A Cooperative Vehicle-Infrastructure System for Road Hazards Detection With Edge Intelligence

Chen Chen, *Senior Member, IEEE*, Guorun Yao, Lei Liu, *Member, IEEE*, Qingqi Pei, *Senior Member, IEEE*, Houbing Song, *Fellow, IEEE*, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*— Road hazards (RH) have always been the cause of many serious traffic accidents. These have posed a threat to the safety of drivers, passengers, and pedestrians, and have also resulted in significant losses to people and even to the economies of countries. Hence, road hazards detection (RHD) could play an essential role in intelligent transportation systems (ITS). The cooperative vehicle-infrastructure systems (CVIS) coordinate the communication between vehicles and roadside infrastructures. Onboard computing devices (OCD), then, make fast analyses and decisions based on road conditions. In this study, an RHD solution based on CVIS is proposed. Firstly, a high-performance heavy action detection model is selected. Using a meta-learning paradigm, critical features are generalized from a few-shot RH data. Secondly, we designed a lightweight RHD model to ensure its smooth inference on an OCD. Thirdly, we use a knowledge distillation (KD) framework to progressively distill the features of the complex model and the privileged information of the data into the lightweight one. Experimental results demonstrate that the model can effectively detect RH and obtain an accuracy of 90.2% with an inference time of 14.7ms.**

***Index Terms*— Cooperative vehicle-infrastructure system, edge intelligence, road hazards detection, meta-learning, knowledge distillation.**

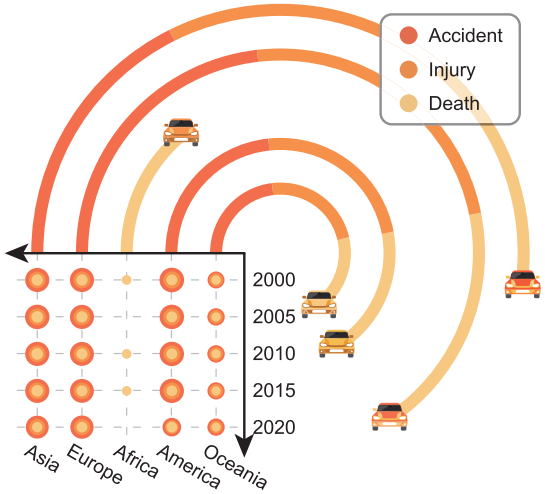

Fig. 1. **Statistical data of worldwide traffic accident conditions.** The bubble chart in the bottom left corner shows the yearly change from 2000 to 2020 in the number of accidents, injuries and deaths worldwide. The circular chart aggregates the data from 1970 to 2020. Some of the statistics are missing, but overall it reflects the harm caused by traffic accidents.

I. Introduction

STATISTICS [1] show that approximately 1.3 million people die each year worldwide caused of road traffic injuries. Another 20 to 50 million people suffer non-fatal harm, and many of them become disabled as a result. Their dependants have to cover the high cost of treatment and even take time off work to look after them. Road traffic accidents cost 3% of the GDP of most nations, which can wreak havoc on a country's economy. Fig. 1 illustrates the statistical data of worldwide traffic accident conditions [2], [3], [4]. In summary, there is an urgent need to address the issue of road safety!

RH are the causative factors of traffic accidents. The RH mentioned in this study refer to *the situations that occur in front of the vehicle which require the driver to reposition the vehicle within a very short duration, including vehicle throwing, emergency lane changing, emergency braking, sudden pedestrian intrusion, sudden animal intrusion, etc.* Fig. 3 illustrates the occurrence and response procedures of RH. The transmission of information in the circuit shown in Fig. 3 causes the driver's response time. Heretofore, traditional RHD methods mainly used specific instruments, such as air duct detectors, ring detectors, and ultrasonic/infrared motion alarms. These methods have many weaknesses, like low detection accuracy, short lifespan, and vulnerability to environmental impacts. Crucially, the mounting of the instruments can also damage the road surface. Furthermore, these instruments

Manuscript received 12 February 2022; revised 8 August 2022, 24 October 2022, and 6 January 2023; accepted 12 January 2023. Date of publication 6 February 2023; date of current version 8 May 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807500; in part by the National Natural Science Foundation of China under Grant 62072360, Grant 62001357, Grant 62172438, and Grant 61901367; in part by the Key Research and Development Plan of Shaanxi Province under Grant 2021ZDLGY02-09, Grant 2023-GHZD-44, and Grant 2023-ZDLGY-54; in part by the Natural Science Foundation of Guangdong Province of China under Grant 2022A1515010988; in part by the Key Project on Artificial Intelligence of Xi'an Science and Technology Plan under Grant 2022JH-RGZN-0003, Grant 2022JH-RGZN-0103, and Grant 2022JH-CLCJ-0053; in part by the Xi'an Science and Technology Plan under Grant 20RGZN0005; and in part by the Xi'an Key Laboratory of Mobile Edge Computing and Security under Grant 201805052-ZD3CG36. The Associate Editor for this article was D. F. Wolf. *(Corresponding authors: Houbing Song; Chen Chen.)*

Chen Chen, Guorun Yao, Lei Liu, and Qingqi Pei are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: cc2000@mail.xidian.edu.cn; gryao@stu.xidian.edu.cn; qqpei@mail.xidian.edu.cn).

Houbing Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250 USA (e-mail: h.song@ieee.org).

Schahram Dustdar is with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria (e-mail: dustdar@dsg.tuwien.ac.at).

Digital Object Identifier 10.1109/TITS.2023.3241251

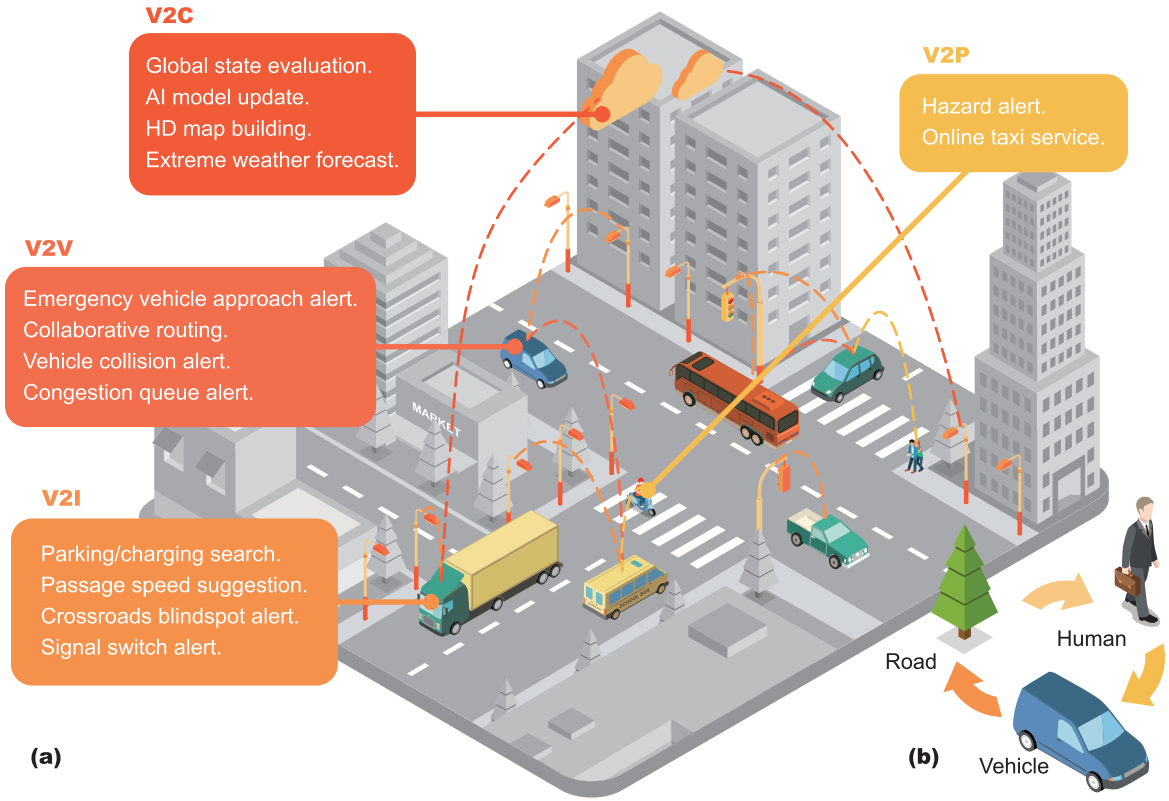

Fig. 2. **Schematic of the traffic scenario. (a)** lists the specific applications of V2C, V2V, V2I, and V2P that exist in real traffic scenarios. The cloud computing center, vehicles, roadside infrastructures, pedestrians, drivers, and roads in the scenario create these application requirements in conjunction. We will explore solutions to these problems in the interrelationship. **(b)** shows the three basic components of a traffic scenario: vehicle, road, and human.

can only collect a single type of data which are difficult to reuse. The high cost and low utilization of the instruments make them unworthy of use.

