IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 18, 2021, accepted April 28, 2021, date of publication May 3, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077120

Efficient Traffic Accident Warning Based on Unsupervised Prediction Framework

YUN-FENG ZHOU[1,2,3], KAI XIE[1,2,3], XIN-YU ZHANG[1,2,3], CHANG WEN[3,4], AND JIAN-BIAO HE[5]

[1]School of Electronic Information, Yangtze University, Jingzhou 434023, China
[2]National Demonstration Center for Experimental Electrical and Electronic Education, Yangtze University, Jingzhou 434023, China
[3]Western Institute of Yangtze University, Karamay 834000, China
[4]School of Computer Science, Yangtze University, Jingzhou 434023, China
[5]School of Computer Science and Engineering, Central South University, Changsha 410083, China

Corresponding author: Kai Xie (pami2009@163.com)

This work was supported in part by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2020D01A131, in part by the Fund of Hubei Ministry of Education under Grant B2019039, in part by the Graduate Teaching and Research Fund of Yangtze University under Grant YJY201909, in part by the Teaching and Research Fund of Yangtze University under Grant JY2019011, and in part by the Undergraduate Training Programs for Innovation and Entrepreneurship of Yangtze University under Grant 2019100 and Grant Yz2020057.

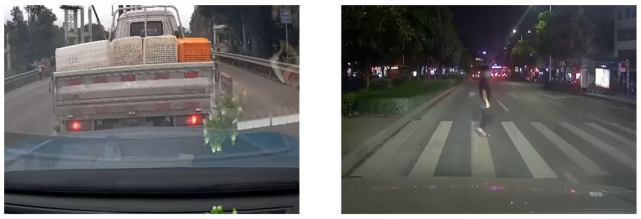
ABSTRACT Recognizing potentially hazardous objects is crucial in the field of transportation, especially in assisted and unmanned driving. However, most existing studies do not focus on defensive driving as they only identify accidents ahead of the vehicle in which they are not involved. In this paper, a driving assistance system is proposed to predict the risk score of potential targets ahead of the vehicle and provide an early warning, which relies on a deep architecture called Fusion-Residual Predictive Network (FRPN) that fused multi-scale residual features and improved adversarial learning. This architecture provides an environment for the generator to perform joint learning from ground-truth images and discriminators with gradient penalty constraints. The deeper convolutional neural network can greatly improve the quality of the image by fusing residual features. Several deep convolutional neural network models were used to evaluate the method on various datasets; among them, the prediction model based on the VGG network, with peak signal-to-noise ratio of 32.67 and structural similarity index of 0.921, respectively, yielded the best results. Subsequently, we utilize the tracking model to design a risk score evaluation method based on the location of the target and it have an improvement in ability to give early warning with 1.95s earlier in the best case. These results prove that our method can effectively reduce the risk of traffic accidents.

INDEX TERMS Generative adversarial network, video prediction, recurrent neural network, convolution neural network, object tracking, traffic warning, unsupervised learning, risk score assessment.

I. INTRODUCTION

Traffic accidents have caused unquantifiable damage to people's lives and property. Globally, countless people have lost their lives annually due to traffic accidents. According to statistics, 90 people die in traffic accidents in the United States every day [1]. Figure 1 shows images of two types of unsuccessful traffic accidents. As the occurrence of an accident is so fast and critical that its prevention time must be short and early. Thus, researchers are supposed to study how to prevent an traffic accidents. Although Tesla and other companies have introduced driverless cars, it is still difficult for the system to make appropriate decisions for drivers in complicated traffic conditions, such as when animals or pedestrians are in front of the car or when unidentified obstacles suddenly appear in front of the car. This is an important research gap. When referring to advanced driver assistance systems (ADAS), laser radar (LiDAR) is typically used to emit laser beams to measure the relative distance between the edge of the object in the field of view (FOV) and the device for accurately capturing contour information to form a point cloud and draw a 3D environment map. However, visual information is also very important in assisted driving. Compared with radar technology, dashcams and other cameras are

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



A B

FIGURE 1. Near-miss traffic incident scenes from Youtube. (A) Close to a vehicle. (B) Close to a pedestrian.

inexpensive. In recent years, almost all new vehicles in China have been equipped with a driving recorder that can record the traffic conditions in front of the car throughout the day. The driving recorder, which has provided a large number of data sources for this study, enables a model to learn more complex situations. However, existing traffic databases, such as the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset [2] and the Dashcam Accident Dataset (DAD) [3], record accidents involving vehicles ahead of the car, which is not the focus of this study. Therefore, this paper proposed a new dataset referred to as the self-accident dataset (SAD), which will be described in detail in a later section.

The key to avoid accidents is to forecast in advance. The work of video prediction can be divided into the three categories. The first structure uses autoencoders to reduce dimensionality and generate videos. Yan *et al.* [4] proposed a deep DynEncoder model that takes the original pixel image as input, which is encoded into a hidden state variable by the encoder. The DynPredictor is then used to dynamically encode the time series. Xue *et al.* [5] proposed a model based on a variational autoencoder and a cross-convolutional network. However, although the model can generate possible future frames from one image, it is not suitable for complex scenes and has low accuracy. Ye *et al.* [6] proposed a pixel-level future prediction approach, which implicitly predicts future states of independent entities while reasoning about their interactions, and composes future video frames using these predicted states. Jasti *et al.* [7] proposed a model based on temporal motion encodings to make it possible to predict any arbitrary number of future frames. The second structure for long-term video prediction is Recurrent Neural Network. Shi *et al.* [8] proposed a convolutional long short-term memory (ConvLSTM) structure based on a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM, proposed by Hochreiter *et al.* [9]) in 2015. Lotter *et al.* [10] proposed PredNet, inspired by the concept of ''predictive coding'' in neuroscience, where each layer of the architecture only performs partial prediction and transmits residuals to the subsequent layers. Wu *et al.* [11] proposed the MotionRNN framework based on transient variation and the motion trend to make it adaptable to complex changes. Wu *et al.* [12] proposed an approach to predict components of the future scene by non-rigid deformation of the background and affine transformation of moving object. Yan *et al.* [13] proposed a method of separating foreground and background images. The model has two encoder inputs, one of which is the motion encoder. The difference in the image sequence is received as the motion input, and LSTM is used to model the motion dynamics. Another encoder receives the last frame of the static image, combines the output of the LSTM with the encoded output of the static image, and decodes it into a predicted image using the decoder. This novel structure makes neural network units originally used for text prediction applicable to images. The third structure is facilitated by the rapid development of the generative adversarial network (GAN) proposed by Goodfellow *et al.* [14], which has led to tremendous progress in image reconstruction. Denton *et al.* [15] proposed a video representation decomposition model that separates the video background content and motion foreground. However, they trained the background content encoder, motion pose encoder, and decoder by generating a confrontation network. Liang *et al.* [16] proposed a dual-learning mechanism. The primal future-frame prediction and dual future-flow prediction form a closed loop, generating informative feedback signals for better video prediction. Recently, Shouno *et al.* [17] proposed a model for natural videos with rapidly changing frames, depth residuals, and a hierarchical structure. Each layer predicts the future state with a different spatial resolution and these predictions from different layers are combined through top-down connections to generate future frames. Lin *et al.* [18] proposed a motion-aware feature enhancement network for video prediction to produce realistic future frames and achieved relatively long-term predictions. These models have been shown to have certain advantages in traffic scene prediction. However, larger structures tend to consume more resources during prediction, thereby degrading the model performance. Luc *et al.* [19] proposed a novel method to improve the performance of the original model by updating the hidden state of the generator loop unit.

