Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8848874, 12 pages
https://doi.org/10.1155/2020/8848874

Research Article

A New Video-Based Crash Detection Method: Balancing Speed and Accuracy Using a Feature Fusion Deep Learning Framework

Zhenbo Lu,[1] Wei Zhou ,[1] Shixiang Zhang,[2] and Chen Wang [1]

[1] *Intelligent Transportation Research Center, Southeast University, Nanjing, China*
[2] *China Design Group Co., Ltd., Nanjing, China*

Correspondence should be addressed to Chen Wang; wkobec@hotmail.com

Received 15 August 2020; Revised 28 September 2020; Accepted 24 October 2020; Published 16 November 2020

Academic Editor: Kun Wang

Quick and accurate crash detection is important for saving lives and improved traffic incident management. In this paper, a feature fusion-based deep learning framework was developed for video-based urban traffic crash detection task, aiming at achieving a balance between detection speed and accuracy with limited computing resource. In this framework, a residual neural network (ResNet) combined with attention modules was proposed to extract crash-related appearance features from urban traffic videos (i.e., a crash appearance feature extractor), which were further fed to a spatiotemporal feature fusion model, Conv-LSTM (Convolutional Long Short-Term Memory), to simultaneously capture appearance (static) and motion (dynamic) crash features. The proposed model was trained by a set of video clips covering 330 crash and 342 noncrash events. In general, the proposed model achieved an accuracy of 87.78% on the testing dataset and an acceptable detection speed (FPS > 30 with GTX 1060). Thanks to the attention module, the proposed model can capture the localized appearance features (e.g., vehicle damage and pedestrian fallen-off) of crashes better than conventional convolutional neural networks. The Conv-LSTM module outperformed conventional LSTM in terms of capturing motion features of crashes, such as the roadway congestion and pedestrians gathering after crashes. Compared to traditional motion-based crash detection model, the proposed model achieved higher detection accuracy. Moreover, it could detect crashes much faster than other feature fusion-based models (e.g., C3D). The results show that the proposed model is a promising video-based urban traffic crash detection algorithm that could be used in practice in the future.

1. Introduction

Traffic crashes can cause property damage, injuries, death, and nonrecurrent congestions. Accurate and fast crash detection can help improve the response speed of incident management, which in turn reduces injuries/fatalities and congestions induced by crash occurrence. Thus, developing such crash detection methods is necessary and important for traffic incident management.

Traditional crash/incident detection methods mostly rely on traffic flow modeling techniques [1–7]. The basic idea of traffic flow modeling is to identify nonrecurrent congestion, based on data from loop detectors, microwaves, and probe. However, nonrecurrent congestion and recurrent congestion can be difficult to be differentiated without enough and sound historical data. Thus, the performance of traffic flow modeling approach heavily depends on the data quality obtained from traffic detectors. Moreover, it could often fail when the traffic environment is too complex (e.g., multimodal traffic in urban area). Thus, detection accuracy of such method is sometimes not guaranteed. Another emerging method is to identify incident based on crowdsourcing data [8]. However, such method could also suffer from underreporting issues when there is no witness around the incident scene. Nowadays, with the development of intelligent transportation systems (ITS), video cameras have been widely installed in many cities and highways. Thanks to their wide coverage, vision-based crash detection techniques have gained increasing research attention in the recent years [9]. Their basic concept is to automatically identify crash scenes based on the features of traffic images/videos through computer-vision techniques. Such techniques, as a promising intelligent crash detection method, are expected to significantly reduce human labors and have achieved relatively high detection accuracy [10–12].

To ensure detection accuracy, a video-based crash detection method needs to be capable of extracting important crash features from traffic images/videos. In general, there are two main types of features of interest: motion (temporal) features and appearance (spatial) features. Appearance features include apparent vehicle damage, vehicle rollovers, and pedestrian fallen-off. Motion features need to be continuously identified, including the intersection of vehicle trajectories and the gathering of pedestrians. From this perspective, current crash detection methods can be classified into two groups: motion feature-based methods and feature fusion-based methods.

Many research works are based on motion features, such as the intersection of vehicle trajectories, the overlap of bounding box detectors, and the speed change of vehicles. Some used background subtraction methods to extract vehicles' motion features (acceleration, direction, and velocity), based on which certain rules and thresholds were applied to identify crashes [9, 13–15]. Maalou et al. [16] tracked vehicles' motion based on optical flow methods and used heuristic methods to find a threshold for crash identification. Sadeky et al. [17] used Histogram of Flow Gradient (HFG) as the motion features and discriminated crash from noncrash, based on logistic regression. Chen et al. [18] developed an Extreme Learning Machine (ELM) for crash identification, based on motion features represented by Scale-Invariant Feature Transform (SIFT) and optical flow. In recent years, with the development of deep learning methods (e.g., Faster R-CNN (Faster Region-based CNN) [19] and YOLO (You Only Look Once) [20–22]), the performance of vehicle detection and tracking has been significantly improved. Vicente and Elian [23] used YOLO model to detect motion features and used support vector machine (SVM) for crash identification. Lee and Shin [24] used Faster R-CNN for vehicle detection and Simple Online and Real-Time tracking (SORT) for vehicle tracking. Based on those motion features, the incident/crashes in tunnels were detected. Paul [25] applied Mask R-CNN (Mask Region-based CNN) for motion feature extraction and used rules for crash detection. Motion feature-based models only depend on vehicle motions. This requires a high precision of object detection and tracking. When the traffic environment is complicated, vehicle detection and tracking performance could be decreased, resulting in low crash detection performance. Furthermore, some crashes may not be detected only based on motions, such as vehicle rollover and pedestrian fallen-off.

Recently, the feature fusion-based (i.e., appearance and motion) crash detection methods have become increasingly popular. There are two types. One is based on unsupervised learning methods. For instance, Singh and Mohan [26] and Yao [27] developed a crash detection model based on autoencoder methods. Another type is based on supervised learning framework, which normally combines a module (e.g., convolutional neural network) for spatial feature extraction and a module (e.g., a recurrent neural network) for temporal feature extraction. Batanina et al. [28] used Convolutional 3D (C3D) model to capture both spatial and temporal crash features from simulated video crashes. Then, a domain adaption (DA) transfer learning was applied to the real-world condition. The accuracy has been improved by 10%. Huang et al. [29] employed two-stream network to separately extract appearance features and motion features, which were then further combined to detect crashes. According to previous literature, the performance of crash detection can be improved by feature fusion methods.

Although feature fusion-based methods have achieved a better performance than motion feature-based methods, some improvements still can be made. To simultaneously capture both motion and appearance features, such models oftentimes have complicated structures and a large number of parameters. As such, those models require a lot of computing resources and long computational time, which prevent them from being used in a real-time traffic environment. Thus, the current fusion-based models need to find a better balance between detection accuracy and speed.

