Bidirectional Posture-Appearance Interaction Network for Driver Behavior Recognition

Mingkui Tan, *Member, IEEE*, Gengqin Ni, Xu Liu, Shiliang Zhang, Xiangmiao Wu, Yaowei Wang, *Member, IEEE*, and Runhao Zeng

***Abstract*— Driver behavior recognition has become one of the most important tasks for intelligent vehicles. This task, however, is very challenging since the background contents in real-world driving scenarios are often very complex. More critically, the difference between driving behaviors is often very minor, making it extremely difficult to distinguish them. Existing methods often rely only on RGB frames (or skeleton data), which may fail to capture the minor differences between behaviors and appearance information of objects simultaneously and thus fail to achieve promising performance. To address the above issues, in this paper, we propose a bidirectional posture-appearance interaction network (BPAI-Net), which simultaneously considers RGB frames and skeleton (*i.e.*, posture) data for driver behavior recognition. Specifically, we propose a posture-guided convolutional neural network (PG-CNN) and an appearance-guided graph convolutional network (AG-GCN) to extract appearance and posture features, respectively. To exploit the complementary information between appearance and posture, we use the appearance features from PG-CNN for guiding AG-GCN to exploit the contextual information (*e.g.*, nearby objects) to enhance posture features. Then, we use the enhanced posture features from AG-GCN to help PG-CNN focus on critical local areas of video frames that are related to driver behaviors. In this sense, we are able to use the interaction between two modalities to extract more discriminative features and thus improve the recognition accuracy. Experimental results on Drive&Act dataset show that our method outperforms state-of-the-art methods by a large margin (67.83% vs. 63.64%). Furthermore, we collect a bus driver behavior recognition dataset and yield consistent performance gain against baseline methods, demonstrating the effectiveness of our method in real-world applications. The source code and trained models are available at github.com/SCUT-AILab/BPAI-Net/.**

***Index Terms*— Driver behavior recognition, multi-modal learning, attention mechanism, graph convolutional networks.**

Manuscript received 27 April 2021; revised 30 September 2021; accepted 5 October 2021. Date of publication 3 November 2021; date of current version 9 August 2022. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62072190, in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B010155002, in part by the Ministry of Science and Technology Foundation Project 2020AAA0106900, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X183, and in part by the Chinese Association for Artificial Intelligence (CAAI) MindSpore Open Fund. The Associate Editor for this article was J. Blum. *(Mingkui Tan and Gengqin Ni contributed equally to this work.) (Corresponding authors: Yaowei Wang; Runhao Zeng.)*

Mingkui Tan, Gengqin Ni, and Xu Liu are with the School of Software Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: mingkuitan@gmail.com; gengqinni@gmail.com; seqdmy@mail.scut.edu.cn).

Shiliang Zhang is with the Department of Computer Science, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: slzhang.jdl@pku.edu.cn).

Xiangmiao Wu is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: xmwu@scut.edu.cn).

Yaowei Wang is with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: wangyw@pcl.ac.cn).

Runhao Zeng is with the College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: runhaozeng.cs@gmail.com).

Digital Object Identifier 10.1109/TITS.2021.3123127

I. Introduction

DRIVER behaviors and decisions are the main factors affecting driving safety. According to the World Health Organization's report [1], more than 1.35 million people die in road traffic accidents every year mainly due to the distraction (*e.g.*, drinking, using mobile phone) of drivers. However, the accident rate can be reduced by using a precise driver behavior monitoring system [2]. In this sense, driver behavior recognition, which aims to recognize the distraction of drivers, has been becoming an important research topic in intelligent vehicles.

Different from general action recognition tasks that focus on recognizing daily behaviors or sports events, driver behavior recognition (DBR) aims at understanding behaviors happening in transportation vehicles. This task is very challenging due to the following reasons: **First**, DBR is a fine-grained action recognition task since the difference between different driver behaviors is very minor and the crucial cue for recognizing each driver behavior may only correspond to some local areas of video frames. For instance, the difference between *fetching an object* and *placing an object* lies in the minor hand (or arm) movements. To distinguish such actions, we need to focus on the visual information among the hands or arms. **Second**, an important factor for recognizing driver behaviors is understanding human-object interactions, which, however, is very difficult. For example, *drinking* and *eating* share similar postures but correspond to different objects (*e.g.*, bottle and food), making it very difficult to distinguish them. **Third**, unlike general action recognition videos which are highly related to the scenes, driver behaviors videos have similar scenes (*i.e.*, the driving area of a vehicle), making it extremely difficult to distinguish driver behaviors relying on scene information.

Conventional driver behavior recognition methods often employ a two-step pipeline, in which the hand-crafted features are extracted from the raw data and classifiers are applied on the hand-crafted features. In the past few years, with the great

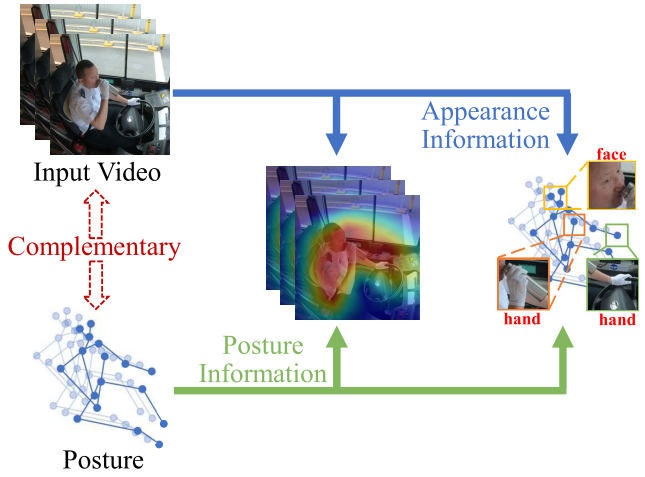


Fig. 1. Existing methods mainly focus on RGB frames (or posture) for extracting driver behavior features. We argue that RGB frames and posture are complementary and simultaneously considering them may help improve recognition accuracy. On the one hand, posture guides appearance feature extraction models to pay more attention to the local regions that are related to driver behaviors. On the other hand, RGB frames provide appearance information to help posture feature extraction models exploit the object contextual information.

success of deep learning models [3]–[6], particularly the convolutional neural networks (CNNs) in computer vision, a variety of deep learning-based researches towards driver behaviors recognition have been done [7]–[12]. In general, these methods can be grouped into two paradigms: appearance-based [7], [9], [10] and posture-based [11], [12]. As for the appearance-based methods, they often adopt some action recognition models (*e.g.*, I3D [13], ResNet [6]), which are designed for general daily actions and sports events classification, to extract appearance features. Although these methods are able to exploit the scene and object information, they often fail to capture the minor difference between driver behaviors (*e.g.*, *fetching an object* and *placing an object*) due to the interference of complex video contents. In terms of posture-based methods, they focus on modeling the dynamics of body key points for recognizing behaviors. They often represent the coordinates of key points as 2D feature maps and use graph convolutional networks (GCNs) to extract posture features. Due to the lack of object information, these methods fail to be aware of the interactions between driver and objects, leading to an inferior performance on distinguishing the behaviors that have similar postures but with different objects, such as *eating* and *drinking*.

We contend that posture and appearance information can be complementary and considering two modalities simultaneously is important for driver behavior recognition (as shown in Figure 1). On the one hand, using posture information performs better at capturing the body dynamics for recognizing fine-grained actions, while the appearance information of objects provides cues for realizing the interaction between objects and drivers. On the other hand, the posture information (*e.g.*, coordinates of body key points) may guide appearance feature extractors to focus on the regions that are highly related to actions and alleviate the interference from irrelevant regions.

In this paper, we propose a bidirectional posture-appearance interaction network (BPAI-Net), where a posture-guided convolutional neural network (PG-CNN) and an appearance-guided graph convolutional network (AG-GCN) is devised to extract appearance and posture features, respectively. Instead of extracting appearance and posture features individually, we let two modalities interact with each other in a **bidirectional** manner during the feature extraction process: **1)** we use the appearance features from PG-CNN for guiding AG-GCN to exploit the contextual object information corresponding to each key point and enhance posture features; **2)** we use the enhanced posture features from AG-GCN to help PG-CNN focus on critical local areas of video frames that are related to driver behaviors in turn. After the above-mentioned bidirectional interactions between appearance and posture, we obtain the enhanced features to classify driver behaviors more precisely. Experimental results show that our method significantly outperforms state-of-the-art methods on Drive&Act dataset. Furthermore, we collect a bus driver behavior recognition dataset and yield consistent performance gain against other compared methods, demonstrating the effectiveness of our method in real-world applications.

To sum up, our contributions are as follows:

- To the best of our knowledge, we are the first to exploit bidirectional posture-appearance interactions for driver behavior recognition in videos.
- We propose a Bidirectional Posture-Appearance Interaction Network (BPAI-Net) to exploit the complementary information between posture and appearance features for driver behavior recognition.
- Experimental results show that BPAI-Net outperforms state-of-the-art methods by a large margin (67.28% vs. 63.64%) on the large-scale Drive&Act dataset.
- We collect and annotate a novel bus driver behavior dataset in a real-world driving scenario.

