Runtime Monitoring of Accidents in Driving Recordings with Multi-Type Logic in Empirical Models

Ziyan An[†], Xia Wang[†], Taylor T. Johnson, Jonathan Sprinkle, and Meiyi Ma

Department of Computer Science, Vanderbilt University, Nashville TN 37235, USA

Abstract. Video capturing devices with limited storage capacity have become increasingly common in recent years. As a result, there is a growing demand for techniques that can effectively analyze and understand these videos. While existing approaches based on data-driven methods have shown promise, they are often constrained by the availability of training data. In this paper, we focus on dashboard camera videos and propose a novel technique for recognizing important events, detecting traffic accidents, and trimming accident video evidence based on anomaly detection results. By leveraging meaningful high-level time-series abstraction and logical reasoning methods with state-of-the-art data-driven techniques, we aim to pinpoint critical evidence of traffic accidents in driving videos captured under various traffic conditions with promising accuracy, continuity, and integrity. Our approach highlights the importance of utilizing a formal system of logic specifications to deduce the relational features extracted from a sequence of video frames and meets the practical limitations of real-time deployment.

Keywords: Logic Specifications for Images and Videos. Systems with Learning-Enabled Components. Runtime Assurance.

1 Introduction

Dashboard cameras that are designed to continuously record video footage have become increasingly popular among vehicle owners [24, 25]. However, this design faces inherent limitations, particularly due to limited device storage capacity. For example, critical video evidence can be overwritten by continuous loop recording dashcams. Thus, there is a growing need for automated accident detection mechanisms that can identify relevant incidents captured in these recordings, such that important evidence can be preserved, while irrelevant sections of longer recordings can be trimmed and discarded on the fly.

Training a neural network to directly predict the anomaly label of each frame for accident detection is a potential approach (e.g. [8, 16]). However, this

[†]These authors made equal contributions to the work, and their order is based on the alphabetical order of their last names.

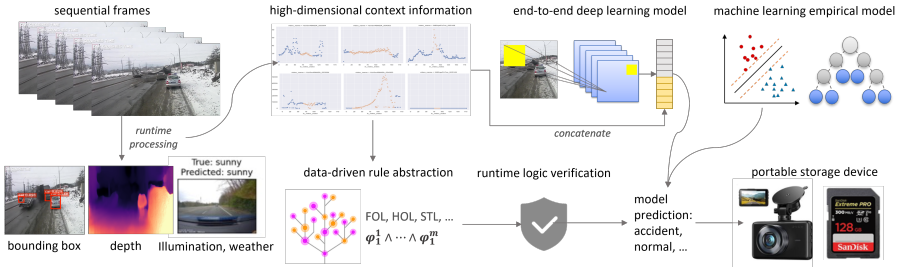

Fig. 1. Our approach leverages runtime logic verification, data-driven detection, and contextual information to accurately identify dangerous scenes in driving videos.

method typically demands a substantial amount of hand-annotated data, which may not always be practical or feasible to obtain [30]. Furthermore, ensuring that the dataset is diverse enough to capture the range of accident scenarios adds another layer of complexity [28].

An additional concern arises when data-driven models produce unrealistic labels while they are deployed online. For example, the labels may exhibit discontinuous behavior, jumping back and forth across consecutive frames. Such predictions contradict the continuous nature of accidents, which have clear starting and ending points. Ideally, the algorithm should produce consistent and continuous predictions in line with our understanding of accidents in reality.

Targeting these challenges, we propose a new framework for detecting driving accidents from dashboard camera recordings that address various accident scenarios without requiring access to additional privacy-sensitive information. Our approach is a lightweight framework that combines data-driven techniques with different types of logical properties, allowing for easy adaptation to different scenarios during deployment.

More specifically, we first demonstrate that high-level time-series features exhibit strong associativity with the occurrence of traffic accidents through supervised learning methods. Additionally, under other conditions being equal, the extensional experiment exhibits that the vision-based deep learning model incorporated with the interpretive time-series abstraction could exceed the pure vision model. From this, we give solid proof that the extracted interpretive contextual features have high discriminability for accident detection indeed. To further improve the accuracy and usability of the detection results, our proposed framework incorporates multiple types of logic properties for verification at runtime. Fig. 1 depicts an overview of our proposed approach.