To fill the gap, we proposed a new feature fusion-based urban traffic crash detection framework, aiming at achieving a good balance between detection accuracy and speed. First, we introduced attention module into residual neural networks to improve the performance of detecting local appearance features. Meanwhile, we linked ResNet with Conv-LSTM model to simultaneously capture crashes' appearance and motion features. The proposed model is expected to achieve high accuracy as well as fast detection speed for crash detection. The remainder of the paper is organized as follows: Section 2 introduces methods used in this study. Section 3 discusses data preparation. Section 4 presents modeling results and discusses research findings. Section 5 provides the research conclusion and future directions.

2. Methodology

In this section, we introduce our proposed model in detail. Figure 1 shows the overall framework of our model. First, the attention module was combined with ResNet to capture the appearance features of the crash images. The ResNet can improve the speed of conventional convolution neural network, while the attention module can enable the model to focus on localized appearance features instead of other irrelevant information to further boost the model. Then, the output feature map is reduced in dimension via a 1×1 convolutional layer, which is then chronologically input into the Conv-LSTM network to further extract the motion features of crashes. Conv-LSTM has an advantage over conventional recurrent neural network (e.g., LSTM) in terms of being lighter and retaining spatial information. Finally, a global pooling layer and a fully connected layer were used to detect a crash (or noncrash). The following is a detailed description of the residual network ResNet, attention module, and Conv-LSTM module in the framework.

2.1. Crash Appearance Feature Module (ResNet + Attention)

2.1.1. Residual Neural Network (ResNet). Residual neural network (i.e., ResNet) was proposed in 2015 [30] and has been widely used in various deep learning-based computer-

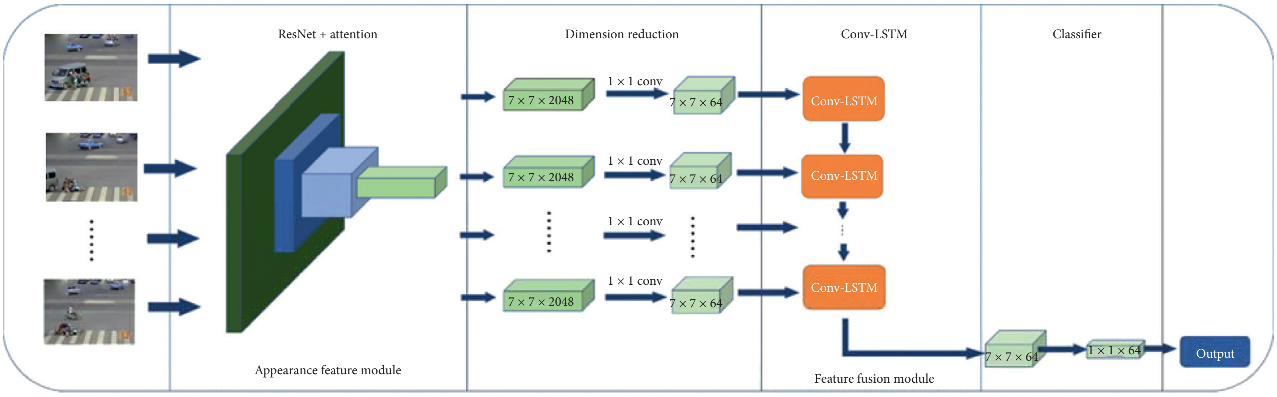

Figure 1: The proposed crash detection model framework.

vision tasks for extracting image features. The purpose of ResNet solves the problem of training difficulties caused by gradient explosion or vanishing in deep convolutional neural networks. Compared to other conventional neural networks (e.g., VGG (Visual Geometry Group Network)) continuously stacking convolutional layers to obtain higher image expression capabilities, ResNet stacks flexible residual modules to obtain stronger expression ability instead.

There are two main types of residual modules. The first type (Figure 2(a)) often appears in a shallow residual network (ResNet 18/34). Each residual module includes two 3 × 3 convolutions, the output of which is the sum of the input (i.e., the output from the last residual module) and its convolution. The ReLU activation function is used to obtain the output of the current residual module, as shown in the following equation:

$$\text{output} = \text{RELU}\left(\text{input} + W_2\left(\text{RELU}\left(W_1\left(\text{input}\right)\right)\right)\right), \tag{1}$$

where W_i is the 3 × 3 convolution operation and i is the layer index.

The second type (Figure 2(b)) often appears in deeper residual networks (ResNet50/101/152). Each residual module includes three convolution layers (1 × 1, 3 × 3, and 1 × 1), the output of which is the sum of the input (i.e., the output from the last residual module) and its convolution.

$$\text{output} = \text{RELU}\left(\text{input} + X_2\left(\text{RELU}\left(W_1\left(\text{RELU}\left(X_1\left(\text{input}\right)\right)\right)\right)\right)\right), \tag{2}$$

where X_i is the 1 × 1 convolution operation.

Since such residual network structure can compensate for the unstable training caused by deep structures [30], it can handle deeper network layers than VGG. Three typical ResNets are 50 layers (ResNet-50), 101 layers (ResNet-101), and 152 layers (ResNet-152). They are similar in structure. The selection of ResNet depends on computational capability and training data amount. Deeper network could be more powerful with adequate training data.

2.1.2. Visual Attention Module. In this paper, we further extend ResNet by integrating visual attention modules. The visual attention module squeeze-and-excitation (SE) Block was first proposed by Hu et al. [31]. This module has been widely used because it is relatively simple and is able to improve the efficiency of many convolutional network models. SE Block belongs to the channel attention mechanism, which gives different weights to different channels of a feature map. As is known, in convolutional neural networks, different channels correspond to different feature extractions. The different classification tasks should lay particular emphasis on different feature selections. The concept is similar to the way that human beings identify objects. For example, people may pay more attention to the shape features when judging cats and dogs, while they may focus on texture features when judging jaguar and leopard (belonging to Felidae). Thus, SE Block improves the ability of feature selection for convolutional neural networks.

As shown in Figure 3, SE Block converts the input X, $X \in \mathbb{R}^{H \times W \times C}$, to U, $U \in \mathbb{R}^{1 \times 1 \times C}$, through a global average pooling operation F_{sq}, as shown in the following equation:

$$U = F_{\text{sq}}(X). \tag{3}$$

For X_i, $X_i \in \mathbb{R}^{H \times W}$, in input tensor $X = (X_1, X_2, \ldots, X_C)$ and U_i, $U_i \in \mathbb{R}^{1 \times 1}$, in output tensor $U = (U_1, U_2, \ldots, U_C)$, the following equation holds.

$$U_C = F_{\text{sq}}(X_C) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_C(i, j). \tag{4}$$

After the global average pooling, the output U is passed through a fully connected layer with a weight of W, $W \in \mathbb{R}^{C \times C}$, that is, $F_{\text{ex}}(\cdot, W)$ in Figure 3, and the result V is as shown in equation (5), where "$*$" refers to matrix multiplication.

