M^{2}DAR: Multi-View Multi-Scale Driver Action Recognition with Vision Transformer

Yunsheng Ma[1], Liangqi Yuan[1], Amr Abdelraouf[2],
Kyungtae Han[2], Rohit Gupta[2], Zihao Li[1], Ziran Wang[1]
[1]Purdue University, College of Engineering [2]Toyota Motor North America, InfoTech Labs
{yunsheng, liangqiy, zihao, ziran}@purdue.edu
{amr.abdelraouf, kyungtae.han, rohit.gupta}@toyota.com

Abstract

Ensuring traffic safety and preventing accidents is a critical goal in daily driving, where the advancement of computer vision technologies can be leveraged to achieve this goal. In this paper, we present a multi-view, multi-scale framework for naturalistic driving action recognition and localization in untrimmed videos, namely M^{2}DAR, with a particular focus on detecting distracted driving behaviors. Our system features a weight-sharing, multi-scale Transformer-based action recognition network that learns robust hierarchical representations. Furthermore, we propose a new election algorithm consisting of aggregation, filtering, merging, and selection processes to refine the preliminary results from the action recognition module across multiple views. Extensive experiments conducted on the 7th AI City Challenge Track 3 dataset demonstrate the effectiveness of our approach, where we achieved an overlap score of 0.5921 on the A2 test set. Our source code is available at https://github.com/PurdueDigitalTwin/M2DAR.

1. Introduction

Distracted driving poses a serious threat to road safety, with approximately 8.6 fatalities occurring each day in the US, a figure that is on the rise according to the National Highway Traffic Safety Administration (NHTSA) [26]. The danger is further amplified by the increased reliance of drivers on automated driving systems, especially those classified as SAE Level 3 [23]. These systems enable drivers to disengage from steering and pedal control, but they must still remain vigilant and prepared to regain control of the vehicle. However, drivers are prone to losing awareness of their surroundings when not actively driving, and engaging in distractions can significantly impair their ability to retake control.

Computer vision (CV) is a crucial tool in detecting distracted driving on the road, but its effectiveness can be limited by factors such as inadequate or poor quality data. To address these challenges, the Track 3 of AI City Challenge 2023 [18] has released a comprehensive dataset and organized a competition on naturalistic driving action recognition (DAR). The dataset features recordings of driver actions in real-world scenarios, captured from multiple camera angles, including instances of drowsy or distracted driving [19]. By analyzing these rich driving data, we can gain valuable insights into driver behavior, which can help in developing more effective driver monitoring to improve road safety. The competition's objective is not only to accurately classify but also to localize action segments within an untrimmed video sequence, a problem known as temporal action localization (TAL).

To tackle the challenges associated with DAR, we present a multi-view, multi-scale framework utilizing Vision Transformers (ViT), namely M^{2}DAR. The primary contributions of this paper include:

- The introduction of a weight-sharing, multi-scale Transformer-based action recognition network that learns robust hierarchical representations across multiple views.
- A novel election algorithm consisting of four crucial steps - aggregation, filtering, merging, and selection - designed to refine the preliminary findings from the action recognition network.
- The achievement of our proposed system, which secured 5th place on the public leaderboard of the A2 test set in the AI City Challenge 2023 Track 3, highlights the effectiveness and efficacy of our approach in accurately recognizing driver distractions.