With the development of artificial intelligence (AI), many researchers use deep neural networks (DNN) to solve the problem of RHD. Reference [5] uses convolutional neural networks (CNN) to predict the severity of traffic accidents based on road, vehicle, and pedestrian factors. It requires carefully labeled data. Reference [6] uses stacked sparse autoencoders (SSAE) to analyze the importance and dependence of causative factors on road traffic injuries. Reference [7] uses gated recurrent units (GRU) and CNN to process onboard video and audio data for collision detection. It requires a synchronization of the data. Reference [8] uses CNN for precise accident prediction for highway-rail grade crossings (HRGC) with unbalanced data. Reference [9] uses multiple machine learning (ML) methods to organize and analyze traffic accident databases for accident prediction on high-risk roads. The major challenges of these studies focus on three main dimensions:

- ***Data requirements.*** The emergence of RH is accidental and not easily reproducible, which makes data collection laborious and costly. The small volumes of the existing datasets preclude DNNs from adequately capturing the data distribution and making the correct feature selections.
- ***Resource requirements.*** DNNs require a large amount of data I/O and cache during the loading and computation process, which raises great requirements for computing, communication and storage resources. Nonetheless, OCDs generally carry limited resources and can only handle typical operators.
- ***Online detection.*** Most studies have been performed based on RHD video files without considering online detection. In real-world situations, RHD needs to be operated online to assist drivers in reacting more immediately based on road conditions.

With the advancement of the 5/6*th* generation wireless communication systems (5/6G) and the ongoing evolution of edge intelligence technologies [10], the CVIS can be used to solve the mentioned problems. The CVIS can realize dynamic real-time information interaction on the links of vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-everything (V2X) [11], [12]. Fig. 2 illustrates the schematic of the links in the traffic scenario. This can facilitate vehicles to offload some computational tasks to roadside computing devices, thereby boosting the systems' effectiveness [13], [14].

Compared to traditional approaches, the CVIS can make better use of the computing resources at the edge of the network. At the same time, the image data captured by the camera contains more information. Researchers can use all sorts of algorithms to extract features of the data to fulfill different tasks. This solution does not need to rely on specialized instruments and can directly utilize existing sensors and computing devices on vehicles and roadside infrastructures. It is convenient and economical, avoiding damage to the road surface. Also, relying on the short distance between the

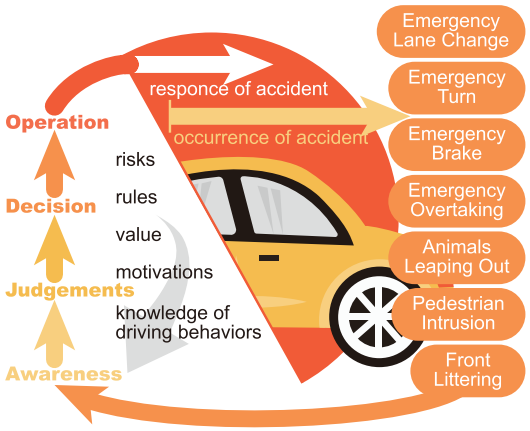

Fig. 3. **The occurrence and response procedures of RH.** The vehicle may encounter numerous RHs including those shown on the right side while traveling on the road. When an RH *occurs*, the driver will make a *judgment* by combining information such as *risk*, *rules*, and *value*. These lead to *decisions* and *operations* that govern the vehicle's reaction. The integrity of the procedure incorporates multiple steps and is influenced by a combination of factors. Therefore, there is a large time delay between the occurrence of the RH and the driver's response.

vehicles and the roadside infrastructures, the latency is greatly reduced.

In this study, we propose an RHD solution based on CVIS. For the model as a whole, we use a knowledge distillation framework to implement the delivery of features. We choose an existing model as the teacher model, which has a complicated structure but high accuracy. Considering the deployment on OCD, we design a lightweight model as the model student model. The student model refers to the state-of-the-art visual backbone and decouples the training and the inference process, thereby balancing the performance of the training and the speed of the inference. We hope that the teacher model performs feature extraction on RH data before passing them to the student model. However, RH occurs much less frequently compared to normal road conditions. Fortunately, the meta-learning paradigm can cope with the problem. The meta-learning paradigm can construct the few-shot data into multiple tasks. It uses several tasks to train the teacher model, which makes the model converge rapidly. This enables the network to learn crucial features utilizing the implicit information in the few-shot data. We perform pre-training and meta-training on the teacher network. The knowledge learned by the teacher model and the privileged information in the data are then progressively distilled into the student model. The tuned student model will undergo reparameterization and be smoothly deployed in the OCD. Experimental results show that we trade the "patience" of the teacher model for the "smartness" of the student model. Our approach can effectively detect RH and conserve computational resources.

Our contributions are as follows:

- ***A meta-learning paradigm is used.*** We use a meta-learning paradigm to support the training of the teacher model, consequently coping with the problem of few-shot RH data.
- ***A lightweight RHD model is designed.*** We design a lightweight RHD model that balances the performance of training and the speed of inference by decoupling the process of training and inference.
- ***A KD framework is adopted.*** We distill the knowledge from the teacher model into the student model. Then, we distilled the privileged information from the data into the student model in two more steps. The performance of the student model is progressively improved in the meantime.
- ***The model is deployed.*** We deployed the ultimate model in OCD. To validate the model's usability, video streams were fed into the system.

This paper is organized as follows: **Section II** describes the development of edge intelligence, action detection algorithms, meta-learning paradigms, and knowledge distillation framework. **Section III** presents the pipeline of our proposed edge intelligence-based RHD method. **Section IV** provides the experimental introduction and data analysis. **Section V** provides a conclusion of this study.

II. Related Works

This section composes four technology areas. First, the latest research progress and applications of edge intelligence techniques, the basis of this study, are announced. Second, the developments of action detection techniques based on different feature extraction methods, the core task of this study, are introduced. Third, different meta-learning paradigms are described. Fourth, multiple knowledge distillation frameworks are presented.

A. Edge Intelligence

Edge intelligence includes onboard and roadside edge intelligence, which together form CVIS [15]. Some AI models can be deployed in multiple layers based on the autonomous driving framework to achieve efficient joint inference between edge intelligent devices [16]. They can achieve assisted driving with the help of CVIS [17]. On the one hand, they can directly achieve effects such as lane change detection [18], and on the other hand, they can share local sensor data with other vehicles through the network to achieve cooperative sensing [19], [20]. When resources are limited, resource management games are performed based on task attributes [21], [22].

At the same time, edge intelligence can provide personalized services for the individual demands of users [23]. The edge intelligence platform can activate, scale, and orchestrate different services according to user density [24]. Based on network slicing technology, isolated virtual content service slices with different QoS requirements are extracted to provide customized services [25]. When tasks such as data collection and content distribution are in demand, multiple cache-enabled edge intelligence devices in various traffic environments can be combined to form an edge caching mechanism [26], [27]. For the data in use, the edge intelligence platform will also provide capabilities such as intrusion detection and privacy protection based on technologies including blockchain and federated learning [28], [29].

B. Action Detection

Action detection is one of the popular research topics in ML [30]. These algorithms are designed to detect and classify actions in videos and have been widely used in video surveillance, live stream, and autonomous driving.

Action detection algorithms can be classified according to different **feature extraction methods**. [31], [32] manually select features, which often lack the generalization ability and need to be adjusted repeatedly in changing scenarios. References [33], [34], [35], and [36] use 3D convolution to simultaneously analyze temporospatial features of the video. References [37], [38], [39], [40], and [41] use two branches to separately extract temporal and spatial features. References [42], [43], and [44] use RNN structures to model the video streams. Video transformer (ViT) structures have been used to implement video detection [45], [46], [47] since [48] adopted transformer structures to enter the vision domain.

Action detection algorithms can be classified according to different **input video format**. When the video can be viewed in its integrity, it is called *offline video*, and vice versa, it is called *online video*. References [49], [50], and [51] regress the beginning and ending time boundaries of actions in offline videos for classification. References [52], [53], [54], and [55] evaluate action beginning times based only on the historical videos observed from online video streams, and make predictions about the categories of future actions.

C. Meta-Learning

Compared with the traditional ML paradigm, meta-learning uses the task as the fundamental unit of study and can make accurate generalizations of data features for few-shot data [56], [57], [58], [59].