$$V = F_{\text{ex}}(U, V) = \sigma(U * V). \tag{5}$$

The above is the activation function, and the result V is also called attention weight. Finally, multiply the attention weight V and the input X by the channel weight to adjust the importance of different channels of the input (equation (6)). Here "$\circ$" refers to the element-wise multiplication (i.e., Hadamard product):

$$Y = F_{\text{scale}}(X, V) = X \circ V. \tag{6}$$

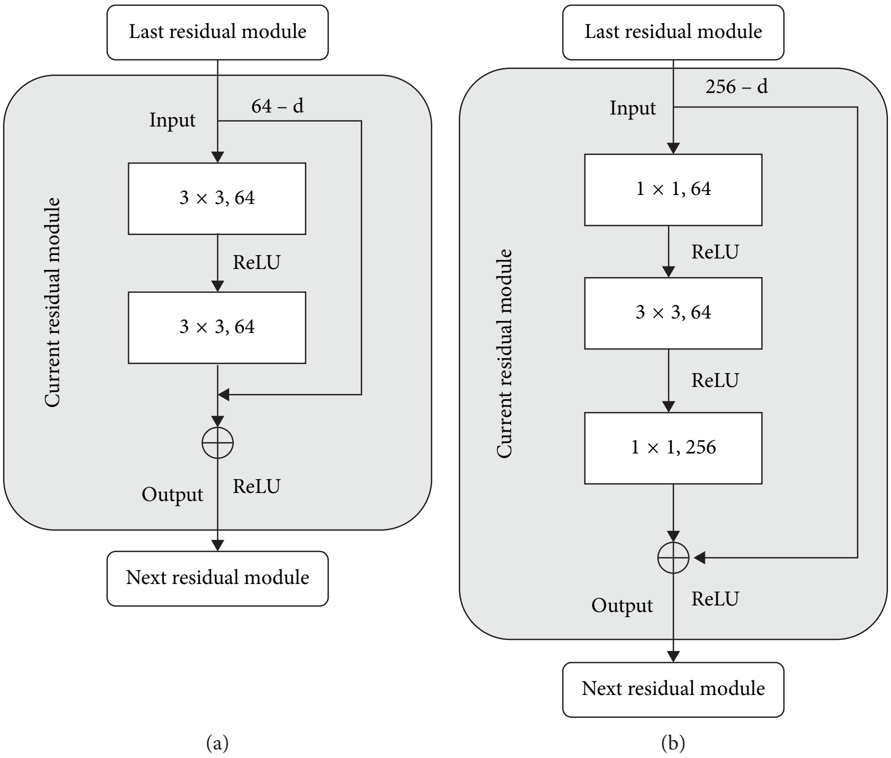

FIGURE 2: Basic components for residual neural network. (a) ResNet 18/34. (b) ResNet 50/101/152.

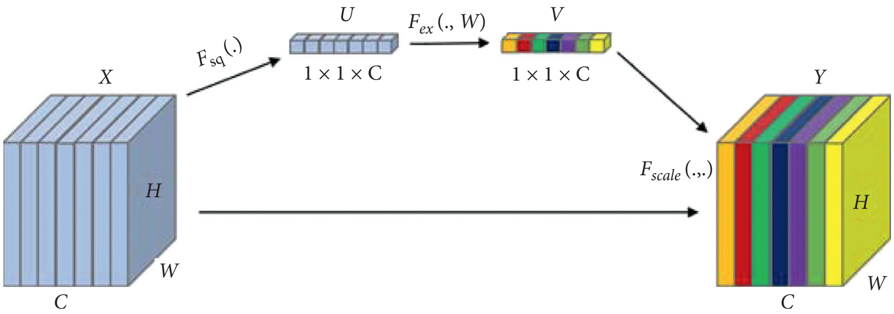

FIGURE 3: SE Block.

An improved visual attention module over SE is called convolutional block attention module (CBAM), which was first proposed by Gupta [32]. Based on the basic channel visual attention module (i.e., SE), CBAM innovatively introduces the spatial visual attention module, as shown in Figure 4. Different from the basic module, the spatial visual attention module initially performs maximum pooling and average pooling operations $F_{s\_\mathrm{sq}}(\cdot)$ on the input X_S by channel and then converts the two-layer feature map to a single-layer feature map through a 1 × 1 convolutional layer with a weight of W, as shown in $F_{s\_\mathrm{ex}}(\cdot, W)$ in Figure 4. Finally, softmax is used to convert the original distribution to a probability distribution and adjust the importance of the model to different spatial positions of the input X_S. The process can be expressed by the three following equations:

$$U_S = F_{s\_\mathrm{sq}}(X_S), \tag{7}$$

$$V_S = F_{s\_\mathrm{ex}}(U_S, W) = \mathrm{soft\,max}(U_S * W), \tag{8}$$

$$Y_S = F_{\mathrm{scale}}(X_S, V_S) = X_S \circ V_S. \tag{9}$$

CBAM module can be embedded into residual modules to improve its feature selection performance. Figure 5 shows how the two modules are integrated.

2.2. Feature Fusion Module (Conv-LSTM). The Conv-LSTM module was first used in precipitation nowcasting [33], the structure of which is shown in Figure 6. Traditional LSTM input requires data flattening, which often causes spatial

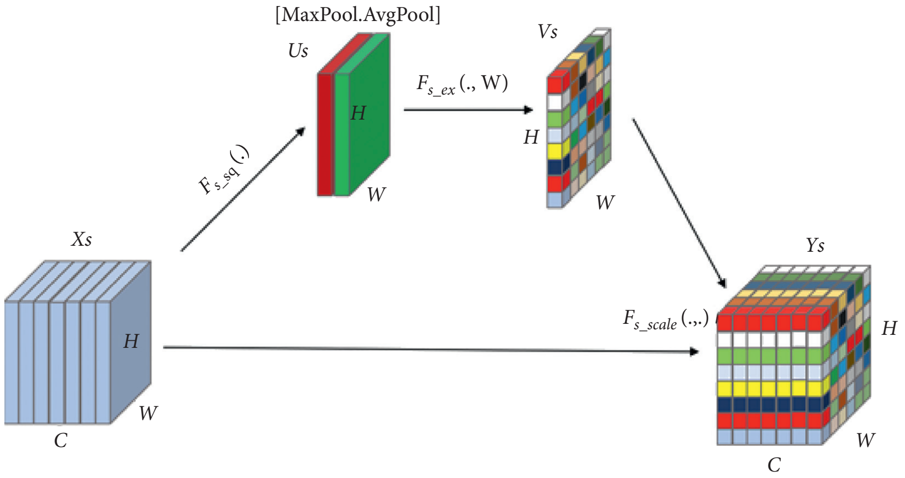

Figure 4: CBAM module.