A representative work on the topic of early event prediction was conducted by Ryoo *et al.* [20], who proposed a method for predicting human behavior. They represented motion as integral histograms of spatio-temporal features, effectively modeling how the distribution of features changes over time. Cai *et al.* [21] established a model for pedestrian motion trajectory prediction from far shot first-person perspective video. In the field of transportation, traffic accident prediction approaches are roughly divided into two categories, based on statistical models and machine learning. In the first category, Ren *et al.* [22] used different types of traffic-related data such as traffic accidents, traffic flow, weather conditions, and air pollution in the same city to build a complex system, and then used recurrent neural networks to predict. However, these results only indicate that accidents may occur under such conditions, but cannot predict the real traffic flow. The other part of the problem is to predict whether other vehicles will have an accident, centered on the came camera. Chan *et al.* [3] proposed the use of dynamic spatial attention and RNN to

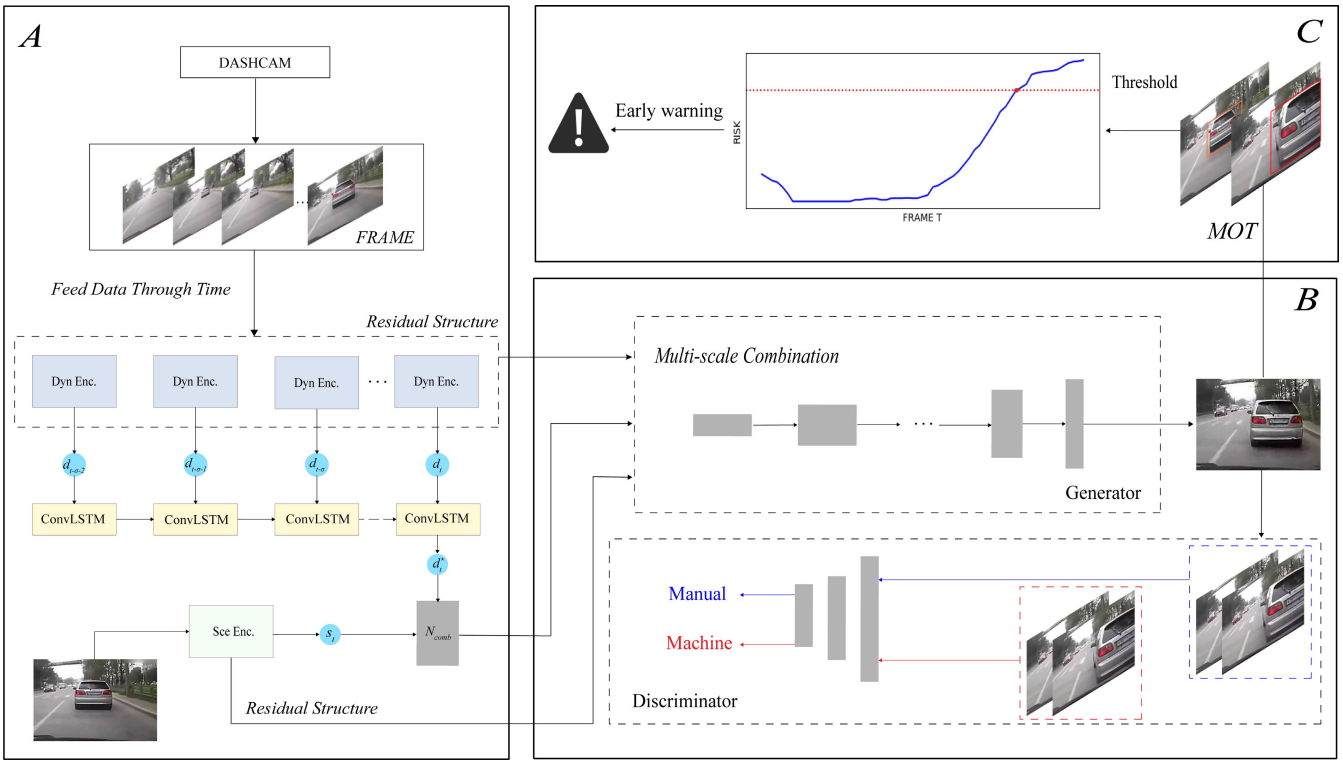


FIGURE 2. Flow of proposed algorithm (for formal unity in visualization, the third section of Part A is displayed in Part B). (A) is the separate encoding and prediction of moving objects and background through a multi-scale residual feature fusion network, (B) is the image reconstruction and optimization through an adversarial neural network with gradient penalty terms and real image constraints, and (C) is the final risk assessment through tracking the target and calculating the risk score.

predict accidents in a dashcam video. Yu Yao *et al.* [23] devised a method for providing early warning by detecting objects and predicting their trajectories. Suzuki *et al.* [24] predicted traffic accidents using adaptive loss and large-scale event databases. Recently, Shouno *et al.* [17] proposed a semi-supervised method using GAN trained on regular sequences to predict future frames and compared these prediction frames with real frames to determine whether an abnormal event occurred. Although these works are similar to ours, they mainly predict the collision of other vehicles in front of the car with dashcam and cannot be used to assist driving or provide early warning.

In this study, a novel early traffic accident prediction method was developed, in addition to collecting new traffic datasets. First, the Fusion-Residual Predictive Network (FRPN) is used to predict the future location of objects in the video from the previous images. Even when there is significant movement between the frames. This method can also accurately predict realistic future frames of natural videos. Then, the tracking model tracks these targets to assess the risk score of them to our own vehicle. Finally, the proposed algorithm with a preset threshold is evaluated to achieve an early warning. The proposed model can be easily applied to existing datasets. Furthermore, unsupervised learning saves a large amount of manual labeling.

The remainder of this paper is organized as follows. Section II introduces the theoretical basis of algorithm of proposed model. The training details, experimental results, and prediction results are presented in Section III. Finally, Section IV concludes the paper.

II. METHOD

The proposed method of early warning of traffic accidents relying on a dashcam in a vehicle to facilitate defensive driving. It is composed of three steps: separated image encoding and prediction using the embedded sequence prediction module, generating clearer and objective images using an improved GAN model, and locating and tracking suspicious objects in view using an additional target detection model with a confidence tracking algorithm. Finally, the trajectory of object is predicted and analyzed. A schematic of this method is presented in Fig. 2.

Before the algorithm, the size of the extracted image is normalized. Several image sequences (including the current image and previous frames) are then input into the corresponding encoder networks to extract the differential and global features, respectively. The difference between two adjacent frames is input into the ConvLSTM network to obtain the predictive state. However, for scene encoding, only the current image is utilized. Subsequently, the resulting predicted differential features and scene distribution coding are reconstructed through a combined network with a multi-scale residual structure to obtain future frames. However, the resulting image become increasingly blurred as the sequence become longer. Therefore, the discriminator is deployed to perform adversarial learning to clarify the

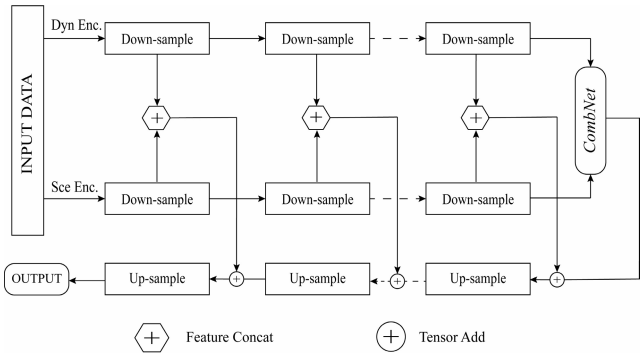


FIGURE 3. Multi-scale residual combination structure based on local dynamic.

blurred image with incomplete information, which makes more similar to the future frame image through constrained learning. Through the above steps, a stable model is trained to generate a future image. Finally, the model automatically generated predictive pictures periodically to provide advance warnings.

A. IMAGE ENCODING AND PREDICTION

Vehicles can move extremely quickly; therefore, determining difference between frames is crucial. This model was inspired by the MCNET framework proposed by Villegas *et al.* [25]. However, the network structure, which is not effective in very complex scenarios, is improved to fit new problem. It will be described in detail in the Experiment section. Each δ-frame was input into the encoder as a sequence for image information extraction, and separately input into the prediction module for spatio-temporal prediction to obtain the predictive feature representation. Finally, it was fused with the scene distribution coding of the current frame and used as the input of the decoder for image reconstruction. Let x_t denote the t-th frame in the input video. The goal of frame prediction is to generate future frames $\hat{x}_{t+1}$, and even more future images from the input $x_{t-\delta+1} : x_t$. The overall structure is illustrated in Fig. 3.