This integration allows our proposed method to demonstrate more stable and robust performance, based on essential patterns captured from not only pure vision data but also interpretive time-series abstraction. For example, significant fluctuations in the sizes of bounding boxes can be observed in abnormal scenes, as compared to normal driving behaviors. Moreover, logic specifications as such are not limited to specific environments, road types, or times of the day. Therefore, the proposed method applies to different driving scenarios and does not

require extra training data for each scenario, unlike most existing data-driven approaches. To summarize, our framework offers the following contributions:

- We propose a two-step system to detect accidents in continuous driving videos. The system leverages vision-based algorithms, contextual information, and runtime supervision with higher-order logic to improve predictions.
- We extract interpretive high-level time-series features which capture important patterns and characteristics of abnormal scenes, enabling accurate and reliable accident detection.
- We further ensure the continuity, robustness, and generalizability of our framework under various driving scenarios by utilizing knowledge refined from the abstraction of high-level logic properties.

2 Related Work

Previously, data-driven approaches have demonstrated remarkable performance in detecting dangerous driving scenes and traffic accidents by leveraging offline-trained deep models (e.g., Zhao et al. [34], Doshi et al. [14]). However, data-driven methods for video-based accident detection face the inherent issue of dataset scarcity [4]. While large-scale vision datasets exist (e.g., Geiger et al. [17]), publicly available traffic anomaly datasets specifically designed for every detection algorithm's objective are limited. On the other hand, formal logic has found widespread application in image-related tasks. For example, Dokhanchi et al. [12] propose Timed Quality Temporal Logic to assess the quality of object detection algorithms in terms of spatio-temporal logical relations. Additionally, Balakrishnan et al. [2, 3] extend this work, enabling the specification and monitoring of logic to evaluate the results of object detection algorithms with objects appearing at different times.

In recent years, the integration of rules into empirical methods has gained traction as a strategy to enhance the robustness of data-driven research [15, 22, 27]. This approach enables researchers to apply well-established theoretical constructs and expert specifications to observational or empirical models, thereby improving the reliability and validity of their findings [33]. Notable examples of this research direction include Bakar et al.'s Tsetlin Machine [1], a lightweight automata learning algorithm utilizing propositional logic, and a recent work by Hashemi et al. [18] exploring the integration of temporal logic into control systems, allowing for the leveraging of a more diverse range of domain knowledge. To the best of our knowledge, this work represents the first investigation into the utilization of logical reasoning as a verification strategy to enhance overall detection performance on video-based driving datasets.

3 Motivating Study

To examine the difficulties of real-time accident detection in long driving videos, we present two motivating studies in this section. Firstly, we find

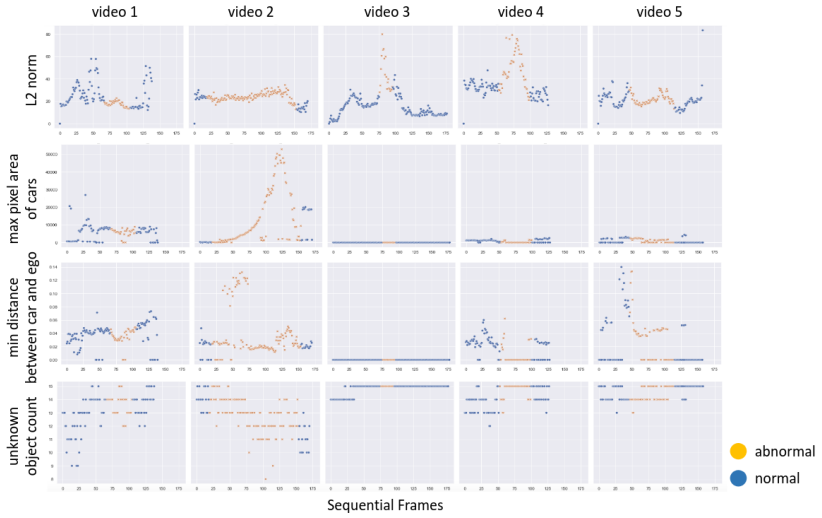

Fig. 2. Each column represents a randomly chosen video, while each row represents a meaningful high-level time-series feature, displayed sequentially. In row 1, a larger discrepancy in the L2 norm distance between two adjacent frames indicates an increased risk. In row 2, a larger pixel area of the car object's bounding box implies a higher level of risk. Row 3 shows a shorter distance (the distance feature is extracted from a thermal map, thus larger value implies a shorter distance) between the detected car object and the ego vehicle signifies a higher level of risk. Lastly, row 4 shows risky situations are usually accompanied by a higher number of unknown detected objects.

that time-series features exhibit a promising ability for anomaly differentiation. Thus, we incorporate this implicit contextual knowledge into empirical models to enhance their performance. Secondly, we observe discontinuity issues in the predictions made by end-to-end algorithms. As a solution, we emphasize the importance of incorporating logic verification to address this practical concern.