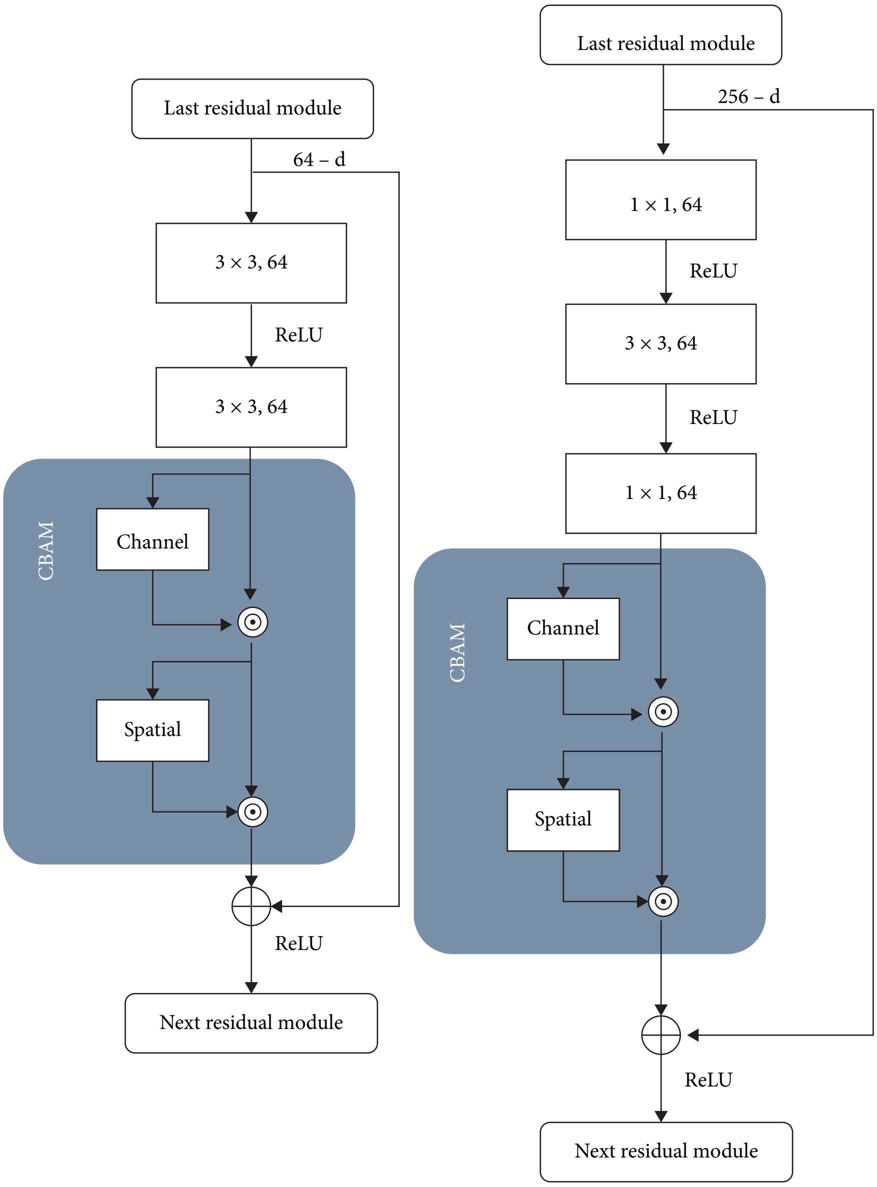

Figure 5: The position of the CBAM module in a residual model.

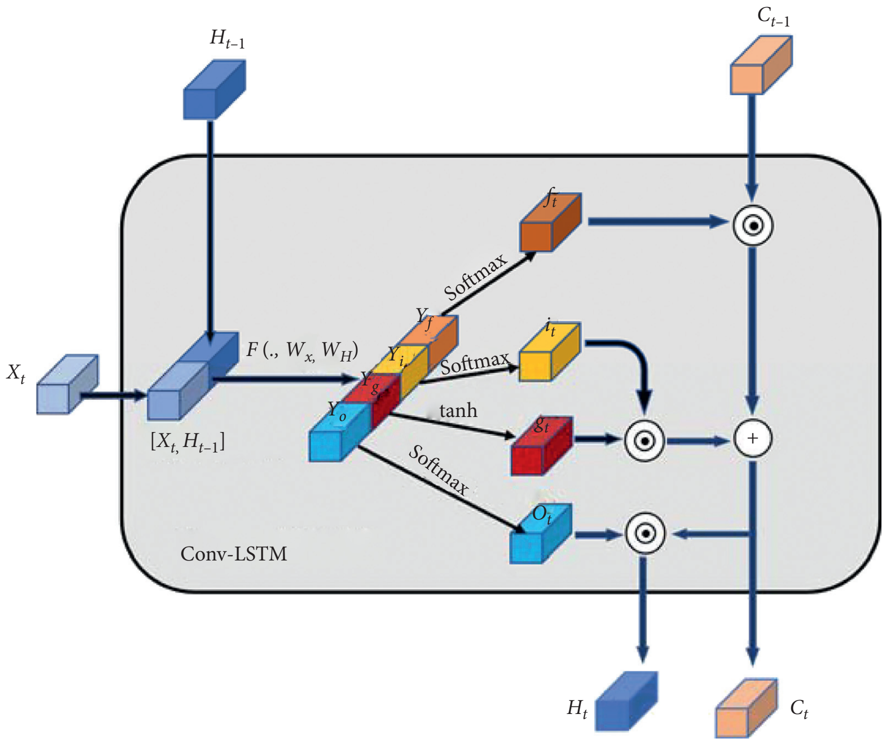

Figure 6: Conv-LSTM model.

information loss. The Conv-LSTM module inherits the gating structure adopted by the traditional LSTM, while it uses convolution neuron as a basic unit to retain spatial features. The data modeling process is as follows.

First, the inputs χ_t and H_{t-1} are stacked along the channel dimension to generate $[\chi_t, H_{t-1}]$; then a one-dimensional convolution $F\,(\cdot; W_\chi, W_H)$ performs convolution operation on $[\chi_t, H_{t-1}]$:

$$\left[Y_f, Y_i, Y_g, Y_o\right] = F\Big(\left[\chi_t, H_{t-1}\right]; W_\chi, W_H\,]\Big). \tag{10}$$

Then, obtain $[f_t, i_t, g_t, o_t]$ by using activation function on $[Y_f, Y_i, Y_g, Y_o]$, as shown in the following equation:

$$\begin{aligned}\left[f_t, i_t, g_t, o_t\right] = \Big[&\text{soft max}\left(Y_f\right), \text{soft max}\left(Y_i\right),\\ &\text{soft max}\left(Y_g\right), \text{soft max}\left(Y_O\right)\Big].\end{aligned} \tag{11}$$

Finally, the outputs C_t and H_t of the Conv-LSTM module at the time step t are obtained by gating operations, as shown in the two following equations:

$$C_t = f_t \circ C_{t-1} + i_t \circ g_t, \tag{12}$$

$$H_t = o_t \circ C_t = o_t \circ \left(f_t \circ C_{t-1} + i_t \circ g_t\right). \tag{13}$$

3. Data Preparation

To the best of our knowledge, there is no public database for crash detection task. Thus, in this study, all data were acquired from local police in China. We prepared two datasets. The first dataset is an urban city traffic image dataset, which contains 5061 traffic crash images and 5573 noncrash traffic images. The crash images include multiple types, such as single-vehicle, multivehicle, and non-motorist-related crashes. Figure 7 shows some examples. Another dataset is an urban surveillance video dataset, which contains 420 crash video clips and 432 noncrash video clips. The duration of each video clip is around 20 seconds, with 24/25 frames per second.

