

## Research Article

# A Vision-Based Video Crash Detection Framework for Mixed Traffic Flow Environment Considering Low-Visibility Condition

Chen Wang <sup>1</sup>, Yulu Dai <sup>1</sup>, Wei Zhou <sup>1</sup> and Yifei Geng<sup>2</sup>

<sup>1</sup>Intelligent Transportation Research Center, Southeast University, Nanjing 211189, China

<sup>2</sup>School of Automation, Southeast University, Nanjing 211189, China

Correspondence should be addressed to Chen Wang; [wkobec@hotmail.com](mailto:wkobec@hotmail.com)

Received 13 September 2019; Accepted 9 October 2019; Published 17 January 2020

Guest Editor: Feng Chen

Copyright © 2020 Chen Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a vision-based crash detection framework was proposed to quickly detect various crash types in mixed traffic flow environment, considering low-visibility conditions. First, Retinex image enhancement algorithm was introduced to improve the quality of images, collected under low-visibility conditions (e.g., heavy rainy days, foggy days and dark night with poor lights). Then, a Yolo v3 model was trained to detect multiple objects from images, including fallen pedestrians/cyclists, vehicle rollover, moving/stopped vehicles, moving/stopped cyclists/pedestrians, and so on. Then, a set of features were developed from the Yolo outputs, based on which a decision model was trained for crash detection. An experiment was conducted to validate the model framework. The results showed that the proposed framework achieved a high detection rate of 92.5%, with relatively low false alarm rate of 7.5%. There are some useful findings: (1) the proposed model outperformed empirical rule-based detection models; (2) image enhancement method can largely improve crash detection performance under low-visibility conditions; (3) the accuracy of object detection (e.g., bounding box prediction) can impact crash detection performance, especially for minor motor-vehicle crashes. Overall, the proposed framework can be considered as a promising tool for quick crash detection in mixed traffic flow environment under various visibility conditions. Some limitations are also discussed in the paper.

## 1. Introduction

Emergency response to roadway crashes is very important for traffic management. On the one hand, people injured in a crash need to be sent to the nearest hospital in the first place to prevent their health condition from being worsened, on the other hand, serious crashes often cause nonrecurrent congestions, if emergency response or clearance is not carried out in time. In order to mitigate those negative impacts, roadway crashes need to be quickly detected.

Crash detection can be conducted by analyzing traffic flow data from roadway detectors, such as loops and microwaves. However, such method is often inaccurate due to systematic errors caused by both algorithms and data quality [1–5]. Thus, in practice, crashes were often detected by human observers through CCTV in Traffic Management Centers (TMC). The advantage of CCTV is that it can directly capture crash scenes within its range. With the development of intelligent transportation system (ITS), more and more CCTVs have been

implemented in big cities and highways. Although human observations through CCTV can be reliable, it is sometime too labor-intensive and time-consuming. Thus, it is very meaningful to develop other reliable automatic crash detection methods based on CCTV [6, 7].

In recent years, computer vision technologies have undergone a fast development and largely utilized in transportation field [8, 9], thanks to the increasing power of computers and deep learning methods. The performance of vision-based object detection, based on deep learning methods, has been significantly improved. Thus, researchers have been focusing on developing crash detection models based on complex deep learning frameworks [10, 11]. Their results also showed the capability of computer vision in crash detection. However, sometimes a complex deep learning framework require high computational costs and difficult to be implemented in practice.

To note, previous literature mainly focused on detecting crashes in motorized traffic environment in developed

countries. In developing countries, a larger number of pedestrians and cyclists could share roadways with automobiles. Thus, crash detection in mixed traffic flow environment could be an even more important task for those countries. Moreover, in order to be used in practice, a vision-based crash detection model needs to be robust to various conditions, especially low-visibility ones such as heavy rain, fog, poor lights, and so on. Sometimes, even deep-learning based vision algorithms did not perform well in those low-visibility conditions [12–14], due to relatively low image quality. Thus, some additional efforts were often added to improve detection performance, such as image enhancement methods [15–19].

Considering these, a vision-based crash detection model framework was developed for mixed traffic flow environment in this study. Regarding low-visibility conditions, an image enhancement method was also introduced to improve image quality so that a deep learning algorithm can better identify moving objects. Regarding quick crash detection, a Yolo v3 model was employed to extract features from images, based on which a decision tree model was trained for detecting various crash types that could occur in mixed traffic flow environment. The paper is organized as follows: the second section discuss previous literature related to vision-based crash detection and image enhancement. Section 3 introduces the Retinex algorithm, Yolo v3, and decision tree-based framework of crash detection. Section 4 discusses the results of an experiment. Section 5 concludes the findings of this research.

## 2. Literature Review

In the past twenty years, researchers have conducted many studies on vision-based traffic crash detection, which can be classified into three categories: (1) modeling of traffic flow patterns; (2) modeling of vehicle interactions; and (3) analysis of vehicle activities [10].

The first method is to compare vehicle trajectories to typical vehicle motion patterns that can be learned from large data samples. In this framework, if a trajectory is not consistent with typical trajectory patterns, it can be considered as a traffic incident [20–22]. However, it is not easy to identify whether this incident is a crash due to limited crash trajectory data that can be collected in the real world. The second method determines crash occurrence based on speed change information, which applies social force model and intelligent driver model to model interactions among vehicles. This method requires a larger number of training samples. The third method largely depends on trackers because it needs to continuously calculate vehicle motion features (e.g., distance, acceleration, direction etc.) [23–27]. As such, aberrant behaviors [28, 29] related to traffic incidents could be detected. However, it is often difficult to be utilized in practice, limited by high computational costs and unsatisfactory tracking performance in congested traffic environment [30]. In general, fruitful results have been achieved for vision-based crash detection. However, most literature focused on motor vehicle crash instead of crashes involving nonmotorized modes, such as bicycle-related and pedestrian-related crashes [7, 23, 31]. Moreover, many models are compute-intensive, by constructing complicated deep learning structures.