3.1 Analysis on High-Level Time-Series Abstraction

To demonstrate the distinguishing feature patterns between accident and non-accident driving video frames, we focus on four crucial high-level time series features. These features have been identified based on the feature importance factor of a trained Random Forest (RF) [5] model, as illustrated in Fig 2.

By combining these observations, we gain a more insightful understanding of how accident frames can be distinguished from normal scenes. In the figure above, we randomly select five videos as columns, where the orange points represent abnormal (accident) frames, and the blue points represent normal (non-accident) frames in the driving video sequence. In particular, the first row demonstrates that large gaps between adjacent frames indicate an elevated risk factor. Considering an accident as an out-of-control scenario, it is easy to correlate these large gaps with reasons such as driving too fast or encountering sudden unexpected

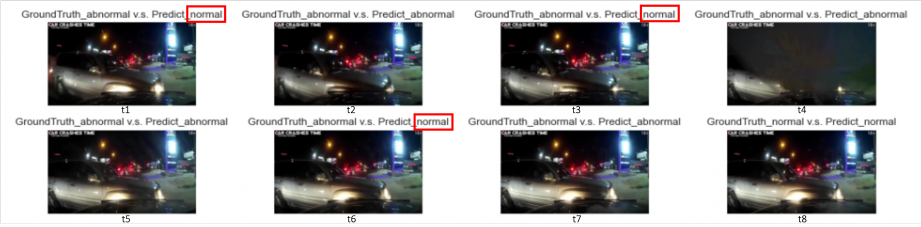

Fig. 3. Discontinious anomaly prediction provided by Random Forest via high-level time-series abstraction of an accident example video.

situations. The second row shows that a large pixel area of the car object in a frame, signifying the ego car's proximity to the leading car, indicates potential risk. This can be considered in conjunction with the third feature. As depicted in the third row of the figure, it is evident that a small distance from the car object to the ego in a frame is associated with a risk factor, and this close proximity beyond the safe boundary projects imminent danger. Finally, an unusually large number of unknown objects in a frame indicates complexity in the road environment, serving as a predictive indicator of risky situations.

3.2 Analysis on Discontinuity of Anomaly Detection

In practice, the occurrence of accidents is usually continuous, and the output of anomaly detection should not jump between normal and abnormal in the time sequence space within the time of the accident, resulting in discontinuous anomaly detection and incomplete interception of accident evidence. Although the high-level time-series features mentioned above can already be employed to obtain anomaly predictions with relevant good performance, we can intuitively observe from Fig 3 that this discontinuity of prediction still has a serious adverse impact on the availability of the anomaly detection system.

4 Proposed Method

Targeting the challenges identified above, our proposed framework (Fig. 1) begins by capturing frames from each first-person video stream using the small storage device. We employ pre-trained state-of-the-art vision models to extract multi-modality scene statistics and gain the differential features between two adjacent frames. Simultaneously, our second module extracts environmental features as time-series data, such as weather and illumination conditions. These two sets of time-series data are concatenated to generate a high-level time-series abstraction. The downstream decision-making component provides anomaly prediction by either utilizing only the abstraction via multiple machine learning models or by combining the abstraction with pure vision information via deep learning models. Finally, the decision is calibrated and enhanced by multiple types of logic specifications and properties.

4.1 High-Level Time-Series Feature Extraction

Object detection feature extraction As the contextual and surrounding data of the ego, we extract the following information from each video:

- Object count: We detect 12 transportation-related classes using YOLOv3 and aggregate the total count for each class in each frame.
- Detection confidence: We gather statistics for the minimum, maximum, and mean confidence scores associated with each object bounding box.
- Object distance: We measure the distance between the ego vehicle and the center of each detected bounding box, then record statistics for minimum, maximum, and mean distances.
- Object size: By calculating the pixel area of each object's bounding box, we determine the minimum, maximum, and mean sizes for each class.

