training (UniformerV2_1 and UniformerV2_2 are two versions that are trained on all 16 classes and three classes, respectively). Next, the action classification probabilities are used for the multi-view ensemble to obtain clip-level classification results. Finally, a post-processing technique is applied to generate temporal action localization results.

empirical analysis, we arrange different camera view selections corresponding to each model and each action class, as listed in Table 1. The camera view selections will directly affect the model ensemble score:

$$S = \alpha * (\frac{\sum_{i=1}^{5} Dash_i}{n} + \frac{\sum_{i=1}^{5} Dash'_i}{n}) \\ + \beta * \frac{\sum_{i=1}^{5} Right_i}{n} + \sigma * \frac{\sum_{i=1}^{5} Rear_i}{n} \tag{1}$$

where n denotes the number of training folds, $Dash, Right, Rear$ are weights of VideoMAE corresponding to each fold of each view, $Dash'$ is the Dashboard weight of UniformerV2_1, and α, β, σ are hyperparameters to control the prediction accuracy of action labels.

Our general ensemble score (Eq. 1) is applied to specific actions such as ”Eating”, ”Picking up from floor” and ”Talking to passenger”. As can be seen in Table 1, we observed that VideoMAE is performs well across all views, while UniFormerV2_1 achieve good results for actions observed from the Dashboard view. Consequently, we combine results from the Dashboard, Rear-view, and Right-side of VideoMAE and UniformerV2's Dashboard to construct Eq. 1.For actions not covered by the general ensemble score, we leverage different perspectives from UniformerV2_2 and X3D, which handle cases typically visible only from the Rear-view and Right-side.

3.3. Multi-step post-processing

Upon receiving a video input, each of the models computes the probability score for the action clips. These initial outputs are then refined through a post-processing phase to determine the action labels and their corresponding temporal boundaries. While the Non-Maximum Suppression (NMS) algorithm [20] is typically used for TAL, the study in [26] has shown that it is ineffective for the AI City Challenge dataset due to the distinct timing of actions within each video and the absence of overlap among them. In our framework, we perform a multi-step post-processing algorithm to generate final action temporal segments.

In *Step 1*, we aggregate the results from the five folds of each model across three different views and calculate their mean across all models and views. Following this, we apply a smoothing operation with a mean filter, as suggested in [19]:

$$\tilde{P}_l(x) = \frac{1}{2w} \sum_{j=l-w}^{l+w} p_l(x) \tag{2}$$

where w is the window size, $p(x)$ and $l(x)$ denote the sequence of probability scores and its length, respectively. The smoothing operation aims to rectify any inaccuracies in clip classification, resulting in a series of contiguous clips. These refined fold-averaged results are then utilized in the multi-view ensemble to calculate the composite probability score for the action clips in accordance with Equation 1.

In *Step 2*, actions that exhibit low probability scores are filtered by reassigning their category to 0, indicative of nor-

Table 1. The summary of selected camera views of each model to predict each distracted driving action.

Activities	VideoMAE			UniformerV2_1			UniformerV2_2			X3D		
	Dash	Rear	Right	Dash	Rear	Right	Dash	Rear	Right	Dash	Rear	Right
Drinking	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Phone Call (Right)	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Phone Call (Left)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Eating	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Texting (Right)	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Texting (Left)	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Reaching behind	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓
Adjusting Control Panel	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓
Picking up from floor (Driver)	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Picking up from floor (Passenger)	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Talking to passenger (Right)	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Talking to passenger (Backseat)	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
Yawning	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Hand on head	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Singing or dance with music	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓

mal driving behavior, provided these scores fall below a predefined threshold λ_1.

In *Step 3*, the linking operation is initiated by first discarding clips having category 0. We then merge the clips sharing the same categories into a unified temporal segment, provided their time gap is less than a pre-defined merging threshold λ_2. For activities that typically span longer durations, such as ”Singing or dance with music” or ”Talking to passenger”, we apply a higher λ_2 to capture both short and long temporal segments of each action.

