Journal of Sensors

WILEY

RESEARCH ARTICLE OPEN ACCESS

A Comparative Evaluation of Deep Learning Models for Road Accident Detection Using CCTV Images and Dashcam Videos

Syed Jamaluddin Ahmad[1] | Hamimah Ujir[2] | Irwandi Hipiny[2] | Syahrul Nizam Junaini[1]

[1]Faculty of Computer Science and Information Technology, University Malaysia Sarawak, Kota Samarahan 94300, Sarawak, Malaysia | [2]Faculty of Computing and Software Engineering, iCATS University CollegeJalan Stampin Timur, Kuching 93350, Sarawak, Malaysia

Correspondence: Syahrul Nizam Junaini (syahruln@unimas.my)

Received: 26 February 2026 | **Revised:** 8 May 2026 | **Accepted:** 13 May 2026

Academic Editor: Pramita Mishra

Keywords: CCTV | CNN | deep learning | LSTM | machine learning | RNN | RTA | SVM | traffic accidents

ABSTRACT

Road traffic accidents (RTAs) remain a major global safety challenge, requiring reliable automated detection systems to support rapid emergency response, traffic management, and smart city surveillance. This study presents a comparative evaluation of deep learning models for road accident detection using heterogeneous visual traffic sources, namely CCTV images and dashcam videos. A unified evaluation framework is introduced to systematically compare traditional machine learning, standard deep learning, and hybrid deep learning models within a consistent experimental setting. Publicly available datasets comprising several 1000 annotated CCTV images and thousands of dashcam video clips were used, covering diverse traffic scenarios, including urban and highway environments, day and night conditions, weather variations, and multiple camera viewpoints. The evaluated models were grouped into baseline approaches, consisting of Support Vector Machine, 2D Convolutional Neural Network (2D-CNN), R-CNN, and Long Short-Term Memory (LSTM) networks, and a proposed hybrid VGG16-LSTM model. This categorization reflects the complementary strengths of spatial and temporal feature extraction for static image-based and dynamic video-based accident detection. For CCTV image analysis, the 2D-CNN achieved the best performance, with 99% accuracy and a 98% F1-score. For dashcam video analysis, the VGG16-LSTM model outperformed competing approaches, achieving 99.53% accuracy and a high area under the receiver operator characteristic (ROC) curve. The findings demonstrate the effectiveness of convolutional models for spatial accident representation and the importance of temporal modeling for video-based detection. The proposed cross-modality framework provides practical insights into model suitability across different traffic data sources and has potential integration into Intelligent Transportation Systems. Future research should examine cross-dataset validation, event-level detection, and robustness in uncontrolled real-world environments.

1 | Introduction

Traffic accidents pose a major public safety issue worldwide, leading to elevated levels of injuries, deaths, and financial damage. As cities expand and the number of vehicles rises, conventional approaches to traffic management and accident prevention find it challenging to address the growing complexities. The emergence of artificial intelligence (AI) presents promising prospects for improving road safety through predictive analytics, real-time data processing, and automated intervention systems.

Approximately 1.3 million fatalities occur annually because of road accidents, according to statistics from the World Health Organization (WHO). There are two categories of causes involved; the first category encompasses pre-accident factors, while the second pertains to post-accident factors. The first category illustrates causal relationships such as adverse weather conditions, inadequate road infrastructure, and insufficient systems for accident detection prior to their occurrence; the second category involves delays in emergency response and medical services [1].

In the event of a traffic accident, alert system produces precise information regarding the locations and circumstances of the incidents [2–4], enabling prompt actions to rectify the imbalance situation. Various digital methods have been implemented in several satellite cities to manage and regulate road conditions. These initiatives have employed a diverse array of technologies, including computer vision, the Internet of Things (IoTs) [5], blockchain [6], and Vehicle Ad hoc Network (VANET).

Numerous studies have employed different methods to effectively identify collisions to support road traffic accident (RTA) management [7]. Two blockchain-based approaches for accident detection introduce an offline detection system for scenarios without internet access [8]. Additionally, a system tailored for autonomous vehicles [9] utilizes a dashboard camera. Another study [10, 11] implemented computer vision techniques to monitor moving objects.