Frame difference feature extraction The frame difference feature is a commonly used technique in video processing and analysis to detect motion in a sequence of frames. It involves computing the difference between two adjacent frames and using this difference as a feature for further analysis. Therefore, in this work, we incorporate the following frame-by-frame difference features:

- L-2 norm is a common method used to measure the difference between two frames. It quantifies the average squared difference between the pixel values of the two frames. Mathematically, the L-2 norm is defined as $L_2(x_1, x_2) = \|x_1 - x_2\|_2^2$. A lower L-2 distance indicates two more similar frames.
- We compute the top 10 eigenvalues of the absolute difference matrix, implying the most striking features of the difference between two consecutive frames.

Illumination and weather feature extraction For illumination and weather prediction tasks, we first perform model selection on part of the labeled dataset using several deep learning models, including ResNet-50 [19], MobileNet [20], VGG-19 [29], Xception [9], and DenseNet-201 [21]. For both prediction tasks, DenseNet-201 performs the best. For illumination prediction tasks, we train the model on a two-class dataset that contains day and night images, and for weather prediction tasks, we train the model on a hybrid five-class image dataset containing sunny, cloudy, rainy, snowy, and foggy weather [31]. Further, we use pre-trained models to predict illumination and weather labels for dashcam driving frames. Finally, we assign the majority of predictions of the entire driving video as the final illumination and weather label for all frames in a video.

4.2 Empirical Model for Accident Classification

We show that our framework, incorporating a machine learning model based on only high-level time-series features, together with a deep learning model utilizing both vision information and high-level time-series abstraction, can provide feasible anomaly predictions.

As there are multiple time-series features extracted from driving dash cam video frames, we employ basic machine learning algorithms to examine the correlation between these features and the accident classes. The abstraction comprises 13 dimensions of frame difference features, 120 dimensions of object detection features, and 2 dimensions of illumination and weather label features, with the dependent variable being the binary label.

We then design a deep neural network architecture that leverages pre-trained ResNet-18 weights as the feature extractor and fully connected layers that process high-level contextual information to detect the occurrence of accidents. The structure consists of the following components. Firstly, raw frames from driving videos are passed through the pre-trained ResNet-18 feature extractor, which outputs learned image representations from the complex model. After that, a linear layer is utilized to reduce the dimension of the learned representation. Then, an additional linear layer equipped with batch normalization and ReLU activation is employed to learn representations from high-level contextual information. The outputs from both the pre-trained ResNet-18 feature extractor and the additional linear layer with contextual information are concatenated into a one-dimensional tensor. The concatenated tensor is then fed into a final linear layer with a Sigmoid activation function that maps the output to a probability range between 0 and 1. The output from the Sigmoid activation is rounded to obtain the final class label.

4.3 Logic Calibration and Verification

FOL and HOL Specifications During the process of deploying machine learning models with high-level time-series features, we have observed that the path logic of the decision tree could provide usable anomaly predictions to some extent. It is familiar that a tree structure could be represented as a tree graph and several mutually exclusive rules, which could be considered as First Order Logic (FOL) for further applications. In this case, each rule has a probability associated with the class based on the majority of labels. However, as discussed above, this data-driven FOL cannot ensure the accuracy, continuity, and integrity of the anomaly detection task. Thus, we propose employing Higher Order Logic (HOL) generated from the FOL.

The design of HOL stems from an idea of simplicity. As mentioned above, each sample (frame) of the dataset will be assigned to a leaf node of the decision tree, thus each sample could apply to one rule, which is the path the sample goes through from the root node to the leaf node. Plus, this sample could inherit the probability associated with the leaf node. It is assumed that the final decision could rely on continuous observation of the sequential frames. Assuming the observation window size is k, the **abnormal impulse**, denoted as **odds**, up to the current frame i can be defined by dividing the combination of probabilities that these frames are abnormal by the combination of probabilities that the frames are normal, as (1). Note the probability of each frame normal is p_0^j, and the probability of abnormal is p_1^j, $p_0^j + p_1^j = 1$. Then, when the odds signal is strong enough, the system recognizes it as the starting point of an accident;

when the odds signal is weak enough, the system recognizes it as the ending point of an accident, and all continuous frames between the starting point and the ending point will be judged as abnormal continuously, the process is shown in Fig. 4. Moreover, the threshold value of the odds signals for the starting and ending points can be obtained by data-driven method or annotated by experts.