1) SEPARATE IMAGE ENCODER

To achieve a more meaningful image reconstruction, the region of interest and background content are calculated separately, and the final image is synthesized through a combination network.

For the proposed algorithm, the differential encoder captures the dynamic scene composition by observing the inter-frame difference sequence data cyclically input to the network and through local dynamics outputs the motion feature of targets using the following formula:

$$[d_t, c_t] = f_{dyn}(x_t - x_{t-1}, d_{t-1}, c_{t-1}) \tag{1}$$

where the memory unit c_t is used to retain the dynamic information in time and d_t is the motion features. Further, f_{dyn} is a neural network that captures the local dynamics of the frame image by element-wise subtraction between the two adjacent inputs, the characteristics and dynamic information of the last time node, thus minimizing redundant calculations. The content of the forecasting work will be discussed in the next section.

The differential image encoder observations of spatially specific objects incorporate detailed cues of moving objects that may be involved in accidents. However, the full-frame content can capture important clues related to the scene or movement of the camera. The scene encoder is mainly aimed to obtain information such as the layout of the scene and the position of the salient object from the sequence input. Using the following equation, the characteristics of a single frame can be obtained:

$$s_t = f_{sce}(x_t) \tag{2}$$

where s_t is the scene features of image background and f_{sce} is CNN focused on learning the content feature extraction of the current single frame, where the pooling operation is consistent with the differential encoder.

2) PREDICTION WITH CONVOLUTIONAL LSTM MODEL

The encoder employs CNN to extract the feature difference between the current and previous frame images. It is also necessary for the ConvLSTM unit which input the hidden state vector output from the previous frame, and the last LSTM unit outputs the prediction result $[d_t^*, c_t]$, where d_t^* is the dynamic feature captured by the encoder and ConvLSTM. The encoded feature information, which can be regarded as a spatio-temporal sequence, is used as the input variable of the time series module to model the time dimension. The corresponding variable is input into each time node, which output the corresponding hidden information and cell state, finally obtaining the structural information of the predicted image.

The structure diagram of the LSTM, a special RNN structure, is shown in Fig. 4(A). The structure stores the state to be memorized by introducing a cell state and adds three gates to the original structure: the forget gate (f_t, decides to when to forget the previous state), input gate (i_t, decides when to add the new state), and output gate (o_t, decides when to combine the cell state and input for output). Unlike the traditional LSTM, the ConvLSTM can be modeled either in time or in space, and the internal structure is shown in Figs. 4(B) and 4(C). This function can be expressed using the following formulae:

$$i_t = \sigma(w_{xi} * z_t + w_{hi} * h_{t-1} + w_{ci} \circ c_{t-1} + b_i) \tag{3}$$
$$f_t = \sigma(w_{xf} * z_t + w_{hf} * h_{t-1} + w_{cf} \circ c_{t-1} + b_f) \tag{4}$$
$$\bar{c}_t = \tanh(w_{xc} * z_t + w_{hc} * h_{t-1} + b_c) \tag{5}$$
$$o_t = \sigma(w_{xo} * z_t + w_{ho} * h_{t-1} + w_{co} \circ c_t + b_o) \tag{6}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \bar{c}_t \tag{7}$$
$$h_t = \tanh(c_t) \circ o_t \tag{8}$$

where "$*$" denotes the convolution operator and "$\circ$" denotes the Hadamard product.

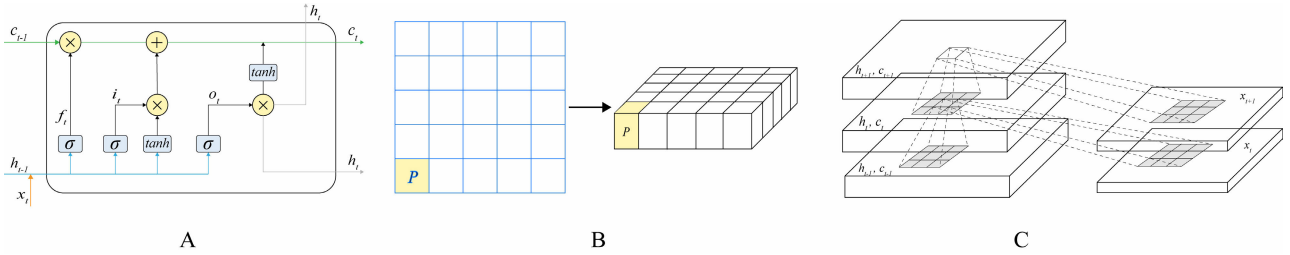


FIGURE 4. (A) LSTM internal unit structure diagram. (B) Transforming 2D image into 3D tensor. (C)Inner structure of ConvLSTM.

3) IMAGE DECODER WITH MUTLI-SCALE COMBINATION

To strengthen the network ability to adapt to images of different scales and provide more information to the decoder, the features obtained before pooling are preserved between operations at two different scales, and the retained residual results are merged with the upsampling results prior to decoding. This residual connection [26] connects the motion and scene features of each scale. The residual features of layer l are given as

$$r_t^l = f_{res}([d_t^l, s_t^l])^l \tag{9}$$

where r_t^l is the residual feature output of layer l obtained by scale layering, and f_{res} is the residual function consisting of a series of convolution layers and a final linear rectification layer.

These high-dimensional feature representations combine the feature information of motion and scene to obtain the feature expression of the final predicted image, using the following formula:

$$o_t = f_{comb}(d_t, s_t) \tag{10}$$

where d_t and s_t are the high-dimensional features of motion and scene respectively, and f_{comb} is a CNN with bottleneck layer [27]. The original high-dimensional space contained redundant and noisy information; therefore, the resultant error generated through dimensionality reduction is reduced to improve the decoding accuracy. The model projects both d_t and s_t onto a lower-dimensional embedding space and returned them to their original size to build the combined feature o_t. Then the o_t and r_t are merged through a decoder to obtain the pixel-level predicted image of the next frame using the following formula:

$$\hat{x}_{t+1} = f_{dec}(o_t, r_t) \tag{11}$$

where f_{dec} uses transposed convolution for upsampling. The deconvolution network [28] has the same number of layers as the foreground encoder. Each scale is corrected and unpooled by adding a residual connection of the motion content after each unpooling operation. The final output layer is passed through a tanh(.) activation function. The formula is given as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{12}$$

B. RECONSTRUCTION AND OPTIMIZATION BY WASSERSTEIN GAN WITH GRADIENT PENALTY

In this section, we combine the Wasserstein GAN with a gradient penalty (WGAN-GP) proposed by Gulrajani *et al.* [29] for adversarial learning. The network consists of two parts: a generator, G, and a discriminator, D. These two parts have the relationship of adversarial learning and parameter sharing. G attempts to generate a predicted image, x, from the coding sequence input, and D improves its discriminatory ability to judge the authenticity of the predicted image. Through these mutual relations, both parts conduct adversarial learning, improve each other, and generate a clearer and more ideal predicted image. In addition, the training detail of whole video prediction is shown in Algorithm 1.

1) WASSERSTEIN DISTANCE

The predicted image generated by the decoder has certain limitations, such as image blur and inconsistent generation. Therefore, the Wasserstein algorithm is added to further optimize the generated image. Instead of the Jensen-Shannon (JS) divergence [30], the earth mover's distance (EMD) [31] is used to measure the distance between the distribution of the real and generated samples, which can solve the GAN training instability. Even when the overlap between the real and generated distributions is nonexistent or extremely small, there is no gradient disappearance due to that the JS divergence is constant.