In *Step 4*, we perform two noise removal operations. First, observation indicates that most actions occur within a 1 to 30-second window. Hence, to prevent overlap with standard actions, we exclude activities whose duration exceeds 32 seconds. Second, training annotations reveal that actions like “drinking,” “reaching behind,” “adjusting the control panel,” “picking up from the floor,” and “yawning” are short-lived, whereas others are more prolonged. For these longer actions, we eliminate any that fall below a predefined noise threshold λ_3 to avoid noisy actions. Notably, λ_1, λ_2, and λ_3 are hyperparameters that are fine-tuned during the inference stage for optimal results.

In *Step 5*, based on the analysis of the results after *Step 4*, we identified a subset of actions with overlapping characteristics that could potentially lead to erroneous identifications by action recognition models. Instances include actions such as “Talking to passenger at the right” and ”Talking to passenger at backseat”, or ”Talking to passenger at the backseat” and ”Singing or dance with music”. To mitigate this, we start a reclassification operation where we compare the prediction scores for corresponding views of each action pair and assign the label with the higher probability score to enhance accuracy. Fig 2 illustrates five main steps of our post-processing algorithm.

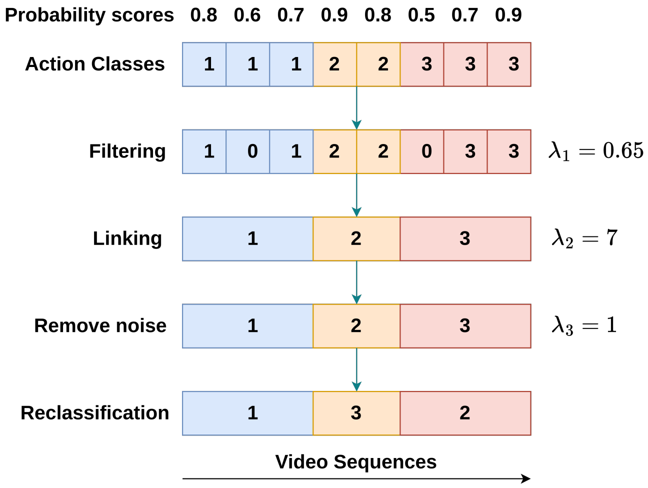

Figure 2. An example of our post-processing algorithm on a video sequence.

4. Experiments

4.1. Dataset

The dataset consists of 594 video clips for about 90 hours in total, which were captured by 99 drivers. In each video, participants must do 16 different activities such as phone, eating, and reaching back in random order. Three cameras were mounted in the car, talking responsibility for recording different angles in synchronization. Furthermore, each driver performs the task twice, one is equipped with no appearance block and the another is equipped with sunglasses or a hat. Thus, each driver has a total of six videos, resulting in 594 videos in total.

In Track 3 of AI City Challenge 2024, these videos are split into three datasets A1, A2, and B, each including 69, 15, and 15 drivers, respectively. For dataset A1, the ground truth labels of the start time, end time, and type of corresponding distracted behaviors were provided. Dataset A2 was given with no labels, and it was used as the first test set to evaluate action recognition algorithms on the online evaluation server. Dataset B is released later and is used as the final test set. The objective of this track is to locate accurate timestamps and the types of distracted behavior from the untrimmed videos.

4.2. Implementation details

Our framework is developed and tested on a workstation running Ubuntu 22.04 with Pytorch 1.11.0. The machine is powered by an Intel Core i9-10980XE @3.00GHZ CPU and 4x RTX A6000 GPUs, each equipped with 48GB of VRAM.

For the VideoMAE model, similar to [33], we use a 16-frame vanilla ViT-L model as the backbone paired with a simple linear classification head. The input size, clip lengths, the number of samples, and the sampling rate are specified as 224, 16, 1, and 4, respectively. Each view undergoes a 35-epoch training cycle per fold, utilizing the lion optimizer, a learning rate of 0.001, weight decay of 0.2, a cosine annealing learning rate schedule, a five-epoch warm-up period, and a layer decay of 0.75.

In the case of the two UniformerV2 models, we employ the vanilla ViT-L model as the backbone with the input size, clip lengths, the number of samples, and the sampling rate specified as 336, 7, 1, and 16, respectively. Training for each view extends over 50 epochs per fold, employing the Adamw optimizer, a learning rate of 0.0004, weight decay of 0.05, a cosine annealing learning rate schedule, no warm-up epochs, and a dropout rate of 0.5.

