Received August 19, 2019, accepted August 30, 2019, date of publication September 11, 2019,
date of current version September 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939532

An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems

DAXIN TIAN, (Senior Member, IEEE), CHUANG ZHANG, XUTING DUAN, AND XIXIAN WANG
Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

Corresponding author: Xuting Duan (duanxuting@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672082 and Grant 61822101, and in part by the Beijing Municipal Natural Science Foundation under Grant 4181002.

ABSTRACT Car accidents cause a large number of deaths and disabilities every day, a certain proportion of which result from untimely treatment and secondary accidents. To some extent, automatic car accident detection can shorten response time of rescue agencies and vehicles around accidents to improve rescue efficiency and traffic safety level. In this paper, we proposed an automatic car accident detection method based on Cooperative Vehicle Infrastructure Systems (CVIS) and machine vision. First of all, a novel image dataset CAD-CVIS is established to improve accuracy of accident detection based on intelligent roadside devices in CVIS. Especially, CAD-CVIS is consisted of various kinds of accident types, weather conditions and accident location, which can improve self-adaptability of accident detection methods among different traffic situations. Secondly, we develop a deep neural network model YOLO-CA based on CAD-CVIS and deep learning algorithms to detect accident. In the model, we utilize Multi-Scale Feature Fusion (MSFF) and loss function with dynamic weights to enhance performance of detecting small objects. Finally, our experiment study evaluates performance of YOLO-CA for detecting car accidents, and the results show that our proposed method can detect car accident in 0.0461 seconds (21.6FPS) with 90.02% average precision (AP). In additionally, we compare YOLO-CA with other object detection models, and the results demonstrate the comprehensive performance improvement on the accuracy and real-time over other models.

INDEX TERMS Car accident detection, CVIS, machine vision, deep learning.

I. INTRODUCTION

According to the World Health Organization, there are about 1.35 million deaths and 20-50 million injuries as a result of the car accident globally every year [1]. Especially, a certain proportion of deaths and injuries are due to untimely treatment and secondary accidents [2], which results from that rescue agency and vehicles around accident cannot obtain quick response about the accident [3], [4]. Therefore, it is vital important to develop an efficient accident detection method, which can significantly reduce both the number of deaths and injuries as well as the impact and severity of accidents [5]. Under this background, many fundamental projects and studies to develop efficient detection method have been launched for developing and testing [6]–[10].

The traditional methods utilize vehicle motion parameters captured by vehicular GPS devices to detect car accident, such as acceleration and velocity. However, these methods based on single type of features cannot meet the performance need of accident detection in the aspect of accuracy and real-time. With the development of computer and communication technologies, Cooperative Vehicle Infrastructure System and Internet of Vehicles have been developed rapidly in recent years [11]–[13]. Moreover, the image recognition based on video captured by intelligent roadside devices in CVIS has become one of research hotspots in the field of intelligent transportation system [14], [15]. For traffic situation awareness, image recognition technology has advantages of high efficiency, flexible installation and low maintenance costs. Therefore, the image recognition has been applied to detection pedestrian, vehicle, traffic sign and so on successfully [16]–[20]. In generally, there are many distinctive image and video features in traffic accidents, such as vehicle collision, rollover and so on. To some extent, these features can be used to detect or predict car accidents. Accordingly, some researchers apply the machine vision technology based on

The associate editor coordinating the review of this manuscript and approving it for publication was Xianye Ben.

deep-learning into methods of car accident detection. These methods extract and process complex image features instead of single vehicle motion parameter, which improves the accuracy of detecting car accidents. However, the datasets of these methods are mostly captured by car cameras or cell phones of pedestrian, which is not suitable for roadside devices in CVIS. In additionally, the reliability and real-time performance of these methods need to be improved to meet the requirements of car accident detection.

In this paper, we propose a data-driven car accident detection method based on CVIS, whose goal is improving efficiency and accuracy of car accident response. With the goal, we focus on such a general application scenario when there is an accident on the road, roadside intelligent devices recognize and locate it efficiently. First, we build a novel dataset, Car Accident Detection for Cooperative Vehicle Infrastructure System dataset (CAD-CVIS), which is more suitable for car accident detection based on roadside intelligent devices in CVIS. Then, a deep learning model YOLO-CA based on CAD-CVIS is developed to detect car accident. Especially, we optimize the network of traditional deep learning models YOLO [21] to build network of YOLO-CA, which is more accurate and fast in detecting car accident. In additionally, considering of wide shooting scope of roadside cameras in CVIS, multi-scale feature fusion method and loss function with dynamic weights are utilized to improve performance of detecting small objects.

The rest of this paper is organized as follows: Section 2 gives an overviews of related work. We present the details of our proposed method in Section 3. The performance evaluation is discussed in Section 4. Finally, Section 5 conclude this paper.

II. RELATED WORK

The car accident detection and notification method is a challenging issue and has attracted a lot of attention from researchers. They have proposed and applied various car accident detection methods. In generally, car accident detection methods are mainly divided into the following two kinds: vehicle running condition-based and accident video features-based.

A. METHOD BASED ON VEHICLE RUNNING CONDITION

When an accident occurs, the motion state of the vehicle will change dramatically. Therefore, many researchers proposed the accident detection method by monitoring motion parameters, such as acceleration, velocity and so on. Reference [22] used On Board Diagnosis (OBD) system to monitor speed and engine status to detect a crash, and utilized smart-phone to report the accident by Wi-Fi or cellular network. Reference [23] developed an accident detection and reporting system using GPS, GPRS, and GSM. The speed of vehicle obtained from High Sensitive GPS receiver is considered as the index for detecting accidents, and the GSM/GPRS modem is utilized to send the location of the accident. Reference [24] presented a prototype system called e-NOTIFY, which monitors the change of acceleration to detect accident and utilize V2X communication technologies to report it. To a certain extent, these methods can detect and report car accidents in short time, and improve the efficiency of car accidents warning. However, the vehicle running condition before car accidents is complex and unpredictable, and the accuracy of accident detection only based on speed and acceleration may be low. In addition, they rely too heavily on vehicular monitoring and communication equipment, which may be unreliable or damaged in some extreme circumstances, such as heavy canopy, underground tunnel, and serious car accidents.

B. METHOD BASED VIDEO FEATURES

With the development of machine vision and artificial neural network technology, more and more applications based on video processing have been applied in transportation and vehicle fields. Under this background, some researchers utilized video features of the car accident to detect it. Reference [25] presented a Dynamic-Spatial-Attention Recurrent Neural Network (RNN) for anticipating accidents in dashcam videos, which can predict accidents about 2 seconds before they occur with 80% recall and 56.14% precision. Reference [26] proposed a car accident detection system based on first-person videos, which detected anomalies by predicting the future locations of car participants and then monitoring the prediction accuracy and consistency metrics. These methods also have some limitations because of low penetration of vehicular intelligent devices and shielding effects between vehicles.