A multitude of studies depend on datasets that are either incomplete, outdated, or fail to reflect real-world situations [12–15]. Additionally, accident data frequently leans toward non-accident occurrences, complicating accurate predictions of rare events [16]. Many sophisticated models, particularly those utilizing deep learning, lack interpretability, which hinders comprehension and trust in their predictions [17, 18]. Furthermore, a majority of studies concentrate on correlation-based predictions while overlooking causal relationships among factors such as weather, road conditions, and driver behavior [19]. Most concerningly, the utilization of personal and sensitive information (such as GPS and in-vehicle monitoring) presents ethical and legal dilemmas [17, 20]. The challenge of reconciling predictive capabilities with data privacy remains an open question. Convolutional Neural Networks (CNNs) and various other deep learning frameworks have demonstrated considerable promise in identifying traffic accidents [17, 18, 20, 21], yet there are numerous aspects that require enhancement to improve their accuracy, dependability, and practical application.

In terms of road accident detection, accuracy is a critical concern. However, many previous studies have not reached very high accuracy levels, which can make it difficult for people to fully trust the proposed models or systems. With higher accuracy in accident detection, it becomes easier to identify the factors that contribute to road accidents and to determine the parties who may be responsible for the incident. In this work, our goal is to improve the accuracy of the model for detecting road accidents.

This study aims to address these challenges. The first objective is to evaluate the effectiveness of 2D-CNN for frame-level accident detection using CCTV imagery, while the second objective is to assess hybrid deep learning models for video-based accident detection using dashcam recordings. A further objective is to compare these approaches under unified performance metrics to provide clearer empirical insight.

This study contributes by: (i) providing a comparative evaluation of image-based and video-based accident detection pipelines using CCTV and dashcam data, (ii) analyzing classifier behavior across multiple performance metrics under identical data conditions, and (iii) highlighting practical limitations of high-accuracy deep learning models when applied to small-scale real-world datasets.

The remainder of this paper is organized as follows. Section 2 reviews related work on accident detection using machine learning and deep learning approaches, while Section 3 describes the datasets, model architectures, and evaluation methodology. Section 4 presents and analyses the experimental results, followed by discussion and limitations in Section 5, and concluding remarks and future research directions in Section 6.

2 | Literature Review

Previous research has shown the capability of AI to improve road safety through a variety of approaches. Throughout the decade of AI from 2014 to 2024, a significant number of studies have focused on employing AI techniques for traffic accident detection. The extensive literature provides a review of the highly illustrative research conducted over the past 10 years, spanning from 2014 to 2024.

Robles-Serrano et al. [12] introduced various studies that concentrate on the use of video footage from traffic cameras to identify unusual events that could suggest an accident. CNNs and Recurrent Neural Networks (RNNs) are frequently employed to examine visual data. These models can be trained using labeled datasets that annotate accident occurrences, enabling them to detect patterns linked to collisions.

Yadav et al. [13] demonstrated that deep learning algorithms, including CNNs and Long Short Term Memory (LSTM) networks, were utilized on time-series data collected from sensors positioned along roadsides, successfully categorizing various types of incidents by analyzing patterns in vehicle movements. While M et al. [17] introduced a system that gathers data from nearby vehicles in proximity and processes this information through machine learning techniques aimed at predicting potential traffic accidents. The system collects data from GPS and GSM sources and employs three algorithms: SVM, RNN, and CNN.

Roy et al. [18] have proposed a system that collects real-time data from CCTV footage, utilizing three CNN models: ResNet50, VGG16, and a custom-built CNN to extract complex features from the frames of the footage. The training dataset was also gathered from Kaggle, and the model was evaluated to determine the occurrence of accidents or non-accidents, thus facilitating a quicker response for road safety.

The concept of VANET is introduced by Sharma et al. [19] and they demonstrated the use of machine learning algorithms for traffic accident detection where a random forest (RF) classifier was employed to collect simulated data. The accuracy of the data obtained from the RF classifier was found to be merely 73.13%.

Furthermore, Desai et al. [21] identified traffic accidents by employing computer vision and AI methods based on images captured from CCTV footage, along with an alert system facilitated by an android application utilizing the YOLOv3 algorithm. In earlier technologies, sensors were utilized for accident detection, and notifications would be dispatched to the driver's relatives. However, sensors may become damaged or fail to operate effectively. To address this issue, computer vision and AI techniques were implemented for accident detection and notifications via an android application.