II. Related Works

A. Driver Behavior Recognition

Driver behavior recognition has attracted much attention in recent years. Previous works conduct driver behavior recognition mainly based on hand-crafted features [14]–[18]. With the development of deep learning, convolutional neural network-based methods [7]–[9] have been applied for this task and achieved promising performance. Existing deep learning approaches for driver behavior recognition basically focus on three modalities of data: RGB frames, optical flow or driver skeletons/key points.

1) Single Modal: Most existing methods rely on single modal data for driver behavior recognition task [7], [11], [13]. Classic models (*e.g.*, VGG-16 [3], AlexNet [4] and etc.) are modified and equipped with regularization techniques [7] or HOG features [19] to extract driver behavior features from RGB frames. Leekha *et al.* [20] design a lightweight network to recognize behaviors based on frames processed by GrabCut. To better leverage the temporal relationship between frames, some researchers use recurrent neural networks [21] or 3D convolutional neural networks [13] to extract driver behavior

features relying on optical flow. Li *et al.* [11] use skeletons to represent the driver's posture and recognize the behavior by using a graph convolutional network (GCN). However, these methods only consider single modal data, which neglects the complementary information between multiple modalities.

2) Multi-Modal: To enhance the driver behavior features, some researchers attempted to use multi-modal data (*e.g.*, RGB frames and optical flows [22]). Kose *et al.* [23] use an early fusion strategy by concatenating RGB and optical flow along the channel dimension as input. Moslemi *et al.* [24]) propose a late fusion strategy, in which the RGB and optical flow features are first extracted separately and then fused before feeding to the classifier. Since the optical flow may be sensitive to camera motion, Behera *et al.* [8] propose to leverage a multi-stream network to separately process RGB and skeletons. However, these works neglect the interaction between modalities, which is crucial for capturing driver behavior features. Different from previous methods, in this paper, we propose to exploit the complementary information between RGB and skeletons for better feature representation, which significantly outperforms the considered methods.

B. Driver Behavior Datasets

Due to the popularity of driver behavior recognition in the computer vision community, many datasets have been proposed in cars [25]–[31] or driving simulators [9], [32]. Previous datasets mainly provide RGB images for recognition, such as StateFarm's Distracted Driver Detection competition on Kaggle [33]. Recently, Martin *et al.* [29] design a new domain-specific Drive&Act benchmark with RGB frames, near-infrared images and depth data for fine-grained recognition of driver behavior. To improve the capability of Drive&Act, Reiß *et al.* [34] re-separate semantic attributes to each action. Martin *et al.* [12] annotate the bounding boxes of the dynamics objects contained in Drive&Act. Different from previous datasets that focus on car drivers, we collect a new dataset from bus drivers since the bus is an important vehicle for public transport. More importantly, our dataset consists of behaviors that are related to complex interactions between the driver and passengers (*e.g.*, *passenger pulling the driver*). To the best of our knowledge, this is the first dataset for the ***BUS*** driver behavior recognition task.

III. Proposed Method

A. Problem Definition

Let $V = \{I_t \in \mathbb{R}^{H\times W\times 3}\}_{t=1}^{T}$ be a video with T frames, where I_t denotes the frame at time slot t with height H and width W. Given a video V, the goal of driver behavior recognition is to classify V into a predefined set of C driver behavior classes, such as *drinking* and *eating*. The key challenge lies in that driver behaviors from different categories only have minor differences. More critically, the crucial cue for recognizing each driver behavior may only correspond to some local areas of video frames and the slight movements of body parts, making it difficult to correctly distinguish driver behaviors.

Existing methods attempt to recognize driver behaviors by inputting the video clip V to a clip-classifier (*i.e.*, some action classification methods [6], [35], [36]), which, however, may fail to focus on the visual regions that are informative and relevant to driver behaviors. Other methods try to model the body dynamics by exploiting neural networks on posture (*i.e.*, body key points). They first extract key points $\mathbf{P} \in \mathbb{R}^{T\times N\times 2}$ from V by calculating $\mathbf{P} = g(V)$, where g denotes some certain pose estimation methods (*e.g.*, OpenPose [37]) and N is the number of key points for each driver in a single frame. Then, a graph convolutional network is applied on $\mathbf{P}$ to classify driver behaviors. Such approaches fail to be aware of the interactions between drivers and objects, thus leading to inferior results.

B. General Scheme

We focus on solving the problem that existing approaches attempt to tackle the driver behavior recognition task relying on only appearance (or posture) individually while neglecting their complementary information, which, however, is important for driver behavior recognition. In this paper, our goal is to consider both posture and appearance information and exploit the interactions between them in a driving behavior recognition system. The intuition is that posture is more suitable for modeling the body dynamics and the locations of key points guide us to focus on the regions that are related to the driver behaviors. On the other hand, appearance is capable of capturing the objects that interact with drivers. Such object information can be used to learn better action representations through posture.

In this paper, we propose a bidirectional posture-appearance interaction network (BPAI-Net), which consists of an appearance feature extractor (*i.e.*, Posture-Guided CNN) and a posture feature extractor (*i.e.*, Appearance-Guided GCN). **First**, given a video V, we obtain the posture information $\mathbf{P}$ (*i.e.*, key points) of drivers through some certain posture estimation methods (*e.g.*, OpenPose [37]) and obtain video frames as the appearance input. **Second**, we input the video frames to the Posture-Guided CNN to obtain the appearance hidden feature $\mathbf{F}_a$, and forward the key points to Appearance-Guided GCN to obtain posture hidden feature $\mathbf{G}_p$. **Third**, with the help of object information provided by $\mathbf{F}_a$, the key point-based RoI module M_{krm} takes $\mathbf{G}_p$, $\mathbf{F}_a$ and key point coordinates as input and obtains the appearance-aware posture feature by computing

$$\mathbf{G}'_p = M_{krm}(\mathbf{P}, \mathbf{G}_p, \mathbf{F}_a). \tag{1}$$

The enhanced posture features $\mathbf{G}'_p$ are further processed by GCN, resulting in $\mathbf{G}''_p$. **Fourth**, the posture feature $\mathbf{G}''_p$ from the Appearance-Guided GCN is fed back to the Posture-Guided CNN. Such posture features are inputted to posture-guided attention module M_{pam} to guide the appearance feature extractor focus on the critical areas that are related to driver behaviors. PAM obtains the posture-guided appearance features $\mathbf{F}'_b$ by computing