The first image dataset was used to train the ResNet plus attention module, while the video dataset was used to train the whole network. By transferring the pretrained ResNet module, the convergence speed of the whole network could be boosted. To note that, images/video clips were manually labeled with either crash or noncrash. As such, the capability of the trained model was expected to identify crashes among normal traffic scenes.

4. Results and Discussion

All experiments in this study were carried out on a laptop equipped with Nvidia GTX 1060 GPU. Some detailed parameters of the laptop are as follows: (1) I7-7700HQ CPU @2.80 GHZ and (2) GTX 1060 (6G) GPU, core frequency: 1506-1709MH, and floating-point operation: 4.4 TFLOPs.

First, a set of deep learning models was compared for differentiating crash images (positive) from noncrash images (negative), with the purpose of finding a best crash appearance feature module, which was further linked to Conv-LSTM. VGG-16 and ResNet-50 were used as baseline models. Four extended models were developed by incorporating SE and CBAM modules into VGG and ResNet. The training dataset included 3861 crash and 4373 noncrash images, while the testing dataset included 1200 images for each category. Table 1 shows the performances of those

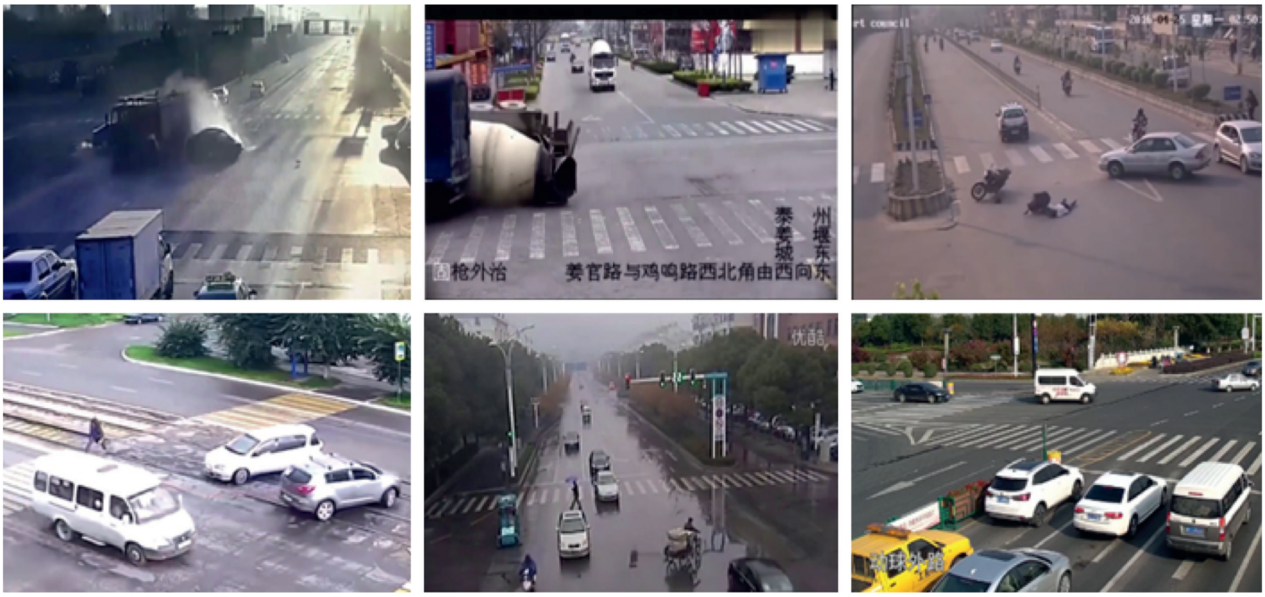

Figure 7: Traffic image dataset.

Table 1: Performance of crash appearance feature extractors.

Model name	True positive (TP)	False negative (FN)	False positive (FP)	True negative (TN)	Accuracy (%)
VGG-16	1056	144	273	927	82.63
ResNet-50	1087	113	233	967	85.58
VGG-16 + SE	1075	125	251	949	84.33
VGG-16-CBAM	1103	97	231	969	86.33
ResNet-50 + SE	1132	68	214	986	88.25
ResNet-50- + CBAM	1135	65	171	1029	90.17

models (i.e., crash appearance feature extraction models) in the test dataset. Compared to VGG-16 and ResNet-50, extended models with attention modules generally had higher detection accuracy. Among those, the ResNet-50 + CBAM model achieved the highest accuracy of 90.17%. Figure 8 shows the testing accuracy for each training epoch for those crash appearance extraction models.

It can be also found that all models had much more false-positive (FP) cases than false-negative (FN) cases. This indicates that those models tend to determine noncrash traffic scenes as crashes. Some traffic conditions (e.g., stopped vehicles and heavy congestions with many overlapping pedestrians and vehicles) could have very similar appearance features to those of crash scenes. Thus, models solely based on appearance features cannot well identify those conditions.

We further visualize those models based on the gradient-weighted class activation mapping (Grad-CAM) technique [34], as shown in Figure 9. ResNet appeared to be better than VGG in terms of focusing on the appearance features of crashes. For example, VGG failed to identify the appearance features of crash *D*, while ResNet identified them correctly. When adding attention modules, the extended models (ResNet 50 + SE/CBAM) could better focus on appearance

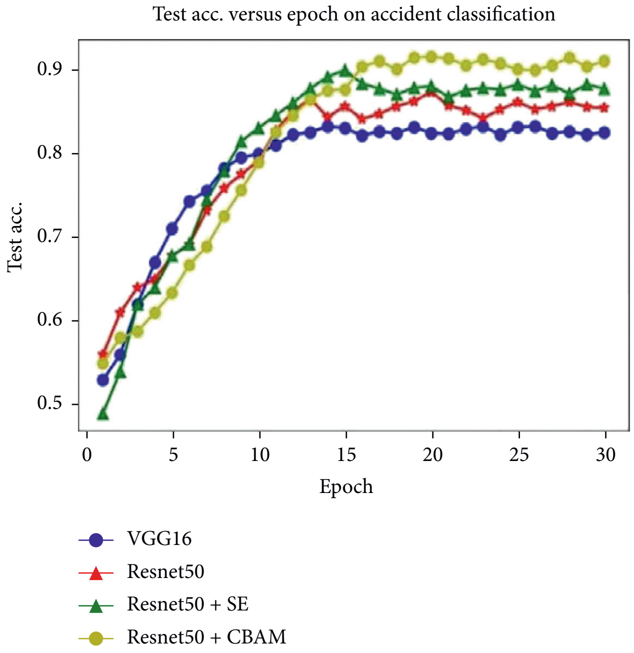

Figure 8: Testing accuracy for each training epoch for crash appearance extractors.