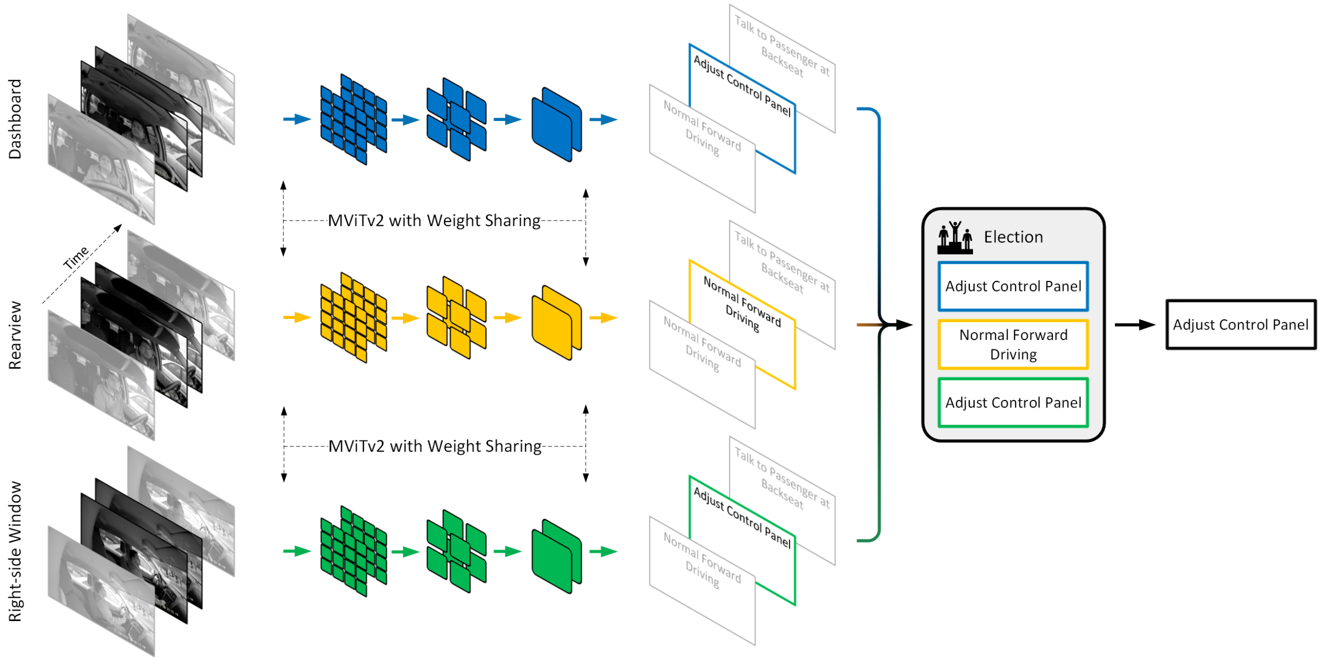

Figure 1. Schematic diagram of our M^{2}DAR system. The system consists of two stages: the action recognition stage and the election stage. In the action recognition stage, the system recognizes driver actions using a weight-sharing recognition network. In the election stage, the system refines the preliminary results obtained from the action recognition stage to generate the final action time chunks.

2. Methodology

In this section, we introduce our M^{2}DAR system, which offers an efficient and effective solution for accurately detecting and recognizing naturalistic driver actions. Our system consists of two stages: the DAR stage and the election stage, as illustrated in Figure 1.

In the *DAR stage*, we use a sliding window technique with temporally overlapping frames to classify video clips of a fixed length into different action categories. This allows us to process long video sequences and identify the actions being performed within them. Our approach is designed to leverage both spatial and temporal information and effectively capture the spatiotemporal characteristics of the actions.

In the *Election stage*, we refine the preliminary results obtained from the action recognition module to arrive at a final prediction. This stage is critical for improving the performance of DAR, as it allows us to consolidate the information from different camera views and select the most reliable action candidates.

2.1. Problem Definition

Our goal is to accurately determine the start and end times and identify the specific actions performed by a driver in each video, using input from multiple camera angles. We represent the number of camera views as M, and a multi-view video as $\mathbf{v} = (\mathbf{x}_1, ..., \mathbf{x}_t, ..., \mathbf{x}_T)$, where $\mathbf{x}_t$ denotes a multi-view frame. Specifically, $\mathbf{x}_t$ is defined as:

$$\mathbf{x}_t = \left\{\mathbf{x}_{t,m} \in \mathbb{R}^{C\times H_m \times W_m}\right\}_{m=1}^{M}, \tag{1}$$

where H_m and W_m denote the height and width of the input image captured from view m, respectively, while C represents the number of color channels.