The metric-based meta-learning paradigm [60], [61], [62], [63] will first learn a kernel function. Accordingly, the data is encoded into an intermediate domain where the data in the same category will be recapitulated. Then the similarity measure of the data is calculated and the data is classified consequently.

The model-based meta-learning paradigm [64], [65], [66], [67] is more focused on finding a model that can update parameters rapidly within a small amount of training. Often external storage or neural network is used to help the network learn efficiently.

The optimization-based meta-learning paradigm [68], [69], [70], [71] addresses the problem of few-shot data from the perspective of optimization algorithms. They model the optimizer and learn a model-independent one that converges the model in a finite number of steps.

D. Knowledge Distillation

The concept of KD was first proposed by [72] for *model compression*. It often involves the participation of two models with different complexity for the same task [73]. In general, the heavy, cumbersome model is termed the teacher model, while the simple, lightweight model is termed the student model. References [74], [75], [76], [77], [78], and [79] makes the student model simulate the logits of the teacher model. References [80], [81], [82], [83], [84], and [85] focuses on minimizing the teacher model and the student model's representational relationships of the intermediate layer parameters.

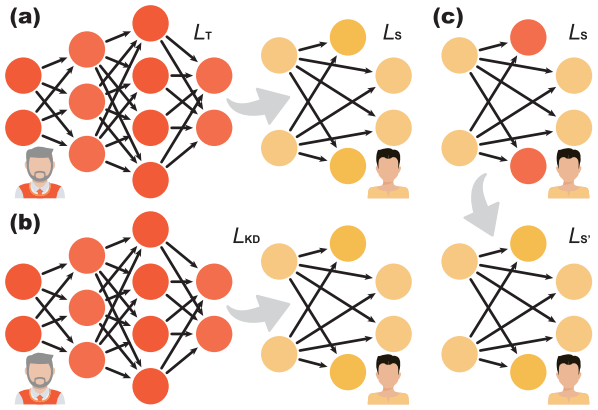

Fig. 4. **KD framework classification. (a)** illustrates the offline KD, which requires a well-trained teacher model. The student model is instructed to emulate the logit output of the teacher model, thus achieving knowledge transference. This strategy typically requires a sophisticated training procedure. Moreover, the capacity gap between the teacher model and the student model needs to be manipulated. **(b)** illustrates the online KD. In this case, both the teacher model and the student model can be untrained. They perform cooperative learning and guided learning to ultimately achieve an end-to-end KD. Online KD has high parallelism. However, this approach struggles to cope with high-volume teacher models. **(c)** illustrates self-KD, which is a special case of online KD. It can enable the sharing of knowledge of different depths in the model.

Apart from model compression, KD can also be used to *extract the privileged information*. Teacher models trained with data containing the privileged information can transfer the implicit information to the student models. References [86], [87], [88], and [89] uses optical flow as privileged information to train the teacher model, which helps the student model avoid the complex computation. References [90] and [91] trains the teacher model with complete actions while training the student model with historical actions, hence allowing the student model to achieve action prediction.

KD can take different training strategies. *Offline KD* is a two-stage strategy that trains the teacher model and the student model successively [72], [75], [77], [78], [79], [80]. *Online KD* uses only one stage to train both the teacher model and the student model simultaneously [74], [76], [91], [92], [93], [94]. *Self-KD* uses the same model as a teacher and student models, and distillation is usually performed between different layers [95], [96], [97], [98], [99]. Fig. 4 illustrates the further indication of KD.

III. Methodology

In this section, we will first provide an overall statement of the methodology corresponding to the existing problems. And then we will make a theoretical explanation of the key technologies involved.

A. Problem Statement

Fig. 5 illustrates the overall architecture of this study. We use the meta-learning paradigm to cope with the shortage

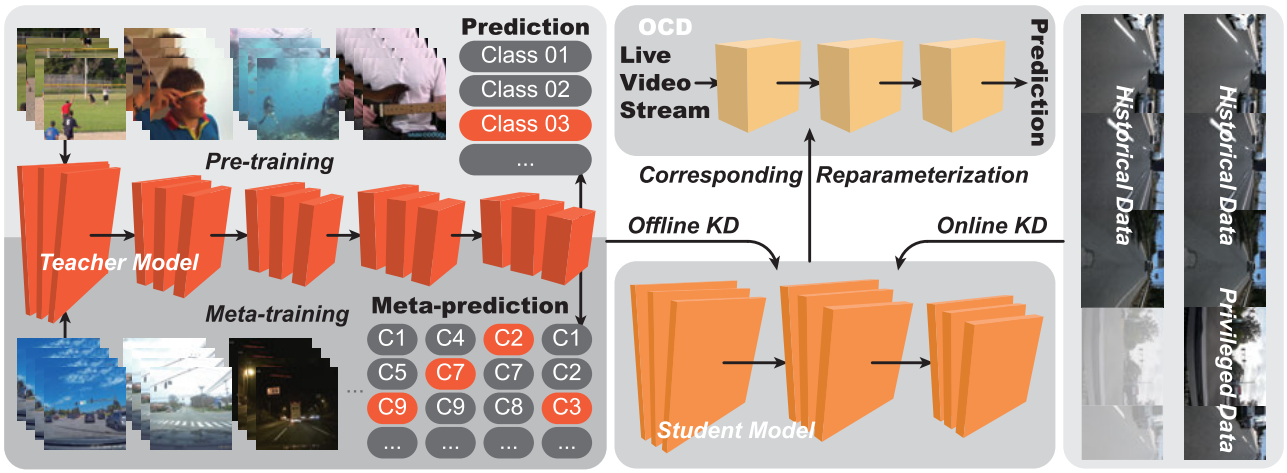

Fig. 5. **Study's overall architecture.** Three major segments are included: the arrangement of the meta-learning paradigm, the design and reparameterization of the lightweight model, and the implementation of the knowledge distillation framework.

of RH data, which mainly includes the shortage of classes and instances, and the presence of implicit classes. The left panel illustrates the training of the teacher model under the meta-learning paradigm. In the first phase, a generic dataset is used for the model pre-training. During this procedure, we expect the model to give the categories of the input videos. In the second phase, we split the dedicated dataset into different tasks based on the N-way K-shot rule. The teacher model is trained using different tasks based on pre-trained parameters. This improves the detection capability of the model for few-shot videos. During this procedure, we expect the model to ascribe the categories of the input videos for the current task.

Next, we design a lightweight RHD model with fewer parametric numbers and less computational volume for the problem of limited OCD resources. The model can economize resources and enhance the speed of inference, but the precision of the model tends to drop in this case. To trade off between speed and precision and to enlarge the privileged knowledge, we use a progressive KD framework. The middle illustrates the construction of the student model. The right side illustrates the production of the video stream. It contains historical video data and video data with privileged information. We use the teacher model to conduct offline KD on the student model so that the student model obtains the capability to detect RH. The privileged data is then used to conduct online KD on the student model so that the student model learns the capability to predict future conditions from the historical data. The online RHD problem is therefore resolved. Eventually, the model is reparameterized to further reduce the parameters and to make it compatible with the typical operators available on OCD.

B. Meta-Learning Paradigm

1) Pre-Training: The first phase of the meta-learning paradigm is the pre-training of the model using the generic Kinetics-400 dataset [100]. The generic dataset is introduced in detail in Section IV-A. We choose X3D [101] as the baseline. X3D is designed based on ResNet [102] architecture and extended from 2D image to 3D video feature extraction. The correlation between deep layer features and shallow layer features in ResNet can be indicated as:

$$f_{l_2} = f_{l_1} + \sum_{l=1}^{l_2-1} \mathcal{H}(f_l, W_l), \tag{1}$$

where

- $l_{(\cdot)}$ denotes the layer index of the model, and l_2 denotes the deeper layer while l_1 denotes the shallower one,
- $f_{(\cdot)}$ denotes the feature map of the model,
- $W_{(\cdot)}$ denotes the weight of the model,
- $\mathcal{H}(\cdot)$ denotes the mapping function of the model.

The residual structure can train the newly added layers into identical mapping functions, thus making the complex neural network a nested function containing the optimum function. X3D simultaneously extracts the spatiotemporal dimension of the video in order to preserve the complete temporal frequency. The spatiotemporal features are fused in the pooling layer, which finally gives the classification probabilities. X3D uses the cross-entropy (CE) loss function, which can be indicated as:

$$\mathcal{L}_{X3D} = -\frac{1}{S}\sum_{s} \iota_s \log(p_s), \tag{2}$$

where

- S denotes the number of samples involved in the calculation,
- $\iota_{(\cdot)}$ denotes the label, $\iota = 1$ if the sample n lies in category c, otherwise $\iota = 0$,
- $p_{(\cdot)}$ denotes the observed probability that the sample n belongs to category c.