$$\mathbf{F}'_b = M_{pam}(\mathbf{G}''_p, \mathbf{F}_b). \tag{2}$$

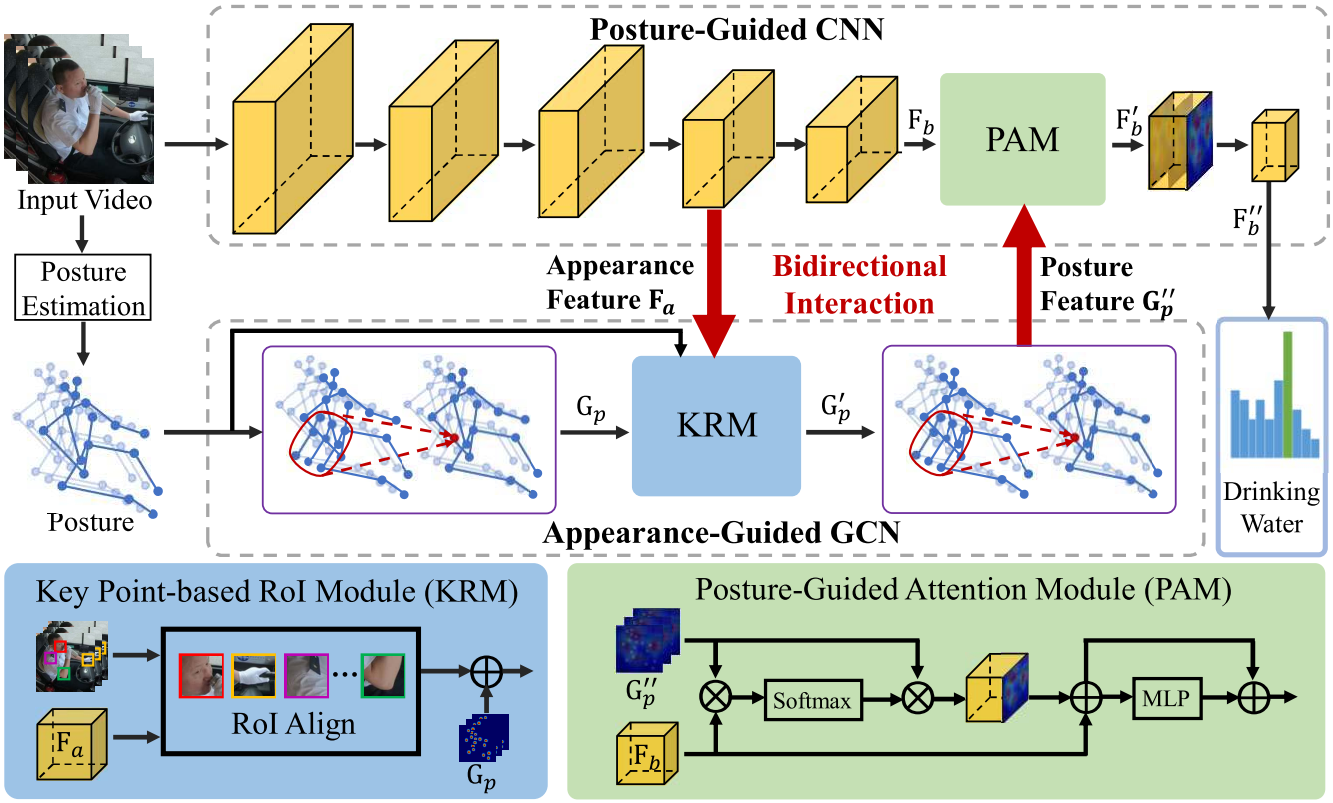


Fig. 2. Schematic of our BPAI-Net. Given a video, we use a posture-guided convolutional neural network (PG-CNN) and an appearance-guided graph convolutional network (AG-GCN) to extract appearance features and posture features, respectively. We propose a KRM module to make AG-GCN perceive the appearance information and enhance posture features. Then, we use a PAM module to exploit the enhanced posture features to guide PG-CNN focus on critical local areas for classifying driver behaviors.

Algorithm 1 Algorithmic Flow of Our BPAI-Net

Require: Video $V = \{I_t \in \mathbb{R}^{H\times W\times 3}\}_{t=1}^{T}$, key points $\mathbf{P} \in \mathbb{R}^{T\times N\times 2}$.

obtain bounding box b using Eqn. (4).
while not converges **do**
  // PG-CNN and AG-GCN
  predict appearance and posture features $\mathbf{F}_a$ and $\mathbf{G}_p$.
  // enhance posture features with appearance (M_{krm})
  obtain enhanced posture feature $\mathbf{G}'_p$ via Eqns. (5)-(7).
  obtain appearance-guided feature $\mathbf{G}''_p$.
  // enhance appearance features with posture (M_{pam})
  obtain enhanced feature $\mathbf{F}'_b$ using Eqns. (8)-(10).
  obtain posture-guided feature $\mathbf{F}''_b$.
  // classifier
  predict driver behavior categories relying on $\mathbf{F}''_b$.
end while

Last, the features $\mathbf{F}''_b$ outputted by posture-guided CNN are used to classify driver behaviors. For better readability, Algorithm 1 depicts the algorithmic flow of our method.

In the following sections, we aim to answer two questions: (1) how to learn more informative posture features with the help of object appearance (Section III-C); (2) how to learn discriminative appearance features with the help of posture information and facilitate driver behavior recognition (Section III-D).

C. Enhance Posture Feature Using Appearance Information

We seek to extract discriminative posture features from the input skeleton data and facilitate downstream driver behavior recognition tasks. To this end, we propose an appearance-guided graph convolutional network (AG-GCN) with GCN being the feature extraction backbone and a key point-based RoI module (KRM) to exploit the appearance information from video frames. The schematic depiction of AG-GCN is shown in Figure 2.

1) Graph Convolutional Network (GCN): The key points of a driver can be represented by a time series of human joint locations in the form of 2D or 3D coordinates. And the structure of the human skeleton can be naturally regarded as a graph, where nodes correspond to joints and edges correspond to bones. Thus, it is straightforward to apply a GCN to capture the motion patterns of key points.

Without loss of generality, we assume the body key points have been obtained beforehand by some pose estimation methods (*e.g.*, the OpenPose method [37]). Given the sequences of body joints, we construct a GCN and the input is the joint coordinate vectors of each node. Considering that the driver behavior recognition task is based on the video that contains spatial and temporal dimensions, we use ST-GCN [38] to

model the motion pattern of key points. ST-GCN consists of a series of ST-GCN blocks and the formulation of each ST-GCN block is as follows

$$\begin{aligned} \mathbf{X}_{out} &= \hat{\mathbf{A}}\mathbf{X}_{in}\mathbf{W}, \\ \hat{\mathbf{A}} &= \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}, \\ \tilde{\mathbf{A}} &= \mathbf{A} + \mathbf{I}_N \end{aligned} \tag{3}$$

where $\mathbf{X}_{in} \in \mathbb{R}^{N\times d_{in}}$ is the input feature and $\mathbf{X}_{out} \in \mathbb{R}^{N\times d_{out}}$ is the output feature. d_{in} and d_{out} are feature channels. $\mathbf{W} \in \mathbb{R}^{N\times d_{out}}$ is the parameter matrix to be learned; $\mathbf{A}$ is the adjacency matrix; $\mathbf{D}$ and $\mathbf{I}_N$ denote the degree matrix and the identity matrix, respectively. The last ST-GCN block is connected to a fully connected layer to obtain the classification results. We refer the readers to [38] for more details.

The above ST-GCN is able to capture the minor difference from the view of the changes of key point locations. However, for certain driver behaviors with similar postures (*e.g.*, *drinking* and *eating*), their behaviors differ only in the objects (*e.g.*, bottle and food) interacted with the driver. In this sense, directly using ST-GCN may not be able to distinguish such similar behaviors due to the lack of appearance information. To resolve this problem, we devise a key point-based RoI module (KRM) to provide appearance guidance for ST-GCN.

2) Key Point-Based RoI Module: We propose to enhance the posture features by additionally considering the local regions that are centered at each key point. The intuition is that the local region around each key point may contain the objects that are important for driver behaviors recognition. For example, the local region of the hand provides indicative object information (*i.e.*, bottle or hamburger) for distinguishing drinking and eating, while only using ST-GCN may fail to classify these two behaviors correctly.

Given a key point coordinates (x, y), we generate a bounding box $b = (x_1, y_1, x_2, y_2)$ with size $d \times d$ by computing

$$\begin{aligned} x_1 &= x - \frac{d}{2}, \quad y_1 = y - \frac{d}{2}, \\ x_2 &= x + \frac{d}{2}, \quad y_2 = y + \frac{d}{2}. \end{aligned} \tag{4}$$

Then, we use the generated bounding box to extract a patch feature $\mathbf{X}_m$ from a feature map $\mathbf{F}_a$, which can be the hidden features from any layer of the posture-guided CNN (will be introduced in Section III-D). Formally, we use the RoI Align [39] technique f to obtain

$$\mathbf{X}_m = f(\mathbf{F}_a, b), \tag{5}$$

where $\mathbf{X}_m \in \mathbb{R}^{C\times T\times N\times H\times W}$ and C is the number of channels. Following Mask R-CNN [39], we further add an average pooling layer h to obtain a feature vector to represent the object information for each key point by computing

$$\mathbf{X}_p = h(\mathbf{X}_m), \tag{6}$$

where $\mathbf{X}_p \in \mathbb{R}^{C\times T\times N\times 1\times 1}$. To fuse the posture feature $\mathbf{G}_p$ from a certain layer of AG-GCN and the object feature $\mathbf{X}_p$, we first feed them into two multi-layer perceptrons (MLPs) respectively and then sum the outputs of MLPs, resulting in appearance-aware posture feature $\mathbf{G}'_p$ by computing

$$\mathbf{G}'_p = \mathrm{MLP}_1(\mathbf{X}_p) + \mathrm{MLP}_2(\mathbf{G}_p), \tag{7}$$

where MLP_1 and MLP_2 are implemented as fully-connected layers. The output feature $\mathbf{G}'_p$ contains the posture information and appearance information to make each node perceive the appearance information around it. Then, we take $\mathbf{G}'_p$ as the input of ST-GCN blocks behind $\mathbf{G}_p$ by using Eq. (3). With the guidance of appearance information, AG-GCN produces the appearance-guided posture feature $\mathbf{G}''_p \in \mathbb{R}^{T\times N\times C}$.

D. Enhance Appearance Feature Using Posture Information

Given a set of video frames, we seek to capture the appearance information of driver behaviors and extract discriminative features. We propose a posture-guided convolutional neural network (PG-CNN), consisting of a CNN and a posture-guided attention module (PAM) M_{pam} to exploit the posture information. We use TSM [35] as the backbone of CNN to extract the appearance features from video frames. It is worth noting that our method is general and compatible with different backbones (will be discussed in Section V).

However, directly using TSM to extract features may not be able to achieve promising results due to the interference of background. The background in the video is often very complex in real-world driving scenarios, making the driver behavior features extremely hard to extract. To tackle this issue, we introduce a posture-guided attention module (PAM) to consider the posture features from AG-GCN and guide TSM to extract the discriminative features.