Let $\mathbf{y}$ represent the ground-truth set of actions performed by the driver in the video, and let $\mathcal{C}$ be the set of predefined action categories. Each element i in the ground-truth set can be expressed as $\mathbf{y}_i = (s_i, e_i, c_i)$, where s_i indicates the starting time, e_i corresponds to the ending time, and $c_i \in \mathcal{C}$ is the associated activity label. Let $\hat{\mathbf{y}}$ be the set of N predictions, where N is the cardinality of $\mathcal{C}$[1]. We evaluate the performance of our system using the average activity overlap score.

To compute this score, we need to find a bipartite matching between the ground-truth set $\mathbf{y}$ and the predicted set $\hat{\mathbf{y}}$, yielding a permutation of N elements $\sigma \in \mathfrak{S}_N$ with the

[1] Assuming that the driver performs each of the 16 different tasks once, in random order, as stated in the challenge statement.

highest overlap score:

$$\hat{\sigma} = \text{argmax}_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} os(\mathbf{y}_i, \hat{\mathbf{y}}_{\sigma(i)}), \tag{2}$$

where $os(\mathbf{y}_i, \hat{\mathbf{y}}_{\sigma(i)})$ is the pair-wise overlap score between the ground truth y_i and a prediction with index $\sigma(i)$. A match is counted if $\mathbf{y}_i$ and $\hat{\mathbf{y}}_{\sigma(i)}$ are of the same class ($c_i = \hat{c}_{\sigma(i)}$) and the start time $\hat{s}_{\sigma(i)}$ and end time $\hat{e}_{\sigma(i)}$ are within 10 seconds before or after the ground-truth activity's start time s_i and end time e_i, respectively. The overlap score between each matched pair of activities is calculated as the ratio of their time intersection to their time union, as defined below:

$$os(\mathbf{y}_i, \hat{\mathbf{y}}_{\sigma(i)}) = \frac{\max(\min(e_i, \hat{e}_{\sigma(i)}) - \max(s_i, \hat{s}_{\sigma(i)}), 0)}{\max(e_i, \hat{e}_{\sigma(i)}) - \min(s_i, \hat{s}_{\sigma(i)})}. \tag{3}$$

If a ground-truth activity has no match or a predicted activity has no ground-truth match, an overlap score of 0 is assigned. The final score is obtained by computing the average overlap score across all matched and unmatched activities.

2.2. Driver Action Recognition Stage

Accurate recognition of distracted driving behaviors demands a robust video classification backbone. While Transformers were initially developed for natural language processing tasks [28], recent progress has demonstrated their versatility beyond language tasks. For instance, ViT [5] have interpreted image patches as visual words, achieving competitive performance with convolutional neural network (CNN) counterparts [9,10]. Transformers are exceptional at modeling global information and long-range dependencies, making them suitable for analyzing video data.

Multiscale Vision Transformers (MViT) have further extended the power of ViT by introducing a pooling attention mechanism that generates a feature hierarchy with multiple stages, gradually reducing from high-resolution to low-resolution. MViT has achieved state-of-the-art performance in video tasks [6]. To leverage these advancements, we have employed the Multiscale Vision Transformer v2 (MViTv2) [12] as the backbone model in our system for DAR. To balance efficiency and performance, we have selected MViTv2-B (B stands for Base) as the backbone model.

The recognition module takes a fixed-length video clip as input at a time. To train the backbone model, we use a temporal data augmentation technique inspired by [12, 13]. Specifically, we extract video clips from the original video data $\mathbf{v}$ and the corresponding variable-length annotation set $\mathbf{y}$, and assign them with corresponding activity labels. We create our training set by taking the union of all video clips from different videos and annotation sets as follows:

$$\mathcal{D}_{\text{train}} = \bigcup_{k=1}^{N_{\text{video}}} \left\{(\mathbf{x}^k_{s_i}, \mathbf{x}^k_{s_i+1}, ..., \mathbf{x}^k_{e_i}), c^k_i\right\}_{i=1}^{N}, \tag{4}$$

where $\mathbf{x}^k_{s_i}, \mathbf{x}^k_{s_i+1}, ..., \mathbf{x}^k_{e_i}$ denote the frames in the video clip and c^k_i is the corresponding activity label from video $\mathbf{v}^k$ and N_{video} is the number of videos available for training. During this process, we discard empty segments (video clips without any annotations) to remove noisy information and ensure high-quality training data.