There are also some other methods which use roadside devices instead of vehicular equipments to obtain and process video. Reference [27] proposed a novel accident detection system at intersection, which composed background images from image sequence and detected accidents by using Hidden Markov Model. Reference [28] outlined a novel method for modeling of interaction among multiple moving objects, and used the Motion Interaction Field to detect and localize car accidents. Reference [29] proposed a novel approach for automatic road accident detection, which was based on detecting damaged vehicles from footage received from surveillance cameras installed in roads. In this method, Histogram of gradients (HOG) and Gray level co-occurrence matrix features were used to train support vector machines. Reference [30] presented a novel dataset for car accidents analysis based on traffic Closed-Circuit Television (CCTV) footage, and combined Faster Regions-Convolutional Neural Network (R-CNN) and Context Mining to detect and predict car accidents. The method in [30] achieved 1.68 seconds in terms of Time-To-Accident measure with an Average Precision of 47.25%. Reference [8] proposed a novel framework for automatic car accident detection, which learned feature representation from the spatio-temporal volumes of raw pixel intensity instead of traditional hand-crafted features. The experiments of method in [8] demonstrated it can detect on average 77.5% accidents correctly with 22.5% false alarms.

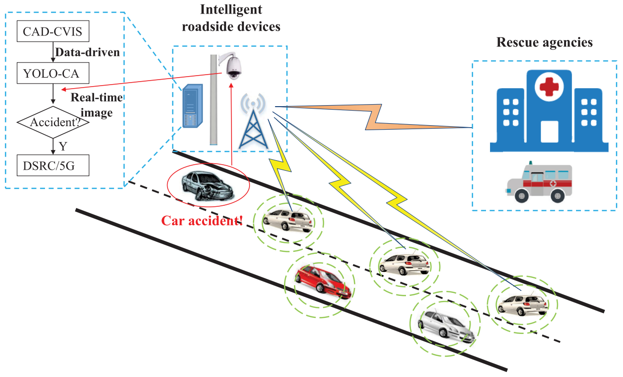


FIGURE 1. The application scenario of the automatic car accident detection method based on CVIS.

Compared with the methods based on vehicle running condition, these methods improve the detection accuracy and some of them even can predict accidents about 2 seconds before they occur. To some extent, these methods are significant in decreasing the accident rate and improving traffic safety. However, the detection accuracy of these methods is low and the error rate is high, and the wrong accident information will have a great impact on the normal traffic flow. Concerning the core issue mentioned above, in order to avoid the drawbacks of vehicular cameras, our proposed method utilizes the roadside intelligent edge devices to obtain traffic video and process image. Moreover, for sake of improving the accuracy of accident detection method based on intelligent roadside devices, we establish the CAD-CVIS dataset based on video sharing websites, which is consisted of various kinds of accident types, weather conditions and accident locations. Moreover, we develop the model YOLO-CA to improve the reliability and real-time performance among different traffic conditions by combining deep learning algorithms and MSFF method.

III. METHODS

A. METHOD OVERVIEW

The Fig. 1 shows the application principle of our proposed car accident detection method based CVIS. Firstly, the car accident detection application program with YOLO-CA model is deployed on the edge server, which is developed based on CAD-CVIS and deep learning algorithms. Then edge server receives and processes the real-time image captured by roadside cameras. Finally, the roadside communication unit will broadcast the accident emergency messages to the relevant vehicles and rescue agencies by DSRC and 5G networks. In the rest of this section, we will present the details of CAD-CVIS and YOLO-CA model.

B. CAD-CVIS

1) DATA COLLECTION AND ANNOTATION

There are two major challenges in collecting car accidents data:(1) Access: access to roadside traffic cameras data is often limited. In addition, the accident data from transportation administration is often not available for public uses because of many legal reasons. (2) Abnormality: car accidents are rare in the road compared with normal traffic conditions. In this work, we try to draw support from video sharing websites to search the videos and images including car accidents, such as news report and documentary. In order to improve the applicability of our proposed method to roadside edge device, we only pick out the videos and images captured from a traffic CCTV footage.

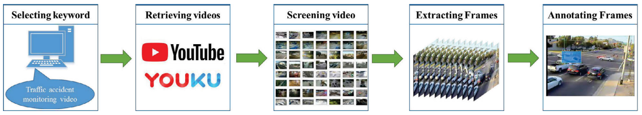


FIGURE 2. Data collection and annotation for the CAD-CVIS dataset.

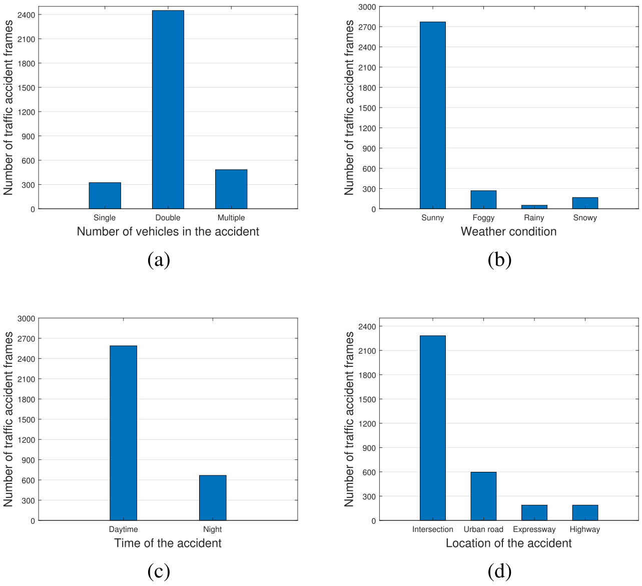


FIGURE 3. Number of accident frames in CAD-CVIS categorized by different indexes. (a) Accident Type (b) Weather condition (c) Accident time (d) Accident location.

Through the above steps, we obtain 633 car accidents scenes, 3255 accident key frames and 225206 normal frames. Moreover, the car accident scene only occupies a small part of each accident frame. We utilize LabelImg [31] to annotate the location of the accident in each frame in detail to enhance the accuracy of locating accident. The high accuracy enables emergency message be sent to the vehicles that are in the same direction as accident more efficiently and decrease the impact to the vehicles that are in the opposite direction. The whole steps of data collection and annotation are shown in Fig. 2. The CAD-CVIS dataset is made available for research use through https://github.com/zzzzzzc/Car-accident-detection.