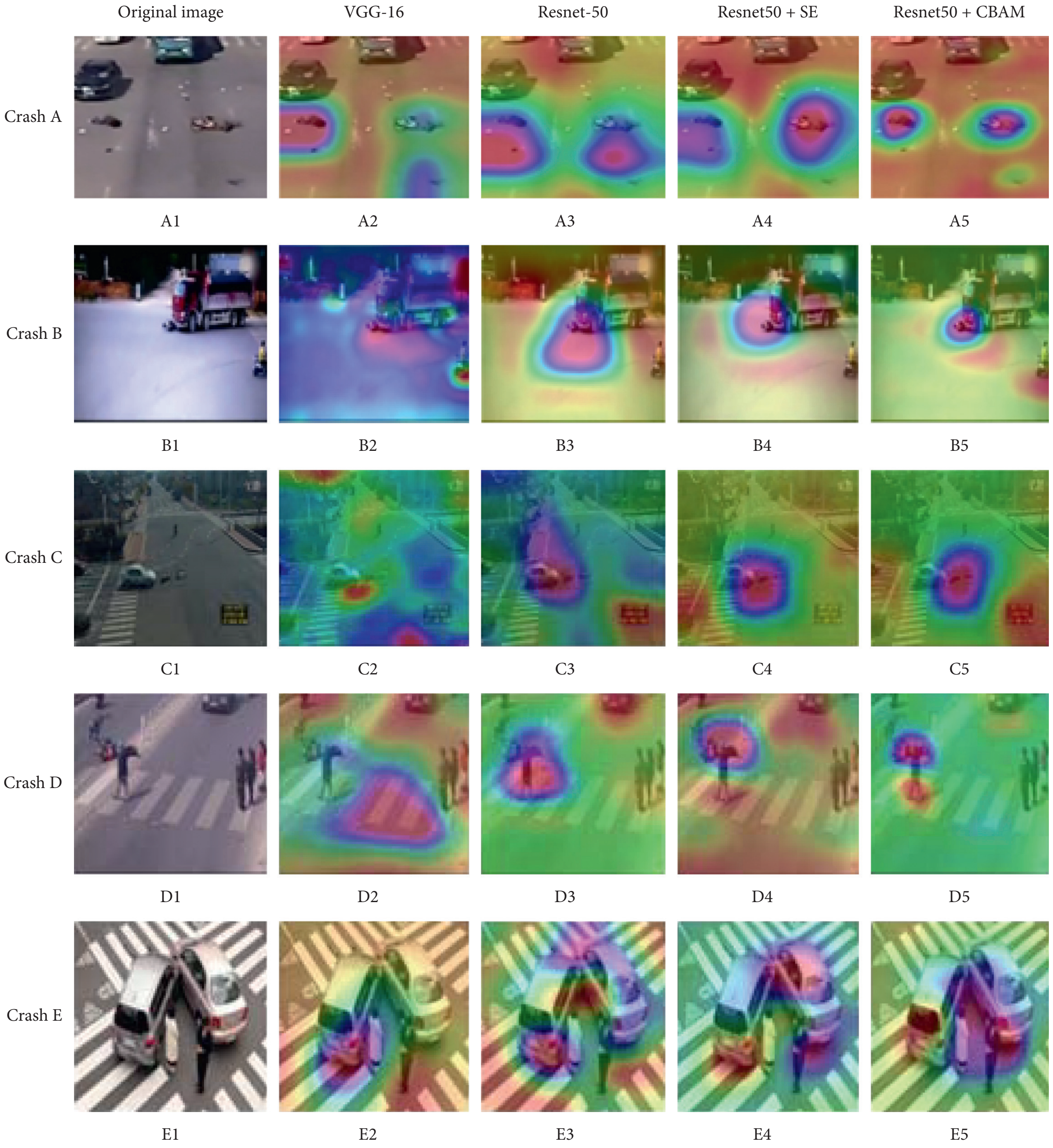

Figure 9: Crash appearance feature extraction visualization.

features. For instance, CBAM improved the performance of ResNet in recognizing the location of fallen people for crash C.

Then, using ResNet50 + CBAM as the pretrained model, we further trained the whole model with 330 crash video clips and 342 noncrash video clips. The testing dataset included 90 crash videos and 90 noncrash videos. To examine the performance of the proposed model, six models were compared. Model 1 determined crashes based on the amount of velocity changes or anomalies of trajectory extracted by Faster R-CNN + SORT. Model 2 identified crashes depending on the number of detected crash frames of a video clip. Models 3–6 utilized Resnet-50 + CBAM/SE to extract appearance features, while they employed LSTM/Conv-LSTM models to extract motion features from videos. Table 2 shows the performance of the six candidate models.

The results indicate that model 1 had the lowest detection accuracy and speed compared to other models. In general, this model had a good performance in detecting multivehicle crashes. However, it largely failed to detect

Table 2: Crash detection models' performances on testing set.

No.	Model	True positive (TP)	False negative (FN)	False positive (FP)	True negative (TN)	Accuracy (%)	FPS
1	Faster R-CNN + SORT + rules [24]	58	32	18	72	72.22	0.73
2	ResNet-50 + CBAM + rules	69	21	12	78	81.67	50
3	ResNet-50 + CBAM + LSTM	70	20	12	78	82.22	27
4	ResNet-50 + SE + Conv-LSTM	74	16	12	78	84.44	35
5	ResNet-50 + CBAM + Conv-LSTM	78	12	11	79	87.22	33
6	ResNet-50 + CBAM + Bi-Conv-LSTM	79	11	11	79	87.78	30

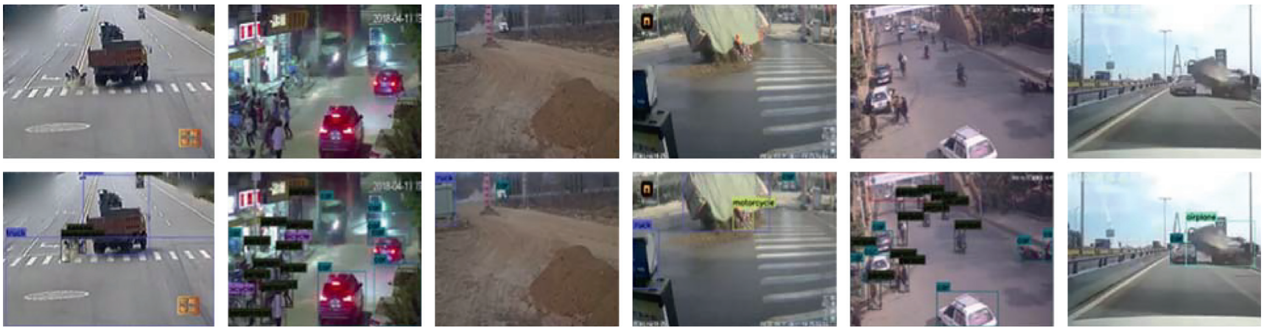

Figure 10: False-negative examples of Model 1.