During training, we pass the data from the three camera views through a weight-sharing recognition network. For each video clip in the training set, we randomly sample a $S \times \tau$ frame segment that contains S frames with a temporal stride of τ, which forms a training batch. The sampling is performed independently for each camera view to ensure diverse training examples. We employ standard cross-entropy loss function to optimize the network parameters.

During inference, we adopt a sliding window approach with overlapping frames to generate predictions for each test video. Specifically, we use a window size of $S \times \tau$, which is the same as the input size of the action recognition backbone model, and slide the window across the entire video with a temporal stride of $S \times \tau/4$. For each window, we feed the corresponding frames from all three camera views through the weight-sharing recognition network and obtain a probability matrix for the action categories. We then average the scores across all frame positions of the entire video to obtain a probability matrix that captures the overall temporal dynamics of the video. Finally, we pass the resulting probability matrix to the election module to generate the final action time chunks.

2.3. Election Stage

To refine preliminary findings obtained from the action recognition module, we propose a novel algorithm called Election. The algorithm leverages a probability matrix $\mathbf{p} \in \mathbb{R}^{T \times |\mathcal{C}| \times M}$ as input, where T, $|\mathcal{C}|$, and M represent the video's duration, the count of pre-defined action categories, and the number of camera views, respectively. The proposed method has four steps.

Aggregation (AGG). In the first step, to capture information from various camera views, we apply a convolution operation to the input probability matrix using convolution kernels. Specifically, the operation is defined as:

$$\mathbf{p}'_{t,c} = \sum_{m=1}^{M} \omega_{c,m} \cdot \mathbf{p}_{t,c,m}, \tag{5}$$