2) STATISTICS OF THE CAD-CVIS

Statistics of the CAD-CVIS dataset can be found in Fig. 3.It can be found that the CAD-CVIS dataset includes various types of car accidents, which can improve the adaptability of our method to different conditions. According to the number of vehicles in the accident, the CAD-CVIS dataset includes 323 Single Vehicle Accident frames, 2449 Double Vehicle Accidents frames and 483 Multiple Vehicle Accidents

TABLE 1. Comparison between CAD-CVIS and related datasets.

Dataset name	Scenes	Frames or Duration	A	R	M
UCSD Ped2	77	1636 frames	×	✓	×
CUHK Avenue	47	3820 frames	×	×	✓
DAD	620	2.4 hours	✓	×	✓
CADP	1416	5.2 hours	×	✓	✓
CAD-CVIS	**632**	**3255+225206 frames**	✓	✓	✓

frames. Moreover, the CAD-CVIS dataset covers a variety of weather conditions, such as 2769 accident frames under sunny condition, 268 frames under foggy condition, 52 accident frames under rainy condition and 166 accident frames under snowy condition. Besides, there are 2588 frames of accidents in the daytime and 667 accident frames at night. In addition, the CAD-CVIS dataset contains 2281 frames of accidents occurring at the intersection, 596 frames in the urban road, 189 frames in the expressway and 189 frames in the highway.

Comparison between CAD-CVIS and related datasets can be found in Table. 1. The A in Table. 1 responses that there is annotation of car accident in the dataset. R responses that the videos and frames captured from the roadside CCTV footage. M responses that there are multiple road conditions in dataset. Compared with CUHK Avenue [32], UCSD Ped2 [33] and DAD [25], CAD-CVIS contains more car accident scenes, which can improve the adaptability of model based on CAD-CVIS. Moreover, the frames of CAD-CVIS are all captured from roadside CCTV footage, which is more suitable for the accident detection methods based on intelligent roadside devices in CVIS.

C. OUR PROPOSED DEEP NEURAL NETWORK MODEL

In the task of car accident detection, we must not only judge whether there is a car accident in the image, but also accurately locate the car accident. That's because the accurate location guarantees that the RSU can broadcast the emergency message to the vehicles affected by the accident. The classification and location algorithms can be divided into two kinds:(1) Two stage model, such as R-CNN [34], Fast R-CNN [35], Faster R-CNN [36] and Faster R-CNN with FPN [37]. These algorithms utilize selective research and Region Proposal Network (RPN) to select about 2000 proposal regions in the image, and then detection objects by the features of these regions extracted by CNN. These region-based models locate objects accurately, but extracting proposals take a great deal of time. (2) One stage model, such as YOLO [21](You Only Look Once) and SSD (Single Shot MultiBox Detector) [38]. These algorithms implement location and classification by one CNN, which can provide end to end detection service. Because of eliminating the process of selecting the proposal regions, these algorithms are very fast and still has guaranteeing accuracy. Considering that accident detection requires high real-time performance, we design the deep neural network based on one-stage model YOLO [21].

1) NETWORK DESIGN

YOLO utilizes its particular CNN to complete classification and location of multiple objects in an image at one time. In the training process of YOLO, each image is divided into $S \times S$ grids. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object [39]. This design can improve the detection speed dramatically and the detection accuracy with reference to global features. However, it also will cause serious detection error when there are more than one objects in one grids. Roadside cameras have a wide scope of shooting, the accident area may be small in the image. Inspired of the multi-scale feature fusion (MSFF) network, in order to improve the performance of model to detect small objects, we utilize 24 layers to achieve image upsampling and obtain two different dimensional output tensors. This new car accident detection model is called as YOLO-CA, and the network structure diagram of YOLO-CA is shown as Fig. 4.

TABLE 2. Composition of YOLO-CA network.

Layer name	Number
Input	1
Convolution	65
Batch Normalization	65
Leaky ReLU	65
Zero Padding	5
Add	23
Upsampling	1
Concatenate	1
Total	**228**

As shown in Fig. 4, YOLO-CA is composed of 228 neural network layers, and the number of each kind of layer is shown in Table. 2. These layers constitute many kinds of basic components of YOLO-CA network, such as DBL and ResN. The DBL is the minimum components of YOLO-CA network, which is composed of Convolution layer, Batch Normalization layer and Leaky ReLU layer. ResN consists of Zero Padding layer, DBL and N Resblock_units [40], which is designed to avoid neural network degradation caused by increased depth. Ups in Fig. 4 is upsampling layer, which is utilized to improve the performance of YOLO-CA to detect small objects. Concat is concatenate layer, which is used to concatenate the layer in Darknet-53 and upsampling layer.

2) DETECTION PRINCIPLE

Fig. 5 shows the detection principle of YOLO-CA, which includes extracting feature map and predicting bounding box. As shown in Fig. 5, YOLO-CA divides the input image into 13×13 grid and 26×26 grid. The first grid is responsible for detecting the large objects, whereas the second grid makes up for the inaccuracy of small target detection in the first grid. The feature extraction networks corresponding to these two grids are different, but the detection models of the objects is similar. For ease of presentation, we regard the first grid as example to explain the training steps of YOLO-CA. The center of car accident region falls into the grid cell $(7, 5)$, so this cell is responsible for detecting this car accident in the whole training process. Then the cell $(7, 5)$ will predict three bounding boxes, and each boxes includes six parameters: x, y, w, h, CS, p. The (x, y) is the center point of the bounding