Posture-Guided Attention Module: Given the posture feature $\mathbf{G}''_p$ from AG-GCN and the appearance feature $\mathbf{F}_b \in \mathbb{R}^{T\times C\times HW}$, we compute the posture-guided attention weight $\mathbf{W}_{att}$ as follows:

$$\mathbf{W}_{att} = \mathrm{Softmax}(\mathbf{G}''_p\mathbf{F}_b), \tag{8}$$

where $\mathbf{W}_{att} \in \mathbb{R}^{T\times N\times HW}$. The softmax function is conducted along the spatial dimension (*i.e.*, HW). In this way, the summation of the attention weights between the n-th key point and all the spatial patches is 1, *i.e.*, $\sum_{l=1}^{HW} W_{att_{n,l}} = 1$. The attention weight $\mathbf{W}_{att}$ automatically learns to represent the relationships between each spatial patch feature of $\mathbf{F}_b$ and each key point feature of $\mathbf{G}''_p$. Such attention weights guide the appearance feature to aggregate posture features from each key point. Relying on $\mathbf{W}_{att}$, we are able to extract more discriminative features that are related to driver behaviors. To do so, we apply the attention weight $\mathbf{W}_{att}$ on the enhanced posture features $\mathbf{G}''_p$ and add a skip connection to obtain the enhanced appearance features by computing

$$\mathbf{X}_{att} = \mathbf{G}''^{\top}_p\mathbf{W}_{att} + \mathbf{F}_b, \tag{9}$$

then $\mathbf{X}_{att} \in \mathbb{R}^{T\times C\times HW}$ will be further processed by an MLP with a skip connection to produce features $\mathbf{F}'_b$:

$$\mathbf{F}'_b = \mathrm{MLP}(\mathbf{X}_{att}) + \mathbf{X}_{att}, \tag{10}$$

where the MLP is implemented by two full-connected layers with a ReLU activation function. Lastly, we input $\mathbf{F}'_b \in \mathbb{R}^{T\times C\times HW}$ to several convolutional layers and obtain the enhanced appearance feature $\mathbf{F}''_b$, which is further processed by an average pooling layer and a fully-connected layer to produce the classification result.

TABLE I

COMPARISONS OF PUBLICLY AVAILABLE DRIVER BEHAVIOR RECOGNITION DATASETS (R/D/I DENOTES RGB/DEPTH/INFRARED)

Dataset	Year	#Drivers	Modalities	#Classes	Temporal annotation	Scenarios
CVRR-Hands [43]	2013	8	R/D	19	Yes	Car
NTHU-DDD [44]	2017	36	R/I	8	NO	Simulator
AUC-DD [45]	2019	44	R	10	NO	Car
Drive&Act [29]	2019	15	R/D/I	83	Yes	Car
DMD [30]	2020	37	R/D/I	93	Yes	Car / Simulator
DAD [31]	2021	31	R/D/I	24	Yes	Simulator
PCL-BDB(ours)	2021	**55**	R/D/I	40	Yes	**Bus**

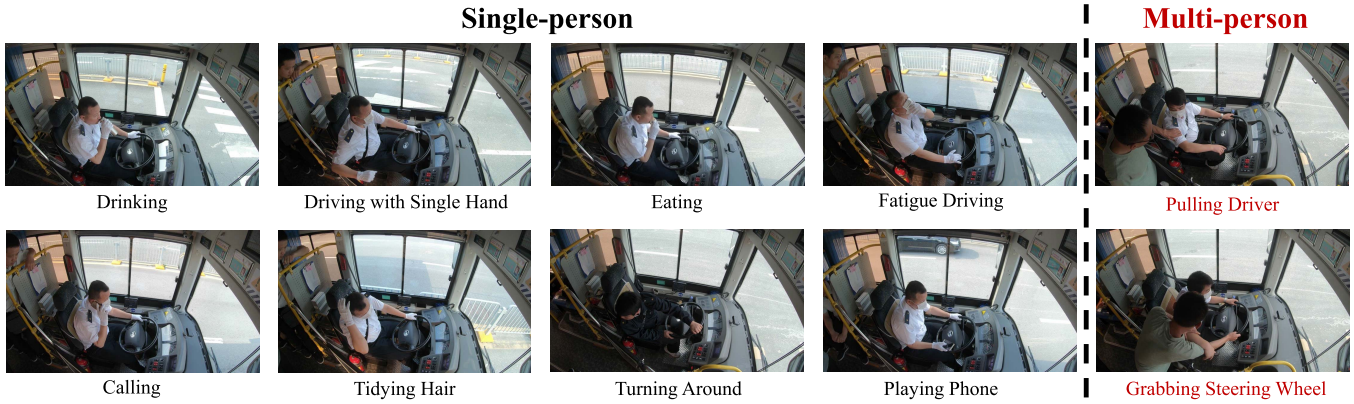


Fig. 3. Examples of 10 abnormal classes on our PCL-BDB dataset. We include 8 single-person abnormal action and 2 multi-person abnormal action (*i.e.*, *pulling driver* and *grabbing steering wheel*). For brevity, we do not show the same action with different hands.

E. Loss Function

To train our BPAI-Net, we follow other driver behavior recognition methods [10]–[12] to use cross-entropy (CE) loss by computing

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_a = -\sum_{i=1}^{C} \mathbb{1}_{gt}^{i} \log \mathbf{p}_{i,p} - \sum_{i=1}^{C} \mathbb{1}_{gt}^{i} \log \mathbf{p}_{i,a}, \quad (11)$$

where C is the number of driver behavior categories. $\mathbb{1}_{gt}^{i}$ is the indicator function, being 1 if category i is the ground truth label and 0 otherwise. $\mathbf{p}_{i,a}$ and $\mathbf{p}_{i,p}$ is the probability of the i-th category outputted by AG-GCN and PG-CNN, respectively. Here, we follow [40]–[42] to use the outputs of two branches (*i.e.*, PG-CNN and AG-GCN) to compute the losses (*i.e.*, $\mathcal{L}_p$ and $\mathcal{L}_a$) simultaneously.

IV. ABNORMAL BUS DRIVER BEHAVIOR DATASET

Recognizing the behavior of bus drivers plays an increasingly important role in public safety. However, it is difficult to obtain a reliable bus driver behavior recognition system due to the following reasons: 1) existing driver behavior datasets are mainly collected in cars or driving simulators [29]–[31]. In this sense, it is difficult to train a well-performed recognition model for bus drivers when only the car driver behavior datasets are available since the scenarios are different. 2) existing datasets are mainly collected in a simple and static environment, *e.g.*, the static driving simulators or cars. However, in a real-world scenario, the driving environment may be more complex due to the crowded people on the bus. For example, passengers will appear in the camera when getting on and off the bus, which may influence the prediction of the recognition models. Moreover, the behavior of passengers may have a great impact on the drivers, *e.g.*, disturbing the bus driver. Such behaviors have not been considered in existing datasets. To address the above issues, we propose to collect an abnormal bus driver behavior dataset with the Peng Cheng Laboratory, resulting in a **P**eng **C**heng **L**aboratory-**B**us **D**river **B**ehavior (PCL-BDB) dataset. Table I shows comparisons between publicly available driver behavior recognition datasets.

A. Data Collection Settings

We collect driver behavior data on four different types of buses. We have 55 participants, including 15 female drivers and 40 male drivers. To ensure the diversity of our dataset, we select the participants with different body heights, weights and different driving styles, wearing different uniforms, masks or sunglasses. As shown in Figures 3 and 5, each driver is instructed to perform 10 abnormal and 1 normal driver behaviors, including Normal Driving, Drinking, Eating, Calling, Playing Phone, Turning Around, Tidying Hair, Driving with Single Hand, Pulling Driver, Grabbing Steering Wheel, and Fatigue Driving. The drivers are also instructed to perform the behavior in different styles. For example, the driver can perform the *calling* behavior with either his left or right hand. More critically, to simulate the real-world bus driving scenario, we collect our PCL-BDB dataset when the bus is moving, and record videos both in the daytime and at night. As a result, there exists luminosity variations on our PCL-BDB dataset (see Figure 4). For each video, we annotate the starting and ending frames of each action instance.

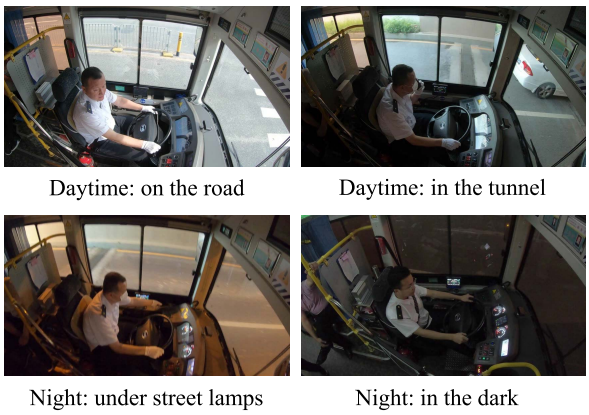


Fig. 4. Illustrative examples of different luminosity on PCL-BDB dataset.