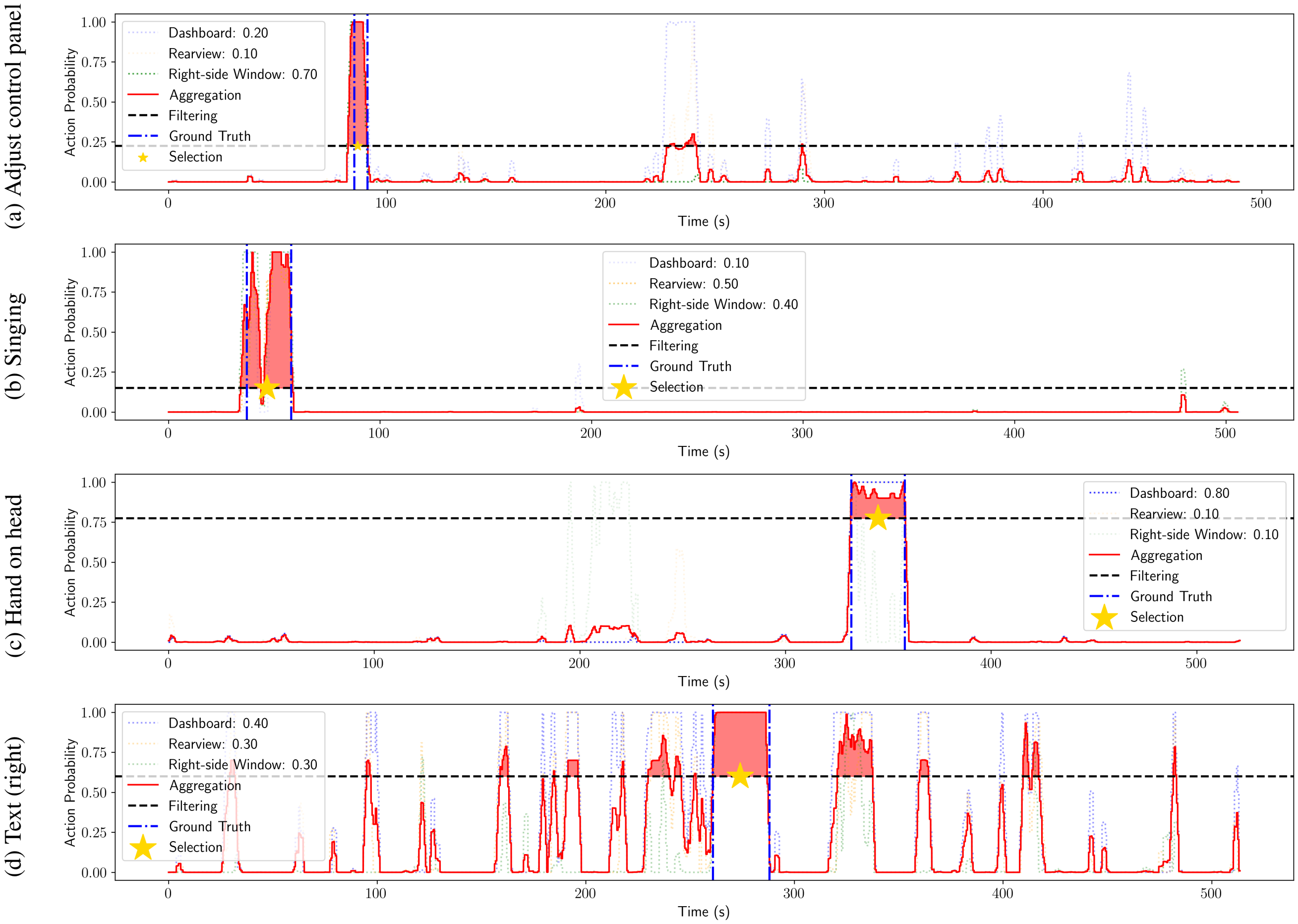

Figure 2. The figure visualizes how our proposed election algorithm consolidates preliminary findings from the action recognition module within our M^{2}DAR framework. The plot displays probability scores from the action recognition module of four action categories: (a) adjust control panel, (b) singing or dancing with music, (c) hand on head, and (d) text (right). The blue, yellow, and green dotted lines represent the outputs from the recognition module, with transparency indicating their respective weights in recognizing the action. The red line shows the probability score after the aggregation step of the election algorithm. The black dashed lines represent the probability thresholds, while the red regions between the black dashed line and the red line are the action candidates. The gold star indicates the outcome of the selection step.

where $\mathbf{p}' \in \mathbb{R}^{T\times|\mathcal{C}|}$ is the aggregated probability scores. To weight the information from each camera view m differently for each action category c, we define the convolution kernels $\omega \in \mathbb{R}^{|\mathcal{C}|\times M}$. The kernel weight $\omega_{c,m}$ is specifically set for each action category c and camera view m to integrate the complementary information from different camera views while focusing on the one containing the most discriminative information. This design is based on the observation that different perspectives have different effects on various behavior recognition, and different behaviors have different characteristics under different perspectives. For example, the action *talk to passenger at backseat* may be difficult to recognize from the dashboard view or the rearview view, but very clear from the right-side window view. Therefore, the convolution operation aims to enhance the quality of the probability scores by integrating the complementary information from different camera views while weighting the views based on their relevance for each specific action category.