$$\text{odds}^i = \frac{\prod_{j=i-(k-1)}^{i} p_1^j}{\prod_{j=i-(k-1)}^{i} p_0^j} \tag{1}$$

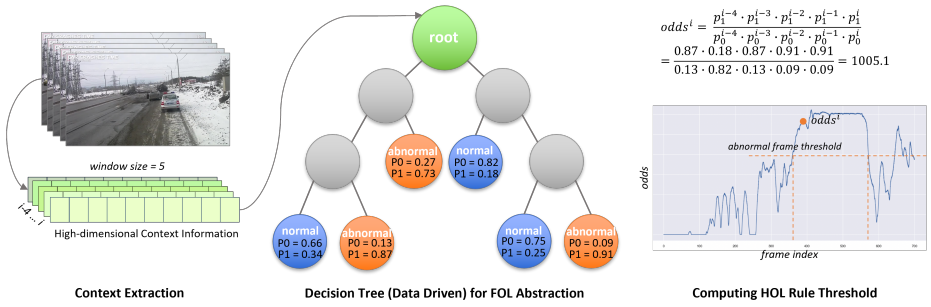

Fig. 4. Process of FOL and HOL calibration. In this example, we set the window size to 5, and we employ an array with dimension 122 to represent context information.

Temporal Logic Specifications Temporal logic is a powerful tool for expressing specifications on time-series data, offering formal symbolism, evaluation mechanisms, and semantics [13]. In our proposed framework, we utilize Signal Temporal Logic (STL) to establish verification criteria [23]. This allows us to consider a range of signal types, such as raw images, extracted time-series features, and factorized data. These verification criteria can either be provided by an expert, derived from empirical observations, or learned from data-driven algorithms. In STL, we represent the temporal range as $[a, b] \in \mathbb{R}_{\geq 0}$, where $a \leq b$. We use the function $\mu : \mathbb{R}^n \to \{\top, \bot\}$ to denote a signal predicate, such as $f(x) \geq 0$, where $x \in \mathcal{X}$ is the signal variable. The temporal operator "always" is denoted by $\Box$, and the temporal operator "eventually" is denoted by $\Diamond$. More specifically, $\Box_{[a,b]}\varphi$ indicates that φ should hold at every future time step within the interval $[a, b]$. $\Diamond_{[a,b]}\varphi$ indicates that φ should hold at some future time step within $[a, b]$. In our framework, specifications such as "the accident frame should satisfy the property that its maximum bounding box sizes of the previous and following three frames are greater than 60% of the maximum bounding box size across all n frames" can be easily specified with the STL formula $\Box_{[1,n]}(\Box_{[t-3,t+3]} max(\mathbf{bbox}) \geq 0.6 \cdot \mathbf{mbox})$, where t specifies the current timestamp, n denotes the total number of frames, "bbox" denotes the bounding

box areas of the current frame, and “mbox" denotes the maximum bounding box size across the video.

5 Evaluation Results

5.1 Evaluation Setup

In this work, we utilized real traffic accident frames obtained from user-uploaded YouTube videos for our evaluations [32]. To be more specific, we downloaded a total of 627 lengthy videos from the internet, each of which could be processed into several hundred frames with corresponding annotations. From these videos, we selected 29 videos (approximately 5%) to form our testing dataset, with each video containing complete sequential frames. Out of the remaining 598 videos, which encompassed a total of 64,563 individual frames, we divided the frames randomly into a training dataset (80%) and a validation dataset (20%). The percentage of accident frames in the dataset was 33.89%. The validation dataset was used for model fine-tuning and model selection processes.

5.2 Baselines

In this work, all deep learning models were implemented and trained using PyTorch framework, utilizing an NVIDIA RTX-3070 GPU. We experimented with two optimization algorithms, namely Stochastic Gradient Descent (SGD) and Adam, with different learning rates of 1e-2, 1e-3, and 1e-4. The models with the best performance during the evaluation process were reported. More specifically, we consider the methods listed below.