Another practical issue for crash detection method is the ability to deal with low-visibility conditions (e.g., fog, heavy rain, dark night). Image enhancement methods were usually utilized to improve the robustness of video detection to low-visibility conditions. Image enhance methods can adjust digital images so that key features are more easily to be identified [32, 33]. Such technology was also used to provide better image quality to improve the performance of crash detection [34, 35]. There are two major types of image enhancement methods: physical model and tensile transformation. The first method usually develops a physical model considering fog formation. Sometimes, it is difficult to guarantee enough accuracy under various conditions. The second method normally uses histogram equalization [36], wavelet transform [37], homomorphic filtering [38] to enhance low-quality images (e.g., those with raindrops and fogs). The robustness of such a method could be limited in some conditions, for it requires large number of parameters and thresholds to be tuned.

## 3. Methods

In this study, a crash detection framework was proposed for mixed traffic flow environment. The framework has three major components. First, Retinex image enhancement algorithm was introduced to enhance image quality. Second, Yolo v3 was utilized to detect moving objects, such as vehicles, pedestrians, and bicyclists/motorcyclists. Third, a decision tree-based framework was proposed to determine various crash scenarios bin mixed traffic flow environment.

*3.1. Retinex Image Enhancement Algorithms.* Retinex is an image enhancement algorithm proposed by Edwin H. Land. The basic theory is that the color of an object is determined by the ability of the object to reflect light from long waves (red), medium waves (green), and short waves (blue), rather than the absolute value of the intensity of the reflected light. The color of an object is not affected by illumination nonuniformity, but possesses consistency. Unlike traditional linear and nonlinear algorithms that only enhance a certain type of image, Retinex algorithm can balance dynamic range compression, edge enhancement and color constancy. Thus, it can be used for the adaptive enhancements of various image types, which is a feasible choice in this research.

Figure 1 shows the theory of Retinex that a given image  $S(x, y)$  can be decomposed into two different images: a reflected image  $R(x, y)$  and a luminance image (also called as incident image)  $L(x, y)$ .

The image can be formulated as:

$$S(x, y) = R(x, y) \cdot L(x, y). \quad (1)$$

Convert it into logarithmic domain:

$$r(x, y) = \log R(x, y) = \log \frac{S(x, y)}{L(x, y)}. \quad (2)$$

And it can be written as:

$$r(x, y) = \log S(x, y) - \log [F(x, y) * S(x, y)]. \quad (3)$$

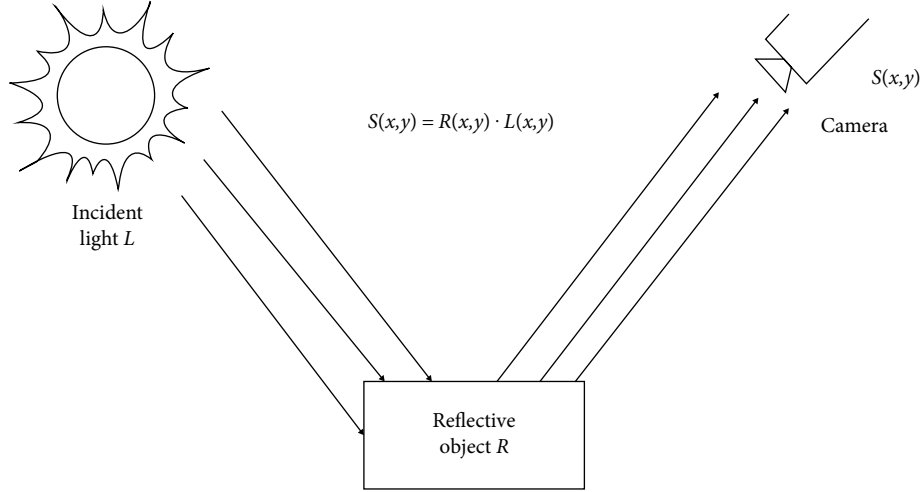


FIGURE 1: Schematic diagram of Retinex principle.

where  $r(x, y)$  is the output image,  $*$  is convolution operator, and  $F(x, y)$  is surround function. The surround function,  $F(x, y)$  is given as:

$$F(x, y) = \lambda e^{-(x^2+y^2)/c^2}, \quad (4)$$

where  $c$  is the scale that control the extent of the surround. Mathematically, solving  $R(x, y)$  is a singular problem that can only be calculated by mathematically approximated estimates. The steps of Retinex are as follows:

- Step 1. Read in the initial image  $S(x, y)$ , and separate  $R$ ,  $G$ , and  $B$  channels of the image;
- Step 2. Convert the pixel values of each channel from integers to floats and convert them to the logarithmic domain;
- Step 3. Input the scale  $c$ , and calculate the value of  $\lambda$  which is equal to  $1/(\iint F(x, y) dx dy)$ ;
- Step 4. Calculate the value of  $r(x, y)$  of each channel;
- Step 5. Convert  $r(x, y)$  from logarithmic domain to real domain;
- Step 6. Stretch  $R(x, y)$  linearly and output in the corresponding format.

3.2. *Yolo v3*. YOU ONLY LOOK ONCE (YOLO) is a state-of-the-art, real-time object detection system. The core idea of Yolo v3 is to use the picture as a network input, which is to return to the position of the bounding box and its subordinate categories (e.g., vehicles, trees, or pedestrians etc.) directly in the output layer. The overall stages of Yolo v3 which is consisted of four periods are illustrated below.

3.2.1. *Bounding Box Prediction*. Sum of squared error loss is used to predict the coordinate value, so the error can be calculated rapidly. Yolo v3 predicts the score of an object for each bounding box by logistic regression. Each bounding box needs four values to represent its position of the images:  $(x_p, y_p, w_p, h_p)$ , which respect separately: (the  $x$  coordinate of center point, the  $y$  coordinate of center point, weight of bounding box, height of bounding box).

$$\begin{aligned} b_x &= \sigma(t_x) + c_x; \\ b_y &= \sigma(t_y) + c_y; \\ b_w &= p_w e^{t_w}; \\ b_h &= p_h e^{t_h}, \end{aligned} \quad (5)$$

where  $c_x, c_y$  are the coordinate offsets of the grid,  $p_w$  and  $p_h$  are the side lengths of the preset anchor box, the resulting frame coordinates are  $b_x, b_y, b_w, b_h$  and the network learning goals are  $t_x, t_y, t_w, t_h$ .