Tamagusko et al. [22] implemented synthetic images in conjunction with transfer learning to identify road accidents, utilizing transfer learning to facilitate the adaptation of one feature to different settings. Transfer learning involves acquiring knowledge from one problem and applying it to a new, analogous problem.

In social networking communication, Ali et al. [23] outlined a framework for real-time monitoring based on social networks, which is proposed for the detection of traffic accidents and the analysis of conditions utilizing ontology, latent Dirichlet allocation (OLDA), and bidirectional LSTM (Bi-LSTM).

Hadi Nassar et al. [24] conducted a comprehensive review of the machine learning techniques employed by researchers to date, including CNN, RNN, SVM, and others, for the purpose of detecting traffic accidents, utilizing datasets from multiple sources. There are lots of models available in machine learning approaches such as CNN, RNN, SVM, LSTM and KNN in Table 1. The specific datasets from CCTV and dashcam were used for each classifier to evaluate the performance, respectively.

3 | Methodology

3.1 | Data Arrays

The data arrays were collected from multiple sources, particularly noting that Kaggle is the most suitable for this research [26]. It includes segments of CCTV visual content pertaining to both crash and non-crash, systematically outlined in Table 2.

It is categorized into four distinct types: vehicle collisions, collisions between cars and motorcyclists, collisions involving pedal cyclists, and collisions between vehicles and pedestrians. The dataset was divided into 70% training, 15% validation, and 15% testing, ensuring that model training, hyperparameter tuning, and final evaluation were conducted on mutually exclusive subsets.

In addition, we have enhanced the results section to improve its technical depth. Beyond reporting performance metrics such as accuracy and F1-score, we now provide a more detailed analysis of model behavior, including comparative performance across models, discussion of generalization capability, and explanation of why certain models (e.g., CNN-based architectures) outperform others. We also clarify the limitations of models such as SVM and KNN in handling high-dimensional image data without effective feature extraction.

TABLE 2 | Quantity of features in the applied data array.

Feature	Accident	Non-accident
Training frames	369	422
Testing frames	47	54
Verification	46	52

TABLE 3 | Number of mounted frames of DAD dataset.

Frames	Number of frames
Crash frames	1500
Normal frames	3000

The dataset collected from Dashcam Accident Datasets (DAD) [27], demonstrates that the implementation of a dashcam affixed to vehicles for recording actual traffic accident footage is essential for the advancement of self-driving technology with safety assurances; the specifics of the dataset are presented in Table 3.

3.2 | Machine Learning Algorithms

Evaluate the accuracy of the classifiers CNN, RNN, R-CNN, KNN, SVM, and LSTM using the datasets from Kaggle [26] and DAD [27]. CNNs are proficient in the automatic extraction of spatial hierarchies and features from images. This ability enables them to identify the intricate patterns linked to traffic accidents, including vehicle collisions and atypical behaviors. By employing multiple layers along with convolutional filters, CNNs can identify features at various scales.

On the contrast, the absence of robust feature extraction and selection can significantly impact the performance of SVM and KNN when handling high-dimensional image data, as these models rely heavily on the quality of input features. This limitation partly explains their inferior performance compared to deep

TABLE 1 | Different machine learning classifiers.

Model	Advantages	Limitations	Why CNN is better	Ref.
CNN	Great for image/video classification	High computational cost	Achieve high accuracy on large datasets	[12]
R-CNN	Great for image or videos classification	Relatively high cost	Achieve high accuracy on large datasets	[24]
SVM	SVM's can model nonlinear decision boundaries	Don't scale well to larger datasets	Not suitable for large scale image dataset	[17]
LSTM	Learns long-range dependencies	Intensive, prone to overfitting	Faster and less prone to overfitting for real-time	[13]
KNN	Simple to understand, fast to train, and effective for nonlinear data	Computationally expensive for large datasets	Better for nonlinear datasets	[25]

learning models such as CNN, which automatically learn hierarchical feature representations directly from raw images.