Table 3: Model 6 versus the C3D model in detection accuracy and speed.

Model	Train accuracy (%)	Test accuracy (%)	Parameters (MB)	FLOPs/per video (B)	FPS
C3D [30]	99.89	67.22	249.99	574.36	14
Model 6	96.58	87.78	24.22	265.26	30

vehicle-pedestrian crashes and single-vehicle crashes. The reason could be that such models can only recognize crash motions (e.g., the intersection of vehicle trajectories, abnormal behaviours of nonmotorists [35, 36]) instead of crash appearance (e.g., fallen people, vehicle rollover, vehicle damage, etc.). Figure 10 shows some crash scenes that were falsely detected by model 1.

Compared to model 1, feature fusion-based models had better performance in detection accuracy. Among feature fusion-based models, the rule-based model (model 2) and the LSTM models (i.e., models 3 and 6) had lower detection accuracy than the Conv-LSTM models (models 4 and 5). The basic idea of the rule-based model is to determine a crash based on the number of detected crash frames of a video clip. Based on experiments, a highest accuracy can be achieved when the threshold is set to 10 (i.e., 10 frames). Since such method requires no sequential information, it may not well identify crash motion features (the FN rate is high). LSTM models require the flattened layout of appearance feature maps, which could lose spatial information. Conv-LSTM can simultaneously detect both motion features and appearance features, while retaining their original information considerably (FN decreases compared to rule-based models).

Regarding detection speed, the proposed model framework considerably outperformed motion-based deep learning models. In order to get a high detection accuracy of motion objects, motion-based models often require powerful deep learning models for vehicle detection and tracking. In general, Conv-LSTM achieved the highest detection accuracy with acceptable detection speed (FPS > 30).

Furthermore, a typical feature fusion-based model (i.e., C3D model) was also compared to our best model (i.e., model 6). As shown in Table 3, an overfitting issue occurred for the C3D model, with a training accuracy of 99.89% and a test accuracy of 67.22%. The reason is that the C3D model has much more parameters (over 10 times) than our proposed model. Since the dataset is limited, the model was easily overfitted. Regarding computational loads and detection speed, the proposed model was also better than the C3D model in terms of FLOP (floating point operations) and FPS.

Of note, the best Conv-LSTM (i.e., model 6) models still have some false-positive cases. Some noncrash scenes (congestions) cannot be well identified by the model, as shown in Figure 11. This is probably due to limited sample size. Another reason could be that the proposed model tends to focus on part of images (thanks to attention module), while ignoring the understanding of the whole traffic scene.

As for misdetection (i.e., FN), some typical cases were discussed here (Figure 12). The first crash is that two vehicles

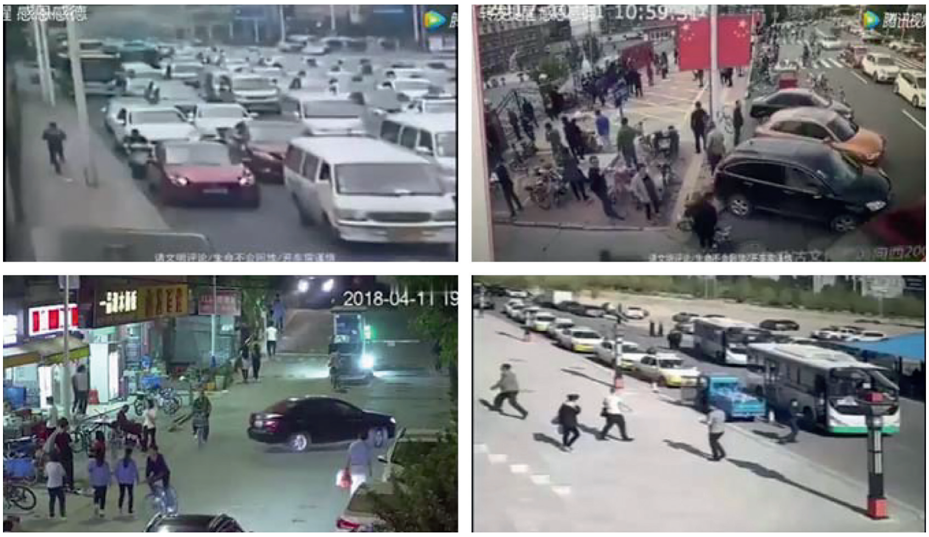

Figure 11: False-positive examples of model 6 (the best model).

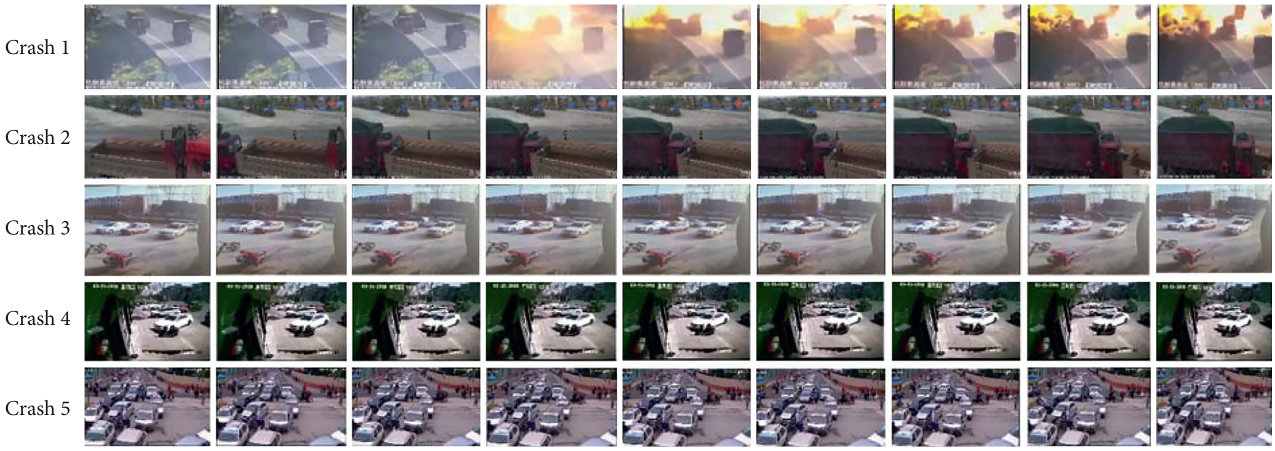

Figure 12: False-negative examples of model 6 (the best model).

collided with each other and led to an explosion. When it happened, the fire quickly covered the whole traffic scene. Such case is very rare in our current dataset, so that the trained model cannot well identify appearance features. The second to fifth crashes all happened in congested or complex traffic environment. In such environment, crash features were blocked or were difficult to be identified, especially when the original image quality is not high.