The model parameter selection of X3D follows the coordinate descent algorithm, which is simple but efficient non-gradient optimization. Subject to certain constraints $\mathcal{C}(\cdot)$, the optimal value of the objective function $\mathcal{P}(\cdot)$ is searched in different dimensions of the variables sequentially. X3D selects suitable parameters by expanding them in different dimensions among the model. The selected parameters are as follows:

- γ_τ denotes the multiplier of the sampling frame-rate,
- γ_t denotes the multiplier of the temporal size, and can be expanded by sampling a longer temporal clip and increasing the frame-rate,

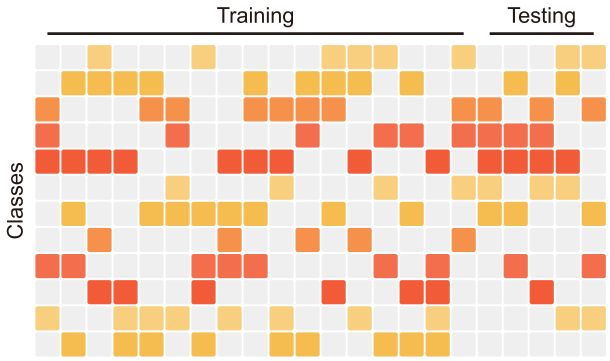

Fig. 6. **Schematic of task construction for the meta-learning paradigm.** In the N-way K-shot meta-training phase, N different categories are randomly selected from all categories in the training set, and K samples are selected for each category to construct different tasks. The highlighted blocks in the figure illustrate the randomly selected categories. Different colors represent different categories. The meta-training will overlay different combinations of categories for training. The meta-testing phase will construct a small number of tasks using the same rules for model fine-tuning before testing with unlabeled data.

γ_s denotes the multiplier of spatial resolution,
γ_d denotes the multiplier of network depth, *i.e.* the number of layers per residual block,
γ_w denotes the multiplier of network width, *i.e.* the channel number of layers,
γ_b denotes the inner channel width of the center filter in each residual block.

A carefully pre-trained model can be more flexible to convert between different tasks. X3D can be applied malleably by determining the appropriate network structure in an iterative mode when dealing with data of different distributions. The process of determining the jth γ in the ith iteration can be calculated as follows:

$$\gamma_j^i = \arg\max_{\gamma_j} \mathcal{P}(\gamma_0^i, \ldots, \gamma_{j-1}^i, \gamma_j, \gamma_{j+1}^{i-1}, \ldots, \gamma_k^{i-1}), \quad (3)$$

2) Meta-Training: The second phase of the meta-learning paradigm is meta-training. We use the RH dataset to enhance the dedicated capabilities of the model. The procedure of the meta-training algorithm is shown in Algorithm 1. As mentioned in Section II-C, meta-learning is performed based on tasks that ultimately result in a well-performing model. Hence an N-way K-shot strategy is used here based on the generic dataset. In each episode, N classes are randomly sampled from the dataset, with K items per class as the support set and Q items as the query set. Fig. 6 illustrates a schematic of task construction for the meta-learning paradigm. The support set and the query set indicate the training set and the test set of the current task, respectively. The weights W are updated with several iterations. With the convergence of the model, the model can be fine-tuned by randomly selecting data from the Road Hazard Stimuli dataset [103] according to the same strategy mentioned above. Then the final teacher model for the RHD is found.

Algorithm 1 Procedure of Meta-Training

Input: Pre-trained weights W; Training dataset $\mathcal{V} = \{(v_1, \iota_1), \ldots, (v_S, \iota_S)\}$, where $v_{(\cdot)}$ and $\iota_{(\cdot)}$ denote the videos and labels in the dataset; Real-world dataset $\mathcal{R} = \{u_1, \ldots, u_X\}$.
Output: Labels of samples in real-world dataset ι_r.

```
Initialize with weights W;
if not converged then
    // Use generic dataset.
    // E denotes total episode.
    for e = 1 to E do
        /* N denotes the number of categories selected at a time. */
        /* S denotes the total number of samples. */
        /* K and Q denote the number of samples per category in the support set and query set, respectively. */
        Sample N < S classes, K + Q items per class;
        Divide support set T and query set Q;
        Use T for training;
        Use Q for evaluating;
        Calculate the mean accuracy for current task;
        Update W;
else
    // Use dedicated dataset.
    Sample N classes, K + Q items per class;
    Fine-tune W;
```

C. Lightweight Road Hazards Detection Model

1) Backbone: After referring to recent vision backbones [104], [105], we noticed that floating point operations (FLOPs) show a weak correlation with the memory resources required by the model, inference speed, etc. On the one hand, many models use shared parameters, which can lead to FLOPs that are distinctly at variance with the number of parameters. On the other hand, the use of cross-layer connections can require additional memory resources to store the intermediate computational results. Furthermore, some tangled activation functions can have an impact on computational efficiency. A prevalent solution for tackling lightweight models is to decouple the architectures at training and inference. Better performance is achieved by using a linearly over-parameterized model while training. The linear structure in the model is reparameterized during inference to ensure its smoothness. Hence we design the reparameterization block. We partition a whole module into multiple branches while training but integrate them into a fully equivalent module during inference.

We designed the lightweight RHD model based on the MobileOne block [104]. Fig. 7 illustrates its structure. To reduce the data movement cost, we use a facile feed-forward structure. The feature information will flow in a pipeline, greatly reducing the memory acquisition cost, which frees the model from the speed bottleneck caused by I/O. By doing so, the number of network layers that the model

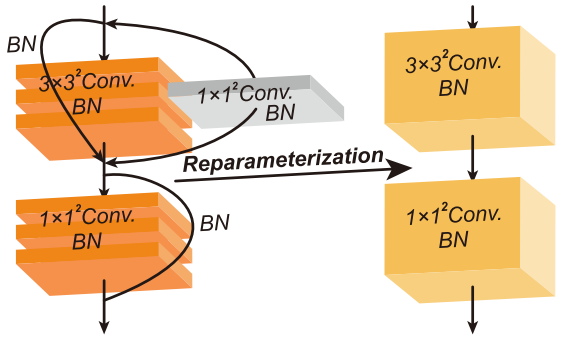

Fig. 7. **The block structure of the lightweight model.** We refer to the block in [104] and generalize it to 3D feature extraction. Compared to the ResNet block [102], this module reduces a bottleneck structure and can merge cross-layer connections by reparameterization operations.

can accommodate is enlarged. Meanwhile, we use existing operators and easy but efficient nonlinear activation functions to ensure the model is compatible with the runtime environment in OCD. The model can also use the coordinate descent method to select the appropriate structural hyperparameters as mentioned in Section III-B.

2) Loss: Our designed loss function contains CE loss and label smoothing regularization. The label smoothing regularization incorporates the loss of both incorrect classification and time location. The classification loss for label smoothing can be indicated as:

$$\hat{\iota} = (1 - \eta)\iota + \eta\epsilon, \tag{4}$$

where

$\hat{\iota}$ denotes the smoothed labels,
η denotes smoothing coefficient,
ϵ denotes a vector with all values 1.

The time location loss of label smoothing can be indicated as:

$$\mathcal{L}_{TL} = \frac{1}{\gamma_t} \sum_{t} \left| \log \hat{\iota}_t - \log \hat{\iota}_{t^*} \right|, \tag{5}$$

where

t denotes RH instant predicted by the model,
t^* denotes the ground truth RH instant marked in the label.

With the loss function $\mathcal{L}_{TL}$, the model is guided to increase the probability of correct predictions while decreasing the incorrect ones. Thus it improves the generalization ability of the model and avoids overfitting or overconfidence. In conjunction, the total loss of the lightweight model controlled by the scaling factor λ can be indicated as:

$$\mathcal{L}_{LW} = \mathcal{L}_{X3D} + \lambda \mathcal{L}_{TL}. \tag{6}$$

D. Knowledge Distillation Framework

1) Different Models: The first phase of KD is the distillation from the teacher model to the student model. The meta-learned X3D model mentioned in Section III-B is regarded as the teacher model, and the redesigned lightweight RHD model in Section III-C is regarded as the student model. We choose the offline KD. In this case, the output of the teacher model is used as a soft label, and what we want is for the student model to imitate the output of the teacher model. The ultimate goal is to minimize the discrepancy between the output of the student and the teacher. This stage is quite straightforward and the loss function can be indicated as:

$$\mathcal{L}_{KD} = \alpha \mathcal{L}_{SF} + \beta \mathcal{L}_{HD}, \tag{7}$$

$$\mathcal{L}_{SF} = -\sum_{s} \phi_s^T \log(\psi_s^T), \tag{8}$$

$$\mathcal{L}_{HD} = -\sum_{s} \iota_s \log(\psi_s^1), \tag{9}$$

where

α denotes the scaling factor of soft target loss $\mathcal{L}_{SF}$,
β denotes the scaling factor of hard target loss $\mathcal{L}_{HD}$,
ϕ^T denotes the result of the teacher model logits processed by softmax function at temperature T,
ψ^T denotes the result of the student model logits processed by softmax function at temperature T.