If the bounding box prior overlaps a ground truth object by more than any other bounding box prior, then the value is 1. If the overlap does not reach a threshold (setting 0.5), the prediction of bounding box will be ignored, and it is displayed as no loss.

3.2.2. *Class Prediction*. To classify different kinds of objections, independent logistic classifiers are used instead of a SoftMax. When training, binary cross-entropy loss is used for the class predictions.

After learning by Logistic regression classifier, there are a set of weights:  $w_0, w_1, \dots, w_n$ , and the  $n$  features of each sample can be written as  $x_1, x_2, \dots, x_n$ , when the data of test samples are input, which can be combined with the weights linearly:

$$x = w_0 + w_1 x_1 + \dots + w_n x_n. \quad (6)$$

The sigmoid function is:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (7)$$

The prediction probability in sigmoid function can be expressed as:

$$P(y = 1|x) = \pi(x) = \frac{1}{1 + e^{-g(x)}}. \quad (8)$$

$$P(y = 0|x) = 1 - P(y = 1|x), \quad (9)$$

where  $g(x)$  is  $x$ .

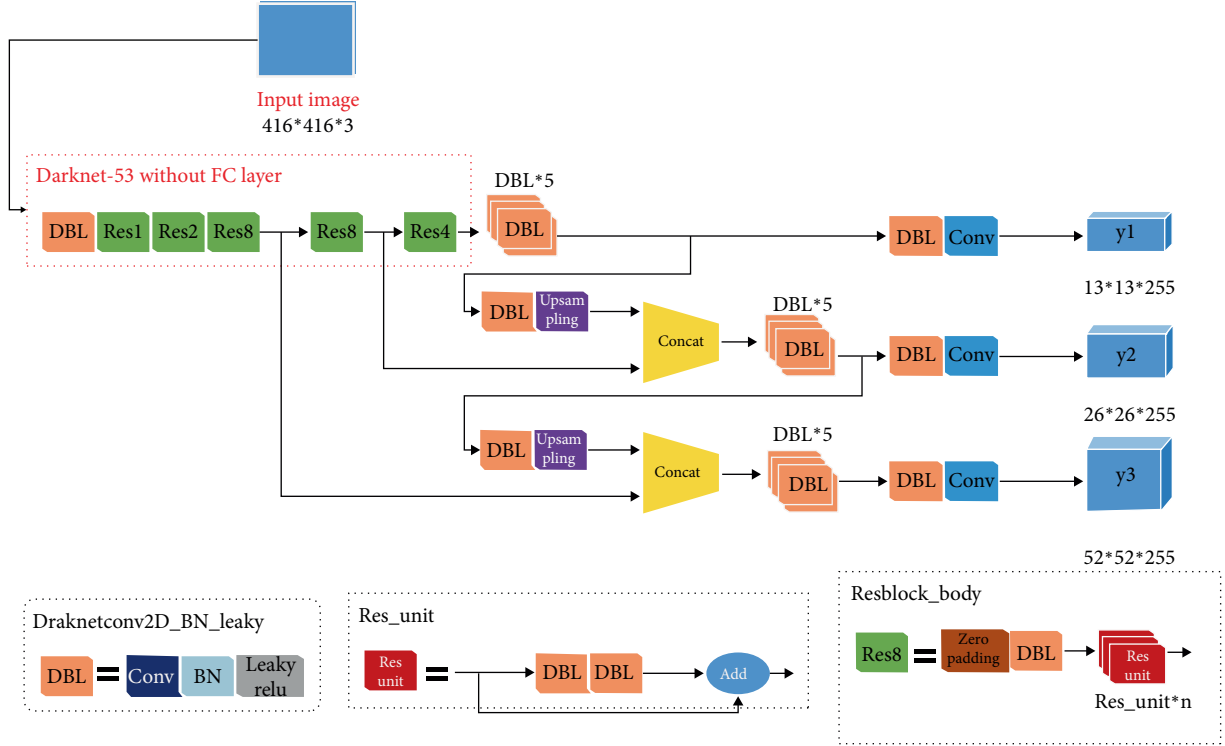


FIGURE 2: Yolo v3 flow chart.

3.2.3. *Predictions Across Scales.* Yolo v3 predicts different boxes at three different scales. Yolo v3 uses FPN (feature pyramid network) to extract feature from scales, and finally predicts a 3-D tensor, containing the bounding box information, object information, and class information.

3.2.4. *Feature Extractor.* Yolo v3 uses a complex network for performing feature extraction, which has 53 convolutional layers, called Darknet-53. This new network is much more powerful than Darknet-19 but still more efficient than ResNet-101 or ResNet-152. The loss function of YOLO is:

$$\begin{aligned}
 \text{Loss function} = & \lambda_{coord} \sum_{i=0}^s \sum_{j=0}^B l_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\
 & + \lambda_{coord} \sum_{i=0}^s \sum_{j=0}^B l_{ij}^{obj} \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \\
 & + \sum_{i=0}^s \sum_{j=0}^B l_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^s \sum_{j=0}^B l_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^s \sum_{c \in \text{classes}} l_{ij}^{obj} (p_i(c) - \hat{p}_i(c))^2.
 \end{aligned} \quad (10)$$

The flow chart of YOLO is shown in Figure 2.

3.3. *Decision-Tree Based Crash Detection Framework.* In mixed traffic flow environment, crashes could occur between motorists and nonmotorists. Thus, a motion-based method (e.g., modeling of vehicle interactions, analysis of vehicle activities, etc.) may not have full capability to detect such

type of crashes. Moreover, since the computational cost of object detection and tracking has already been high, an even more complex framework by integrating other deep learning models (e.g., recursive neural network) would become too compute-intensive. Thus, in this paper, we consider a simplified framework for quick crash detection that can be implemented in practice.