EMD is also known as the Wasserstein distance. This WGAN model uses the simplified Wasserstein distance equation of the Kantorovich–Rubinstein duality [32] to obtain the minimum cost of transmission quality when the actual distribution P_r is converted to the generated distribution, P_g, by the Wasserstein distance. The formula is as follows:

$$W(P_r, P_g) = \sup_{||f||_L \leq 1} E_{x\sim \mathrm{Pr}}[f(x)] - E_{x\sim \mathrm{Pg}}[f(x)] \tag{13}$$

where sup is the minimum upper limit, and a deep network is used to learn an optimal function 1- Lipschitz, to optimize the Wasserstein distance.

2) DISCRIMINATOR

To generate higher-quality images, in addition to taking the original picture as the learning target, the discriminator is also used to make the model more robust. Although the

Algorithm 1 Training Detail of Prediction Model

Require: Input samples $I > 2$, Preditive results $P > 0$.
1: Initialize the parameters of the *ConvLSTM*, Generator G, and Discriminator D
2: Samples $\{x_1, \ldots, x_n\}$ from the training set
3: **for** $i = 1$ to I **do**
4: Obtain the coding of motion (d_t) and multi-scale residual results (d_t^l)
5: Predict the state of the motion (d_t^*) via *ConvLSTM*
6: **for** $p = 1$ to P **do**
7: Compose predictive frames and previous frames into a new sequence as training samples
8: Obtain the coding of scene (s_t) and multi-scale residual results (s_t^l)
9: Combine predictive motion features (d_t^*), scene features (s_t) and residual results (d_t^l), (s_t^l)
10: Generate the predictive frame by deconvolution
11: **for** *number of training iterations* **do**
12: **for** *number of steps to app1y to the D* **do**
13: Update D by ascending:
14: $L_D(x, x^*) = -E_{x,x^*\sim \mathrm{P}r}(\mathrm{D}(x,x^*)) + E_{x\sim \mathrm{P}g}$ $[\mathrm{D}(x, \mathrm{G}(x))] + \lambda_d E_{\hat{x}\sim Penalty}(\mathrm{D}(x, x^*))\,[(||\nabla_{\hat{x}}\mathrm{D}(\hat{x})||_2 - 1)^2]$
15: Update G by descending:
16: $L_G(x, y) = -\lambda_d E_{x\sim \mathrm{P}r}[\mathrm{D}(x, \mathrm{G}(x))] + \lambda_{img} L_{img}$

discriminator is similar to the GAN model, noise is not as an input. The task of training the discriminator ceases to be a binary classification task. However, the gradient of the discriminator is used to optimize the network parameters to approximate the Wasserstein distance, which is a regression task. Therefore, the sigmoid function is canceled in the last layer of the network. As long as the difference exists, the model will continue to learn until the difference as small as possible, while maximizing the discriminator objective through iterative optimization.

The WGAN-GP resolves the gradient explosion and disappearance that occurred during training to a great extent. After each update of the parameters of the discriminator, it is extremely unscientific to truncate their absolute values within a fixed constant range. However, when most of the values crossed this limit, the values are clustered at $-c$ and c. Therefore, an additional gradient penalty term is set in the original discriminator loss function such that the gradient of the discriminator does not exceed K, which avoids truncating the parameter of Lipschitz constraint leading to their values go to the extreme and limiting the discriminatory ability of D, and deteriorates the quality of G. Based on the above description, the final loss function is as follows:

$$L_D(x, x^*) = -E_{x,x^*\sim \mathrm{P}_r}(\mathrm{D}(x, x^*)) + E_{x\sim \mathrm{P}_g}[\mathrm{D}(x, \mathrm{G}(x))] + \lambda_d E_{\hat{x}\sim \mathrm{P}_{penalty}}[(||\nabla_{\hat{x}}\mathrm{D}(\hat{x})||_2 - 1)^2] \quad (14)$$

where x^* is the future real sample of input frame x, λ_d is the penalty coefficient to control the gradient penalty, and $P_{penalty}$ represents the area between the real and generated distributions. Therefore, real and generated samples, x_r and x_g are chosen, and a gradient penalty sample x_p is selected on the line between them.

3) GENERATOR

The result predicted by the LSTM may not be ideal, adding the comparison with the real image in the loss function can constrain the final generated image to be close to reality. When training the generator, the weights of the discriminator trained in the previous stage are fixed, and through a series of deconvolution operations, complete future video frames separated by a period of time were generated. Further, it is necessary for generator to learn how to generate accurate images and to keep learning to deceive discriminator, generate a matrix similar to the real image data, attempt to shorten the Wasserstein distance between the samples, and minimize the generator objective function through iterative optimization.

However, the predicted and real sample images are often dissimilar and blurry. Inspired by Mathieu *et al*. [33], a loss function is developed that combined adversarial loss and L_{img} into the generator. By adding this additional loss function to compel the distribution of the predicted image to be consistent with the actual image distribution, the formula for L_{img}, a loss in the image space, is as follows:

$$L_{img} = L_p(x, x^*) + L_{gdl}(x, x^*) \quad (15)$$

$$L_p(x, x^*) = (\sum_{l=1}^{n} |x^l - (x^*)^l|^p)^{\frac{1}{p}} \quad (16)$$

$$L_{gdl}(x, x^*) = \sum_{i,j} |(|x_{i,j} - x_{i-1,j}| - |x^*_{i,j} - x^*_{i-1,j}|)|^{\alpha} + |(|x_{i,j-1} - x_{i,j}| - |x^*_{i,j-1} - x^*_{i,j}|)|^{\alpha} \quad (17)$$

where x^* is the future real sample of the generated frame, x.

L_{img} makes the network generate the correct sequence through input, L_p is controlled by hyperparameter p to directly guide the network to approximate real images, and L_{gdl} is set based on hyperparameter α, to sharpen the predicted image following the gradient difference loss.

Combining the above losses, the final following objective function was obtained:

$$L_G(x, y) = -\lambda_g E_{x\sim \mathrm{P}_r}[\mathrm{D}(x, \mathrm{G}(x))] + \lambda_{img} L_{img} \quad (18)$$

As the learning object of the generator, the optimal loss function can be adjusted using λ_g and λ_{img}.

C. FINAL TRACKING PREDICTION

In this section, a traffic hazard assessment system based on a driving recorder is introduced. A score is assigned to every suspicious object that appeared in the FOV, and a predetermined threshold is used for advanced judgment.

1) MULTI-OBJECT TRACKER WITH IDentity POOL

It is necessary to detect each object and give them a unique identity by performing data association. The joint detection

and embedding (JDE) model proposed by Wang *et al.* [34] has achieved good results in multi-target tracking. Similar to most structures, it sequentially performs detection and association. It adopts a feature pyramid network [35] for multi-scale prediction, combines detection and embedding learning, extracts an embedding vector from the feature map, adopts the automatic balance loss to obtain the best loss weight, and adds a triplet loss similar to cross entropy as the goal of embedding learning. This is represented by the following equation:

$$L_{CE} = -\log \frac{\exp(f^T g^+)}{\exp(f^T g^+) + \sum_i \exp(f^T g_i^-)} \tag{19}$$

where f^T is a sample of the mini-batch selected as the anchor, and g^+ and g_i^- are the class-wise weights of the positive and negative classes, respectively.

The JDE model processes the predicted frame and outputs the bounding box and the corresponding appearance embedding. The target in the FOV will be added to the identity pool, and the Hungarian algorithm is used for association matching, which is limited by the Kalman filter to obtain a higher match rate. At each time step, update the observed tracker and initialize a new identity when a new object is detected. If the target disappeared from the FOV, it was marked as ''lost''(denoted as '0', otherwise '1'). If the target was lost for more than a given temporal threshold, it was deleted from the current identity pool or re-acquired in the allocation step.

2) DEFINE RISK SCORE FOR OBJECT

Traffic conditions are always complex and changeable; therefore, judgments based on the simple target detection tend to be inaccurate and dangerous. We performed batch analysis on the key frames of traffic accidents using the D^2-City dataset [36] and Dashcam Accident Dataset (DAD) [3], and established a risk assessment system from three aspects based on the FOV of the front dashcam: the size variation in the frame, movement of the center of gravity, and angle of the center of the high-risk area. Our subsequent model evaluation is based on predicting the trajectory of the object to assess the degree of danger posed by its current position.