Filtering (FLTR). In the second step, the system identifies initial action candidates by extracting continuous frames with probability scores that exceed a predefined threshold for each action category. These frames are considered as potential action segments that may contain the target actions. The threshold is set empirically based on the probability distribution on the validation data to ensure a balance between recall and precision. Frames with probabilities below this threshold are discarded as they are unlikely to represent a valid action. The resulting clips are

ID	Description	ID	Description
0	Forward Driving	8	Adjust control panel
1	Drinking	9	Pick up from floor (D)
2	Phone Call (R)	10	Pick up from floor (P)
3	Phone Call (L)	11	Talk to pax at the right
4	Eating	12	Talk to pax at backseat
5	Text (R)	13	Yawning
6	Text (L)	14	Hand on head
7	Reaching behind	15	Singing or dancing

Table 1. List of 16 driver actions defined in Track 3 of the AI City Challenge 2023: L, R, D, and P represent Left, Right, Driver, and Passenger, respectively.

considered as initial action candidates and are used as input for the subsequent candidate merging step.

Merging (MRG). In the third step, we merge clips that have a small temporal gap (e.g., less than 0.5 seconds) between them. This design is based on the observation that some driver actions may have a significant pause during their occurrence, which can result in two separate action segments. By merging these clips, we aim to eliminate the influence of those interruption to the final action localization results. The merging process is performed by iteratively comparing the temporal distance between each pair of adjacent action candidate clips, and merging them if the distance is less than the predefined gap threshold. This process is repeated until no further merging is possible.

Selection (SEL). After the merging step, we compute the average score of all merged candidates for each action category. If there are multiple candidates for an action category, we choose the one with the highest average score as the final action candidate. The algorithm outputs $|\mathcal{C}|$ final action candidates, one for each action category. We round the start and end times of the final action candidates to the nearest second and output them as the system's final prediction. This process ensures that the final prediction is based on a comprehensive evaluation of all the merged candidates and their scores, resulting in a more accurate and reliable prediction of the driver's actions.

3. Experiments

3.1. Dataset Description

Track 3 of the AI City Challenge 2023 [18] involves the analysis of synthetic naturalistic driver data captured from three different camera views positioned inside the vehicle: dashboard, right-side window, and rearview mirror, while drivers simulate driving scenarios. In addition, the drivers have three different natural driving appearances, including none, sunglasses, and hat.

Step				Overlap Score
SEL	FLTR	MRG	AGG	
✓	✗	✗	✗	0.4683
✓	✓	✗	✗	0.5347
✓	✓	✓	✗	0.5565
✓	✓	✓	✓	**0.5921**

Table 2. Ablation study comparing the effectiveness of individual steps in the M^{2}DAR system. SEL, FLTR, MRG, and AGG refer to the selection step, filtering step, merging step, and aggregation step, respectively. The scores shown are obtained by uploading the inference results to the evaluation server.

The dataset consists of 34 hours of videos captured from 35 drivers performing 16 distinct driving tasks defined in Tab. 1 [21]. Each video has a length of approximately 8 minutes, with a frame rate of 30 fps and a resolution of 1920×1080. The driver videos are divided into three subsets: A1, A2, and B. The A1 dataset is used for training and includes ground truth labels for start time, end time, and the types of distracted behaviors. The remaining two subsets, A2 and B, each containing videos from five drivers, are used for testing.

3.2. Implementation

We implemented the proposed system using PySlowFast [8], an open-source codebase for video understanding. We set a consistent input size of 448×448 for all three camera views, since their resolutions are identical. For training, we used all the data in the A1 subset for leaderboard submissions. We utilized a pretrained MViTv2-B model on Kinetics-700 [11], with a frame length of $S = 16$ and a sample rate of $\tau = 4$. We employed the AdamW [16] optimizer with a weight decay of 0.0001, and a cosine learning rate scheduler [15], with a base learning rate of 5×10^{-6}, a warm-up period of 30 epochs, and a total of 200 epochs. We conducted all training and inference processes on a NVIDIA A100 GPU.

3.3. Main Results

Our proposed framework achieved an overlap score (refer to Eq. (3)) of 0.5921 on the A2 test set. The proposed Election algorithm consists of four crucial steps: Aggregation (AGG), Filtering (FLTR), Merging (MRG), and Selection (SEL). To further validate the effectiveness of each step, we conducted an ablation study, wherein we modified one module at a time while maintaining the other modules unchanged.