Decision tree model was considered for crash classification, based on features obtained from Yolo v3. It has several advantages: (1) the cost of using the tree (i.e., predicting data) is logarithmic; (2) it requires little data preparation and can handle both numerical and categorical data; (3) it is simple to understand and to interpret.

Given training features  $X_i$  and label  $y$ , a decision tree recursively partitions the space:

$$\begin{aligned}
 Q_{left}(\theta) &= (x, y) | x_j \leq t_m, \\
 Q_{right}(\theta) &= \frac{Q}{Q_{left}(\theta)},
 \end{aligned} \quad (11)$$

where  $Q$  representing the data at node  $m$ , is a candidate split consisting of a feature  $j$  and threshold  $t_m$ ,  $Q_{left}$  and  $Q_{right}$  are subsets partitioned by the decision tree at node  $m$ .

The impurity at  $m$  can be calculated by an impurity function  $H()$ , the choice of which is based on the task being considered:

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)). \quad (12)$$

If it is a classification task with outcomes from 0 to  $k$  for node  $m$ , representing a region  $R_m$  with  $N_m$  observations, let

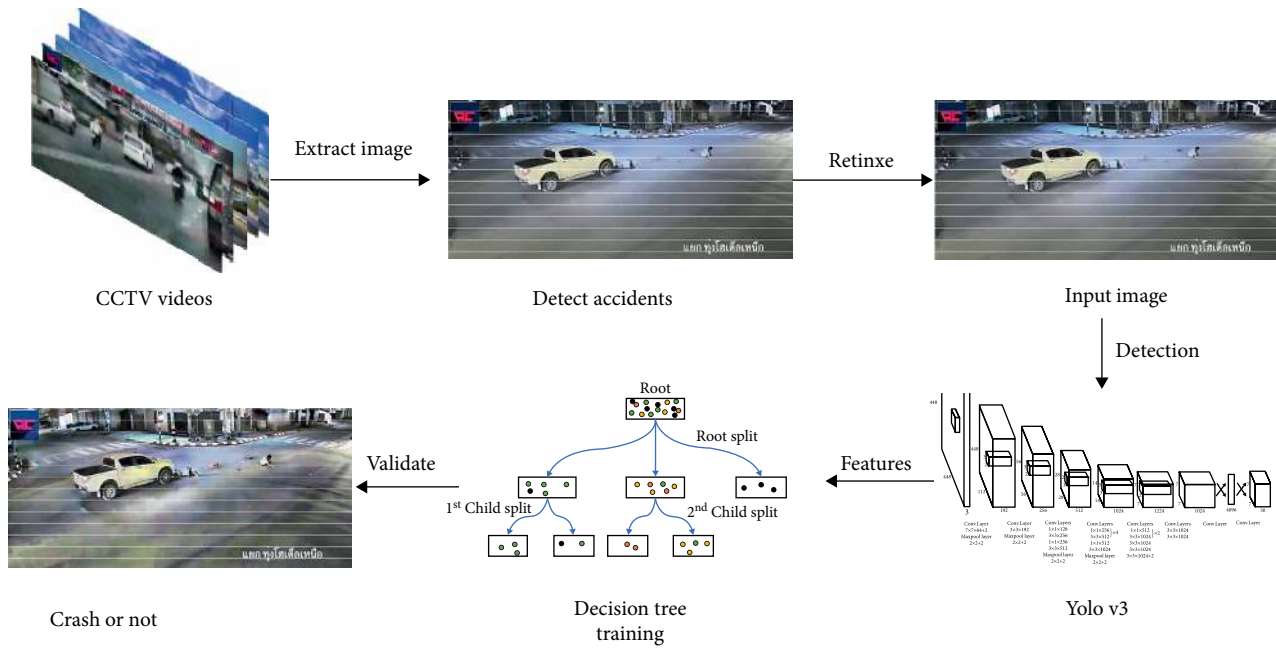


FIGURE 3: Crash detection framework for mixed traffic flow environment.

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k), \quad (13)$$

be the proportion of class  $k$  observations in node  $m$ .

Gini Index is often used to measure impurity:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}). \quad (14)$$

Entropy is another commonly used indicator of impurity:

$$H(X_m) = - \sum_k p_{mk} \log(1 - p_{mk}), \quad (15)$$

where  $X_m$  is the training data in node  $m$ .

Parameters are selected such that the impurity can be minimized:

$$\theta^* = \arg \min_{\theta} G(Q, \theta). \quad (16)$$

The framework is shown in Figure 3.

## 4. Experimental Evaluation

In order to validate the framework, an experiment was conducted on a computer with specification Intel(R) Core (TM) i5-4200 CPU @ 2.50 GHz (4 CPUs), ~2.5 GHz, 8 GB RAM with NVIDIA Corporation GeForce 840 M.

**4.1. Dataset Used.** We collected large number of CCTV videos from online since there is no public database for crash detection. Figure 4 shows the samples of the video data. In general, a video clip records 10–20 s before and after a crash. Our dataset has 127362 frames, in which 45214 contain crash scenes and 82148 are normal frames. There are various crash

types observed, including multi-vehicle crashes, pedestrian-vehicle crashes, and cyclist-vehicle crashes. Moreover, many low-visibility conditions were included in the dataset, such as dark night with poor lights, heavy rains, and foggy days. In this study, 15000 crash frames and 40000 normal frames were used to create training samples, while the remaining frames were used for model testing.

**4.2. Results and Discussion.** First, Retinex was utilized to improved image quality. Figure 5 provides some examples of image enhancement. It can be found that more image details can be seen after the enhancement.

After image enhancement, Yolo v3 was used for detecting objects from images. In the training dataset, crash samples were extracted from videos including fallen people, fallen bicycle/motorcycle, and vehicle rollovers. Those samples were then distorted and scaled to further enlarge the crash sample size. Normal people, bicycle, motorcycle, and vehicles were also collected as normal samples. Figure 6 provides some examples in the training dataset.