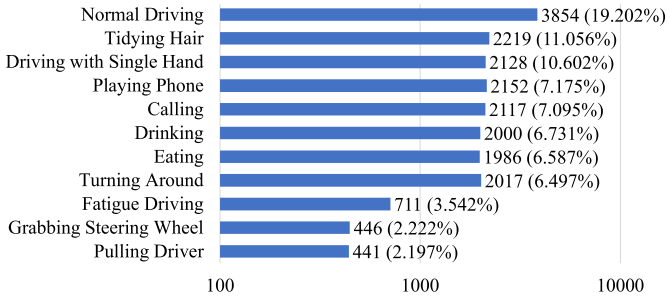


Fig. 5. The statistic information of the driver behavior category on our PCL-BDB dataset.

Camera Setup: We use three types of cameras on each bus: 1) Go Pro Camera (3840 × 1920 pixel at 30 Hz); 2) Near-infrared Camera (1920 × 1080 pixel at 30 Hz); 3) Binocular Depth Camera (1280 × 720 pixel at 30 Hz). We acquire the data in a video stream with three modalities, including color-data, infrared-data, and depth-data. We put the cameras on the top right of the drivers, which is consistent with the real-world camera setting on buses.

B. Data Splits

Our PCL-BDB dataset contains 268 videos with 9622 bus behavior instances over 11 classes. To verify the generalization ability of bus behavior recognition models on unseen drivers, we divide our dataset into training and testing sets based on the identity of the drivers. Specifically, we randomly select 44 drivers for training and the rest 11 drivers for testing. To summarize, we divide the dataset into a training set which contains 190 videos (6812 instances), and a testing set with 78 videos (2810 instances).

V. Experiments

In this section, we compare our BPAI-Net with existing driver behavior recognition methods on both Drive&Act [29] and our PCL-BDB datasets.

A. Datasets

1) Drive&Act [29]: Is a multi-modal benchmark for driver behavior recognition in automated vehicles, which consists of three modalities and six camera views. Each camera view contains 29 long videos with three modalities: infrared, depth and RGB. This dataset contains 83 annotated hierarchical activity categories, which are divided into three levels: Scenarios, Fine-grained Activities, and Atomic Action Units. The ratio of categories between three levels is 12:34:37. We conduct all Drive&Act experiments on the second level annotation with 34 driver behavior classes.

2) PCL-BDB: Contains 11 categories of driver behaviors originally. To increase the difficulty of recognizing fine-grained driver behaviors, we divide each behavior instance into three stages (*i.e.*, starting, middle, ending) and regard the behavior that performed by left or right hands as two different behavior categories. With the above-mentioned process, we obtain 40 fine-grained categories in total. In this paper, experiments on PCL-BDB dataset follow the 40-category setting unless otherwise specified.

B. Implementation Details

1) Posture-Guided CNN: We use 2D and 3D CNN (TSM [35] and I3D [13]) pretrained on Kinetics [13] as the backbones of appearance feature extraction model to verify the effectiveness of our method. Moreover, we implement TSM with different backbones, including MobileNet-V2 [36] and ResNet-50 [6]. For TSM, we use a sparse sampling strategy [46] to sample 8 frames from a video clip as input. For I3D, we randomly sample consecutive 64 frames from a video clip in the training phase, and we sample 64 frames in the middle of the video clip in the testing phase.

2) Appearance-Guided GCN: We use ST-GCN [38] as the backbone of posture feature extraction model and remove the temporal downsampling to make the temporal dimension of appearance feature and posture feature consistent. Following Drive&Act, we only use 13 key points on every frame instead of 25 key points. The number of key points we use is different from the model pretrained on Kinetics, we thus remove some incompatible pretraining parameters and the first batch normalization layer. The patch size d is set to 3.

3) Training Details: In general, each video frame is cropped and resized to 224 × 224 and the batch size is 8. Specifically, for TSM, we finetune it for 50 epochs with an initial learning rate of 0.001 (decays by 0.1 at epoch 30). The momentum and weight decay are set to 0.9 and 5e-4, respectively. For I3D, we finetune it for 100 epochs and set the initial learning rate to 0.01. The learning rate will be divided by a factor of 10 after 50 epochs. The momentum and weight decay are 0.9 and 1e-7, respectively. The experimental settings of AG-GCN are the same as PG-CNN, *e.g.*, the learning rate and momentum.

4) The Case of Multi-Person Behavior Recognition: Given a video that contains more than two persons, we first select the person that is more likely to be a driver. To this end, we first compute the mean coordinates of each key point in the training set of PCL-BDB dataset. Then, we find the smallest box that can contain all the key points' mean coordinates. In this way, the person with the largest number of key points that fall into the area of the predefined 2D spatial box is selected as the driver.

TABLE II

COMPARISONS WITH STATE-OF-THE-ART DRIVER BEHAVIOR RECOGNITION METHODS ON DRIVE&ACT. "POS" AND "APP" DENOTE POSTURE AND APPEARANCE, RESPECTIVELY

Model	Backbone	Modality	Recall (%)
Two-Stream [47]	-	POS	45.39
Three-Stream [48]	-	POS	46.95
C3D [49]	C3D	APP	43.41
P3D [50]	P3D	APP	45.32
TSM [35]	MobileNet-V2	APP	58.61
Ours	MobileNet-V2	APP + POS	**64.03**$_{(+5.42)}$
TSM [35]	ResNet-50	APP	61.77
Ours	ResNet-50	APP + POS	**65.34**$_{(+3.57)}$
I3D [13]	Inception-V1	APP	63.64
Ours	Inception-V1	APP + POS	**67.83**$_{(+4.19)}$

TABLE III

COMPARISONS WITH BASELINE MODELS ON THE VALIDATION SET OF PCL-BDB. "POS" AND "APP" DENOTE POSTURE AND APPEARANCE, RESPECTIVELY

Model	Backbone	Modality	Recall (%)
ST-GCN [38]	-	POS	70.72
TSM [35]	MobileNet-V2	APP	83.09
Ours	MobileNet-V2	APP + POS	**85.92**$_{(+2.83)}$
TSM [35]	ResNet-50	APP	81.63
Ours	ResNet-50	APP + POS	**85.84**$_{(+4.21)}$

C. Evaluation Metric

We follow Drive&Act [29] to use mean per-class accuracy (Recall) as the evaluation metric in all experiments for fair comparisons. The Recall evaluation metric indicates to what extent the model correctly identifies True Positives.

D. Comparisons With State-of-the-Arts

1) Drive&Act: We compare our proposed BPAI-Net with state-of-the-art driver behavior recognition methods in Table X. Our proposed BPAI-Net reaches the highest recall over the compared methods, implying that our method recognizes actions much more accurately than any other method. In particular, when using the same backbone (Inception-V1 [5]) as that in the previous best method (*i.e.*, I3D reported by [29]), our BPAI-Net outperforms I3D by a large margin (4.19% absolute performance gain). Moreover, compared with I3D, our BPAI-Net still achieves better recognition performance (64.03% vs. 63.64%) when using a lightweight backbone (*i.e.*, TSM with MobiletNet-V2). These experimental results verify the effectiveness of our method and the importance of exploiting the complementary information between appearance and posture modalities. More importantly, our method yields consistent improvements with different driver behavior frameworks and backbones, concluding that our method is generally effective and is not limited to the specific backbone.

2) PCL-BDB: Table III reports the driver behaviors recognition results of different methods (*e.g.*, ST-GCN [38] and TSM [35]). Our proposed method outperforms the baseline model with the same backbone by 2.83% and 4.21%, respectively. ST-GCN does not achieve promising results since it only relies on the posture information while neglecting the important cues from the appearance information. These results verify our motivation for considering the complementary information between two modalities. Note that we apply the same experimental settings (*e.g.*, hyper-parameters) on Drive&Act to our PCL-BDB dataset and achieve consistent improvements, indicating our method is generally effective.

TABLE IV

ABLATION STUDY OF THE KRM MODULE ON DRIVE&ACT

Model	Backbone of PG-CNN	Recall (%)
ST-GCN [38]	-	45.34
ST-GCN + KRM	MobileNet-V2	47.47$_{(+2.13)}$
ST-GCN + KRM	ResNet-50	47.63$_{(+2.29)}$
Ours w/o KRM	MobileNet-V2	62.15
Ours	MobileNet-V2	64.03$_{(+1.88)}$
Ours w/o KRM	ResNet-50	63.25
Ours	ResNet-50	65.34$_{(+2.09)}$
Ours w/o KRM	Inception-V1	67.29
Ours	Inception-V1	67.83$_{(+0.54)}$

VI. ABLATION STUDIES

In this section, we perform complete and in-depth ablation studies to evaluate the impact of each component (*e.g.*, PG-CNN and AG-GCN) of our method. We use ResNet-50 as the backbone of CNN and evaluate the performance on the testing set of Drive&Act.