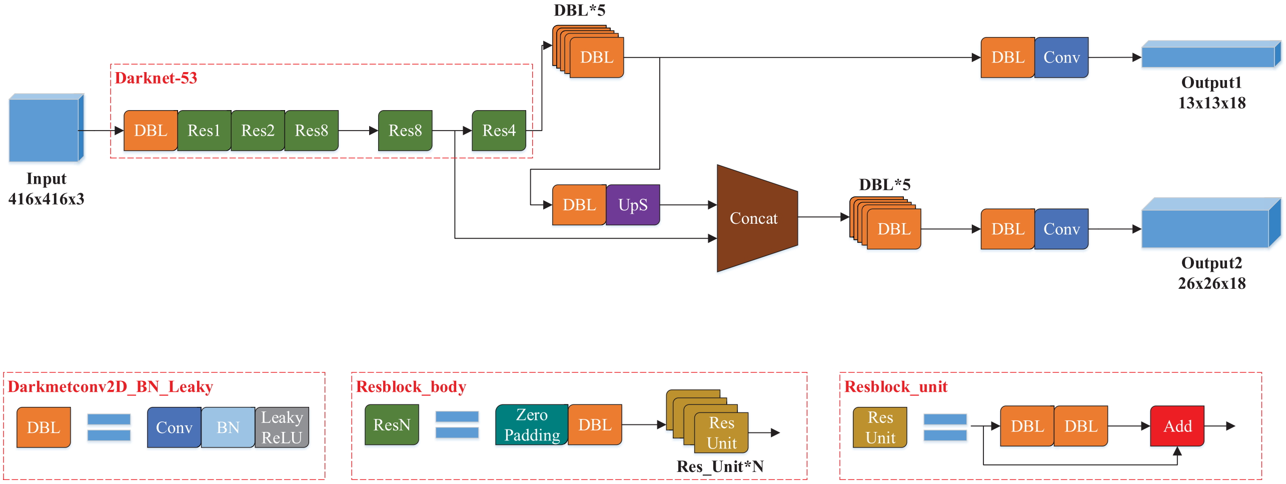


FIGURE 4. The network structure of YOLO-CA.

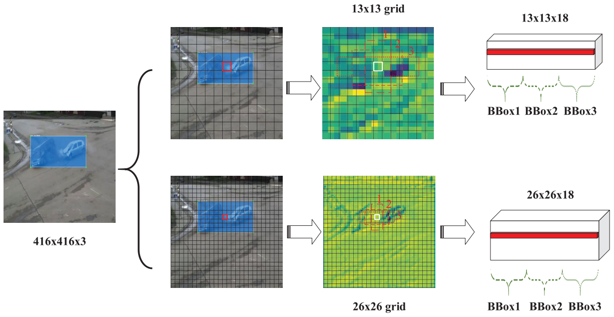


FIGURE 5. The detection principle of YOLO-CA.

box, and the (w, h) is the ratio of width and height of the bounding box to the whole image. The *CS* is confidence score of bounding box, which represents how confident the model is that the bounding box contains an object and how accurate it thinks the box is that it predicts. Lastly, each bounding box will predict class probability of car accident *p*.

After the training of a batch of images, the loss of model will be calculated, which is utilized to adjust the weights of parameters. In the calculation of loss, let the ground truth of an object is $x^*, y^*, w^*, h^*, CS^*, p^*$. $S \times S$ is the number of cells in grid, and *B* is the number of predicted bounding boxes of each grid cell. For each grid cell, the $Pr(Objects)$ equals 1 when the cell contains center of object, whereas it equals 0 when there is not center of object in the cell. For each image, the loss of YOLO-CA is divided into the following four parts:

- Loss of (x, y), which is calculated by (1). Where *BCL* is binary cross entropy loss function, and the *areaTure* is defined as $w^* * h^*$.

$$\begin{aligned} Loss_{xy} &= \sum_{i=1}^{S\times S}\sum_{j=1}^{B} Pr\left(Objects\right)\left(2 - areaTure_{ij}\right) \\ &\quad * \left[BCL\left(x_{ij}\right) + BCL\left(y_{ij}\right)\right] \\ BCL\left(x_{ij}\right) &= x_{ij}^* \log x_{ij} + \left(1 - x_{ij}^*\right)\log\left(1 - x_{ij}\right) \\ BCL\left(y_{ij}\right) &= y_{ij}^* \log y_{ij} + \left(1 - y_{ij}^*\right)\log\left(1 - y_{ij}\right) \end{aligned} \tag{1}$$

- Loss of (w, h), which is calculated by (2). Where *SD* is square difference function. Especially, the $(2 - areaTure_{ij})$ in (1) and (2) is utilized to increase the error punishment of small objects. Because that the same errors of x, y, w, h cause more serious impact on the detection effect of small object than that of large object.

$$\begin{aligned} Loss_{wh} &= \sum_{i=1}^{S\times S}\sum_{j=1}^{B} Pr\left(Objects\right)\left(2 - areaTure_{ij}\right) \\ &\quad * \frac{1}{2}\left[SD\left(w_{ij}\right) + SD\left(h_{ij}\right)\right] \\ SD\left(w_{ij}\right) &= \left(w_{ij} - w_{ij}^*\right)^2 \\ SD\left(h_{ij}\right) &= \left(h_{ij} - h_{ij}^*\right)^2 \end{aligned} \tag{2}$$

- Loss of *CS*, which is calculated by (3). The loss of *CS* can be divided into two parts: the confidence loss of foreground and confidence loss of background.

$$\begin{aligned} Loss_{CS} &= \sum_{i=1}^{S\times S}\sum_{j=1}^{B} Pr\left(Objects\right) * BCL\left(CS_{ij}\right) \\ &\quad + \left(1 - Pr\left(Objects\right)\right) * BCL\left(CS_{ij}\right) \\ BCL\left(CS_{ij}\right) &= CS_{ij}^* \log CS_{ij} \\ &\quad + \left(1 - CS_{ij}^*\right)\log\left(1 - CS_{ij}\right) \end{aligned} \tag{3}$$

where CS^* is defined by (4). In additionally, the IoU_p^t is defined in Fig. 6, which equals the intersection over union (IoU) between the predicted bounding box and the ground truth.

$$CS^* = \Pr(Objects) * IoU_p^t \tag{4}$$

- Loss of *p*, which is calculated by (5)

$$\begin{aligned} Loss_{CS} &= \sum_{i=1}^{S\times S}\sum_{j=1}^{B} Pr\left(Objects\right) BCL\left(p_{ij}\right) \\ BCL\left(p_{ij}\right) &= p_{ij}^* \log p_{ij} + \left(1 - p_{ij}^*\right)\log\left(1 - p_{ij}\right) \end{aligned} \tag{5}$$

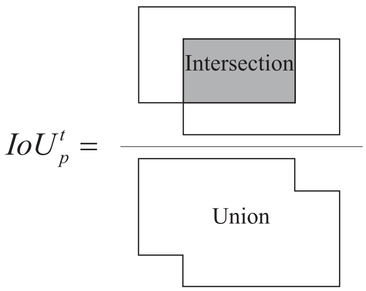


FIGURE 6. The definition of IoU.

For each image in training set, the total loss is defined as (6). Especially, because that the multi-scale feature fusion is used in YOLO-CA, the loss is the sum of conditions under $S = 13$ and $S = 26$. In additionally, the loss of each batch of images is defined as (7).

$$Loss\_img = Loss_{xy} + Loss_{wh} + Loss_{CS} + Loss_p \quad (6)$$

$$Loss = \frac{1}{b}\sum_{k=1}^{b} Loss\_img_k \quad (7)$$

where the b in (7) is the size of batch.