After 5000 iterations, the model became convergent. Figure 7 provides the real-time detection performance based on Yolo v3. The training and testing accuracy of the Yolo model are shown in Figure 8. According to the graph, the training model has no overfitting issue.

Three crash types were observed in the current video dataset, including:

- (1) Pedestrian/cyclist related crash: If this type occurs, fallen people, fallen cyclists, stopped vehicle, stopped people, and stopped cyclists could be detected in the scene.
- (2) Minor motor-vehicle crash: If this type occurs, vehicle overlapping, stopped vehicles, and stopped people/cyclists could be detected in the scene.





FIGURE 4: Sample frames from the video dataset used to evaluate the accident. The dataset contains videos of accidents during the various environmental conditions such as (a) daytime (b) night (c) rain or fog, as well as from different cameras and view angles.



FIGURE 5: Images before and after Retinex.

- (3) Serious motor-vehicle crash: If this type occurs, vehicle rollover, stopped vehicles, and stopped people/cyclists could be detected in the scene.

In order to detect those three crash types, a set of features were developed based on Yolo v3 outputs, including: number of moving vehicle (the number of moving vehicles), number of stopped vehicle (the number of stopped vehicles), number of stopped people (the number of moving pedestrians and cyclists), number of moving people (the number of stopped pedestrians and cyclists), fallen people (the number of fallen people), vehicle rollover (the number of vehicle rollover), intersection of union (IOU), and IOU duration. IOU is often used to measure the overlap between two bounding boxes (e.g., two vehicles). Note that in this study, IOU represents the maximum IOU values that remain unchanged over the observation period, while IOU duration indicates the longest time period that IOU remains changed. A decision tree was trained using these features as inputs, as shown in Figure 9. The average precision is 0.95. According to entropy and information

gain of the tree model, some features were found as important including: fallen people, IOU duration, stopped people, stopped vehicle, and vehicle rollover.

Based on the findings, three empirical rule-based models were also developed as follows:

- Rule 1:* If fallen people or fallen nonmotorized vehicle is continuously detected during a period (e.g., 10 s), the condition can be determined as a crash.
- Rule 2:* If two vehicles are detected as overlapped during a period (e.g., 10 s), and other stopped people are detected around the vehicles, the condition can be determined as a crash.
- Rule 3:* If a car rollover is detected during a period (e.g., 2 s), the condition can be determined as a crash.

Rule 1 model could detect crash types related to pedestrians and cyclists (e.g., bicycles, motorcycles). A relative long period time detection may avoid miss-detection of those occasionally fallen off. Rule 2 model was designed for nonserious



FIGURE 6: Training samples of Yolo v3. (a) Crash samples and (b) normal samples.



FIGURE 7: Yolo v3 results.

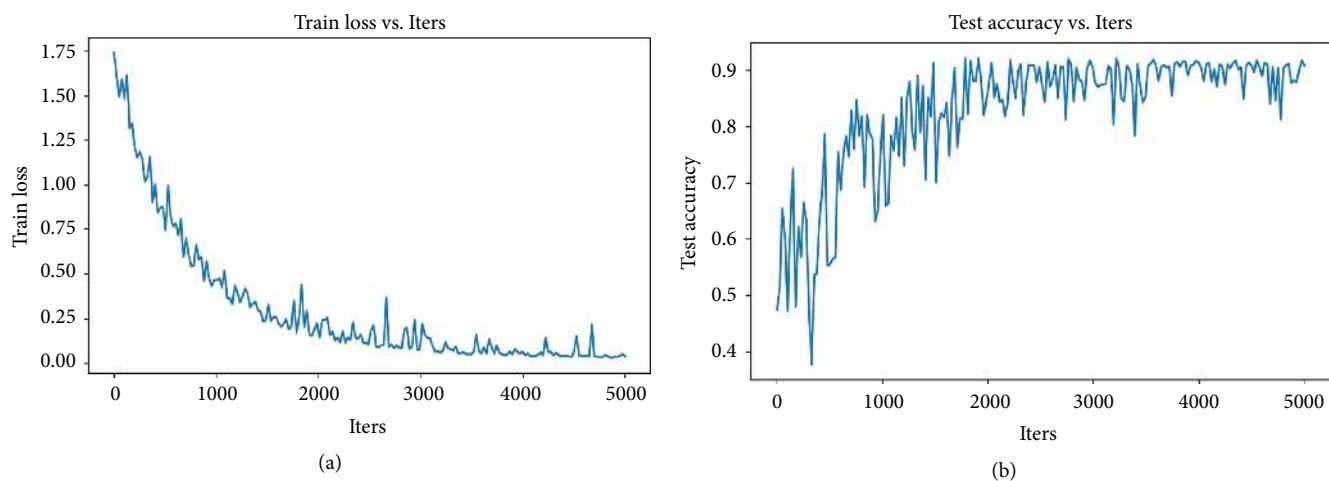


FIGURE 8: (a) Train loss within 5000 iterations; (b) test accuracy with 5000 iterations.