In this instance, CNNs will be utilized to detect patterns in images for the identification of objects, classes, and categories that may signify an accident or a non-accident. Both the training and test datasets will be fed into the CNNs to ascertain whether the outcome is an accident or a non-accident as shown in Figure 1.

The block diagram of 2D-CNN was showed as in Figure 2 and architecture of 2D-CNN model as in Figure 3, respectively.

This information will subsequently be relayed to the appropriate authorities as an alert, prompting immediate and effective actions to enhance traffic safety.

The proposed 2D-CNN is designed to learn hierarchical spatial features from CCTV images and dashcam frames for binary classification (accident vs. non-accident). The architecture consists of four sequential convolutional blocks, each comprising a Conv2D layer with 3×3 kernels and ReLU activation, followed by max-pooling 2×2 for spatial downsampling, with the number of filters increasing progressively from 128 to 512 to capture increasingly complex features. The extracted feature maps are then flattened into a one-dimensional vector and passed through a fully connected dense layer with 1024 neurons and ReLU activation, followed by a dropout layer (rate = 0.5) to reduce overfitting.

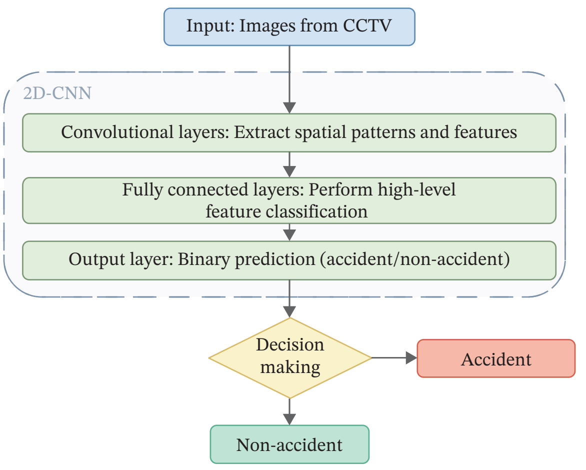

FIGURE 1 | System architecture for RTA detection using 2D-CNN.

Finally, a softmax output layer with two neurons produces class probabilities. The model takes input images of size $224 \times 224 \times 3$ and is trained using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss, ensuring an efficient and reproducible design for accident detection. Keras [28] and TensorFlow [29] are robust frameworks that aid in the creation, training, and implementation of CNNs for a range of applications, such as identifying traffic accidents. Below is an explanation of their role in this process.

The proposed VGG16-LSTM architecture combines spatial and temporal feature learning for accident detection from video sequences. A pretrained VGG16 network (with top layers removed) is used as a feature extractor, where input frames of size $224 \times 224 \times 3$ are processed individually to produce high-level spatial feature vectors; earlier layers are frozen while the final convolutional block is fine-tuned. These features are then passed through a Time Distributed wrapper and fed into an LSTM layer with 128 units to capture temporal dependencies across a sequence of frames (e.g., 10 frames per clip).

The output of the LSTM is further processed by a dense layer with 64 neurons and ReLU activation, followed by a dropout layer (rate = 0.5) to reduce overfitting, and finally a softmax output layer with two neurons for binary classification (accident vs. non-accident). The model is trained using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss, ensuring an effective and reproducible spatiotemporal learning framework.

3.3 | Evaluation of the Classifiers

Analytical procedure of the data encompasses model evaluation, which enables us to assess how effectively the model categorizes data, while also analyzing its strengths and weaknesses by comparing the performance with real data. The receiver operator characteristic (ROC) curve is commonly employed in the examination of binary outcomes to illustrate the effectiveness of a model or algorithm [30].