1. ResNet18-ft: We employ a pre-trained ResNet18 on ImageNet and freeze the last two layers for fine-tuning, which solely relies on vision information.
2. ResNet18-ts: This model is proposed in this work, which incorporates both raw images and contextual information into the accident detection process.

Table 1. Comparison of model performance.

Methods	**Testing Performance**			
	Accuracy	**Recall**	**F1**	**Runtime (s)**
XGB [7]	76.36%	37.46%	47.78%	0.012
LR [11]	69.88%	0.44%	0.84%	0.003
DT [6]	63.43%	51.05%	44.64%	**0.002**
RF [5]	74.98%	44.64%	50.75%	0.071
SVM [10]	70.64%	1.22%	2.34%	12.99
NB [26]	70.64%	1.22%	2.34%	13.12
RF-HOL (Ours)	**85.74%**	**68.29%**	**73.44%**	3.197
ResNet-ft [19]	60.94%	47.73%	41.38%	2.155
ResNet-ts (Ours)	**73.48%**	50.61%	**52.43%**	2.125
ResNet-ts-STL (Ours)	67.80%	**59.12%**	51.47%	**0.017**

3. Supervised ML models: We implement multiple machine learning models, such as XGBoost (XGB), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), SVM, and Naive Bayes (NB).

5.3 Evaluation of Model Performance

To evaluate the performance of our proposed algorithm, we consider the following metrics: accuracy, recall rate, F1 score, and runtime. More specifically, accuracy is defined as the ratio of correct predictions to the total number of instances. Recall Rate is defined as the ratio of true positives (TP) to the sum of TP and false negatives (FN). F1 Score is the harmonic mean of precision and recall. And precision is the ratio of TP to the sum of TP and false positives (FP). The performance of our accident detection module is demonstrated in Table 5.1.

With HOL calibration, the Random Forest model outperforms other models in terms of accuracy, recall rate, and F1 score. The accuracy of the ResNet finetuning model, which solely relies on vision information, was enhanced from 60.94% to 73.48% by incorporating the concatenated features of image representation and high-level time-series features proposed in this work. Moreover, when additionally equipped with Signal Temporal Logic (STL), the recall performance of ResNet also improved from 50.61% to 59.13%. It is worth noting that the post-hoc verification process only required 0.017 seconds. Additionally, the decision tree (DT) model demonstrates exceptional runtime effectiveness. In scenarios where real-time requirements are extremely demanding, the decision tree model demonstrated acceptable anomaly detection effectiveness.

6 Conclusion

In this paper, we investigate runtime verification for detecting accidents in driving footage with contextual information and raw dash cam videos. Moreover, we demonstrate that by integrating the abstraction of high-level time-series features, which exhibit distinguishable patterns under supervised learning methods, the performance of image-based deep learning models for accident detection can be significantly enhanced. Additionally, our proposed model leverages multiple logic specifications to verify and calibrate anomaly predictions. This approach ensures that the predicted accident labels maintain exceptional accuracy, continuity, and integrity, while also guaranteeing a stable and robust system across various accident scenarios.

Acknowledgment

This work was supported, in part, by the National Science Foundation under Grant 2151500, 2028001, and 2220401.