The softmax function at temperature T can be indicated as:

$$\mathcal{H}_{SM}^T(x_s) = \frac{\exp(x_s/T)}{\sum_s \exp(x_s/T)}. \tag{10}$$

2) Different Data: The second phase of KD is the distillation from global data to historical data. We input the data with a global time range and the data with only a historical time range into the student model after the previous phase of KD. In contrast to the data with a historical time range, the data with a global time range carries more privileged information. It can help the model to anticipate the movements of the action implicitly. We use online KD to learn privileged information.

To enhance the globality of the data, we combine the PREVENTION dataset with the Road Hazard Stimuli dataset. Fig. 8 illustrates three examples of generated video streams. Using the former as a carrier, the latter is interjected to artificially produce a video stream of RH random occurrences. The occurrence can be regarded as a Poisson process with a large arrival rate, which can be indicated as:

$$P(t, n) = \frac{(\rho t)^n}{n!} e^{-\rho t}, \tag{11}$$

where

t denotes the time range of the observed video,
n denotes the number of occurrences in RH during t,
ρ denotes the occurrence rate of RH.

IV. Experiments

This section introduces the datasets, evaluation metrics, and implementation details applied in the experiments, and also analyzes the experimental results.

A. Datasets

1) The Kinetics-400 Dataset: Is generic. It is a collection of clips taken from different YouTube videos. The clips contain human actions, human-object interactions such as playing musical instruments, assembling computers, etc., and human-human interactions such as hugging, shaking

Fig. 8. **Examples of generated video streams.** We produced a series of videos using the PREVENTION dataset as the background and the Road Hazard Stimuli dataset as the foreground, using different ρs according to the Poisson process. The gray areas represent the timeline, the highlighted blocks represent the occurrence of RH, and their different colors represent the different RH categories. The video series generated above can simulate the video streams captured by vehicles driving in the real scenario to some extent. The model can thus be tested under the online video stream.

hands, etc. The dataset contains about 400 human action categories, at least 400 video clips per category, and about 10 s long per clip. Each sample is labeled with a class name, YouTube ID, timestamp, etc.

2) The PREVENTION Dataset [106]: Is dedicated. Data are collected using LiDAR, millimeter-wave radar, and cameras on instrumented vehicle driving under natural conditions. The dataset encapsulates both urban and highway driving scenarios. To keep the driving style from being too monotonous, different drivers take turns driving. The total duration as well as the total distance traveled by the vehicle are 356 minutes and 540 km, respectively. The dataset is labeled with vehicle trajectories, lane changes, traffic participant categories, etc.

3) The Road Hazard Stimuli Dataset: Is RHD-dedicated. It is also a selection of RH or nearly-RH video clips from YouTube videos, with non-RH clips incorporated. The RH videos are 253 compared to 250 non-RH videos. The dataset covers a wide range of road scenarios, weather conditions, and RH categories. The selected videos are captured from the perspective of a car recorder, which fits well with the experimental scenario of this study. The videos are unified and cropped to a duration of 8 s, and a period before the occurrence of RH is preserved. The dataset was annotated with the timestamp of RH and the categories of RH.

B. Evaluation Metrics

1) Mean Average Precision (mAP) : We first calculate the Precision and Recall that the model performs on the data based on its frame-by-frame prediction of the input video, which can be indicated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{13}$$

where

TP denotes the true positives among the predicted frames,
FP denotes the false positives among the predicted frames,
FN denotes the false negatives among the predicted frames.

The average precision (AP) can be indicated as the average of the Precision values associated with different Recall values. Then, mAP is calculated in terms of the mean AP of each class.

2) Mean Calibrated Average Precision (mcAP) : To have a better metric of online video stream RHD, we refer to the mcAP in [107]. The frequency of online video stream RH captured by the onboard cameras is much less than the normal driving condition. We consider RH as the foreground of action recognition, while normal driving as the background. Compared with mAP, mcAP can better balance the ratio of foreground and background in the video stream, which can be indicated as:

$$\text{cPrecision} = \frac{\text{TP}}{\text{TP} + \text{FP}/\omega}, \tag{14}$$

$$\text{cAP} = \frac{\sum \mathcal{I} \times \text{cPrecision}}{\text{TP}}, \tag{15}$$

where

ω denotes the ratio of foreground to background in the online video stream,
$\mathcal{I}$ equals 1 when the current frame is a true positive, otherwise equals 0.

3) Model Parameters : Model parameters are evaluated in terms of the volume of parameters needed to be trained. In addition to the number of parameters it contains, the volume also indicates the variation in memory occupation due to the utilization of storage of different precision. During the inference procedure of the model, the parameters will be loaded into the memory. Thus, the spatial complexity of the model can be monitored to a certain extent. This allows us to adapt the model structure.

4) FLOPs: FLOPs can indicate the number of operations required for inference of the model. Since this study is about model inference on an OCD with limited computational capacity, the number of floating point operations can largely reflect the time complexity of the model.

5) Model Runtime Latency : Model runtime latency contains the actual time consumed throughout the process of

TABLE I
PLATFORM PARAMETERS

	Desktop System	Jetson TX2
CPU	Intel Core i7-10700	Dual-Core NVIDIA Denver 2 & Quad-Core ARM Cortex-A57
GPU	NVIDIA GeForce 3090 24GB	NVIDIA 256 CUDA cores
Memory	32GB DDR4	8GB 128-bit LPDDR4 Memory
Storage	3TB TOSHIBA HDWD130	32GB eMMC 5.1

TABLE II
MODEL PARAMETERS

Layers	Filters	Output Sizes $T \times S^2$
Data Layer	stride γ_τ, 1^2	$1\gamma_t \times (112\gamma_s)^2$
conv_1	$1 \times 3^2, 3 \times 1, 24\gamma_w$	$1\gamma_t \times (56\gamma_s)^2$
mbone3D_2	$\begin{bmatrix} 3 \times 3^2, 24\gamma_b\gamma_w \\ 1 \times 1^2, 24\gamma_w \end{bmatrix} \times \gamma_d$	$1\gamma_t \times (28\gamma_s)^2$
mbone3D_3	$\begin{bmatrix} 3 \times 3^2, 48\gamma_b\gamma_w \\ 1 \times 1^2, 48\gamma_w \end{bmatrix} \times 2\gamma_d$	$1\gamma_t \times (14\gamma_s)^2$
mbone3D_4	$\begin{bmatrix} 3 \times 3^2, 96\gamma_b\gamma_w \\ 1 \times 1^2, 96\gamma_w \end{bmatrix} \times 4\gamma_d$	$1\gamma_t \times (7\gamma_s)^2$
conv_4	$1 \times 1^2, 96\gamma_b\gamma_w$	$1\gamma_t \times (4\gamma_s)^2$
pool_4	$1\gamma_t \times (4\gamma_s)^2$	$1 \times 1 \times 1$
fc_1	$1 \times 1^2, 2048$	$1 \times 1 \times 1$
fc_2	1×1^2, #classes	$1 \times 1 \times 1$

parameter loading, I/O, model inference, etc. We can depend on this metric to visually evaluate the temporal performance of the model.

C. Implementation Details

TABLE I demonstrates the parameters of the platform we used. The procedures of training and parameter tuning of the model are operated in the desktop system. Since these processes are accomplished before the model processes the live video stream, they can be seen as taking place offline. We chose the Jetson TX2, which is small in size with a certain amount of computational capacity, as the OCD based on its pervasiveness. The inference process of the models is tested on it. This particular procedure can be seen as taking place online. TABLE II demonstrates the configuration of our parameters for the lightweight model. The layers

conv. denotes convolutional layers,
mbone3D. denotes 3D mobile one blocks illustrated in Section III-C,
pool. denotes pooling layers,
fc. denotes fully Connected layers.