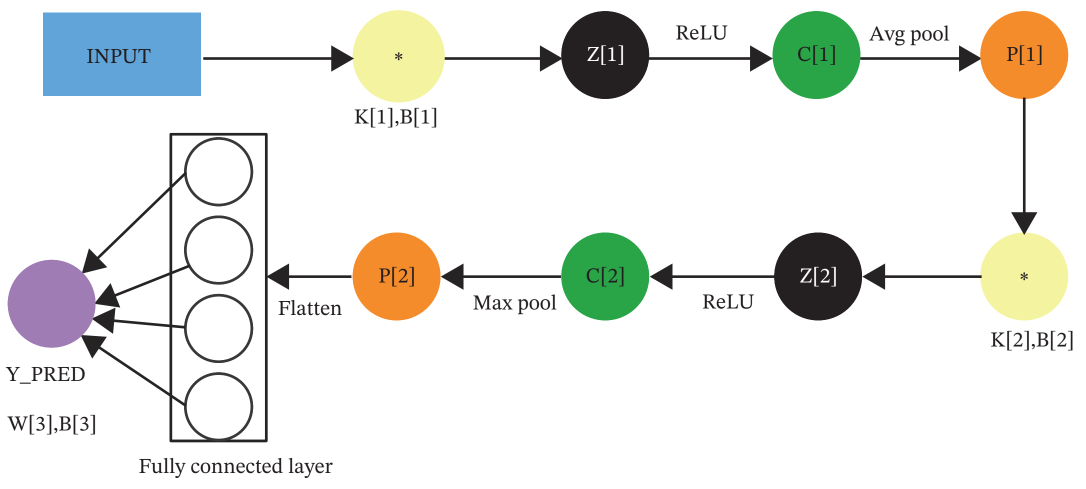

FIGURE 2 | Block diagram of the 2D-CNN. *Means convolution operation.

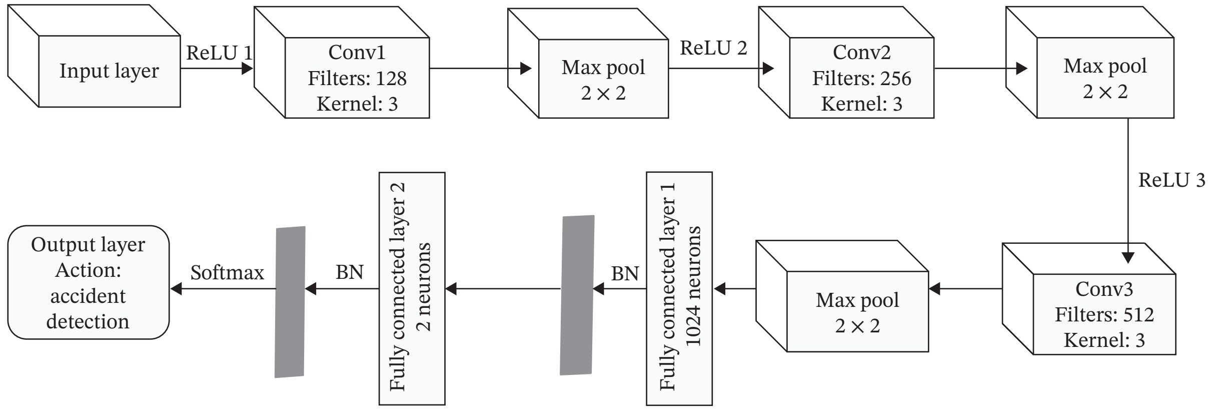

FIGURE 3 | Architecture of the 2D-CNN model.

In order to obtain insights into performance across various criteria, this ROC can be distilled into a solitary quantitative data, known as the area under the curve (AUC). The confusion matrix is used to generate these metrics, which include false negatives (FNs), true negatives (TNs), false positives (FPs), and true positives (TPs). Multiple metrics are connected with the confusion matrix including specificity-FPR, sensitivity-TPR, accuracy, precision, recall, F1-measure, and ROC (AUC), which demonstrates the link between 1-specificity and sensitivity.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{1}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{2}$$

$$\mathrm{F1} - \mathrm{measure} = \frac{2 \times \mathrm{Precision} \times \mathrm{recall}}{\mathrm{Precision} + \mathrm{recall}}, \tag{3}$$

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{4}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{5}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{6}$$

Overfitting is a critical concern, particularly in deep learning models applied to image-based datasets. To mitigate this risk and ensure realistic performance, several measures were incorporated into the experimental pipeline. Specifically, the dataset was divided into distinct training and testing sets, and all reported results are based exclusively on evaluation using unseen test data.

In addition, data augmentation techniques such as rotation, flipping, and scaling were applied to increase variability in the training set. Furthermore, regularization strategies such as dropout and/or early stopping were employed during model training to improve generalization. These strategies collectively reduce the likelihood of overfitting and help ensure that the reported performance reflects the model's ability to generalize rather than memorize the training data.