In the absence of the aggregation step, we employed the baseline method, which directly averages the probability scores from the three camera views. If the filtering step was omitted, a uniform threshold was applied to all action categories to obtain action candidates. Without the merging step, the process proceeded directly to the selection step.

The ablation study results are summarized in Tab. 2, which presents the scores obtained after submitting the inference results to the evaluation server. As demonstrated in Tab. 2, incorporating all four proposed modules resulted in the highest score.

3.4. Election Visualization

We present a visualization of how the proposed election algorithm consolidates the preliminary findings from the action recognition module in our M^{2}DAR framework. The visualization is shown in Fig. 2, where the figures (from top to bottom) represent the probability score signals from the action recognition module on randomly selected videos from the validation set of four action categories: *adjust control panel*, *singing or dancing with music*, *hand on head*, and *text (right)*, respectively. The transparent blue, yellow, and green dot lines depict the outputs from the action recognition module, which are displayed transparently according to their weights in recognizing the particular action. The red line shows the probability score after the aggregation step of the election algorithm. The black dashed lines represent the probability thresholds. The red regions between the black dashed line and the red line are the action candidates, while the dash-dot lines represent the ground truth start and end times of the action. The gold star in the figure refers to the outcome of the selection step.

This visualization confirms the impact of different camera perspectives on recognizing different behaviors. For instance, comparing the first and third rows, we observe that the right-side window view plays a crucial role in recognizing the *adjust control panel* action (first row), but can introduce significant noise in recognizing the *hand on head* action (third row), which is resolved by the aggregation and filtering steps. Additionally, the second row of the *singing or dancing with music* action demonstrates a pause that affects the recognition, which is addressed by the merging step of the election algorithm. Finally, the last row of the *Text (Right)* action highlights how the selection step evaluates all merged candidates and their scores to ensure a comprehensive final prediction.

This visualization illustrates how our proposed algorithm effectively improves the accuracy of action recognition in multi-view videos by leveraging the complementary information from different camera views and selecting the most reliable action candidates. The integration of convolution kernels tailored for specific action categories and the merging of overlapping candidates overcomes the limitations of individual camera views and effectively captures the temporal characteristics of the actions, resulting in improved action recognition and localization performance.

4. Related Work

Driver Action Recognition. DAR has received significant attention from researchers due to its potential to effectively monitor driver distraction and evaluate risky driving behaviors, thereby reducing the risk and severity of traffic accidents [1, 13, 17, 20, 27, 29–31]. In the 6th AI City Challenge [19], Stargazer utilized Transformer to exploit rich temporal features about human behavioral information with a simple action temporal localization framework [13], and PAND [31] proposed a strategy that uses a multi-branch network and a post-processing method for selecting and correcting temporal ranges. Additionally, the heterogeneity of driver behavior is also a major challenge for the generalization ability of the task [30]. ViT-DD proposed a semi-supervised framework to make use of driver emotion as an additional clue in recognizing driver behaviors [17]. Our solution for the AI City Challenge utilizes a weight-sharing approach to effectively leverage contextual information across different views, leading to better performance in multi-view scenarios.

Temporal Action Localization. TAL is a challenging task that involves accurately classifying and localizing specific activities within an untrimmed video sequence. There are two main categories of methodologies used to address this task: bi-stage methods [3, 4, 25] and uni-stage methods [2,14,24]. Bi-stage methods are similar to two-stage object detection approaches [3, 22], where frame or segment-level classification is performed initially, followed by a post-processing stage to consolidate the preliminary findings into a final prediction output. In contrast, uni-stage methods integrate both the temporal localization and activity categorization aspects into a single process, but the resulting complexity can lead to increased training and inference times, difficulty in scaling to handle large-scale video datasets, and meticulous fine-tuning of hyperparameters for optimal performance [7]. Our solution for the AI City Challenge adopts a bi-stage method by dividing the multi-view video into clips and processing them using Transformer with weight-sharing, achieving high accuracy while also reducing complexity and enabling efficient processing of large-scale video datasets.