D. Ablation Study

TABLE III demonstrates the performance comparison of different models and the ablation experimental procedure. We first prove that pre-training is essential in the meta-learning paradigm. The pre-training phase can help the meta-learning paradigm to improve the precision by 10.1%. Furthermore, pre-training is more conducive to the learning of lightweight models than random parameter initialization. The precision of the lightweight model after pretraining is improved by 6.9%. Definitely, compared with the teacher model, the student model has 30.8%, 41.2% and 37.7% reductions in parameters, FLOPs and latency, respectively. After the model KD, the performance of the lightweight model improves substantially. Its precision only lags behind the teacher model by 0.23%. This reveals that the model KD can effectively distill the knowledge from the teacher model to the student model. Because of the lightweight model design in Section III-C, the model structure was fine-tuned during model KD to fit the model capacity. Although it causes a slight rebound in model complexity, it trades off for a significant rise in performance. However, after the first data KD, the performance of the model first decreased by 0.45%. After the second data KD, the model performance rebounded by 1.1%, exceeding the accuracy before the data KD. This shows that the teacher's "patience" can be exchanged for the student's "smart". Eventually, the reparameterization of the model further improved its speed of the model. Compared with the latest action detection models, our model achieves an advanced level of precision and speed. At the same time, we tuned the online video stream, which makes our model more dedicated. Fig. 9 illustrates the effectiveness of the model application.

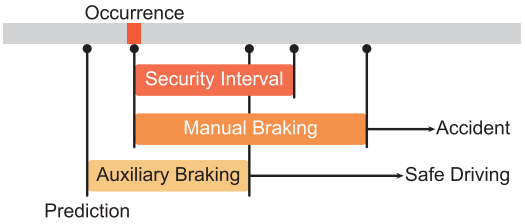

Fig. 9. **The effectiveness of the model application.** [103] shows that there is a safe response time interval after an accident. Safe driving can only be ensured if the driver reacts within this interval. However, according to the statistics in [103], the time consumed by manual braking for many drivers is up to 605 ms, which exceeds the safe time interval of 500 ms. According to the analysis of the experimental results, fortunately, with the auxiliary of the algorithm prediction, the vehicle can alert the driver before the accident occurs and help the driver to brake quicker within the safe time interval.

We compare the final model to be deployed with several representative state-of-the-art action detection models to demonstrate the superiority of our method. The F^2G model utilizes a large number of parameters to guarantee the accuracy of future frame generation to boost the model's performance. Thus, it has a parametric count of 174 M, which puts a huge requirement on both computational resources and memory size, which is not in line with the characteristics of OCD in CVIS scenarios. In contrast, our model can achieve better precision with 10.2% of the parameters. Compared to the SF-Ad model, our reparameterized model effectively reduces the number of FLOPs by 55.9% while improving the precision by 11.0%. The TSN-SD model performs well in live video stream action recognition containing the human body. However, for the RHD task, our model increases precision by 14.6%, reduces the parameters by 64.7%, and reduces latency by 35.8%. Our model precision increases by 15.6%, parameter size decreases by 35.0%, and latency decreases by 49.7% compared to the baseline model X3D.

TABLE III
MODEL PERFORMANCES

Methods	mAP (%) K400	mAP (%) RHS	mcAP (%) Com.	Params. (M)	FLOPs (G)	Latency (ms)
F^2G [54]	84.9	79.4	-	174	-	-
SF-Ad [108]	85.0	79.9	-	-	234.0	-
TSN-SD [109]	86.7	77.4	-	43.6	-	22.9
X3D	85.6	76.7	-	23.7	194.6	29.2
X3D						
+ meta-learning	-	80.4	-	23.7	194.6	29.2
+ meta-learning*	86.1	88.5	-	23.7	194.6	29.2
+ lightweight model	-	56.6	-	16.4	114.4	18.2
+ lightweight model*	-	60.5	-	16.4	114.4	18.1
+ model KD	-	88.3	-	17.8	120.0	18.1
+ data KD 1	-	87.9	90.2	17.8	120.0	18.1
+ data KD 2	-	88.9	91.3	17.8	120.0	18.1
+ reparameterization	-	88.7	90.8	15.4	103.1	14.7

V. CONCLUSION

First, we select a model with a complicated structure but high accuracy as the teacher model. We use a meta-learning paradigm for pre-training the teacher model to cope with the few-shot data of RH. Then, we designed a lightweight model as the student model. This model decouples the processes of training and inference, thus balancing the performance of training and the speed of inference. We adopt a knowledge distillation framework to progressively distill the knowledge obtained by the teacher model and the privileged information in the dataset to the student model. The trained student model will undergo reparameterization and be deployed in an OCD. Experimental results show that we trade the "patience" of the teacher network for the "smartness" of the student network. Our approach can efficiently detect RH and conserve computational resources.

We will continue to promote this study. On the one hand, we will keep up with the technology development trend and continuously update the vision feature extraction method. On the other hand, we will explore model parameter compression techniques in depth to maximize the capabilities of the hardware.

REFERENCES

[1] World Health Organization. (Jun. 2022). *Road Traffic Injuries*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] Our World in Data. (Feb. 2020). *Death Rate Due to Road Traffic Injuries*. [Online]. Available: https://ourworldindata.org/grapher/death-rate-road-traffic-injuries?time=2019&country= OWID_WRL

[3] Organisation for Economic Cooperation and Development. (Jun. 2021). *Road Accidents*. [Online]. Available: https://data.oecd.org/transport/road-accidents.htm

[4] Statista Explainer. (Aug. 2021). *Estimated Impact of Vehicle Automation on Collision Rates in 2030, by Automation Level*. [Online]. Available: https://www.statista.com/statistics/1238242/impact-of-vehicle-automation-on-collision-rates/

[5] M. A. Rahim and H. M. Hassan, "A deep learning based traffic crash severity prediction framework," *Accident Anal. Prevention*, vol. 154, May 2021, Art. no. 106090.

[6] Z. Ma, G. Mei, and S. Cuomo, "An analytic framework using deep learning for prediction of traffic accident injury severity based on contributing factors," *Accident Anal. Prevention*, vol. 160, Sep. 2021, Art. no. 106322.

[7] J. G. Choi, C. W. Kong, G. Kim, and S. Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras," *Exp. Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115400.

[8] L. Gao, P. Lu, and Y. Ren, "A deep learning approach for imbalanced crash data in predicting highway-rail grade crossings accidents," *Rel. Eng. Syst. Saf.*, vol. 216, Dec. 2021, Art. no. 108019.

[9] D.-J. Lin, M.-Y. Chen, H.-S. Chiang, and P. K. Sharma, "Intelligent traffic accident prediction model for Internet of Vehicles with deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2340–2349, Mar. 2022.

[10] M. Zhao, C. Chen, L. Liu, D. Lan, and S. Wan, "Orbital collaborative learning in 6G space-air-ground integrated networks," *Neurocomputing*, vol. 497, pp. 94–109, Aug. 2022.

[11] H. Yang, Z. Wei, Z. Feng, X. Chen, Y. Li, and P. Zhang, "Intelligent computation offloading for MEC-based cooperative vehicle infrastructure system: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7665–7679, Jul. 2022.

[12] C. Chen, J. Jiang, Y. Zhou, N. Lv, X. Liang, and S. Wan, "An edge intelligence empowered flooding process prediction using Internet of Things in smart city," *J. Parallel Distrib. Comput.*, vol. 165, pp. 66–78, Jul. 2022.

[13] C. Chen, Y. Zeng, H. Li, Y. Liu, and S. Wan, "A multi-hop task offloading decision model in MEC-enabled Internet of Vehicles," *IEEE Internet Things J.*, early access, Jan. 18, 2022, doi: 10.1109/JIOT.2022.3143529.

[14] Y. Zhang, C. Chen, L. Liu, D. Lan, H. Jiang, and S. Wan, "Aerial edge computing on orbit: A task offloading and allocation scheme," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 11, pp. 1–11, Sep. 2022.

[15] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, and A. Mouzakitis, "A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6206–6221, Jul. 2022.

[16] B. Yang et al., "Edge intelligence for autonomous driving in 6G wireless system: Design challenges and solutions," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 40–47, Apr. 2021.

[17] Y. Wang et al., "Cooperative perception of roadside unit and onboard equipment with edge artificial intelligence for driving assistance," *Connected Cities for Smart Mobility toward Accessible Resilient Transp. Center*, vol. 1, no. 1, pp. 1–2, Nov. 2021.

[18] B. Fan, Y. Wu, Z. He, Y. Chen, T. Q. S. Quek, and C.-Z. Xu, "Digital twin empowered mobile edge computing for intelligent vehicular lane-changing," *IEEE Netw.*, vol. 35, no. 6, pp. 194–201, Nov. 2021.

[19] Q. Yang, S. Fu, H. Wang, and H. Fang, "Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 3, pp. 96–101, May 2021.