The size of the video produced by different dashcam configurations various, so this algorithm uses the proportion calculation method to avoid errors caused by the uniqueness of the parameters. The upper-left corner of the frame is defined as the origin of the coordinates, and W_f as well as H_f stand for the width and height of frames respectively. Further, Δs is the area size change of the target between two adjacent frames, p is the position of the center of gravity of the target in the FOV, and θ is the angle between the target line and the lower horizontal coordinate axis, where the target line is the line connecting the crosshair (H_f, W_f / 2) and the target point. The calculation formulae for these parameters are as follows:

$$\Delta s = S_n - S_{n-1}$$

$$p = 1 - \sqrt{\frac{(W_f/2 - x_n)^2 + (H_f - y_n - H_{box}/2)^2}{(W_f/2)^2 + H_f^2}}$$

$$\theta = arctan\left(\frac{H_f - y_n}{W_f/2 - x_n}\right) \tag{20}$$

where S_n is the current area of the target, and its area in the last frame is S_{n-1}. (x_n, y_n) is the center of mass of current position of the target. H_{box} is the height of the target box.

In the dashcam, a distant vehicle is always in an upper position of images. As the vehicle approaches, its area gradually increases and its position gradually moves to the lower part of the image. Subsequently, it disappears in the lower left or right corner of the FOV. Vehicles with abnormally behaviour, which is potential to cause accidents, tend to move to the lower center of the image. At this time, the risk score gradually increases. When the vehicle is stationary or moves at a steady speed, the relative area and position of the target in the image remain unchanged. Therefore, three hyperparameters are defined to calculate the risk score by weighting based on the following formula:

$$r = \alpha g(\Delta s) + \beta h(p) + \gamma j(\theta) \tag{21}$$

here, $g(\Delta s)$ is the function of area change, $h(p)$ is the function of location change (degree of danger), $j(\theta)$ is the function of angle change, and r is the risk index between [0, 1]. Further, α, β, and γ are the proportion parameters that control the three risk scores. Seven road conditions are comprehensively analyzed: normal vehicle from the opposite direction (NVOD), abnormal vehicle from the opposite direction (AVOD), vehicle crossing from the left or right (VCFLoR), vehicle stopped in front (VSF), normal vehicle from the same direction (NVSD), similar-speed following vehicle (SSFV), and normal overtaking vehicle (NOverV), as shown in Table 1.

Prior to experiment, these parameters are predetermined and used to evaluate the risk generated by the trajectory change using the following state function:

$$g(\Delta s) = \begin{cases} 1, & \Delta s > \Delta S_T \\ 0, & \Delta s \le \Delta S_T \end{cases}$$

$$h(p) = \begin{cases} 1, & p = p_{worst} \\ 0, & p = p_{best} \end{cases}$$

$$j(\theta) = \begin{cases} 1, & \theta = \theta_{\max} \\ 0, & \theta = \theta_{\min} \end{cases} \tag{22}$$

where ΔS_T is the area size change threshold and p_{best} is the safest position in the FOV, that is, the upper left and right corners of the image. At this point, the dangerous state function value is zero, meaning that p_{worst}, the dangerous state function value, is one. It should be noted that this function is not a discrete integer function, but its value is of type float. Furthermore, the formulae indicate only extreme cases.

III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the algorithm system is introduced in detail through the experimental parameters and procedures. The entire experiment is divided into three steps. First, we provide

TABLE 1. The changing trend of the state function in different situations.

Index	*NVOD*	*AVOD*	*VCFLoR*	*VSF*	*NVSD*	*SSFV*	*NOverV*
g(s)	↗	↗	↗	↗	↘	–	↘
h(p)	↘	↗	↗	↗	↘	–	↘
j(θ)	↘	↗	↗	–	↗	–	↗

the model quality used by the evaluation metrics for quantitative verification and basic experimental settings. We then determine the key experimental parameters during the training process. Finally, these evaluation metrics were used to evaluate the results of video-based accident risk prediction. The framework described in this section is illustrated in Fig. 5.

A. EXPERIMENTAL CRITERION

In the following experiment, the results of the experiment were evaluated based on three aspects of the video prediction evaluation criteria: similarity, clarity, and accuracy. The similarity was comprehensively evaluated from two aspects: peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). The real frame is defined as Y, and the predicted frame is defined as $\hat{Y}$. These standards are calculated using the following formulae:

$$PSNR(Y,\hat{Y}) = 10\log_{10}\frac{\max_{\hat{Y}}^2}{\frac{1}{N}\sum_{i=0}^{N}(Y_i-\hat{Y}_i)^2} \tag{23}$$

where $\max_{\hat{Y}}^2$ is the maximum possible value of pixels in the predicted frame. The larger the PSNR value is, the less the distortion effect of the image is.

$$SSIM(Y,\hat{Y}) = \frac{(2\mu_Y\mu_{\hat{Y}}+c_1)(2\sigma_{Y\hat{Y}}+c_2)}{(\mu_Y^2+\mu_{\hat{Y}}^2+c_1)(\sigma_Y^2+\sigma_{\hat{Y}}^2+c_2)} \tag{24}$$

where $(\mu_Y, \mu_{\hat{Y}})$ and $(\sigma_Y, \sigma_{\hat{Y}})$ are the mean and variance of Y and $\hat{Y}$ respectively; $\sigma_{Y\hat{Y}}$ is the covariance of $Y\hat{Y}$, $c_1 = (0.01L)^2$; $c_2 = (0.03L)^2$; and L is the dynamic range of the pixel values in the picture frame. The larger the SSIM value is, the greater the similarity of the images is.

The accuracy was obtained using the confusion matrix, which was used to quantitatively evaluate the final accident risk prediction algorithm. These formulae are as follows:

$$Accuracy(\%) = \frac{TP+TN}{TP+TN+FP+FN} \tag{25}$$

$$Recall(\%) = \frac{TP}{TP+FN} \tag{26}$$

$$Precision(\%) = \frac{TP}{TP+FP} \tag{27}$$

Here, TP, TN, FP, and FN are the probabilities of true positive, true negative, false positive and false negative in the confusion matrix, respectively. The detailed divisions are listed in Table2.

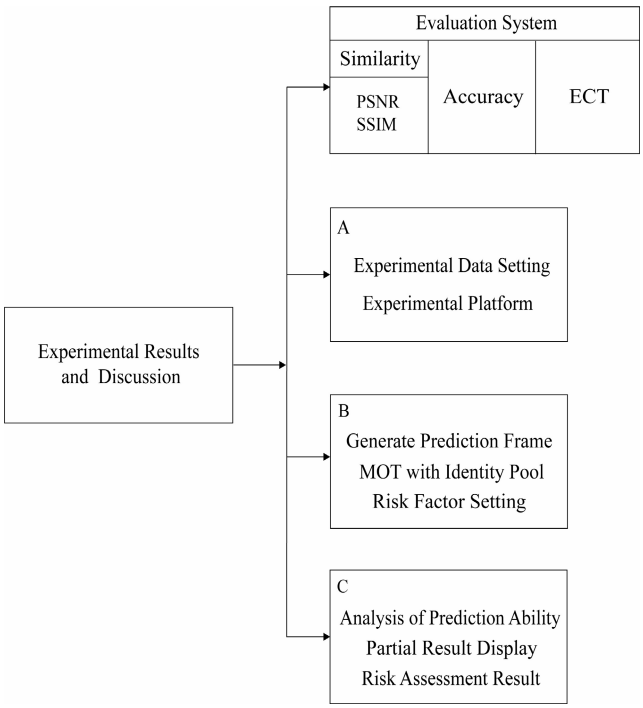


FIGURE 5. Overview of experimental results and discussion. Experimental criterion is divided into three parts and discussed separately (A) Experimental setting (B) Training details (C) Experimental result and analysis.

Several indicators are employed in the standard CLEAR-MOT metric [37] to evaluate the tracking accuracy of the entire MOT system.