5. Conclusion

Detecting crash in a timely and accurate manner is important for traffic incident management. Previous video-based crash detection models suffer from low detection accuracy (e.g., some motion-based models) or high computational costs (e.g., large feature fusion-based models). To fill the gap, in this paper, we proposed a new feature fusion-based deep learning model framework with the purpose of achieving a balance between accuracy and speed for urban traffic crash detection. To this end, ResNet with attention modules was developed to capture the appearance features of crash images. ResNet has faster speed than conventional convolution neural network, while the attention module enables ResNet to focus on localized appearance features other than irrelevant information to further boost the model's speed. Conv-LSTM was introduced to link to ResNet to simultaneously capture appearance and motion features. Compared to conventional recurrent neural network (e.g., LSTM), Conv-LSTM can retain most of the spatial information with relatively fewer parameters.

Based on modeling results, the ResNet with attention modules can improve the performance of detecting localized appearance feature of crashes. Compared to simple rules and LSTM, the Conv-LSTM can better capture the motion features of crashes. The proposed model achieved the overall accuracy of 87.78% with relatively fast detection speed (FPS > 30), which outperformed conventional motion-based models and existing feature fusion-based models. Thus, the proposed method is a promising crash detection method, achieving a good balance between speed and accuracy.

Admittedly, the proposed model also has some limitations. First, the model falsely detected some congested traffic scenes as crashes. An understanding of the whole traffic scene may need to be considered. In the future research, we will attempt to improve the model framework. Second, it still has some misdetections when traffic environment/crash scenes are complicated, rare, and ambiguous. Thus, the model still needs more data and other supplementary methods (e.g., multiangle cameras or few-shot learning) to further improve its robustness. Third, the model needs to be further improved to identify different types/severity levels of crashes. The authors recommended that future research should be focused on those topics.

Data Availability

The data used were acquired from the local traffic police in China. Since those data are from video surveillance (many are crash-related), they can only be accessed with permission from the local government.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (grant no. 71971061), the National Key R&D Program of China (grant no. 2018YFE0102700), and Transportation Science Research Project in Jiangsu Province (grant no. 2019Z02).

References

[1] F. Zhang, "On traffic flow prediction and incident detection of freeway in cities based on kalman filtering," Ph.D. thesis, Beijing Jiaotong University, Beijing, China, 2012.

[2] Md.S. Amin, "Accident detection and reporting system using GPS, GPRS and GSM technology," in *Proceedings of the 2012 International Conference on Informatics, Electronics & Vision (ICIEV)*, Dhaka, Bangladesh, May 2012.

[3] A. Kandari, A. Abdullah Mohammad, and I. F. Alshaikhli, "Accident Detection System and Method for Accident Detection," *US Patent 8903636B1*, 2013.

[4] D. Cogswell, "Accident alarm system using GSM, GPS and accelerometer," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 4, pp. 3506–3511, 2015.

[5] C. Wang, C. Xu, J. Xia, and Z. Qian, "A combined use of microscopic traffic simulation and extreme value methods for traffic safety evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 281–291, 2018.

[6] R. Fahmida, X. Zhang, and C. Mei, "Evaluate probe speed data quality for congestion performance measures," in *Proceedings of the Transportation Research Board (TRB) 99th Annual Meeting*, Washington, DC, USA, January 2019.

[7] M. Li, Z. Li, C. Xu, and T. Liu, "Short-term prediction of safety and operation impacts of lane changes in oscillations with empirical vehicle trajectories," *Accident Analysis & Prevention*, vol. 135, Article ID 105345, 2020.

[8] Xu Zhang, "Identifying secondary crashes using text mining techniques," *Journal of Transportation Safety & Security*, vol. 12, no. 10, pp. 1–21, 2019.

[9] X. Gu, M. Abdel-Aty, Q. Xiang, Q. Cai, and J. Yuan, "Utilizing UAV video data for in-depth analysis of drivers' crash risk at interchange merging areas," *Accident Analysis & Prevention*, vol. 123, pp. 159–169, 2019.

[10] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108–118, 2000.

[11] P. Bai and J. Li, "A video-based method of traffic accident detection," *Journal of University of Jinan (Science and Technology*, vol. 3, no. 26, pp. 282–286, 2012.

[12] C. Lu, C. C. Xu, and Y. Dai, "A crash prediction method based on bivariate extreme value theory and video-based vehicle trajectory data," *Accident Analysis & Prevention*, vol. 123, pp. 365–373, 2018.

[13] Y.-K. Yong-Kul and D.-Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 188–194, 2007.

[14] Y. K. Ki, "Accident detection system using image processing and MDR," *International Journal of Computer Science and Network Security(IJCSNS)*, vol. 7, no. 3, pp. 35–39, 2007.

[15] S. S. Thomas, S. Gupta, and V. K. Subramanian, "Event detection on roads using perceptual video summarization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2944–2954, 2018.

[16] B. Maaloul, "Adaptive video-based algorithm for accident detection on highways," in *Proceedings of the 2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)*, Toulouse, France, June 2017.

[17] S. Sadeky, "Real-time automatic traffic accident recognition using HFG," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, August 2010.

[18] Y. Chen, "A vision based traffic accident detection method using extreme learning machine," in *Proceedings of the 2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, Macau, China, August 2016.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[20] J. He, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[21] J. Redmon and F. Ali, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, July 2017.

[22] J. Redmon and F. Ali, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[23] E. M. A. Vicente and L. R. Elian, "Fast car crash detection in video," in *Proceedings of the 2018 XLIV Latin American Computer Conference (CLEI)*, Sao Paulo, Brazil, January 2018.

[24] K.B. Lee and H.S. Shin, "An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels," in *Proceedings of the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, Istanbul, Turkey, August 2019.

[25] E. Paul, "Computer vision-based accident detection in traffic surveillance," in *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, July 2019.

[26] D. Singh and C. K. Mohan, "Deep spatio-temporal representation for detection of road accidents using stacked autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 879–887, 2019.

[27] Y. Yao, "Unsupervised traffic accident detection in first-person videos," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, November 2019.

[28] E. Batanina, "Domain adaptation for car accident detection in videos," in *Proceedings of the 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Istanbul, Turkey, November 2019.

[29] X. Huang, P. He, A. Rangarajan, and S. Ranka, "Intelligent intersection," *ACM Transactions on Spatial Algorithms and Systems*, vol. 6, no. 2, pp. 1–28, 2020.

[30] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proceedings of the Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2016.

[31] J. Hu, "Squeeze-and-excitation networks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[32] S. Gupta, "CBAM: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[33] X. Shi, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1, Granada, Spain, June 2015.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020.

[35] Y. Guo, Z. Li, Y. Wu, and C. Xu, "Exploring unobserved heterogeneity in bicyclists' red-light running behaviors at different crossing," *Accident Analysis & Prevention*, vol. 115, pp. 118–127, 2018.

[36] Y. Guo, Z. Li, Y. Wu, and C. Xu, "Evaluating factors affecting electric bike users' registration of license plate in China using Bayesian approach," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 59, pp. 212–221, 2018.