[20] Z. Yu, Z. Si, X. Li, D. Wang, and H. Song, "A novel hybrid particle swarm optimization algorithm for path planning of UAVs," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22547–22558, Jun. 2022.

[21] X. Zhu, Y. Luo, A. Liu, N. N. Xiong, M. Dong, and S. Zhang, "A deep reinforcement learning-based resource management game in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2422–2433, Mar. 2022.

[22] S. Shen, C. Yu, K. Zhang, and S. Ci, "Adaptive artificial intelligence for resource-constrained connected vehicles in cybertwin-driven 6G network," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16269–16278, 2021.

[23] Y. Hui et al., "Personalized vehicular edge computing in 6G," *IEEE Netw.*, vol. 35, no. 6, pp. 278–284, Nov. 2021.

[24] W. Qi, Q. Li, Q. Song, L. Guo, and A. Jamalipour, "Extensive edge intelligence for future vehicular networks in 6G," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 128–135, Aug. 2021.

[25] Y. Wu, X. Fang, C. Luo, and G. Min, "Intelligent content precaching scheme for platoon-based edge vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20503–20518, Oct. 2022.

[26] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.

[27] C. Wang, C. Chen, Q. Pei, Z. Jiang, and S. Xu, "An information centric in-network caching scheme for 5G-enabled Internet of Connected Vehicles," *IEEE Trans. Mobile Comput.*, early access, Dec. 21, 2021, doi: 10.1109/TMC.2021.3137219.

[28] H. Liu et al., "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6073–6084, Jun. 2021.

[29] Y. Li, S. Xie, Z. Wan, H. Lv, H. Song, and Z. Lv, "Graph-powered learning methods in the Internet of Things: A survey," *Mach. Learn. Appl.*, vol. 11, Mar. 2023, Art. no. 100441.

[30] X. Hu, J. Dai, M. Li, C. Peng, Y. Li, and S. Du, "Online human action detection and anticipation in videos: A survey," *Neurocomputing*, vol. 491, pp. 395–413, Jun. 2022.

[31] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th ACM Int. Conf. Multimedia*, Sep. 2007, pp. 357–360.

[32] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.

[33] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 305–321.

[34] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.

[35] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 203–213.

[36] K. Li, X. Li, Y. Wang, J. Wang, and Y. Qiao, "CT-Net: Channel tensorization network for video classification," 2021, *arXiv:2106.01603*.

[37] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6202–6211.

[38] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 803–818.

[39] 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.

[40] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang, "TEA: Temporal excitation and aggregation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 909–918.

[41] Z. Wang, Q. She, and A. Smolic, "ACTION-Net: Multipath excitation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13214–13223.

[42] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 354–363.

[43] L. Zhang and X. Xiang, "Video event classification based on two-stage neural network," *Multimedia Tools Appl.*, vol. 79, nos. 29–30, pp. 21471–21486, Aug. 2020.

[44] F. Ali Dharejo et al., "FuzzyAct: A fuzzy-based framework for temporal activity recognition in IoT applications using RNN and 3D-DWT," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 11, pp. 4578–4592, Nov. 2022.

[45] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6836–6846.

[46] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. ICML*, vol. 2, no. 3, 2021, p. 4.

[47] L. Li and L. Zhuang, "Mevit: Motion enhanced video transformer for video classification," in *Proc. Int. Conf. Multimedia Model.* Cham, Switzerland: Springer, 2022, pp. 419–430.

[48] A. Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[49] P. Zhao, L. Xie, C. Ju, Y. Zhang, Y. Wang, and Q. Tian, "Bottom-up temporal action localization with mutual regularization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 539–555.

[50] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han, "Revisiting anchor mechanisms for temporal action localization," *IEEE Trans. Image Process.*, vol. 29, pp. 8535–8548, 2020.

[51] C. Lin et al., "Learning salient boundary feature for anchor-free temporal action localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3320–3329.

[52] M. Xu, M. Gao, Y.-T. Chen, L. Davis, and D. Crandall, "Temporal recurrent networks for online action detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5532–5541.

[53] M. Gao, M. Xu, L. Davis, R. Socher, and C. Xiong, "StartNet: Online detection of action start in untrimmed videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5542–5551.

[54] D.-H. Yoon, N.-G. Cho, and S.-W. Lee, "A novel online action detection framework from untrimmed video streams," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107396.

[55] H. Eun, J. Moon, J. Park, C. Jung, and C. Kim, "Learning to discriminate information for online action detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 809–818.

[56] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1920–1930.

[57] Y. Ma, S. Zhao, W. Wang, Y. Li, and I. King, "Multimodality in meta-learning: A comprehensive survey," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 108976.

[58] S. Luo, Y. Li, P. Gao, Y. Wang, and S. Serikawa, "Meta-seg: A survey of meta-learning for image segmentation," *Pattern Recognit.*, vol. 126, Jun. 2022, Art. no. 108586.

[59] C. Simon, P. Koniusz, and M. Harandi, "Meta-learning for multi-label few-shot classification," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3951–3960.

[60] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 6, 1993, pp. 1–8.

[61] G. Koch et al., "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, Lille, France, 2015, pp. 1–30.

[62] H. Song, Y. Jin, Y. Cheng, B. Liu, D. Liu, and Q. Liu, "Learning interlaced sparse sinkhorn matching network for video super-resolution," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108475.

[63] K. Wu, J. Tan, and C. Liu, "Cross-domain few-shot learning approach for lithium-ion battery surface defects classification using an improved Siamese network," *IEEE Sensors J.*, vol. 22, no. 12, pp. 11847–11856, Mar. 2022.

[64] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2014, *arXiv:1410.3916*.

[65] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1842–1850.

[66] T. Li et al., "Memory-augmented meta-learning on meta-path for fast adaptation cold-start recommendation," *Connection Sci.*, vol. 34, no. 1, pp. 301–318, Dec. 2022.

[67] B. Yu, X. Li, J. Fang, C. Tai, W. Cheng, and J. Xu, "Memory-augmented meta-learning framework for session-based target behavior recommendation," *World Wide Web*, vol. 26, pp. 1–19, Mar. 2022.

[68] A. Srinivasan, A. Bharadwaj, M. Sathyan, and S. Natarajan, "Optimization of image embeddings for few shot learning," in *Proc. 10th Int. Conf. Pattern Recognit. Appl. Methods*, 2017, pp. 1–6.

[69] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.

[70] S. Fu et al., "Adaptive multi-scale transductive information propagation for few-shot learning," *Knowl.-Based Syst.*, vol. 249, Aug. 2022, Art. no. 108979.

[71] S. Mahmud and K. H. Lim, "One-step model agnostic meta-learning using two-phase switching optimization strategy," *Neural Comput. Appl.*, vol. 34, pp. 1–9, Aug. 2022.

[72] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[73] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, Mar. 2021.

[74] J. Choi, H. Cho, S. Jeong, and W. Hwang, "ORC: Network group-based knowledge distillation using online role change," 2022, *arXiv:2206.01186*.

[75] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11953–11962.

[76] G. Wu and S. Gong, "Peer collaborative learning for online knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 10302–10310.

[77] L. Wei, A. Xiao, L. Xie, X. Zhang, X. Chen, and Q. Tian, "Circumventing outliers of autoaugment with knowledge distillation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 608–625.

[78] Y. Li, L. Liu, G. Wang, Y. Du, and P. Chen, "EGNN: Constructing explainable graph neural networks via knowledge distillation," *Knowl.-Based Syst.*, vol. 241, Apr. 2022, Art. no. 108345.

[79] Z. Xu, K. Wu, Z. Che, J. Tang, and J. Ye, "Knowledge transfer in multi-task deep reinforcement learning for continuous control," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 15146–15155.

[80] M. Sepahvand, F. Abdali-Mohammadi, and A. Taherkordi, "Teacher–student knowledge distillation based on decomposed deep feature representation for intelligent mobile applications," *Exp. Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117474.

[81] L. Gao, K. Xu, H. Wang, and Y. Peng, "Multi-representation knowledge distillation for audio classification," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5089–5112, Feb. 2022.

[82] M. Kim, S. Tariq, and S. S. Woo, "FReTAL: Generalizing deepfake detection using knowledge distillation and representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1001–1012.

[83] S. Kang, D. Lee, W. Kweon, and H. Yu, "Personalized knowledge distillation for recommender system," *Knowl.-Based Syst.*, vol. 239, Mar. 2022, Art. no. 107958.

[84] Y. Liu, K. Wang, G. Li, and L. Lin, "Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 5573–5588, 2021.

[85] J. Zhu et al., "Complementary relation contrastive distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9260–9269.