Regarding the X3D model, we apply X3D-L model, selecting width and depth multipliers as 2.0 and 5.0, respectively. The input size, clip lengths, the number of samples and, the sampling are set to 448, 8, 1, and 4, respectively. The learning rate is initialized as 0.0005 with Adam optimizer. The cosine annealing schedule is also applied with a learning rate of 0.0005.

To train all the above models, we follow the format of the Kinetics dataset by splitting input untrimmed videos into multiple small segments corresponding to specific classes. For VideoMAE, we utilized the pre-trained weight from [33]. For UniformerV2, the pre-trained model is used from its original repository [11]. For X3D, we use its pre-trained model on the Kinetic dataset, which was published on the PySlowFast library repository.

In the post-processing step, the k smoothing values are set to 1, 2, 1, 1 for VideoMAE, UniformerV2_1, UniformerV2_2, and X3D. The threshold values of λ_1, λ_2, and λ_3 are 0.3, 7, and 3. The α, β, and σ values are chosen as 0.3, 0.4, and 0.3, respectively.

4.3. Evaluation metric

Video action recognition. In video action recognition, we evaluate our model using two common classification metrics: Accuracy and F1 score. Specifically, the Accuracy can be computed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP, TN, FP, and FN represent the number of true positive, true negative, false positive and false negative, respectively.

For the F1 score, it is calculated as follows:

$$\begin{aligned} F1 &= \frac{2 \times precision \times recall}{precision + recall} \\ &= \frac{TP}{TP + \frac{1}{2}(FP + FN)} \end{aligned} \tag{4}$$

Temporal action localization. In temporal action localization, we measure the performance of our model using average activity overlap score. Let g be the ground truth activity with start time gs and end time ge, we aim to find the closest predicted activity match p of the same class as g with the highest overlap score os. The start time ps and the end time pe are allowed in the range $[gs - 10s, gs + 10s]$, and $[ge - 10s, ge + 10s]$, respectively. The overlap between g and p is defined as follows:

$$os(p, g) = \frac{max(min(ge, pe) - max(gs, ps), 0)}{max(ge, pe) - min(gs, ps)} \tag{5}$$

After matching each ground truth and prediction activities by start times, all unmatched activities between the ground truth and the prediction have an overlap score of 0. The final score is the average overlap score among all matched and unmatched activities.

4.4. Results

Video action recognition. For each camera view, the 5-fold cross-validation is applied across all drivers in the training set. Table 2 presents the results from different camera views across five folds. It can be observed that the accuracy and F1 scores exhibit slight variations when we alter the camera angle for drivers. For example, the Dashboard view performs exceptionally well in fold 1, surpassing the Rear-view and Right-side by approximately 2.85%. However, in fold 3, it exhibits the poorest accuracy, lagging behind the Rear-view and Right-side by approximately 7.79%. To enhance model performance, we employed ensemble techniques that consider all camera angles rather than relying solely on a single view.

Table 2. Results from 5-fold cross-validation on the validation set for each fold.

Camera View	Fold	Accuracy	F1 Score
Dashboard	1	95.20	91.27
	2	94.93	91.0
	3	87.41	86.62
	4	89.78	88.71
	5	91.66	89.50
Rear-view	1	94.33	90.85
	2	92.09	89.55
	3	95.86	91.70
	4	95.71	91.70
	5	88.91	87.92
Right-side	1	90.36	87.86
	2	91.79	89.22
	3	88.73	87.55
	4	82.84	84.47
	5	90.48	88.68

Table 3 displays the accuracy of each action recognition model for individual classes. As can be seen, the performance varies across different models when evaluated using folds. In fold 2, VideoMAE, UniformerV2_1, and the ensemble version exhibit comparative results. However, across the remaining folds, the ensemble version consistently outperforms the other models for all classes. In class 12 for all folds, the ensemble version's results are relatively lower, indicating the need for further hyperparameter optimization.

Temporal action localization. Table 4 shows the top teams from the public leaderboard of the challenge, evaluated on test set A2. Our proposed method secured the 5th position with a 0.7789 mOS score. We outperformed half of the other teams with an average performance difference of about 20%, showcasing the effectiveness and strong generalization ability of our approach.