4 | Results and Analysis

The comparison of the classifiers' results is presented in this section. The test set used has 791 instances from CCTV footages, which are categorized as accidents and non-accidents. While the CCTV test set consists of 791 instances, several measures were implemented to mitigate the risk of overfitting and ensure the robustness of the models.

First, we adopted a structured data split strategy, where the dataset was divided into training, validation, and testing sets (e.g., 70:15:15), ensuring that the test set remained completely unseen during training. Second, data augmentation techniques, including rotation, flipping, and scaling were applied to increase the diversity of the training data and improve generalization. Third, regularization methods such as dropout (rate = 0.5) were incorporated into the network architecture to reduce model overfitting.

In addition, for the VGG16-LSTM model, transfer learning using VGG16 was employed, with earlier layers frozen to reduce the number of trainable parameters, thereby lowering the risk of overfitting on limited data. Model performance was also monitored on a validation set during training to ensure that no significant divergence between training and validation accuracy occurred. Furthermore, consistent performance across multiple evaluation metrics (accuracy, precision, recall, and F1-score) suggests that the models generalize well and do not simply memorize the dataset.

KNN and RNN can be used to detect traffic accidents from CCTV footage, especially when combined with other models like CNN. Therefore, KNN and RNN cannot go individually and have a significantly lower level of accuracy than other classifiers. Furthermore, KNN struggles with high-dimensional image data, as it relies on distance-based similarity measures that become less effective in large feature spaces and are sensitive to noise and irrelevant features. Similarly, RNN is not well-suited for spatial feature extraction, as it is primarily designed for sequential data; when applied directly to image frames without a convolutional backbone, it fails to capture discriminative spatial patterns effectively. Therefore, these classifiers were excluded.

CNN, R-CNN, SVM, and LSTM were identified as having the highest accuracy for performance. The test results revealed the efficiencies of the models as shown in Table 4. Specifically, the results show that CNN achieves the lowest FN (FN = 7) and lowest FP (FP = 4) among all models, indicating its strong

TABLE 4 | Confusion matrix of: (a) CNN, (b) R-CNN, (c) SVM, and (d) LSTM.

Actual	Predicted: Accident	Predicted: Non-accident
(a)		
Accident	True positive (TP) = 364	False negative (FN) = 7
Non-accident	False positive (FP) = 4	True negative (TN) = 416
(b)		
Accident	True positive (TP) = 321	False negative (FN) = 48
Non-accident	False positive (FP) = 27	True negative (TN) = 395
(c)		
Accident	True positive (TP) = 308	False negative (FN) = 61
Non-accident	False positive (FP) = 40	True negative (TN) = 382
(d)		
Accident	True positive (TP) = 201	False negative (FN) = 173
Non-accident	False positive (FP) = 165	True negative (TN) = 252

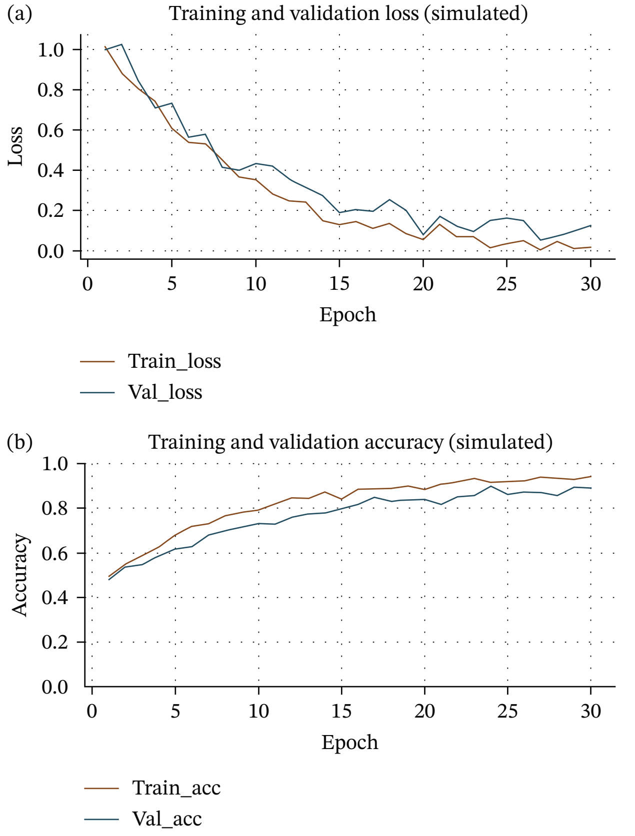

FIGURE 4 | (a) CNN loss on training and validation datasets. (b) CNN accuracy on training and validation datasets.