A. The Effectiveness of Our KRM

As illustrated in Section III-C, we use KRM to obtain the object features by cropping the appearance feature maps and using such features to provide guidance for posture feature extraction. To verify the effectiveness of our KRM, we conduct an ablation study by designing two baselines, namely "**ST-GCN + KRM**" and "**Ours w/o KRM**". 1) As for ST-GCN + KRM, we insert a KRM into the ST-GCN [38] model. To obtain the appearance features for KRM, we adopt a modified PG-CNN (with PAM being removed) as the appearance feature extractor. We also add a global average pooling layer followed by a linear classifier on the output of ST-GCN. The final driver behavior category is predicted by ST-GCN. By comparing the performance of ST-GCN + KRM with ST-GCN, we can justify the importance of our KRM module. From Table IV, with the help of KRM, ST-GCN + KRM consistently outperforms the baseline by a large margin with different backbones (*i.e.*, 2.13% with MobileNet-V2 and 2.29% with ResNet-50). 2) As for "Ours w/o KRM", we directly remove KRM from our proposed BPAI-Net. From Table IV, the performance drops significantly when KRM is removed. These results reveal that the appearance features provide useful cues for recognizing driver behaviors, especially those behaviors that share similar posture but with different objects.

B. Different Fusion Strategies in KRM

To facilitate posture feature extraction, we use KRM to fuse object appearance features and posture features by directly adding each patch feature vector to its corresponding key

TABLE V

ABLATION STUDY OF DIFFERENT FUSION STRATEGY IN KRM MODULE ON DRIVE&ACT

Setting	Fusion Strategy	Additional FLOPs	Recall (%)
without KRM	none	-	63.25
with KRM	addition	+ 0.2M	65.34
with KRM	transformer encoder	+ 115.8M	66.28

TABLE VI

ABLATION STUDY OF THE PAM MODULE ON DRIVE&ACT

Model	Backbone	Recall (%)
TSM [35]	MobileNet-V2	58.61
TSM + PAM	MobileNet-V2	$62.15_{(+3.54)}$
TSM [35]	ResNet-50	61.77
TSM + PAM	ResNet-50	$63.25_{(+1.48)}$
I3D [13]	Inception-V1	63.64
I3D + PAM	Inception-V1	$67.29_{(+3.65)}$

point feature vector. To verify the effectiveness of our fusion strategy, we implement an attention-based strategy to fuse two features. Specifically, we design a baseline named "transformer encoder" by replacing the addition operation with the transformer encoder in [51]. We obtain query features relying on the posture features and obtain key and value features relying on the patch features. Besides, we follow [51] to add multiple heads, multiple layers and positional encoding to the baseline method. Please refer to [51] for more details. From Table V, the transformer-based fusion strategy performs slightly better than our addition-based version (66.28% vs. 65.34%), while significantly increasing the computation cost (115.8M vs. 0.2M). Considering both effectiveness and efficiency, we lastly choose to use the addition operation as the default setting of our experiments.

C. The Effectiveness of Our PAM

Our proposed PAM fuses the posture features from AG-GCN and the appearance features from PG-CNN relying on an attention mechanism and thus leverages body key points to guide the appearance feature extraction. To verify the effectiveness of PAM, we conduct experiments by first removing KRM from AG-GCN and then forwarding $\mathbf{G}_p''$ to PAM. We have two kinds of baselines including: (1) **TSM + PAM** with two different backbones (*i.e.*, MobileNet-V2 and ResNet-50); (2) **I3D + PAM**. As shown in Table VI, our PAM helps to lift the driver behavior recognition performance from 58.61% to 62.15% (MobileNet-V2) and from 61.77% to 63.25% (ResNet-50). Meanwhile, I3D + PAM outperforms the I3D baseline by 3.65%. The experimental results indicate that our proposed PAM module is effective and validate the importance of using body dynamic cues to guide the appearance feature extraction process in driver behavior recognition.

D. Different Positions of PAM

Our PAM module is a flexible module that can be inserted into any layer of PG-CNN. Here, we are interested in where the best position is to use PAM. To this end, we first determine the layers of output feature maps $\mathbf{F}_a$ and $\mathbf{G}_p$. In the first few layers of CNN and GCN, the models extract low-level features which may not contain sufficient appearance and posture cues. While in the last few layers, the extracted high-level features may lose some important spatial and location information. Therefore, we simply use the middle layers of CNN (2nd layer) and GCN (5th layer) to obtain $\mathbf{F}_a$ and $\mathbf{G}_p$, respectively. Then, we conduct experiments by varying the layer L of PG-CNN to insert PAM with the range of $\{2, 3, 4\}$. From Table VII, our BPAI-Net achieves the best performance (65.34%) with $L = 4$. We observe that $L = 2$ achieves comparable performance but incurs a larger computational cost due to a larger spatial dimension of feature maps compared with $L = 4$. Considering both effectiveness and efficiency, we insert PAM into the 4-th layer of PG-CNN $L = 4$ unless otherwise specified.

TABLE VII

ABLATION STUDY OF THE DIFFERENT POSITIONS OF PAM MODULE ON DRIVE&ACT

L	2	3	4 (last layer)
Recall (%)	65.28	62.10	**65.34**

TABLE VIII

ABLATION STUDY OF DIFFERENT FUSION STRATEGY IN PAM MODULE ON DRIVE&ACT

Model	Backbone	Strategy	Recall (%)
BPAI	MobileNet-V2	addition	60.73
BPAI	MobileNet-V2	attention	$64.03_{(+3.3)}$
BPAI	ResNet-50	addition	61.26
BPAI	ResNet-50	attention	$65.34_{(+4.08)}$

E. Different Fusion Strategies in PAM

Our PAM fuses the features from two modalities by using an attention mechanism. To verify the effectiveness of the attention-based fusion strategy, we design an addition-based variant of BPAI-Net. Specifically, we first apply an average pooling layer on $\mathbf{G}_p''$ and $\mathbf{F}_b'$ respectively and add up the resultant feature vectors. From the experimental results in Table VIII, the attention-based strategy outperforms the addition-based version remarkably on both MobileNet-V2 (3.3% absolute gain) and ResNet-50 (4.08% absolute gain) backbones. This is probably because that the attention module may learn to find the posture features that are helpful for each location on the appearance feature maps and boost the performance eventually.

F. Online Applicability of Our Method

For fair comparisons, we follow [10]–[12] to conduct experiments under an offline setting (*i.e.*, a video is segmented into video clips beforehand). To classify the behavior category of each clip, we use a sparse sampling strategy [46] to uniformly sample N frames from a video clip as the input. However, in real-life applications, the video is not segmented beforehand. Thus, we conduct experiments under an online setting [52] to study the online applicability of our method. Specifically, we maintain a working memory S_N which contains N frames as the input. With every N new frames from the video stream arriving, we sample $N/2$ frames from the working memory S_N and the N new frames, respectively. The sampled N frames are used to update S_N and also fed into our method for recognizing driver behavior. Please refer to