5. Conclusion

In this paper, we have presented a multi-view multi-scale framework for DAR and localization in untrimmed videos, which addresses the challenges of detecting distracted driving behaviors in a naturalistic setting. The proposed M^{2}DAR framework employs a weight-sharing recognition network and an election module consisting of four steps: aggregation, filtering, merging, and selection. Our system has achieved an overlap score of 0.5921 on the A2 test set of the AI City Challenge 2023 Track 3. Our proposed framework has the potential to aid in the development of more effective driver monitoring systems and improve road safety.

References

[1] Munirah Alyahya, Shahad Alghannam, and Taghreed Alhussan. Temporal Driver Action Localization Using Action Classification Methods. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3319–3326, 2022. 6

[2] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association and Society for Pattern Recognition, 2017. Accepted: 2020-06-09T13:34:00Z. 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 Computer Vision and Pattern Recognition*, pages 1130–1139, 2018. 6

[4] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S. Davis, and Yan Qiu Chen. Temporal Context Network for Activity Localization in Videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017. 6

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 3

[6] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 3

[7] Junyu Gao, Mengyuan Chen, and Changsheng Xu. Fine-Grained Temporal Contrastive Learning for Weakly-Supervised Temporal Action Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19999–20009, 2022. 6

[8] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast, 2020. 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[10] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, San Diego, Apr. 2015. Computational and Biological Learning Society. arXiv:1409.1556 [cs]. 3

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset, May 2017. arXiv:1705.06950 [cs]. 5

[12] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MViTv2: Improved Multiscale Vision Transformers for Classification and Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 3

[13] Junwei Liang, He Zhu, Enwei Zhang, and Jun Zhang. Stargazer: A Transformer-Based Driver Action Detection System for Intelligent Transportation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3160–3167. IEEE, 2022. 3, 6

[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. 6

[15] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*, 2017. 5

[16] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. 5

[17] Yunsheng Ma and Ziran Wang. ViT-DD: Multi-Task Vision Transformer for Semi-Supervised Driver Distraction Detection. In *IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 2023. arXiv:2209.09178 [cs]. 6

[18] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023. 1, 5

[19] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Alice Li, Shangru Li, and Rama Chellappa. The 6th AI City Challenge. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3346–3355. IEEE Computer Society, June 2022. 1, 6

[20] Chuong Nguyen, Ngoc Nguyen, Su Huynh, Vinh Nguyen, and Son Nguyen. Learning Generalized Feature for Temporal Action Detection: Application for Natural Driving Action Recognition Challenge. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3249–3256, 2022. 6

[21] Mohammed Shaiqur Rahman, Jiyang Wang, Senem Velipasalar Gursoy, David Anastasiu, Shuo Wang, and Anuj Sharma. Synthetic Distracted Driving (SynDD2) dataset for analyzing distracted behaviors and various gaze zones of a driver, 2022. arXiv:2204.08096 [cs]. 5

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 6

[23] SAE On-Road Automated Vehicle Standards Committee and others. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE International: Warrendale, PA, USA*, 2018. 1

[24] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5734–5743, 2017. 6

[25] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal Action Localization in Untrimmed Videos via Multi-Stage CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016. 6

[26] Timothy Stewart. Overview of Motor Vehicle Crashes in 2020. Technical report, National Highway Traffic Safety Administration, 2022. 1

[27] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3168–3173, 2022. 6

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[29] Arpita Vats and David C. Anastasiu. Key Point-Based Driver Activity Recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3274–3281, 2022. 6

[30] Liangqi Yuan, Lu Su, and Ziran Wang. Federated transfer-ordered-personalized learning for driver monitoring application. *arXiv preprint arXiv:2301.04829*, 2023. 6

[31] Hangyue Zhao, Yuchao Xiao, and Yanyun Zhao. PAND: Precise Action Recognition on Naturalistic Driving. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3291–3299, 2022. 6