References

1. Bakar, A., Rahman, T., Shafik, R., Kawsar, F., Montanari, A.: Adaptive intelligence for batteryless sensors using software-accelerated tsetlin machines. In: Proceedings of SenSys (2022)
2. Balakrishnan, A., Deshmukh, J., Hoxha, B., Yamaguchi, T., Fainekos, G.: Percemon: online monitoring for perception systems. In: Runtime Verification: 21st International Conference, RV 2021, Virtual Event, October 11–14, 2021, Proceedings 21. pp. 297–308. Springer (2021)
3. Balakrishnan, A., Puranic, A.G., Qin, X., Dokhanchi, A., Deshmukh, J.V., Amor, H.B., Fainekos, G.: Specifying and evaluating quality metrics for vision-based perception systems. In: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1433–1438. IEEE (2019)
4. Bashetty, S.K., Amor, H.B., Fainekos, G.: Deepcrashtest: Turning dashcam videos into virtual crash tests for automated driving systems. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 11353–11360. IEEE (2020)
5. Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)
6. Breiman, L.: Classification and regression trees. Routledge (2017)
7. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
8. Choi, J.G., Kong, C.W., Kim, G., Lim, S.: Car crash detection using ensemble deep learning and multimodal data from dashboard cameras. Expert Systems with Applications **183**, 115400 (2021)
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions (2017)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**, 273–297 (1995)
11. Cox, D.R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society Series B: Statistical Methodology **20**(2), 215–232 (1958)
12. Dokhanchi, A., Amor, H.B., Deshmukh, J.V., Fainekos, G.: Evaluating perception systems for autonomous vehicles using quality temporal logic. In: Runtime Verification: 18th International Conference, RV 2018, Limassol, Cyprus, November 10–13, 2018, Proceedings 18. pp. 409–416. Springer (2018)
13. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: International Conference on Formal Modeling and Analysis of Timed Systems. pp. 92–106. Springer (2010)
14. Doshi, K., Yilmaz, Y.: An efficient approach for anomaly detection in traffic videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4236–4244 (2021)
15. Dreossi, T., Fremont, D.J., Ghosh, S., Kim, E., Ravanbakhsh, H., Vazquez-Chanlatte, M., Seshia, S.A.: Verifai: A toolkit for the formal design and analysis of artificial intelligence-based systems. In: Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I 31. pp. 432–442. Springer (2019)
16. Du, Y., Qin, B., Zhao, C., Zhu, Y., Cao, J., Ji, Y.: A novel spatio-temporal synchronization method of roadside asynchronous mmw radar-camera for sensor fusion. IEEE transactions on intelligent transportation systems **23**(11), 22278–22289 (2021)
17. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)

18. Hashemi, N., Hoxha, B., Yamaguchi, T., Prokhorov, D., Fainekos, G., Deshmukh, J.: A neurosymbolic approach to the verification of temporal logic properties of learning-enabled control systems. In: Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023). pp. 98–109 (2023)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
20. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017)
21. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2018)
22. Ma, M., Gao, J., Feng, L., Stankovic, J.: Stlnet: Signal temporal logic enforced multivariate recurrent neural networks. Advances in Neural Information Processing Systems **33**, 14604–14614 (2020)
23. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems. pp. 152–166. Springer (2004)
24. Rea, R.V., Johnson, C.J., Aitken, D.A., Child, K.N., Hesse, G.: Dash cam videos on youtube™ offer insights into factors related to moose-vehicle collisions. Accident Analysis & Prevention **118**, 207–213 (2018). https://doi.org/https://doi.org/10.1016/j.aap.2018.02.020, https://www.sciencedirect.com/science/article/pii/S0001457518300824
25. Richardson, A., Sanborn, K., Sprinkle, J.: Intelligent structuring and semantic mapping of dash camera footage and can bus data. In: 2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS). pp. 24–30 (2022). https://doi.org/10.1109/DI-CPS56137.2022.00010
26. Rish, I., et al.: An empirical study of the naive bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. vol. 3, pp. 41–46 (2001)
27. Seshia, S.A., Sadigh, D., Sastry, S.S.: Towards verified artificial intelligence. arXiv preprint arXiv:1606.08514 (2016)
28. Shah, A.P., Lamare, J.B., Nguyen-Anh, T., Hauptmann, A.: Cadp: A novel dataset for cctv traffic camera based accident analysis. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–9. IEEE (2018)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
30. Štitilis, D., Laurinaitis, M.: Legal regulation of the use of dashboard cameras: Aspects of privacy protection. Computer Law & Security Review **32**(2), 316–326 (2016)
31. Xiao, H., Zhang, F., Shen, Z., Wu, K., Zhang, J.: Classification of weather phenomenon from images by using deep convolutional neural network. Earth and Space Science **8**(5), e2020EA001604 (2021)
32. Yao, Y., Wang, X., Xu, M., Pu, Z., Atkins, E., Crandall, D.: When, where, and what? a new dataset for anomaly detection in driving videos (2020)
33. Zhao, Y., An, Z., Gao, X., Mukhopadhyay, A., Ma, M.: Fairguard: Harness logic-based fairness rules in smart cities. arXiv preprint arXiv:2302.11137 (2023)
34. Zhao, Y., Wu, W., He, Y., Li, Y., Tan, X., Chen, S.: Good practices and a strong baseline for traffic anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3993–4001 (2021)