IV. EXPERIMENT

In this section, we evaluate our proposed model YOLO-CA on the CAD-CVIS dataset. First, we give the training results of YOLO-CA, which include the change process of several performance indexes. Then, we show the results of some comparative experiments between YOLO-CA and other detection models. Finally, the visual results are demonstrated among various types and scales of car accident objects.

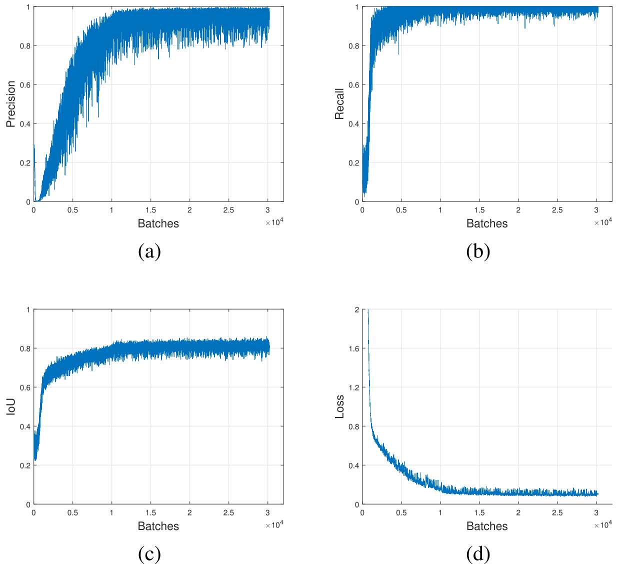
A. IMPLEMENTATION DETAILS

We implement our model in TensorFlow [41] under the operating system Ubuntu 18.04 and perform experiments on a system with Nvidia Titan Xp GPU. We divide the CAD-CVIS dataset into three parts: (1) Training set (80%), which is used to train the parameter weight of network. (2) Validation set (5%), which is utilized to adjust hyperparameters, such as learning rate and drop out rate. (3) Test set (15%), which is used to evaluate the performance of different algorithms for detecting car accident. In additionally, each part of dataset contains all types of accident in Fig. 3. The batch size is set to 64, and the models are trained for up to 30000 iterations. The initial learning rate is set to 0.001, and updating with iteration parameter of 0.1/10000 iterations. The SGD optimizer with a momentum of 0.9 is utilized to adjust parameters of network. Moreover, we use a weight decay of 0.0005 to prevent model overfitting.

B. RESULTS AND ANALYSIS

1) TRAINING RESULTS OF YOLO-CA

Fig. 7 shows the training results of YOLO-CA, including the changes of precision, recall, IoU and loss of each batch in iteration process. In the training process of YOLO-CA, we regard the prediction result with IoU over 0.5 and right classification as true result, and other predictions are all false results. As shown in Table. 3, the prediction results can be divided into four parts: (1) TP: Truth Positive. (2) FP: False Positive. (3) FN: False Negative. (4) TN: True Negative. The precision is defined as $precision = \frac{TP}{TP+FP}$ and recall is defined as $recall = \frac{TP}{TP+FN}$.



(a) (b) (c) (d)

FIGURE 7. The training results of YOLO-CA. (a) Precision (b) Recall (c) IoU (d) Loss.

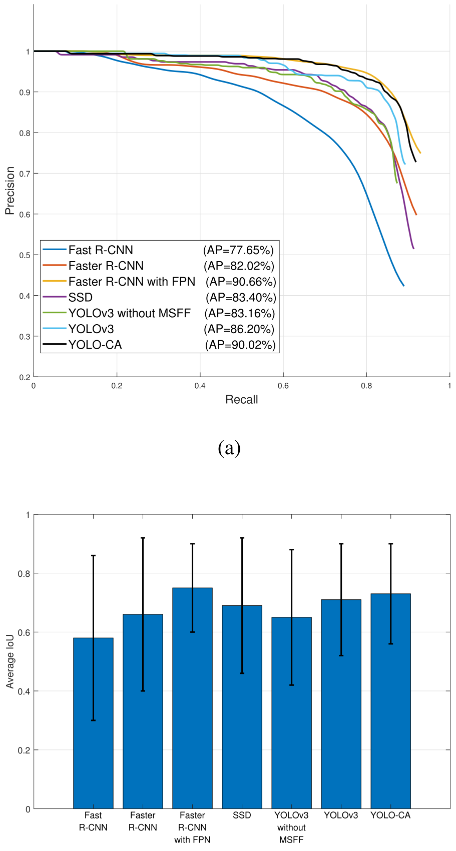
TABLE 3. Distribution map of prediction results.

		Ground truth	
		Positive	Negative
Prediction Result	Positive	TP	FP
	Negative	FN	TN

As shown in Fig. 7a, with the increasing of iterations, the precision of YOLO-CA is increasing gradually and converge over 90%. Moreover, recall eventually converges to more than 95%. In terms of locating performance of YOLO-CA in training set, IoU finally stabilizes above 0.8. The Fig. 7d shows the decreasing process of loss of YOLO-CA in (7), and the final convergence of loss is less than 0.2.

2) COMPARATIVE EXPERIMENTS AND VISUAL RESULTS

The comparative experiments are conducted for comparing seven detection models: (1) One-stage models: SSD, our proposed YOLO-CA, traditional YOLO-v3 and YOLO-v3 without MSFF (Multi-Scale Feature Fusion). (2) Two-stage models: Fast R-CNN, Faster R-CNN and Faster R-CNN with FPN. In order to comparatively demonstrate the validation of YOLO-CA as well as confirm its strength in terms of the comprehensive performance on the accuracy and real-time, the following indexes are selected for comparison among the seven models:



(a)

(b)

FIGURE 8. The AP and IoU results of different models. (a) Precision-Recall curve (b) Average IoU.

- Average Precision (AP) that is defined as the average value of precision under different recall, which can be changed by adjusting threshold of classification confidence. AP index evaluate the accuracy performance of detection models. The average precision can be calculated by (8).

$$AP = \int_0^1 precision(r) \tag{8}$$

where r is recall.

- Average Intersect over Union (Average IoU) that is utilized to evaluate the object locating performance of detection models. The Average IoU is the average value of IoUs between every prediction bounding box and corresponding ground truth.
- Frames Per Second (FPS). Inference time is defined as the average time cost of detecting a frame among test set. FPS is the reciprocal of inference time, which is defined as the average number of frames that can be detected in one second. This indexes evaluate real-time performance of detection models.