The ECT can be regarded as the overall response capacity of the model. If the time of the accident was noted as T_i, and our model provides a collision warning at time T_j, the time interval is the early warning capability for accidents. The ECT is calculated as follows:

$$ECT(s) = T_i - T_j \tag{28}$$

B. EXPERIMENTAL SETTING

1) DATABASE INTRODUCTION

KITTI. The more complex video prediction dataset KITTI [2] was used to objectively evaluate our model. The dataset

Algorithm 2 Traffic Accident Warning System

Require: Video clip v_1 from the test set.
1: Initialize the parameters of the traffic accident warning system
2: $\{x_1, \ldots, x_n\} \leftarrow v_1$
3: Generate predictive frames sequence $\{X_{n+1}, \ldots, X_t\}$ through prediction model
4: **while** *not end of* X_t **do**
5: Track these targets $\{T_1, \ldots, T_n\}$ in this frame via *JDE*
6: **while** *not end of* T_n **do**
7: **if**(*lable* = '0' **and** *lost time* < *continuous threshold*) **or** *lable* = '1' **then**
8: Keep its ID in the identity pool
9: Update its risk score by:
10: $r = \alpha g(\Delta s) + \beta h(p) + \gamma j(\theta)$
11: **if** $r >=$ *risk threshold* **then**
12: Alert
13: **else**
14: Delete this target from current identity pool

TABLE 2. Division of four different results in confusion matrix.

Confusion Matrix		
	Label	Prediction
TP	Accident occured	Accident occured
TN	Non-accident occured	Non-accident occured
FP	Non-accident occured	Accident occured
FN	Accident occured	Non-accident occured

was created by the Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago in 2012 and updated in 2015. It was captured by a car-mounted camera on a car driving in an urban environment in Germany. The dataset includes three categories: city, residential and road. Each category contains 57 videos. These frames were randomly divided into training, validation and testing sections. The training, validation and test sets consist of 40,678, 458 and 1,246 frames, respectively.

D^2-City Dataset. The D^2-City dataset [36] dataset was collected from Didi operating vehicles operating in five cities in China. It covers a total of 12 types of driving and road-related target categories, different weather, roads, and traffic conditions, especially extremely complex and diverse traffic. It is composed of a total of 1000 videos recorded by the driving recorder, the original data provided were stored as short 30-s videos with a frame rate of 25fps, and every frame in the dataset was annotated. There were 700 videos for training, 100 for validation, and 200 for testing.

SAD. To evaluate the proposed method in realistic traffic scenarios, a new dataset SAD was compiled. The dashcam is an inexpensive aftermarket camera that can be installed in a vehicle to record occurrences in front of the vehicle from the driver's perspective. We selected some datasets from the DAD and obtained videos from the YouTube channel and other online channels. Our dataset is composed of cyclists, pedestrians, and vehicles, and reflects various weather conditions (e.g., sunny, rain, and snow) and locations (e.g., city, and country), in circumstances which forced our vehicles to stop. It consisted of 458 positive-sample accident videos and 1137 negative-sample normal videos, of which 1145 videos were used for training (343 positive and 802 negative videos) and 450 videos for evaluation(115 positive and 335 negative videos).

2) EXPERIMENTAL PLATFORM

In this study, all experiments were carried out on a platform with a Windows 10 operating system, an NVIDIA GeForce RTX 2080Ti with 11 GB graphics memory, and Intel Core i9-9900K with 16 GB memory. The software platform was Python 3.5.9 and the Tensorflow 1.12.0 framework.

C. TRAINING DETAILS

1) IMAGE ENCODER AND PREDICTION

Before the implementation of the algorithm, the frames extracted from the video were resized for preprocessing. Owing to hardware limitations, the minimum batch size was set to 4 or 8, to fully exploit the GPU hardware acceleration. Considering the complexity of the scene content, the features from the image content were extracted by using various representative models and different unit numbers as the global feature extractor.

The inception structure used was proposed by Google [38]. The number of filters in each inception block can be expressed by the following equation: $2^{\text{block}-1} \times 64$. After all the convolutional layers, we performed batch normalization, followed by the implementation of a parameterized rectified linear unit [39]. This parameter is a linear function defined in pieces that can produce good results and stable gradients. After several inception blocks, a fixed 1024-dimension feature vector and a container that retained the scene convolution results of several scales from each frame were obtained. In addition, VGG16 [40] and ResNet50 [26] were trained separately to obtain the best model. The differential encoder was similar to the ResNet network; however, only shortcut connection thoughts were selected. The filter size of all the convolution operations was 3×3 and obtained an output of the same size as before and a container that retained the foreground convolution results on three scales. The combined layer consisted of three consecutive 3×3 convolutions with 1024, 512, and 1024 channels. This layer was responsible for combining the encoded content of the differential and scene content. The multi-scale residuals were composed of two consecutive 3×3 convolutions, connecting the convolution results of two containers of different scales. The ConvLSTM combined convolutional layers, instead of fully connected layers or LSTM units.

2) IMAGE RECONSTRUCTION AND OPTIMIZATION

The decoder used for image reconstruction doubly upsampled the image fused by the encoder on the original multi-scale structure and deconvoluted the residual results through

TABLE 3. Evaluation of the quality of multi-step predictions on 20% test set on KITTI Dataset with the different models.

Model	1^{st} frame		5^{th} frame	
	PSNR	SSIM	PSNR	SSIM
PredNet	27.34	0.823	25.18	0.728
VGG-GAN	31.41	0.912	28.74	0.852
FRPN+InceptionNet	30.24	0.901	27.82	0.843
FRPN +VGGNet	32.67	0.921	30.57	0.894
FRPN +ResNet	28.13	0.892	26.15	0.814

FIGURE 6. Qualitative results of multi-step video prediction on KITTI database(Inc means InceptionNet, VGG means VGGNet and Res means ResNet).

element addition, performed thrice, twice, and once. The convolution and filter sizes were all 3×3, and the result was obtained using the hyperbolic tangent tanh(.).

For all experiments, the Adam optimizer was used in the loss function of the generator with a fixed learning rate of 1e-4, $\beta\_1 = 0.5$, $\alpha = 1$, $\lambda_{img} = 1$, $\lambda_g = 0.02$, and p = 2 as the limiting score. In the loss function of the discriminant, the Adam optimizer with a learning rate of 1e-4 and $\beta\_1 = 0.$, $\beta\_2 = 0.999$, epsilon = 1e-8, the last two parameters from the exploration of Kingma *et al.* [37]. In addition, the penalty coefficient λ_d was set to 10, and 100,000 iterations of training were performed using the aforementioned parameters.

3) MOT WITH IDENTITY POOL

The model was trained on the D^2-City dataset to detect suspicious targets in predicted frames {cyclists, pedestrians, vehicles} using the CSPDarknet53 [42] as the backbone network; the set class was 3. The model was trained for 100 epochs using a standard stochastic gradient descent optimizer, and the initial learning rate was 1e-2, which decreased by 0.1 at the 50th and 75th epochs.

TABLE 4. Experimental results of JDE model on SAD, KITTI and D^2-City Datasets in terms of MOT Metric.

Dataset	MOTA	MOTP	MT	ML	IDF1
SAD	62.2	80.2	31.8%	19.0%	60.2
KITTI	63.5	78.3	35.1%	18.2%	57.3
D^2-City	61.7	79.6	29.5%	20.1%	55.2

D. EXPERIMENTAL RESULT AND ANALYSIS

1) ASSESSMENT OF THE PREDICTIVE PERFORMANCE OF FRPN

The quality of video prediction determines the efficiency of the driving assistance method. We compare FRPN with the VGG-GAN model proposed by Shouno *et al.* [17] and the PredNet model proposed by Villegas *et al.* [25], as well as the FRPN model trained with three different global feature extractors. The results are presented in Table 3.