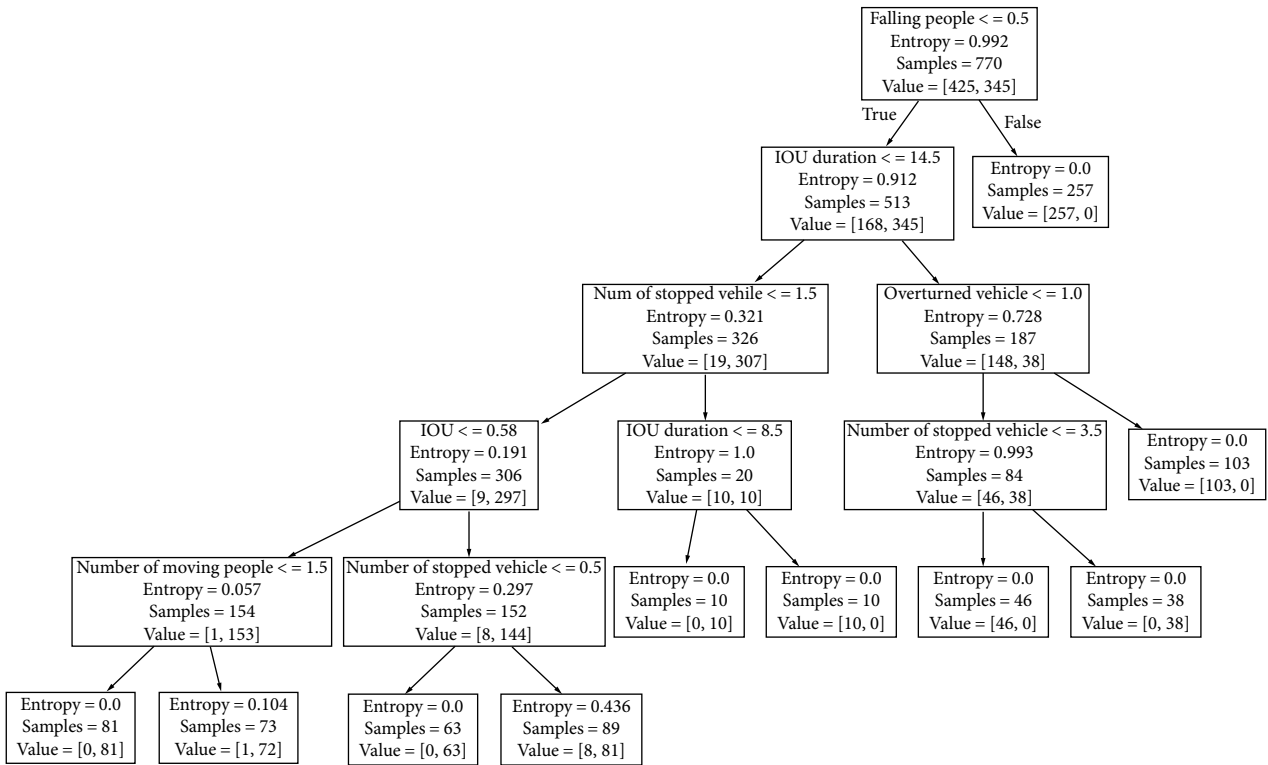


FIGURE 9: Trained decision tree model.

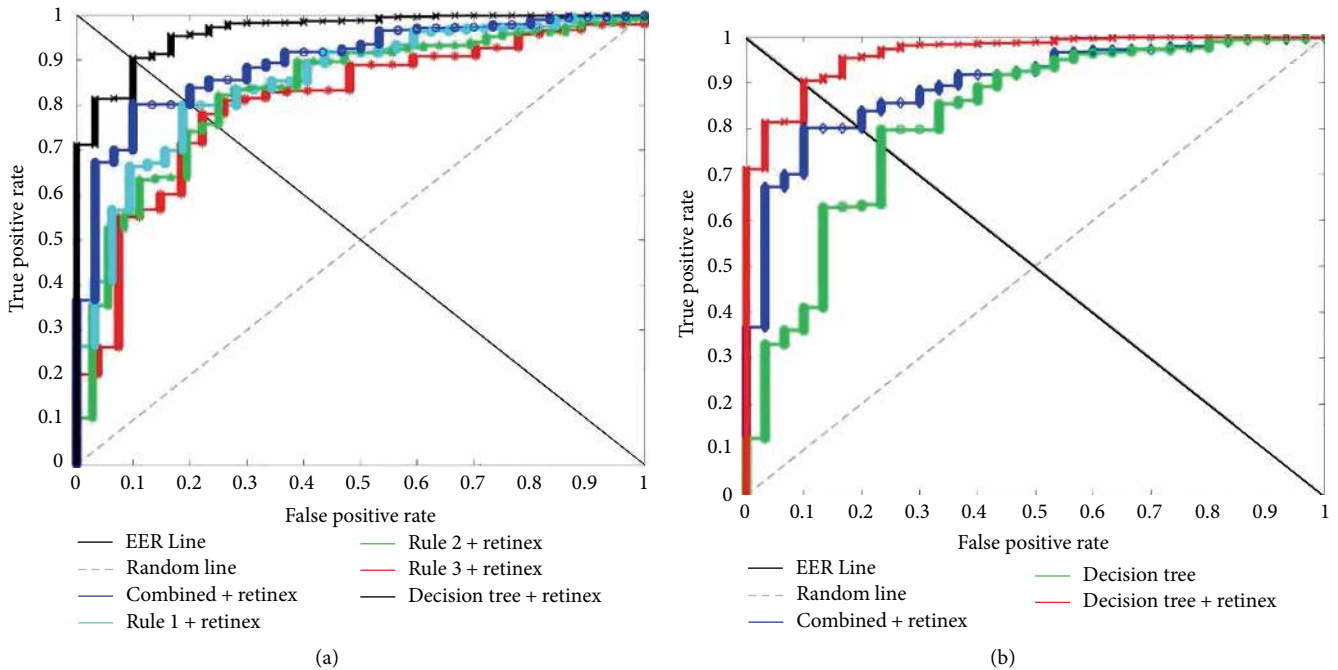


FIGURE 10: AUC comparison among multiple crash detection models. (a) AUC comparison among decision tree, combined-rule and single-rule detection models. (b) AUC comparison between crash detection models with/without Retinex enhancement.

crash types, including minor multivehicle and single-vehicle crashes. In those situations, vehicles may not be damaged seriously or no fallen objects could be detected. According to previous literature, such types could be detected by analyzing

the intersection of vehicle motions. Rule 3 model can detect serious motor-vehicle crash types.

All model performances were compared, as shown in Figure 10. Figure 10(a) provides the ROC curves of all those



TABLE 1: Area under curve (AUC) for multiple crash detection models.