The Fig. 8a shows the Precision-Recall curve of detection models among test set. It can be found that Faster R-CNN with FPN and our proposed YOLO-CA have obvious advantages in accuracy performance than the other models. In additionally, our-proposed YOLO-CA can achieve 90.02% of AP, which is slightly lower than that of Faster R-CNN with FPN (90.66%) and higher than those of Fast R-CNN (77.66%), Faster R-CNN (82.02%), SSD (83.40%), YOLOv3 without MSFF (83.16%) and YOLOv3 (86.20%). Average IoU is a vital important index to evaluate locating performance of detection models. Moreover, accurate location is critical to car accident detection and notification, and higher locating performance can improve the safety of the vehicles around accident. As shown in Fig. 8b, YOLO-CA can achieve about 0.73 of Average IoU, which is lower than that of Faster R-CNN with FPN (0.75) and higher than those of Fast R-CNN (0.58), Faster R-CNN (0.66), SSD (0.69), YOLOv3 without MSFF (0.65) and YOLOv3 (0.71).

In order to compare and analysis the performance of models in details, the objects of test set is divided into three parts according to different scales of objects:(1) Large: the area of object is larger than one tenth of image size. (2) Medium: the area of object is over the interval [1/100, 1/10] of image size. (3) Small: the area of object is less than one-hundredth of image size.

The Table. 4 shows the AP and IoU results of the seven models among different scales of object. We can intuitively see that the scales of objects significantly affect the accuracy and locating performance of detection models. It can be found that our proposed YOLO-CA has obvious advantages in AP and Average IoU than Fast R-CNN, Faster R-CNN and YOLOv3 without MSFF, especially among small scale of objects. There is not MSFF process in the above three models, which results in that they detection the objects only rely on the top-level features. However, although there is rich semantic information in top-level features, the location information of objects is rough, which does not benefit to locate the bounding box of objects correctly. On the contrary, there is little semantic information in low-level features with high resolution, but the location information of objects is accurate. For small scales of objects, they make up a small proportion of the whole frame, and their location information is easily lost through multiple convolution processes. YOLO-CA utilizes MSFF to combine top-level features and low-level features, and then makes a prediction in each fused feature layer. This process reserves the rich semantic information and accurate location information simultaneously, so YOLO-CA has better performance in AP and Average IoU than Fast R-CNN, Faster R-CNN and YOLOv3 without MSFF. For SSD, it uses pyramidal feature hierarchy to obtain multi-scale feature maps. But to avoid using low-level features SSD foregoes reusing already computed layers and instead builds the pyramid starting from high up in the network and then by adding several new layers. So SSD misses the opportunity to reuse the higher-resolution maps of feature hierarchy, which are vital important for detecting small objects [37]. Moreover, the performance of backbone of YOLO-CA (Darknet53) is better than that of SSD (VGG-16) because of using residual networks to avoid degradation problem of deep neural network.

TABLE 4. AP and IoU results of different models among different scales of object.

	AP(%)			Average IoU		
Method	Large	Medium	Small	Large	Medium	Small
Fast R-CNN	82.11	80.3	49.20	0.68	0.63	0.46
Faster R-CNN	89.66	86.79	56.79	0.71	0.69	0.51
Faster R-CNN with FPN	93.07	92.15	78.63	0.74	0.72	0.68
SSD	91.37	89.65	60.59	0.73	0.71	0.58
YOLOv3 without MSFF	90.18	90.40	58.89	0.72	0.68	0.50
YOLOv3	90.45	91.17	67.78	0.74	0.73	0.60
YOLO-CA	**93.87**	**91.51**	**76.51**	**0.76**	**0.74**	**0.64**

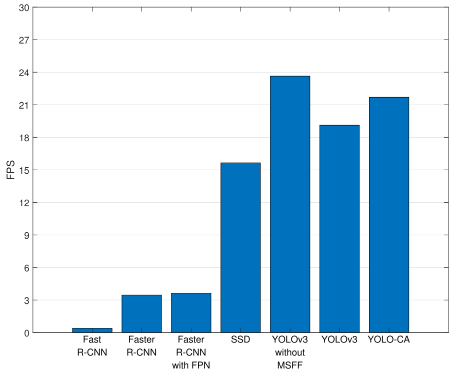


FIGURE 9. The FPS of different models.

Therefore, YOLO-CA can achieve better results of AP and Average IoU than SSD. Compared with YOLOv3, YOLO-CA utilizes loss function with dynamic weights to balance the influence of location loss among different scales of objects. This process increases the error punishment of small objects, because that the same errors of x, y, w, h cause more serious impact on the detection effect of the small object than that of the large object. Consequently, YOLO-CA has obvious advantages in AP and Average IoU of small objects than YOLOv3. The MSFF processes of Faster R-CNN with FPN and YOLO-CA are similar, feature pyramid networks is used to extract feature maps of different scales and fuse these maps to obtain features with high-semantic and high-resolution. Faster R-CNN utilizes RPN to select about 20000 proposal regions, whereas there are only $13 * 13 * 3 + 26 * 26 * 3 = 2535$ candidate bounding boxes in YOLO-CA. This difference results in Faster R-CNN has slight advantages in accuracy performance than YOLO-CA, but also causes serious disadvantages in real-time performance.

Fig. 9 shows the FPS results of different models among test set. It can be found that the FPS of one-stage models is obviously higher than that of two-stage models. This low performance of the two-stage models results from a great deal of time cost of selecting proposal regions.

Fast R-CNN utilizes time-consuming selective research algorithm to select proposal regions based on color and texture features, which results in that Fast R-CNN only achieves 0.4 of FPS. Faster R-CNN uses the RPN that share convolutional layers with state-of-the-art object detection networks instead of selective research to generate proposals. Benefiting from RPN, Faster R-CNN achieve about 3.5 of FPS among test set (Faster R-CNN:3.5, Faster R-CNN with FPN:3.6).

Although Faster R-CNN obtains significantly improvement of real-time performance compared with Fast R-CNN, there is still a big gap with one-stage models. That is because one-stage models abandon the process of selecting proposal regions and utilize one CNN to implement location and classification of objects. As shown in Fig. 9, SSD can achieve 15.6 of FPS among test set. The other three models based on YOLO utilize the backbone Darknet-53 instead of VGG-16 in SSD, and computation of the former network is significantly less than the latter because of using the residual networks. Therefore, the real-time performance of SSD is lower than YOLO-based models in our experiments. In additionally, our proposed YOLO-CA simplifies the MSFF networks of YOLOv3. So YOLO-CA can achieve 21.7 of FPS, which is higher than that of YOLOv3 (about 19.1). Because of lacking MSFF process in YOLOv3 without MSFF, it has better real-time performance (about 23.6 of FPS) than YOLO-CA, but this lacking results in serious performance penalties of AP.