It can be observed from the comparison results in the table that, compared with the PredNet structure proposed by Villeagas *et al.*, the deeper convolutional structure provides more feature representations of complex scenes. In the loss function, the constraint of the real image is added, and the experimental results intuitively show that this method can minimize the difference from the real image and significantly improve the SSIM of the image. However, Other models have a good effect on the results of the single-step prediction. the effect after multiple time steps is not good. It is difficult to address our follow-up early accident warning task. Therefore, the FRPN based on VGGNet is selected as the supporting plate for the multi-target tracking experiment.

2) PARTIAL RESULTS DISPLAY

Fig. 6 shows some examples of sequence images generated by the trained generator and several ground truths, which

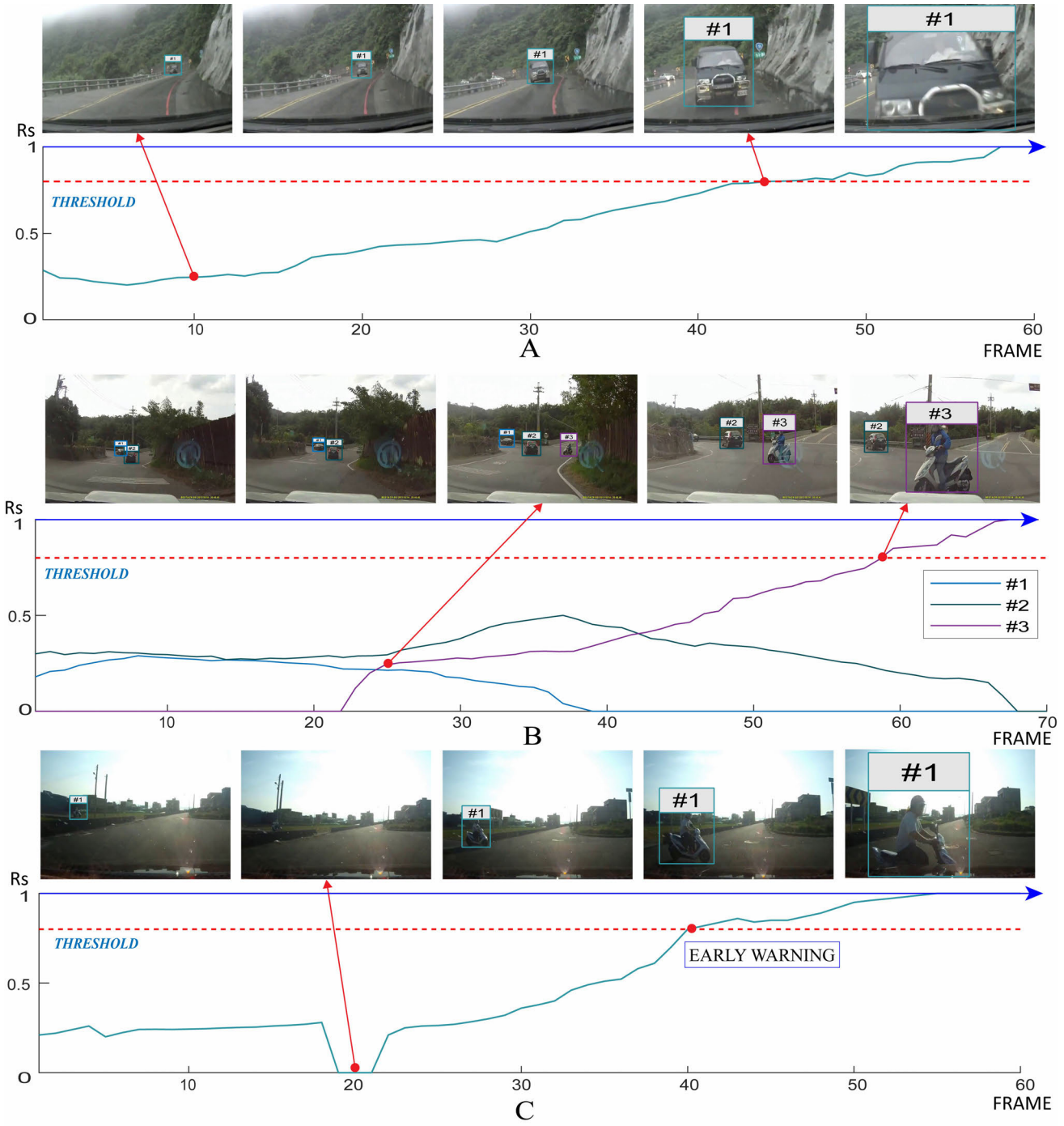


FIGURE 7. Positive examples of our method when the threshold was set to 0.8. The horizontal axis represents the number of video frames, the vertical axis represents the risk score (Rs) of the identified target in the field of view, and the red arrow refers to the risk factor of the target in the frame. (A) is the single-target score calculation of the video frame under real situation, (B) is the multi-target score calculation of the video frame under real situation, and (C) is the single-target score calculation of the video frame under prediction situation.

are the same frames obtained from the video. As can be observed, these models could make certain predictions for future pictures in future time steps without exception. However, owing to uncertainty, entities with a large range of motion will spread, and the image will gradually become blurred. Nevertheless, despite distortion, our model provided more refined and accurate results. The other sequence models clearly suffered from the disappearance of gradients and explosions. Furthermore, the previous methods were unable to adapt when the frame changed significantly. To prevent the gradient explosion in the experiment, a gradient penalty was adopted for the discriminator training. The qualitative results proved that our method avoided these phenomena by adopting a multi-scale fusion structure for adversarial training.

TABLE 5. Experimental results of the early warning capability on SAD Dataset with evaluation system (ES) based on different models.

Model	Accuracy(%)	Precision(%)	Recall(%)	F1 score(%)	ETC(s)
VGG-GAN	65.4	77.5	60.4	64.1	1.63
PredNet	54.6	63.1	55.4	59.0	1.22
FRPN +InceptionNet	58.7	66.5	59.2	62.6	1.62
FRPN +VGGNet	68.2	75.1	65.3	69.9	1.95
FRPN +ResNet	70.3	77.4	64.5	70.4	1.03

3) ASSESSMENT OF THE FINAL RESULT

As shown in Table 4, the CLEAR-MOT metric [37] was used to evaluate the trained JDE model on three different datasets (where MOTA means Multiple Object Tracking Accuracy, MOTP means Multiple Object Tracking Precision, MT means Mostly Tracked targets, ML means Mostly Lost targets, and IDF1 means ID F1 Score. The indicators of MOTA, MOTP, MT, and IDF1 in the table are better for higher values, and the ML is better for lower values). The results indicate that the detection-tracking model achieved relatively stable results in different environments, in spite of with some subtle differences. In relatively simple dataset KITTI, the trained JDE model can reach a normal level with the value of MOTA up to 63.5 and the value of MT up to 31.5%.

The final early warning capabilities of the different backbones under good tracking conditions are presented in Table 5. It can be seen that the different models and the complex scenes do cause the increase of the response time of the early warning system to issue warnings, but the proposed method has clear advantages over the other methods. In the best case, there is a 15.7% increase in accuracy. GAN-VGG proposed by Shouno *et al.* [17] has achieved good results in advance warning time, but in more complex scenes, the accuracy is lower. Compared with different global feature extraction models, ResNet achieved better results in terms of recognition the accuracy was 70.3%, but the complexity of its network and so many parameters cause scene prediction delays. The structure of InceptionNet significantly reduced the amount of calculation. However, in complex scenarios, the feature extraction capability is still lacked, and the accuracy of its risk score is only higher than that of PredNet, which has the simplest structure. Although the model based on VGGNet was not the most accurate, it could issue a warning a long time in advance, which is a compromise choice. Therefore, this method itself has limitations, with two extreme cases existing. The more complex the scene is, the more GPU resources will be consumed, and the harder it was to issue a warning in advance. However, in reality, drivers always pay attention to traffic conditions ahead. Therefore, the proposed method can only be used as an auxiliary means to effectively predict accidents, specifically providing notice to drivers to brake, thereby avoiding accidents.