5. Conclusion

In this study, we presented a solution for Track 3 of the AI City Challenge 2024, which is viewed as a temporal action localization task for Naturalistic Driving Action Recognition. Concretely, the proposed framework includes three main steps. Firstly, we train four pretrained action recognition models, VideoMAE, UniformerV2_1, UniformerV2_2, and X3D for a clip-level classification. Secondly, we employ multi-view ensemble techniques to improve the prediction results. Finally, a non-trivial post-processing algorithm is given to locate precise temporal boundaries and remove noisy actions for short and long temporal correlations in untrimmed videos. The experimental results on the A2 dataset showed that our framework achieved a good performance, with an mOS score of 0.7798, and ranked fifth in the challenge. In our future work, we plan to examine replacing VideoMAE with VideoMAEv2, an improved version of VideoMAE that introduces a dual masking strategy for efficient pre-training. Additionally, we aim to enhance the model's versatility by exploring optical flow features and automatic image colorization techniques.

References

[1] Armstrong Aboah, Ulas Bagci, Abdul Rashid Mussah, Neema Jakisa Owor, and Yaw Adu-Gyamfi. Deepsegmenter: Temporal action localization for detecting anomalies in untrimmed naturalistic driving videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5358–5364, 2023. 1

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

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

[4] Xiaodong Dong, Ruijie Zhao, Hao Sun, Dong Wu, Jin Wang, Xuyang Zhou, Jiang Liu, Shun Cui, and Zhongjiang He. Multi-attention transformer for naturalistic driving action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5434–5440, 2023. 1

[5] 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. 3

[6] National Center for Statistics and Analysis. Distracted driving in 2022. Technical report, National Highway Traffic Safety Administration, 2022. 1

[7] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 3

[8] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018. 3

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola,

Table 3. The accuracy (%) of distracted driving classes. All results are evaluated on the validation set for each fold. The best results are bolded and the second-best are underlined.

Method								Fold 1							
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12	Clas 13	Class 14	Class 15
VideoMAE	**100.0**	**95.65**	**100.0**	73.91	88.46	92.31	96.15	**100.0**	95.65	**100.0**	84.62	70.83	88.46	**100.0**	92.0
UniformerV2_1	80.95	95.45	**100.0**	47.37	**100.0**	**100.0**	**100.0**	**100.0**	92.0	**100.0**	82.61	**80.0**	**91.67**	**100.0**	45.45
X3D	52.17	70.83	75.00	47.37	71.43	65.00	76.19	75.00	72.72	83.33	50.00	66.67	0.0	96.00	86.36
Our ensemble	95.45	**95.65**	**100.0**	**86.96**	92.31	96.15	**100.0**	**100.0**	**100.0**	96.15	**85.71**	55.56	84.62	**100.0**	**100.0**
								Fold 2							
VideoMAE	83.33	**100.0**	**100.0**	**87.50**	**100.0**	**90.0**	**100.0**	**100.0**	75.0	**88.89**	**90.0**	**90.0**	**100.0**	**100.0**	**100.0**
UniformerV2_1	71.43	**100.0**	**100.0**	33.33	**100.0**	80.0	**100.0**	**100.0**	**87.50**	77.78	77.78	77.78	**100.0**	**100.0**	**100.0**
X3D	44.44	**100.0**	**100.0**	75.00	**100.0**	**90.0**	71.43	90.0	33.33	20.0	37.50	42.86	0.0	88.89	77.78
Our ensemble	**87.50**	**100.0**	**100.0**	**87.50**	**100.0**	**90.0**	**100.0**	**100.0**	75.0	77.78	77.78	40.0	**100.0**	**100.0**	**100.0**
								Fold 3							
VideoMAE	**87.50**	90.0	**100.0**	70.0	77.78	**100.0**	**90.0**	90.0	88.89	**100.0**	87.50	88.89	80.0	**90.0**	90.0
UniformerV2_1	100.0	**100.0**	**100.0**	30.0	**100.0**	85.71	**90.0**	**100.0**	**100.0**	**100.0**	**90.0**	**100.0**	**100.0**	**90.0**	90.0
X3D	33.33	**100.0**	77.78	42.86	25.0	33.33	70.0	57.14	57.14	40.0	40.0	80.0	10.0	70.0	70.0
Our ensemble	77.78	90.0	**100.0**	**80.0**	87.5	**100.0**	**90.0**	**100.0**	**100.0**	**100.0**	75.0	25.0	**100.0**	**90.0**	**100.0**
								Fold 4							
VideoMAE	**100.0**	**100.0**	**100.0**	77.78	80.0	**90.0**	**100.0**	90.0	**88.89**	88.89	80.0	70.0	77.78	90.0	70.0
UniformerV2_1	75.0	**100.0**	**100.0**	37.50	**100.0**	88.89	90.0	**100.0**	85.71	80.0	80.0	**90.0**	**80.0**	**100.0**	**100.0**
X3D	30.0	87.50	50.0	44.44	44.44	71.43	50.0	87.50	71.43	66.67	66.67	75.0	0.0	90.0	85.71
Our ensemble	88.89	**100.0**	**100.0**	**88.89**	80.0	**90.0**	**100.0**	90.0	**88.89**	**90.0**	**88.89**	50.0	70.0	88.89	77.78
								Fold 5							
VideoMAE	97.30	**98.72**	**97.33**	74.60	88.31	96.0	93.51	98.72	86.84	84.51	64.10	74.03	84.62	97.40	90.91
UniformerV2_1	95.24	98.68	**97.33**	62.50	**94.44**	93.51	**97.22**	96.10	90.41	**92.31**	74.03	**80.0**	**90.41**	**98.68**	71.43
X3D	28.77	84.21	72.60	44.93	57.14	82.43	59.65	68.25	50.0	50.94	51.06	56.25	6.41	92.11	87.50
Our ensemble	**98.67**	98.71	94.74	**85.25**	88.31	**97.40**	93.51	**100.0**	**93.42**	91.67	**74.65**	42.86	84.62	98.67	**92.11**