Fig. 10 show some visual results of the seven models among different scales of objects. It can be found that there is a false positive in the large objects detection results of Fast R-CNN, but the other six models all have high accuracy and locating performance in large objects in Fig. 10. However, the locating performance of Fast R-CNN, Faster R-CNN, SSD, and YOLOv3 without MSFF decrease significantly in medium object frame (1), and the prediction bounding box cannot fitting out the contour of car accident. Moreover, Fast R-CNN, SSD, and YOLO-without MSFF cannot detect the car accident in small object frame (1). In additionally, except for Faster R-CNN with FPN and YOLO-CA, other models have serious location error in small object frame (3).

3) COMPARISON OF COMPREHENSIVE PERFORMANCE AND PRACTICALITY

As analyzed above, it can be found that our proposed YOLO-CA has performance advantages of detecting car accident than Fast R-CNN, Faster R-CNN, SSD, and YOLOv3 in terms of accuracy, locating and real-time performance. For YOLOv3 without MSFF, the FPS of it (23.6) is higher than that of YOLO-CA (21.7), and this difference is acceptable in the practical application of detecting car accident. However, the AP of YOLO-CA is significantly higher than that

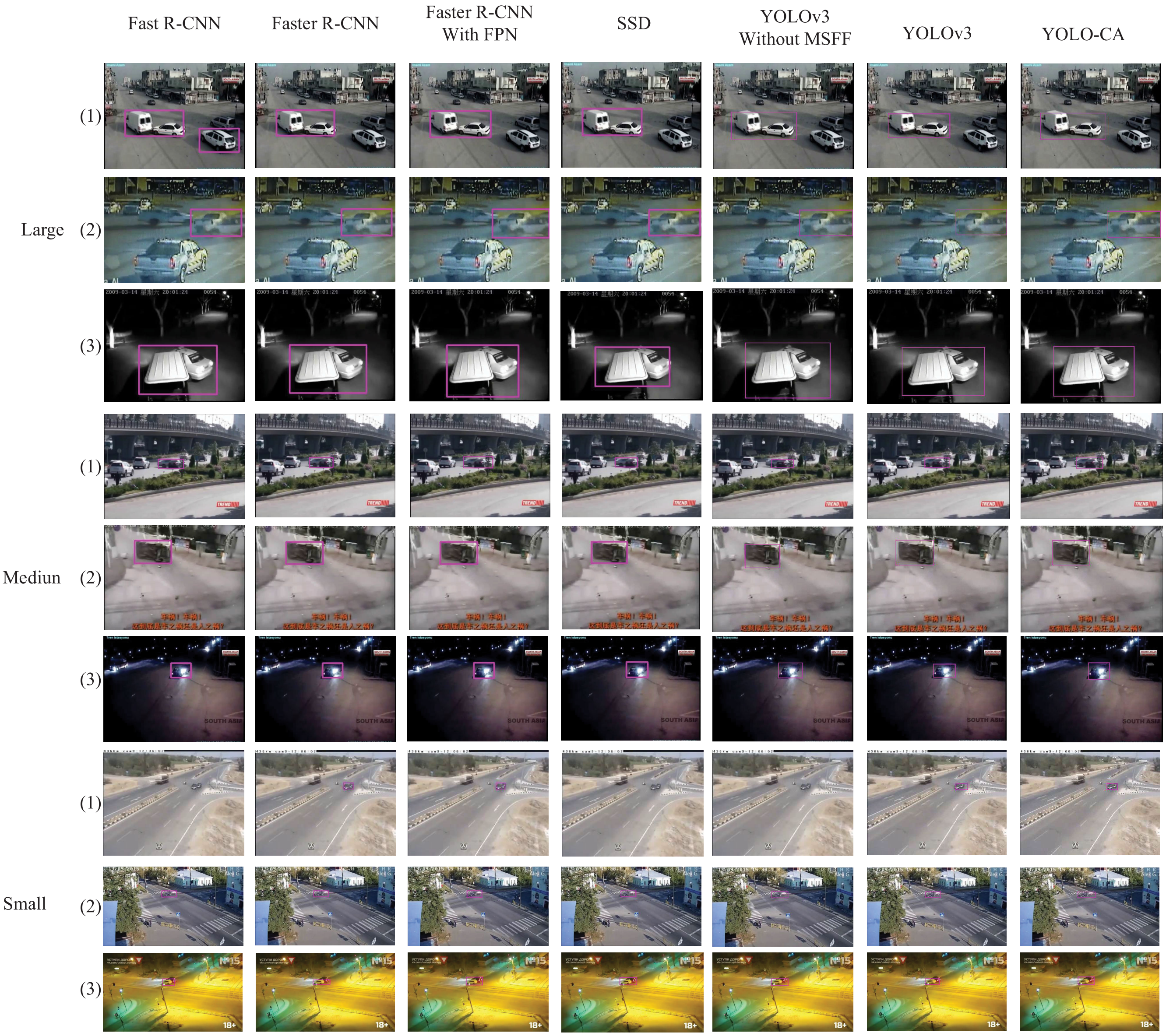


FIGURE 10. Some visual results of the seven models among different scales of objects.

of YOLOv3 without MSFF, especially for small scales of object (76.51% vs 58.89%). Compared with Faster R-CNN with FPN, YOLO-CA can approach the AP of it (90.66% vs 90.03%) with an obvious speed advantage. Faster R-CNN cost about 277ms on average to detect one frame, whereas YOLO-CA only need 46 ms, which illustrates the speed of YOLO-CA is about 6× faster than Faster R-CNN with FPN. Car accident detection in CVIS requires high real-time performance because of the high dynamics of vehicles. To summarize, our proposed YOLO-CA have higher practicality and comprehensive performance on accuracy and real-time.

4) COMPARISON WITH OTHER CAR ACCIDENT DETECTION METHODS

Although other car accident detection methods utilize a small private collection of datasets and do not make them public so comparing them may not be fair at this stage. But still, we list the performance achieved by these methods on their individual datasets. ARRS [3] achieve about 63% AP with 6% false alarms. The method of [27] achieve 89.50% AP. DSA-RNN [25] achieve about 80% recall and 56.14% AP. The method in [30] achieve about 47.25% AP. The method of [8] achieve 77.5% AP and 22.5% false alarms. Moreover, the number of accident scenes of the datasets utilized in these methods is limited, which will result in poor adaptability for new scenarios.