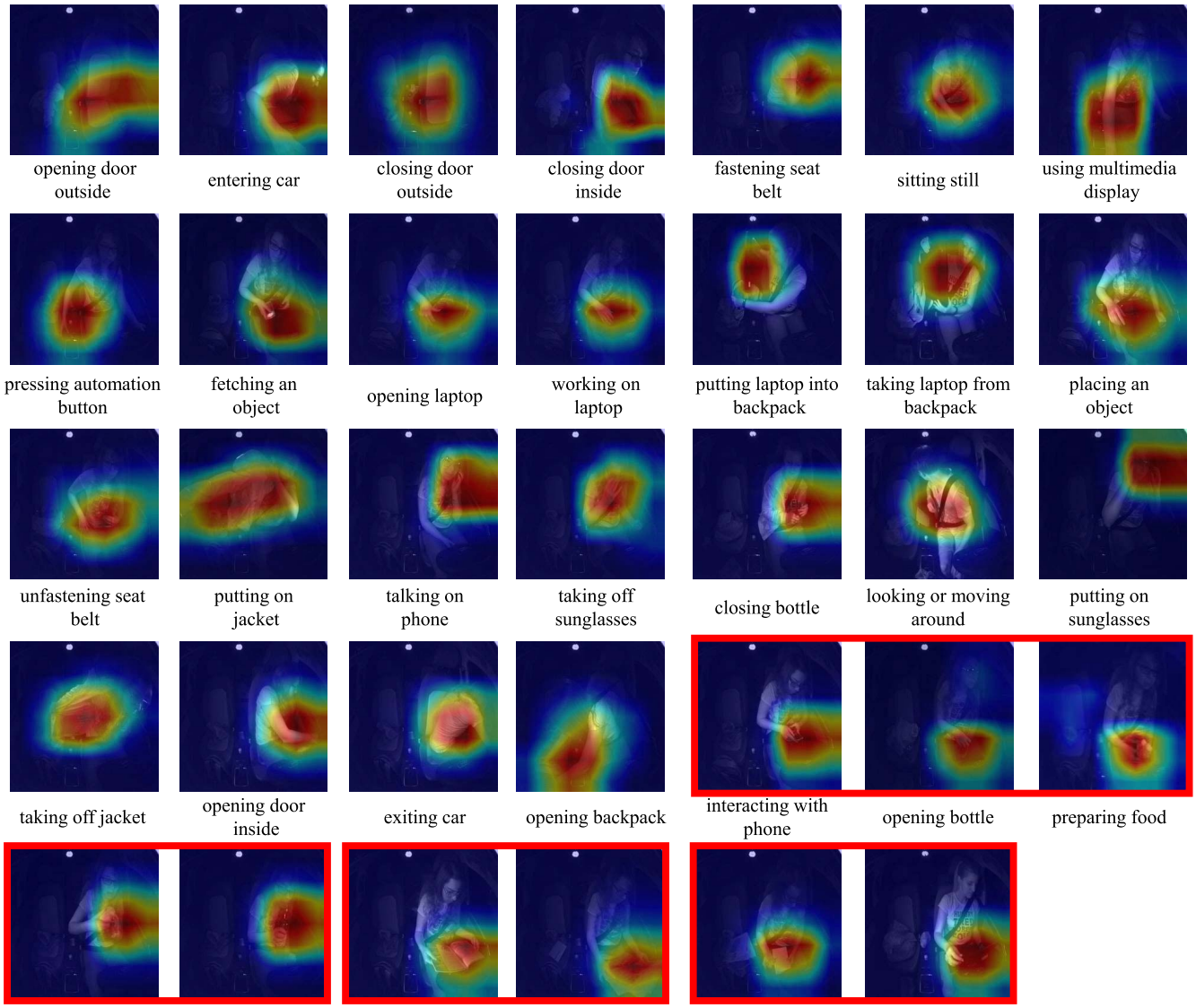


Fig. 6. The class activation maps of 34 action classes from our BPAI-Net on Drive&Act. Red boxes group the behaviors with similar postures. Our method learns to capture the differences between such behaviors by finding the regions with crucial cues.

TABLE IX
COMPARISONS WITH BASELINE MODELS UNDER ONLINE SETTING ON DRIVE&ACT

Model	Backbone	Latency (ms)	Recall (%)
TSM [35]	MobileNet-V2	11.4	54.70
Ours	MobileNet-V2	16.4	**57.91**
TSM [35]	ResNet-50	16.9	58.23
Ours	ResNet-50	19.2	**59.05**

TABLE X
COMPUTATION COMPLEXITY ON DRIVE&ACT IN TERMS OF FLOPS. "BI" DENOTES FOR BIDIRECTIONAL INTERACTION

Model	Backbone	FLOPs	Recall (%)
TSM [35]	ResNet-50	32.9G	61.77
TSM + BI (Ours)	ResNet-50	34.9G$_{(+2.0G)}$	**65.34**$_{(+3.57)}$
I3D [13]	Inception-V1	111.3G	63.64
I3D + BI (Ours)	Inception-V1	112.5G$_{(+1.2G)}$	**67.83**$_{(+4.19)}$

ECO [52] for more details. From Table IX, with the help of bidirectional interaction between appearance and posture, our method still outperforms the baseline methods over a large margin under the online setting. These results suggest that our method can be used to recognize driver behaviors from online video streams.

G. Compute Performance Measurements of Our Method

Our proposed BPAI-Net is a general and flexible framework that is compatible with different appearance and posture feature extractors (*i.e.*, backbones). To measure the compute performance of our method, we choose floating-point operations (FLOPs) as the evaluation metric. From Table X, the bidirectional interaction between appearance and posture only incurs a relatively small additional computation cost (6% with TSM and 1% with I3D) but is able to boost the performance significantly. These results imply that the improvements are brought by the effectiveness of exploiting bidirectional interactions rather than costly operations.

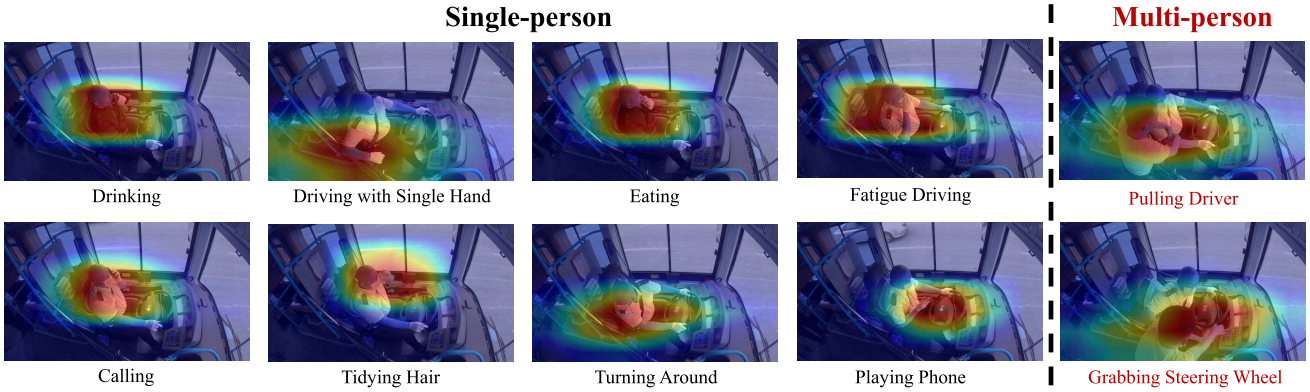


Fig. 7. The class activation maps of 10 abnormal classes from our BPAI-Net on PCL-BDB dataset. Our model learns to focus on the discriminative regions (*e.g.*, the bottle for *Drinking*, the food for *Eating*, and the phone for *Calling*) for recognizing driver behaviors.

H. Qualitative Results

Given the significant improvements brought by our method, we attempt to find out how our BPAI-Net helps to recognize driver behaviors more correctly. To this end, we use the Gradient-weighted Class Activation Mapping (Grad-CAM) technique [53] to obtain visualization results on Drive&Act and PCL-BDB dataset. Specifically, we compute the activation values of each location of the feature maps conditioned on the ground-truth behavior. The location that contributes more to the correct behavior prediction will have a higher activation value. We show the visualization results on Drive&Act in Figure 6. Note that the behaviors with similar postures are highlighted by red boxes. Our BPAI-Net focuses on the regions that are critical to identifying the category (*e.g.*, *closing bottle* and *unfastening seat belt*). For behaviors that involve driver-object interactions, our BPAI-Net is able to focus on the specific interacted object to identify the behavior category.

We also visualize the activation maps of 10 abnormal action categories on our PCL-BDB dataset. As shown in Figure 7, our BPAI-Net is able to focus on the critical area which is the key to recognize driver behaviors. For example, BPAI-Net pays more attention to the driver's right hand to identify the behavior *Driving with Single Hand*. Moreover, for multi-person categories (*i.e.*, *Grabbing Steering Wheel* and *Pulling Driver*), our method learns to concentrate on the key regions where passengers interfere with the driver.

VII. Future Work

Different from existing driver behavior recognition datasets that have not considered the interaction between driver and passengers, our PCL-BDB dataset additionally considers two behavior categories that involve multi-person (*i.e.*, pulling driver and grabbing steering wheel). As discussed in Section V-B, we attempt to select the person that is more likely to be a driver when the input video contains more than one person. It is interesting and important to explore how to detect both the driver and the passengers and exploit their relationships but we leave it for our future work. We hope our proposed BPAI-Net and the PCL-BDB dataset could guide other researchers to study the multi-person driver behavior recognition problem.

VIII. Conclusion

In this paper, we have proposed a Bidirectional Posture-Appearance Interaction Network (BPAI-Net) to simultaneously consider two modalities (*i.e.*, RGB and posture) for driver behavior recognition. To exploit the complementary information between RGB and posture, we have proposed a key point-based RoI module to exploit the contextual object information to facilitate posture feature extraction. We also have proposed a posture-guided attention module to help CNN focuses on critical local areas of video frames that are related to driver behaviors. Experimental results on Drive&Act dataset showed that our method outperformed state-of-the-art methods by a large margin (67.83% vs. 63.64%). Furthermore, we have collected a bus driver behavior recognition dataset and yielded consistent performance gain against existing methods.