Fig. 7 shows the example results of our best method in some cases on the SAD dataset, where the threshold represents the lowest value judged to be dangerous. The assessment of the degree of risk depends on the accurate prediction and tracking of the target location. In the case of roughly the same predictive performance, accurate tracking determines the continuity of the risk score. The increase in the area of the target in the FOV implies that it is approaching the vehicle, and its risk score will increase, but the loss of the target cannot be ruled out. As shown in Figure 7(C), the target was lost due to occlusion, and its risk score was temporarily lost; therefore, this limitation cannot be ignored. Evidently, the proposed algorithm can more accurately calculate the risk value generated by the position of the vehicle in the front view and can allow the system to perform the earliest from when traffic accidents are expected.

IV. CONCLUSION AND FUTURE WORK

In this study, an unsupervised deep learning framework FRPN for traffic accident video prediction based on first-person is established, and a risk scoring evaluation method is proposed. This video prediction model fuses multi-scale residual features, and combines improved adversarial learning with real image constraints and gradient penalty as training targets. Then via this multi-target tracking model JDE, FRPN can predict the trajectory of abnormal targets in front of the vehicle accurately. Subsequently, in order to avoid the occurrence of accidents or reduce the damage caused by accidents, it calculates risk scores based on the position to provide early warning while targets have reached the risk threshold. The experiments indicate that the proposed model can be adapted to more complex environments on different datasets and achieved a high performance in the new dataset SAD.

In recent research, the application of 2D images has gradually become more mature, and the prediction of video sequences containing time information is developing rapidly. Video prediction is crucial in many fields, such as traffic accident prediction, fire prediction, typhoons, and rainfall.

Although unmanned driving provides automatic operation for the driver, assisted driving still has considerable research significance, especially in sudden abnormal conditions, which could substantially reduce the degree of damage and issue rescue notifications immediately. Limited by the network scale and hardware equipment, video prediction still has a large room for development. We hope to achieve more accurate and long-term forecasting results and better tracking results. Although our final evaluation algorithm cannot make a completely accurate risk judgment, our overall framework can still be widely used to reduce the risk as much as possible.

ACKNOWLEDGMENT

(Yun-Feng Zhou, Kai Xie, and Xin-Yu Zhang contributed equally to this work.)

V. AUTHOR CONTRIBUTIONS

Yun-Feng Zhou conceived the algorithms, and designed the experiments; Kai Xie reviewed the paper; Xin-Yu Zhang checked the spelling and made suggestions; Chang Wen conducted the comparative experiment on images; Jian-Biao He is responsible for data collection.

REFERENCES

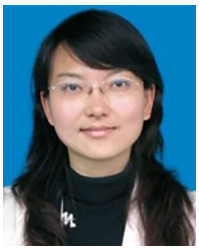
[1] *2012 Motor Vehicle Crashes: Overview*, Nat. Highway Traffic Saf. Admin., Washington, DC, USA, 2013.
[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[3] F. H. Chan, Y. T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 136–153.
[4] X. Yan, H. Chang, S. Shan, and X. Chen, "Modeling video dynamics with deep dynencoder," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 215–230.
[5] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, pp. 2236–2250, 2016.
[6] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani, "Compositional video prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10353–10362.
[7] R. Jasti, "Multi-frame video prediction with learnable temporal motion encodings," Univ. California, Merced, CA, USA, 2020. [Online]. Available: https://escholarship.org/uc/item/1n3761rc
[8] X. Shi, Z. Chen, H. Wang, and D. Y. Yeung, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," 2016, *arXiv:1605.08104*. [Online]. Available: http://arxiv.org/abs/1605.08104
[11] H. Wu, Z. Yao, M. Long, and J. Wang, "MotionRNN: A flexible model for video prediction with spacetime-varying motions," 2021, *arXiv:2103.02243*. [Online]. Available: http://arxiv.org/abs/2103.02243
[12] Y. Wu, R. Gao, J. Park, and Q. Chen, "Future video synthesis with object motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5539–5548.
[13] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in *Proc. Eur. Conf. Comput. Vis.* Cham Springer, 2016, pp. 776–791.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Algorithms*, 2014, pp. 2672–2680.
[15] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.
[16] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1744–1752.
[17] O. Shouno, "Photo-realistic video prediction on natural videos of largely changing frames," 2020, *arXiv:2003.08635*. [Online]. Available: http://arxiv.org/abs/2003.08635
[18] X. Lin, Q. Zou, X. Xu, Y. Huang, and Y. Tian, "Motion-aware feature enhancement network for video prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 688–700, Feb. 2021.
[19] P. Luc, A. Clark, S. Dieleman, D. de Las Casas, Y. Doron, A. Cassirer, and K. Simonyan, "Transformation-based adversarial video prediction on large-scale data," 2020, *arXiv:2003.04035*. [Online]. Available: http://arxiv.org/abs/2003.04035
[20] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1036–1043.
[21] Y. Cai, L. Dai, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 28, 2021.
[22] H. Ren, Y. Song, J. Wang, Y. Hu, and J. Lei, "A deep learning approach to the citywide traffic accident risk prediction," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3346–3351.
[23] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," 2019, *arXiv:1903.00618*. [Online]. Available: http://arxiv.org/abs/1903.00618
[24] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident DB," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3521–3529.
[25] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," 2017, *arXiv:1706.08033*. [Online]. Available: http://arxiv.org/abs/1706.08033
[26] 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.
[27] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[28] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2018–2025.
[29] I. Gulrajani, F. Ahmed, M. Arjovsky, and V. Dumoulin, "Improved training of Wasserstein GANs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5769–5779.
[30] B. Fuglede and F. Topsoe, "Jensen-Shannon divergence and Hilbert space embedding," in *Proc. Int. Symp. onInformation Theory,. ISIT. Proceedings.*, 2004.
[31] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
[32] C. Villani, *Optimal Transport: Old and New*, vol. 338. Berlin, Germany: Springer, 2008.
[33] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015, *arXiv:1511.05440*. [Online]. Available: http://arxiv.org/abs/1511.05440
[34] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," 2019, *arXiv:1909.12605*. [Online]. Available: http://arxiv.org/abs/1909.12605
[35] T.-Y. Lin, P. Dollar, 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. 2017, pp. 2117–2125.
[36] Z. Che, G. Li, T. Li, B. Jiang, X. Shi, X. Zhang, Y. Lu, G. Wu, Y. Liu, and J. Ye, "D^2-City: A large-scale dashcam video dataset of diverse traffic scenarios," 2019, *arXiv:1904.01975*. [Online]. Available: https://arxiv.org/abs/1904.01975
[37] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Mar. 2008.
[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[42] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

YUN-FENG ZHOU was born in Hubei, China, in 2000. He joined the National Demonstration Center for Experimental Electrical and Electronic Education, in 2019, with the intent to research machine learning and image processing. He is currently an Assistant Researcher with Yangtze University, Jingzhou, China. His research interests include image processing, software development, and machine learning.

KAI XIE received the M.S. degree in electronic engineering from the National University of Defense Technology, Changsha, China, in 2003, and the Ph.D. degree in pattern recognition and intelligent system from Shanghai Jiao Tong University, Shanghai, China, in 2006. He is currently a Professor with the School of Electronic Information, Yangtze University, Jingzhou, China. His current research interests include image processing and signal processing.

XIN-YU ZHANG was born in Sichuan, China, in 2001. She is currently an Assistant Researcher with Yangtze University, Jingzhou, China. Her research interests include image processing and artificial intelligence. She joined Laboratory with the intent to research deep learning and image processing. She has been conducting research projects on image recognition and video prediction.



CHANG WEN received the B.S. degree in computer science from the Naval University of Engineering, Wuhan, China, in 2002, and the M.S. degree in computer science from Yangtze University, Jingzhou, China, in 2008. She is currently an Assistant Professor with the School of Computer Science of Yangtze University. Her current research interests include image processing and signal processing.

JIAN-BIAO HE received the B.S. and M.S. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1986 and 1989, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Central South University. His research interests include artificial intelligence, the Internet of Things, pattern recognition, mobile robots, and cloud computing.