suitability for safety-critical applications where minimizing missed accident detections and false alarms is essential. In comparison, R-CNN (FN = 48, FP = 27) and SVM (FN = 61, FP = 40) exhibit moderate performance, while LSTM (FN = 173, FP = 165) performs the weakest, with substantially higher error rates in both categories.

TABLE 5 | Confusion metrics parameters of CNN classifier.

Type		Recall	F1measure	Precision	TPR	FPR
Accident	CNN	0.99	0.98	0.98	0.98	0.01
Non-accident		0.98	0.98	0.99	0.99	0.02

4.1 | Assessment of the CNN Classifier

The ability of CNNs to detect features at different scales is useful in identifying accidents that may differ greatly in size and context. Edge devices such as cameras or drones can be used to deploy optimized CNN models, resulting in instant analysis without the need for extensive backend processes. The training and validation accuracy values are closely matched, demonstrating that there is no evidence of overfitting in Figure 4 and confusion metrics parameters as shown in Table 5. In addition, the CNN framework was designed in detail for detecting road accidents as in Figure 5 [31].

4.2 | Assessment of the R-CNN Classifier

The R-CNN classifier that was proposed merges a convolutional layer alongside an LSTM layer. The results from the fitting process are shown in Tables 6 and 7, revealing a relatively small count of negative and FP predictions. This confirms the model's capability to effectively distinguish between frames of traffic accidents and those of non-accidents.

4.3 | Assessment of the SVM Classifier

Evaluation findings of the SVM-based classifier as shown in Tables 8 and 9 illustrate as the classifier is successful in differentiating among frames that feature accidents and those that do not.

4.4 | Assessment of the LSTM Classifier

As shown in Table 10, the evaluation of LSTM classifier indicates that this model is not as effective as CNN, R-CNN, and SVM.

As shown in Table 11, an illustrated summary of the various metrics employed for the study is given, including recall, F1-score, accuracy, precision, TPR and FPR, and also demonstrated these comparisons through bar diagram in Figure 6 both for accident and non-accident. Given the abundance of intricate and sizable datasets from CCTV footages, CNN has performed better in this investigation in terms of accuracy than other classifiers which achieved the highest accuracy 99%.

The observed performance differences among the evaluated models can be attributed to their inherent architectural capabilities in

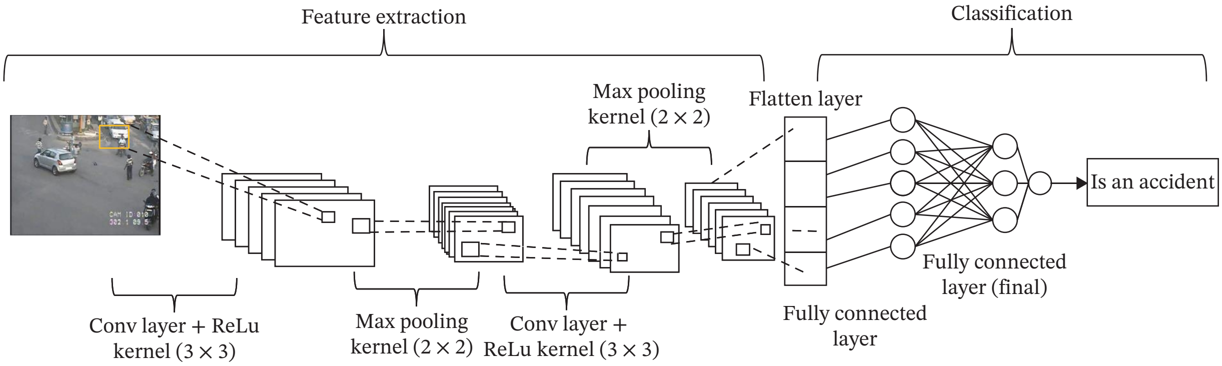

FIGURE 5 | The CNN framework designed for detecting road accidents.