Models	Retinex + decision tree	Retinex + combined	Decision tree	Retinex + rule 1	Retinex + rule 2	Retinex + rule 3
AUC	0.9632	0.8744	0.8201	0.8519	0.8039	0.8295

models with Retinex enhancement in mixed traffic flow environment, which indicates the relationship between sensitivity (True Positive) and specificity (False Positive). It can be seen from Figure 10(a), the decision tree model had the best performance than other models, according to ROC curves. The combined rule model outperformed each single-ruled model. According to Figure 10(b), Retinex enhancement made a considerable improvement on crash detection performance for the decision tree model. Without Retinex, the overall performance of the decision tree model appeared to be lower than the combined rule-based model with Retinex.

Overall, the proposed framework can correctly detect 92.5 % of crashes in the testing dataset. The false alarm rate is 7.5 %. The AUC values for all crash detection models are listed in Table 1.

In general, decision tree-based model appeared to be better than empirical rule-based models. Although the proposed framework achieved relatively high detection accuracy, there are still some issues:

- (1) Fallen pedestrians/cyclists can sometimes be blocked by other objects, increasing false alarm rate.
- (2) In highly congested mixed traffic flow environment, crashes can also be falsely alarmed. This could be due to inaccurate detection of Yolo v3 model (e.g., bounding boxes prediction).
- (3) Retinex can handle most low-visibility conditions in this study. However, when video quality is too low, crashes can still be missed or false alarmed. For example, fast-moving vehicle could be sometimes falsely detected as fallen vehicles, according to our observation.

**4.3. Comparison with the Existing Methods.** Due to the lack of public database, limited research has been identified on this topic. Since most studies were based on private datasets that cannot be accessed, their results are somewhat incomparable. However, we still listed the results here. Yun [39] achieved a detection rate of 0.8950 for crash detection. RTADS [20] reported a hit rate of 92% and false alarm rate of 0.77%. ARRS [21] presented a true positive rate of 0.63 with a false alarm rate of 0.06. Singh [27] reported a hit rate of 77.5% and a false alarm rate of 22.5%. Sadek [40] achieved 99.6% detection rate and 5.2% false alarm rate.

Although some literature has reported high detection rates, such could encounter over-fitting issues due to limited sample size. Second, some literature created complicated deep learning structures, requiring high computational capability. Third, limited literature has focused on crash detection in mixed traffic flow environment under low-visibility conditions.

## 5. Conclusions

The paper proposed a vision-based crash detection framework for mixed traffic flow environment considering low-visibility conditions. Retinex algorithm was introduced to enhance image quality of low-visibility conditions, such as night, foggy, and rainy days. A deep learning model (i.e., Yolo v3) was trained to detect objects in mixed traffic flow environment and a decision tree model was developed for crash detection, considering various crash scenarios between motorized and nonmotorized traffic. The proposed method achieved a hit rate of 92.5% and a false alarm rate of 7.5%. Interesting findings include: (1) the proposed model outperformed empirical rule-based detection models; (2) image enhancement method can largely improve crash detection performance under low-visibility conditions; (3) the accuracy of object detection (e.g., bounding boxes prediction) can impact crash detection performance, especially for minor motor-vehicle crashes.

Overall, the results are encouraging and the framework is promising. Admittedly, there are still some issues that can be further addressed. First, different image enhancement methods could be tried to improve the overall performance. Second, other deep learning method can be used and compared to original Yolo v3 model. Third, other more complex deep learning structure can be examined and compared to the current framework, in terms of accuracy and computational speed.

## Data Availability

Data were large number of video clips.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research has been supported by the National Key R&D Program of China (2018YFE0102700).

## References

- [1] G. Yuan, X. Zhang, Q. Yao, and K. Wang, "Hierarchical and modular surveillance systems in ITS," *IEEE Intelligent Systems*, vol. 26, no. 5, pp. 10–15, 2011.
- [2] S. Xia, J. Xiong, Y. Liu, and G. Li, "Vision-based traffic accident detection using matrix approximation," in *2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World, ASCC 2015, IEEE*, pp. 1–5, Kota Kinabalu, Malaysia, 2015.