Table 4. The summary of top 10 public leaderboard of AI City Challenge Track 3 2024.

Rank	Team ID	Team Name	mOS
1	155	TelaAI	0.8282
2	189	supermonkey	0.8213
3	32	yptang	0.8149
4	207	Rockets	0.8045
5	5	**SKKU-AutoLab (our)**	**0.7798**
6	136	Bumblebee_AIO	0.7624
7	17	boat	0.6844
8	165	MCPRL	0.6080
9	156	zzl	0.5963
10	125	USTC-IAT-United	0.2307

Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[11] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. 3, 6

[12] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19948–19960, 2023. 2

[13] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

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

[15] Rongchang Li, Cong Wu, Linze Li, Zhongwei Shen, Tianyang Xu, Xiao-jun Wu, Xi Li, Jiwen Lu, and Josef Kittler. Action probability calibration for efficient naturalistic driving action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5269–5276, 2023. 1

[16] 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

[17] 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

[18] 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

[19] Alberto Montes, Amaia Salvador, Santiago Pascual, and Xavier Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv preprint arXiv:1608.08128*, 2016. 4

[20] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, pages 850–855. IEEE, 2006. 4

[21] 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. *Data in Brief*, 46:108793, 2023. 1

[22] 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. 2023. 1

[23] Zhensheng Shi, Ju Liang, Qianqian Li, Haiyong Zheng, Zhaorui Gu, Junyu Dong, and Bing Zheng. Multi-modal multi-action video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13678–13687, 2021. 2

[24] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 3

[25] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2

[26] 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. 3, 4

[27] Binglu Wang, Yongqiang Zhao, Le Yang, Teng Long, and Xuelong Li. Temporal action localization in the deep learning era: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[28] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh Chakraborty, Sanjita Prajapati, Quan Kong, Norimasa Kobori, Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Ganzorig Batnasan, Fady Alnajjar, Ping-Yang Chen, Jun-Wei Hsieh, Xunlei Wu, Sameer Satish Pusegaonkar, Yizhou Wang, Sujit Biswas, and Rama Chellappa. The 8th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 1

[29] 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. 3

[30] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10156–10165, 2020. 2

[31] Liangqi Yuan, Yunsheng Ma, Lu Su, and Ziran Wang. Peer-to-peer federated continual learning for naturalistic driving action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5249–5258, 2023. 1

[32] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2914–2923, 2017. 3

[33] Wei Zhou, Yinlong Qian, Zequn Jie, and Lin Ma. Multi view action recognition for distracted driver behavior localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5374–5379, 2023. 1, 3, 6