References

[1] "Global status report on road safety 2018: Summary," World Health Org., Geneva, Switzerland, Tech. Rep. WHO/NMH/NVI/18.20, 2018.
[2] T. A. Dingus *et al.*, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 10, pp. 2636–2641, 2016.
[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.
[5] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[7] B. Baheti, S. Gajre, and S. Talbar, "Detection of distracted driver using convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1032–1038.
[8] A. Behera, A. Keidel, and B. Debnath, "Context-driven multi-stream LSTM (M-LSTM) for recognizing fine-grained activity of drivers," in *Proc. 40th German Conf. Pattern Recognit. (GCPR)*, Stuttgart, Germany, Oct. 2018, pp. 298–314.
[9] D. Tran, H. M. Do, W. Sheng, H. Bai, and G. Chowdhary, "Real-time detection of distracted driving based on deep learning," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1210–1219, Dec. 2018.
[10] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," 2017, *arXiv:1706.09498*.
[11] P. Li, M. Lu, Z. Zhang, D. Shan, and Y. Yang, "A novel spatial-temporal graph for skeleton-based driver action recognition," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3243–3248.

[12] M. Martin, M. Voit, and R. Stiefelhagen, "Dynamic interaction graphs for driver activity recognition," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7.

[13] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.

[14] I. Jegham, A. B. Khalifa, I. Alouani, and M. A. Mahjoub, "Safe driving: Driver action recognition using SURF keypoints," in *Proc. 30th Int. Conf. Microelectron. (ICM)*, Dec. 2018, pp. 60–63.

[15] X. Zhang, N. Zheng, F. Wang, and Y. He, "Visual recognition of driver hand-held cell phone use based on hidden CRF," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Jul. 2011, pp. 248–251.

[16] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 35–43.

[17] C. H. Zhao, B. L. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Intell. Transp. Syst.*, vol. 6, no. 2, pp. 161–168, 2012.

[18] C. H. Zhao, B. L. Zhang, X. Z. Zhang, S. Q. Zhao, and H. X. Li, "Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers," *Neural Comput. Appl.*, vol. 22, no. 1, pp. 175–184, 2013.

[19] M. R. Arefin, F. Makhmudkhujaev, O. Chae, and J. Kim, "Aggregating CNN and HOG features for real-time distracted driver detection," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–3.

[20] M. Leekha, M. Goswami, R. R. Shah, Y. Yin, and R. Zimmermann, "Are you paying attention? Detecting distracted driving in real-time," in *Proc. IEEE 5th Int. Conf. Multimedia Big Data (BigMM)*, Sep. 2019, pp. 171–180.

[21] Z. Wharton, A. Behera, Y. Liu, and N. Bessis, "Coarse temporal attention network (CTA-Net) for driver's activity recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, p. 1278.

[22] J.-C. Chen, C.-Y. Lee, P.-Y. Huang, and C.-R. Lin, "Driver behavior analysis via two-stream deep convolutional neural network," *Appl. Sci.*, vol. 10, no. 6, p. 1908, Mar. 2020.

[23] N. Kose, O. Kopuklu, A. Unnervik, and G. Rigoll, "Real-time driver state monitoring using a CNN based spatio-temporal approach," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3236–3242.

[24] N. Moslemi, R. Azmi, and M. Soryani, "Driver distraction recognition using 3D convolutional neural networks," in *Proc. 4th Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Mar. 2019, pp. 145–151.

[25] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by convolutional neural networks," *IET Comput. Vis.*, vol. 10, no. 2, pp. 103–114, Mar. 2016.

[26] M. Torstensson, B. Duran, and C. Englund, "Using recurrent neural networks for action and intention recognition of car drivers," in *Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, Feb. 2019, pp. 232–242.

[27] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver distraction identification with an ensemble of convolutional neural networks," 2019, *arXiv:1901.09097*.

[28] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F.-Y. Wang, "Driver activity recognition for intelligent vehicles: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5379–5390, Jun. 2019.

[29] M. Martin *et al.*, "Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2801–2810.

[30] J. D. Ortega *et al.*, "DMD: A large-scale multi-modal driver monitoring dataset for attention and alertness analysis," in *Proc. Comput. Vision ECCV Workshops*, Glasgow, U.K., Aug. 2020, pp. 387–405.

[31] O. Kopuklu, J. Zheng, H. Xu, and G. Rigoll, "Driver anomaly detection: A dataset and contrastive learning approach," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 91–100.

[32] C. Ou, C. Ouali, and F. Karray, "Transfer learning based strategy for improving driver distraction recognition," in *Proc. 15th Int. Conf. Int. Conf. Image Anal. Recognit.*, Póvoa de Varzim, Portugal. Springer, 2018, pp. 443–452.

[33] StateFarm. (2016). *State Farm Distracted Driver Detection*. [Online]. Available: https://www.kaggle.com/c/state-fatm-distracted-driver-detection

[34] S. Reis, A. Roitberg, M. Haurilet, and R. Stiefelhagen, "Activity-aware attributes for zero-shot driver behavior recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3950–3955.

[35] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7083–7093.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[37] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7291–7299.

[38] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–10.

[39] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[40] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014, *arXiv:1406.2199*.

[41] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.

[42] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, "Two-stream 3-D convNet fusion for action recognition in videos with arbitrary size and length," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 634–644, Mar. 2018.

[43] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, Dec. 2014.

[44] C.-H. Weng, Y.-H. Lai, and S.-H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *Proc. Asian Conf. Comput. Vis. (ACCV)*. Taipei, Taiwan: Springer, 2016, pp. 117–133.

[45] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver distraction identification with an ensemble of convolutional neural networks," *J. Adv. Transp.*, vol. 2019, pp. 1–12, Feb. 2019.

[46] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands. Springer, 2016, pp. 20–36.

[47] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 499–508.

[48] M. Martin, J. Popp, M. Anneken, M. Voit, and R. Stiefelhagen, "Body pose and context information for driver secondary task detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 2015–2021.

[49] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[50] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5533–5541.

[51] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:1706.03762*.

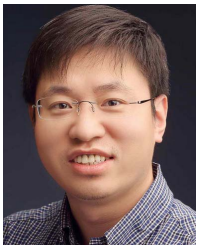
[52] M. Zolfaghari, K. Singh, and T. Brox, "ECO: Efficient convolutional network for online video understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 695–712.

[53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

Mingkui Tan (Member, IEEE) received the bachelor's degree in environmental science and engineering and the master's degree in control science and engineering from Hunan University, Changsha, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014. From 2014 to 2016, he worked as a Senior Research Associate on computer vision with the School of Computer Science, The University of Adelaide, Australia. He is currently a Professor with the School of Software Engineering, South China University of Technology. His research interests include machine learning, sparse analysis, deep learning, and large-scale optimization.

Gengqin Ni received the B.S. degree in software engineering from the Guangdong University of Finance and Economics, Guangzhou, China, in 2019. He is currently pursuing the M.S. degree with the School of Software Engineering, South China University of Technology. His research interest includes video action recognition.

Xu Liu received the B.S. degree in software engineering from the South China University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the M.S. degree with the School of Software Engineering. His research interests include action recognition and model compression.



Shiliang Zhang received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2012. He was a Post-Doctoral Scientist with NEC Laboratories America and a Post-Doctoral Research Fellow with The University of Texas at San Antonio. He is currently a Tenure-Track Assistant Professor with the School of Electronics Engineering and Computer Science, Peking University. He was a recipient of the Outstanding Doctoral Dissertation Awards from the Chinese Academy of Sciences and the Chinese Computer Federation, the President Scholarship from the Chinese Academy of Sciences, the NEC Laboratories America Spot Recognition Award, and the Microsoft Research Fellowship. He was also a recipient of the Top 10% Paper Award at the IEEE MMSP 2011.

Xiangmiao Wu received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2002. He is currently a Senior Engineer with the School of Computing Science and Engineering, South China University of Technology. His research interests include machine learning, data mining, and embedded systems.

Yaowei Wang (Member, IEEE) received the Ph.D. degree in computer science from the University of Chinese Academy of Sciences in 2005. He was a Professor with the National Engineering Laboratory for Video Technology Shenzhen (NELVT), Peking University Shenzhen Graduate School, in 2019. From 2014 to 2015, he worked as an Academic Visitor at the Vision Laboratory, Queen Mary University of London. He worked with the Department of Electronics Engineering, Beijing Institute of Technology, from 2005 to 2019. He is currently an Associate Professor with the Peng Cheng Laboratory, Shenzhen, China. He is the author or coauthor of over 70 refereed journals and conference papers. His research interests include machine learning and multimedia content analysis and understanding. He is a member of CIE, CCF, and CSIG. He was a recipient of the second prize of the National Technology Invention in 2017 and the first prize of the CIE Technology Invention in 2015. His team was ranked as one of the best performers in the TRECVID CCD/SED tasks from 2009 to 2012 and in PETS 2012.

Runhao Zeng received the bachelor's degree in automation science and engineering, the master's degree in pattern recognition and intelligent systems, and the Ph.D. degree in software engineering from the South China University of Technology, Guangzhou, China, in 2015, 2017, and 2021, respectively. His research interests include machine learning, deep learning, and their applications in video understanding.