V. CONCLUSION

In this paper, we have proposed an automatic car accident detection method based on CVIS. First of all, we present the application principles of our proposed method in the CVIS. Secondly, we build a novel image dataset CAD-CVIS, which is more suitable for car accident detection method based on intelligent roadside devices in CVIS. Then we develop the car accident detection model YOLO-CA based on CAD-CVIS and deep learning algorithms. In the model, we combine the multi-scale feature fusion and loss function with dynamic weights to improve real-time and accuracy of YOLO-CA. Finally, we show the simulation experiments results of our method, which demonstrates our proposed methods can detect car accident in 0.0461 seconds with 90.02% AP. Moreover, the comparative experiments results show that YOLO-CA has comprehensive performance advantages of detecting car accident than other detection models, in terms of accuracy and real-time.

REFERENCES

[1] WHO. *Global Status Report on Road Safety 2018*. Accessed: Dec. 2018. [Online]. Available: https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/

[2] H. L. Wang and M. A. Jia-Liang, "A design of smart car accident rescue system combined with WeChat platform," *J. Transp. Eng.*, vol. 17, no. 2, pp. 48–52, Apr. 2017.

[3] Y. K. Ki and D. Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 188–194, Jun. 2007.

[4] W. Hao and J. Daniel, "Motor vehicle driver injury severity study under various traffic control at highway-rail grade crossings in the United States," *J. Saf. Res.*, vol. 51, pp. 41–48, Dec. 2014.

[5] J. White, C. Thompson, H. Turner, H. Turner, and D. C. Schmidt, "Wreckwatch: Automatic traffic accident detection and notification with smartphones," *Mobile Netw. Appl.*, vol. 16, no. 3, pp. 285–303, Jun. 2011.

[6] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "Real-time automatic traffic accident recognition using HFG," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3348–3351.

[7] A. Shaik, N. Bowen, J. Bole, G. Kunzi, D. Bruce, A. Abdelgawad, and K. Yelamarthi, "Smart car: An IoT based accident detection system," in *Proc. IEEE Global Conf. Internet Things (GCIoT)*, Dec. 2018, pp. 1–5.

[8] D. Singh and C. K. Mohan, "Deep spatio-temporal representation for detection of road accidents using stacked autoencoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 879–887, Mar. 2019.

[9] M. Zheng, T. Li, R. Zhu, J. Chen, Z. Ma, M. Tang, Z. Cui, and Z. Wang, "Traffic accident's severity prediction: A deep-learning approach-based CNN network," *IEEE Access*, vol. 7, pp. 39897–39910, 2019.

[10] L. Zheng, Z. Peng, J. Yan, and W. Han, "An online learning and unsupervised traffic anomaly detection system," *Adv. Sci. Lett.*, vol. 7, no. 1, pp. 449–455, 2012.

[11] Y. Fangchun, W. Shangguang, L. Jinglin, L. Zhihan, and S. Qibo, "An overview of Internet of vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.

[12] C. Ma, W. Hao, A. Wang, and H. Zhao, "Developing a coordinated signal control system for urban ring road under the vehicle-infrastructure connected environment," *IEEE Access*, vol. 6, pp. 52471–52478, 2018.

[13] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 26–32, Sep. 2018.

[14] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, 2018.

[15] G. Wu, F. Chen, X. Pan, M. Xu, and X. Zhu, "Using the visual intervention influence of pavement markings for rutting mitigation—Part I: Preliminary experiments and field tests," *Int. J. Pavement Eng.*, vol. 20, no. 6, pp. 734–746, 2019.

[16] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1025–1032.

[17] T. Qu, Q. Zhang, and S. Sun, "Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21651–21663, 2017.

[18] D. Dooley, B. McGinley, C. Hughes, L. Kilmartin, E. Jones, and M. Glavin, "A blind-zone detection method using a rear-mounted fisheye camera with combination of vehicle detection methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 264–278, Jan. 2016.

[19] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, "A traffic sign detection algorithm based on deep convolutional neural network," in *Proc. IEEE Int. Conf. Signal Image Process. (ICSIP)*, Aug. 2016, pp. 676–679.

[20] S. Zhang, C. Bauckhage, and A. B. Cremers, "Efficient pedestrian detection via rectangular features based on a statistical shape model," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 763–775, Apr. 2015.

[21] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Aug. 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[22] J. Zaldivar, C. T. Calafate, J. C. Cano, and P. Manzoni, "Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones," in *Proc. IEEE 36th Conf. Local Comput. Netw. (LCN)*, Oct. 2011, pp. 813–819.

[23] M. S. Amin, J. Jalil, and M. B. I. Reaz, "Accident detection and reporting system using GPS, GPRS and GSM technology," in *Proc. Int. Conf. Inform., Electron. Vis.*, 2012, pp. 640–643.

[24] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Automatic accident detection: Assistance through communication technologies and vehicles," *IEEE Veh. Technol. Mag.*, vol. 7, no. 3, pp. 90–100, Sep. 2012.

[25] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Computer Vision-ACCV 2016*. Springer, 2017, pp. 136–153.

[26] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," Mar. 2019, *arXiv:1903.00618*. [Online]. Available: https://arxiv.org/abs/1903.00618

[27] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 108–118, Jun. 2000.

[28] K. Yun, H. Jeong, K. M. Yi, S. W. Kim, and J. Y. Choi, "Motion interaction field for accident detection in traffic surveillance video," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 3062–3067.

[29] V. Ravindran, L. Viswanathan, and S. Rangaswamy, "A novel approach to automatic road-accident detection using machine vision techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 235–242, 2016.

[30] A. P. Shah, J.-B. Lamare, T. Nguyen-Anh, and A. Hauptmann, "CADP: A novel dataset for CCTV traffic camera based accident analysis," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Nov. 2018, pp. 1–9.

[31] Tzutalin. (2015). *Labelimg*. Git Code. [Online]. Available: https://github.com/tzutalin/labelImg

[32] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2720–2727.

[33] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1975–1981. [Online]. Available: http://ieeexplore.ieee.org/document/5539872/

[34] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[35] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 936–944.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[40] 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.

[41] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," Mar. 2015, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467

DAXIN TIAN is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligent.

CHUANG ZHANG is currently pursuing the master's degree with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include multimedia communications and processing and machine learning.

XUTING DUAN is currently a Lecturer with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include connected vehicles, vehicular ad hoc networks, and vehicular localization.

XIXIAN WANG is currently pursuing the master's degree with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include image processing and machine learning.