- [3] B. Li, Y. Liu, J. Ren, Y. Chen, L. Xin, and J. Shi, "Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment," *IET Intelligent Transport Systems*, vol. 10, no. 6, pp. 428–437, 2016.
- [4] C. Xu, Y. Wang, P. Liu, W. Wang, and J. Bao, "Quantitative risk assessment of freeway crash casualty using high-resolution traffic data," *Reliability Engineering and System Safety*, vol. 169, pp. 299–311, 2018.
- [5] 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.
- [6] X. Qu, J. Zhang, and S. Wang, "On the stochastic fundamental diagram for freeway traffic: model development, analytical properties, validation, and extensive applications," *Transportation Research Part B Methodology*, vol. 104, pp. 256–271, 2017.
- [7] M. Zhou, X. Qu, and X. Li, "A recurrent neural network based microscopic car following model to predict traffic oscillation," *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 245–264, 2017.
- [8] C. Wang, C. Xu, J. Xia, and Z. Qian, "Modeling faults among e-bike-related fatal crashes in China," *Traffic Injury Prevention*, vol. 18, no. 2, pp. 175–181, 2017.
- [9] B. Dong, X. Ma, F. Chen, and S. Chen, "Investigating the differences of single- and multi-vehicle accident probability using mixed logit model," *Journal of Advanced Transportation*, vol. 2018, Article ID 2702360, 9 pages, 2018.
- [10] C. S. Regazzoni, A. Cavallaro, Y. Wu, and J. Konrad, "Of video surveillance systems continues bound to take a approach to video," in *IEEE Signal Processing Magazine*, IEEE, pp. 16–17, 2010.
- [11] X. Zhu, Z. Dai, F. Chen, X. Pan, and M. Xu, "Using the visual intervention influence of pavement marking for rutting mitigation II: visual intervention timing based on the finite element simulation," *International Journal of Pavement Engineering*, vol. 20, no. 5, pp. 573–584, 2019.
- [12] G. Wu, F. Chen, X. Pan, D. M. Xu, and X. Y. Zhu, "Using the visual intervention influence of pavement markings for rutting mitigation-part I: preliminary experiments and field tests," *International Journal of Pavement Engineering*, vol. 20, no. 6, pp. 734–746, 2019.
- [13] A. Yoneyama, C. H. Yeh, and C. C. JayKuo, "Robust vehicle and traffic information extraction for highway surveillance," *Eurasip Journal on Applied Signal Processing*, vol. 2005, no. 14, pp. 2305–2321, 2005.
- [14] A. Dopfer and C. C. Wang, "What can we learn from accident videos?" in *2013 CACS International Automatic Control Conference (CACS)*, pp. 68–73, IEEE, Nantou, Taiwan, 2013.
- [15] C. Wang, C. Xu, J. Xia, Z. Qian, and L. Lu, "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.
- [16] C. Wang, C. Xu, and Y. Dai, "A crash prediction method based on bivariate extreme value theory and video-based vehicle trajectory data," *Accident Analysis and Prevention*, vol. 123, pp. 365–373, 2019.
- [17] Y. Guo, Z. Li, Y. Wu, and C. Xu, "Exploring unobserved heterogeneity in bicyclists' red-light running behaviors at different crossing facilities," *Accident Analysis and Prevention*, vol. 115, pp. 118–127, 2018.
- [18] C. Xu, H. Li, J. Zhao, J. Chen, and W. Wang, "Investigating the relationship between jobs-housing balance and traffic safety," *Accident Analysis and Prevention*, vol. 107, pp. 126–136, 2017.
- [19] Z. Liu, S. Wang, B. Zhou, and Q. Cheng, "Robust optimization of distance-based tolls in a network considering stochastic day to day dynamics," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 58–72, 2017.
- [20] H. Tan, J. Zhang, J. Feng, and F. Li, "Vehicle speed measurement for accident scene investigation," in *2010 IEEE 7th International Conference on E-Business Engineering*, pp. 389–392, IEEE, Shanghai, China, 2010.
- [21] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006.
- [22] X. Li, A. Ghiasi, Z. Xu, and X. Qu, "A piecewise trajectory optimization model for connected automated vehicles: exact optimization algorithm and queue propagation analysis," *Transportation Research Part B: Methodology*, vol. 118, pp. 429–456, 2018.
- [23] I. J. Lee, "An accident detection system on highway using vehicle tracking trace," in *2011 International Conference on ICT Convergence, ICTC 2011*, pp. 716–721, IEEE, Seoul, South Korea, 2011.
- [24] Z. Hui, Y. Xie, M. Lu, and J. Fu, "Vision-based real-time traffic accident detection," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 1035–1038, IEEE, Shenyang, China, 2015.
- [25] J. W. Hwang, Y. S. Lee, and S. B. Cho, "Hierarchical probabilistic network-based system for traffic accident detection at intersections," in *Proceedings - Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC and ATC 2010 Conferences, UIC-ATC 2010*, pp. 211–216, IEEE, Xian, Shaanxi, China, 2010.
- [26] Y. K. Ki 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.
- [27] 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.
- [28] Z. Xu, W. Tao, S. Easa, X. Zhao, and X. Qu, "Modeling relationship between truck fuel consumption and driving behavior using data from internet of vehicles," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 3, pp. 209–219, 2018.
- [29] C. Ma, W. Hao, X. Wang, and W. Yan, "The impact of aggressive driving behavior on driver injury severity at highway-rail grade crossings accidents," *Journal of Advanced Transportation*, vol. 2018, Article ID 9841498, 10 pages, 2018.
- [30] N. Wang, L. Qi, J. Dong, H. Fan, X. Chen, and H. Yu, "Two-stage underwater image restoration based on a physical model," in *Proceedings of SPIE 10225, Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, SPIE, Tokyo, Japan, 2017.
- [31] Q. Wang, J. Gao, and Y. Yuan, "A joint convolutional neural networks and context transfer for street scenes labeling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1457–1470, 2018.
- [32] Q. Li, F. Xu, X. Zhao, Y. Yao, and Y. Li, "High resolution image restoration algorithm of wavefront coding system based on

- wiener filter and wavelet de-noising,” in *Optoelectronic Imaging and Multimedia Technology IV*, SPIE, Beijing, China, 2016.
- [33] Y. S. Zhang, F. Zhang, and B. Z. Li, “Image restoration method based on fractional variable order differential,” *Multidimensional Systems and Signal Processing*, vol. 29, no. 3, pp. 999–1024, 2018.
- [34] Y. Zhang, L. Sun, C. Yan, X. Ji, and Q. Dai, “Adaptive residual networks for high-quality image restoration,” *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3150–3163, 2018.
- [35] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, “Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4520–4533, 2017.
- [36] Y. Wang and Z. Pan, “Image contrast enhancement using adjacent-blocks-based modification for local histogram equalization,” *Infrared Physics and Technology*, vol. 86, pp. 59–65, 2017.
- [37] W. He, Q. Wu, and S. Li, “Medical X-Ray image enhancement based on wavelet domain homomorphic filtering and CLAHE,” in *2016 International Conference on Robots & Intelligent System (ICRIS)*, pp. 249–254, IEEE, Zhangjiajie, China, 2016.
- [38] B. Shalchian, H. Rajabi, and H. Soltanian-Zadeh, “Assessment of the wavelet transform in reduction of noise from simulated PET images,” *Journal of Nuclear Medicine Technology*, vol. 37, no. 4, pp. 223–228, 2009.
- [39] K. Yun, H. Jeong, K. M. Yi, S. W. Kim, and J. Y. Choi, “Motion interaction field for accident detection in traffic surveillance video,” in *2014 22nd International Conference on Pattern Recognition*, pp. 3062–3067, IEEE, Stockholm, Sweden, 2014.
- [40] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, “Real-time automatic traffic accident recognition using HFG,” in *2010 20th International Conference on Pattern Recognition*, pp. 3348–3351, IEEE, Istanbul, Turkey, 2010.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