[86] C. Zhang, Y. Zou, G. Chen, and L. Gan, "EAR: Efficient action recognition with local-global temporal aggregation," *Image Vis. Comput.*, vol. 116, Dec. 2021, Art. no. 104329.

[87] Z. Guo, X. Zheng, X. Chun, and S. Ying, "ADCI-Net: An adaptive discriminative clip identification strategy for fast video action recognition," *J. Electron. Imag.*, vol. 30, no. 2, Apr. 2021, Art. no. 023030.

[88] X. Yang, L. Kong, and J. Yang, "Unsupervised motion representation enhanced network for action recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2445–2449.

[89] L. Huang, L. Wang, and H. Li, "Multi-modality self-distillation for weakly supervised temporal action localization," *IEEE Trans. Image Process.*, vol. 31, pp. 1504–1519, 2022.

[90] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3556–3565.

[91] P. Zhao, L. Xie, J. Wang, Y. Zhang, and Q. Tian, "Progressive privileged knowledge distillation for online action detection," *Pattern Recognit.*, vol. 129, Sep. 2022, Art. no. 108741.

[92] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5191–5198.

[93] C. Tan and J. Liu, "Online knowledge distillation with elastic peer," *Inf. Sci.*, vol. 583, pp. 1–13, Jan. 2022.

[94] S. Li et al., "Distilling a powerful Student model via online knowledge distillation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 7, 2022, doi: 10.1109/TNNLS.2022.3152732.

[95] G. Li, R. Togo, T. Ogawa, and M. Haseyama, "Self-knowledge distillation based self-supervised learning for COVID-19 detection from chest X-ray images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1371–1375.

[96] S. Park, J. Kim, and Y. S. Heo, "Semantic segmentation using pixel-wise adaptive label smoothing via self-knowledge distillation for limited labeling data," *Sensors*, vol. 22, no. 7, p. 2623, Mar. 2022.

[97] X. Wang et al., "Self-knowledge distillation for the object segmentation based on atrous spatial pyramid," *J. Phys., Conf.*, vol. 2294, no. 1, 2022, Art. no. 012023.

[98] C. Xu, W. Gao, T. Li, N. Bai, G. Li, and Y. Zhang, "Teacher–student collaborative knowledge distillation for image classification," *Appl. Intell.*, pp. 1–13, Mar. 2022.

[99] S. Yun, J. Park, K. Lee, and J. Shin, "Regularizing class-wise predictions via self-knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 53, Jun. 2020, pp. 13876–13885.

[100] 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.

[101] W. Kay et al., "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[102] 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.

[103] B. Wolfe, B. Seppelt, B. Mehler, B. Reimer, and R. Rosenholtz, "Rapid holistic perception and evasion of road hazards," *J. Exp. Psychol., Gen.*, vol. 149, no. 3, p. 490, 2020.

[104] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "An improved one millisecond mobile backbone," 2022, *arXiv:2206.04040*.

[105] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[106] R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "The PREVENTION dataset: A novel benchmark for prediction of vehicles intentions," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3114–3121.

[107] R. D. Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 269–284.

[108] M. Biparva, D. Fernández-Llorca, R. I. Gonzalo, and J. K. Tsotsos, "Video action recognition for lane-change classification and prediction of surrounding vehicles," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 569–578, Sep. 2022.

[109] C. Li, J. Zhang, and J. Yao, "Streamer action recognition in live video with spatial–temporal attention and deep dictionary learning," *Neurocomputing*, vol. 453, pp. 383–392, Sep. 2021.

Chen Chen (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in telecommunication from Xidian University, Xi'an, China, in 2000, 2006, and 2008, respectively. He is currently a Professor with the Department of telecommunication, Xidian University, and a member of the State Key Laboratory of Integrated Service Networks, Xidian University. He is also the Director of the Xi'an Key Laboratory of Mobile Edge Computing and Security and the Director of the Intelligent Transportation Research Laboratory, Xidian University. He was a Visiting Professor at the Department of EECS, University of Tennessee, and the Department of CS, University of California. He serves as the general chair, the PC chair, the workshop chair or a TPC member for a number of conferences. He has authored/coauthored two books, over 130 scientific papers in international journals and conference proceedings. He has contributed to the development of five copyrighted software systems and invented over 100 patents. He is also a Senior Member of China Computer Federation (CCF) and China Institute of Communications (CIC).

Guorun Yao received the B.S. degree in communication engineering from Xidian University in 2020, where he is currently pursuing the M.S. degree majoring in transportation engineering. His research interests include vehicular networking, machine learning, and edge computing.

Lei Liu (Member, IEEE) received the B.Eng. degree in electronic information engineering from Zhengzhou University, Zhengzhou, China, in 2010, and the M.Sc. and Ph.D. degrees in communication and information systems from Xidian University, Xi'an, China, in 2013 and 2019, respectively. From 2013 to 2015, he worked with a subsidiary of China Electronics Corporation. From 2018 to 2019, he was supported by China Scholarship Council to be a Visiting Ph.D. Student with the University of Oslo, Oslo, Norway. From 2020 to 2022, he was a Lecturer with the School of Telecommunications Engineering, Xidian University, where he is currently an Associate Professor with the Xidian Guangzhou Institute of Technology. His research interests include vehicular ad hoc networks, intelligent transportation, edge intelligence, and distributed computing.

Qingqi Pei (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in computer science and cryptography from Xidian University in 1998, 2005, and 2008, respectively. He is currently a Professor and a member of the State Key Laboratory of Integrated Services Networks. He is also a Professional Member of ACM and a Senior Member of Chinese Institute of Electronics and China Computer Federation. He is also the Director of the Shaanxi Key Laboratory of Blockchain and Secure Computing. His research interests include digital contents protection and wireless networks and security.

Houbing Song (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in August 2012.

He is currently a Tenured Associate Professor and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab) (www.SONGLab.us), University of Maryland, Baltimore County (UMBC), Baltimore, MD, USA. Prior to joining UMBC, he was a Tenured Associate Professor in electrical engineering and computer science at Embry-Riddle Aeronautical University, Daytona Beach, FL, USA. He is the editor of eight books, the author of more than 100 articles, and the inventor of two patents. His research interests include cyber-physical systems/the Internet of Things, cybersecurity and privacy, and AI/machine learning/big data analytics. His research has been sponsored by federal agencies (including National Science Foundation, U.S. Department of Transportation, and Federal Aviation Administration, among others) and industry. His research has been featured by popular news media outlets, including IEEE GlobalSpec's Engineering360, Association for Uncrewed Vehicle Systems International (AUVSI), Security Magazine, CXOTech Magazine, Fox News, U.S. News & World Report, The Washington Times, and New Atlas.

Dr. Song is an ACM Distinguished Member and an ACM Distinguished Speaker. He is a Highly Cited Researcher identified by Clarivate in 2021 and 2022 and a Top 1000 Computer Scientist identified by Research.com. He received Research.com Rising Star of Science Award in 2022 (World Ranking: 82; U.S. Ranking: 16). He was a recipient of more than ten Best Paper Awards from major international conferences, including IEEE CPSCom-2019, IEEE ICII 2019, IEEE/AIAA ICNS 2019, IEEE CBDCom 2020, WASA 2020, AIAA/ IEEE DASC 2021, IEEE GLOBECOM 2021, and IEEE INFOCOM 2022. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE since 2023, IEEE INTERNET OF THINGS JOURNAL since 2020, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS since 2021, and IEEE JOURNAL ON MINIATURIZATION FOR AIR AND SPACE SYSTEMS since 2020. He was an Associate Technical Editor of *IEEE Communications Magazine* (2017–2020).

Schahram Dustdar (Fellow, IEEE) is currently a Professor in computer science with the Distributed Systems Group, TU Vienna, Austria. He was an Honorary Professor in information systems at the University of Groningen, The Netherlands, from 2004 to 2010, a Visiting Professor at the University of Sevilla, Spain, from 2016 to 2017, and a Visiting Professor at the University of California at Berkeley, USA, in 2017. He is also an Elected Member of the Academia Europaea, where he is the Chairperson of the Informatics Section. He was a recipient of the ACM Distinguished Scientist Award in 2009, the IBM Faculty Award in 2012, and the IEEE TCSVC Outstanding Leadership Award for outstanding leadership in services computing in 2018. He is the Co-Editor-in-Chief of the *ACM Transactions on Internet of Things* and the Editor-in-Chief of *Computing* (Springer). He is also an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the *ACM Transactions on the Web*, and the *ACM Transactions on Internet Technology*. He serves on the Editorial Board for IEEE INTERNET COMPUTING and the *IEEE Computer* magazine.