TABLE 6 | Confusion metrics parameters of R-CNN classifier.

Type		Recall	F1-measure	Precision	TPR	FPR
Accident	R-CNN	0.94	0.91	0.89	0.93	0.13
Non-accident		0.87	0.89	0.92	0.86	0.06

TABLE 7 | Classification assessment of the R-CNN classifier.

Assessment	Precision	Recall	F1-score	Support
0	0.92	0.87	0.89	369
1	0.89	0.94	0.91	422
Accuracy	—	—	0.91	791
Macro avg	0.91	0.91	0.91	791
Weighted avg	0.90	0.91	0.91	791

TABLE 8 | Confusion metrics parameters of SVM classifier.

Type		Recall	F1-measure	Precision	TPR	FPR
Accident	SVM	0.91	0.88	0.86	0.91	0.17
Non-accident		0.83	0.86	0.89	0.83	0.09

TABLE 9 | Classification assessment of the SVM classifier.

Assessment	Precision	Recall	F1-score	Support
0	0.89	0.83	0.86	369
1	0.86	0.91	0.88	422
Accuracy	—	—	0.87	791
Macro avg	0.88	0.87	0.87	791
Weighted avg	0.87	0.87	0.87	791

capturing spatial and temporal features. CNN-based models achieve the highest accuracy (up to 99%) due to their ability to learn hierarchical spatial representations, enabling effective detection of visual cues such as vehicle deformation, collision patterns, and scene context in CCTV images.

TABLE 10 | Confusion metrics parameters of LSTM classifier.

Type		Recall	F1-measure	Precision	TPR	FPR
Accident	LSTM	0.60	0.59	0.59	0.60	0.46
Non-accident		0.54	0.54	0.55	0.54	0.40

In contrast, the SVM model exhibits comparatively lower performance, as it relies on handcrafted or shallow features that are less expressive for complex visual patterns. The R-CNN model benefits from region-based object localization, improving detection in cluttered scenes; however, its multistage processing introduces computational overhead and may limit efficiency.

Meanwhile, the standalone LSTM model shows relatively modest performance on image-based data, as it is primarily designed for temporal sequence modeling rather than static feature extraction. The proposed 2D-CNN model outperforms individual approaches by effectively combining spatial feature extraction with temporal dependency learning, making it particularly suitable for static-image analysis. This demonstrates that integrating complementary model strengths is critical for robust and real-world accident detection.

Recall, F1-score, precision, TPR, and FPR for CNN classifier show a significant difference in pick value compared to other classifiers, which is characterized as its best efficiency. In contrast, R-CNN is in the next position after CNN, which depicted better accuracy than SVM and LSTM. However, LSTM is not up to the mark in this study.

Among the evaluated classifiers, the 2D-CNN demonstrated the highest classification performance on the CCTV dataset, achieving an accuracy of 99% for both accident and non-accident categories under the given experimental conditions. Figure 6 provides a visual comparison of classifier behavior across multiple metrics, highlighting the consistently low FP rates achieved by the CNN relative to other models. ROC curves were used to assess the discriminative capability of each classifier across varying decision thresholds. As shown in Figure 7, CNN exhibits a higher AUC, indicating superior class separability on the CCTV dataset.

As illustrated in Table 12, it is evident that, within the constraints of frame-level CCTV analysis and the evaluated dataset size, the

TABLE 11 | Comparisons of F1-score, recall, precision, TPR, and FPR among the models.

Type	Classifiers	Recall	F1-measure	Precision	TPR	FPR
Accident	2D-CNN	0.99	0.98	0.98	0.98	0.01
	R-CNN	0.94	0.91	0.89	0.93	0.13
	SVM	0.91	0.88	0.86	0.91	0.17
	LSTM	0.60	0.59	0.59	0.60	0.46
Non-accident	2D-CNN	0.98	0.98	0.99	0.99	0.02
	R-CNN	0.87	0.89	0.92	0.86	0.06
	SVM	0.83	0.86	0.89	0.83	0.09
	LSTM	0.54	0.54	0.55	0.54	0.40

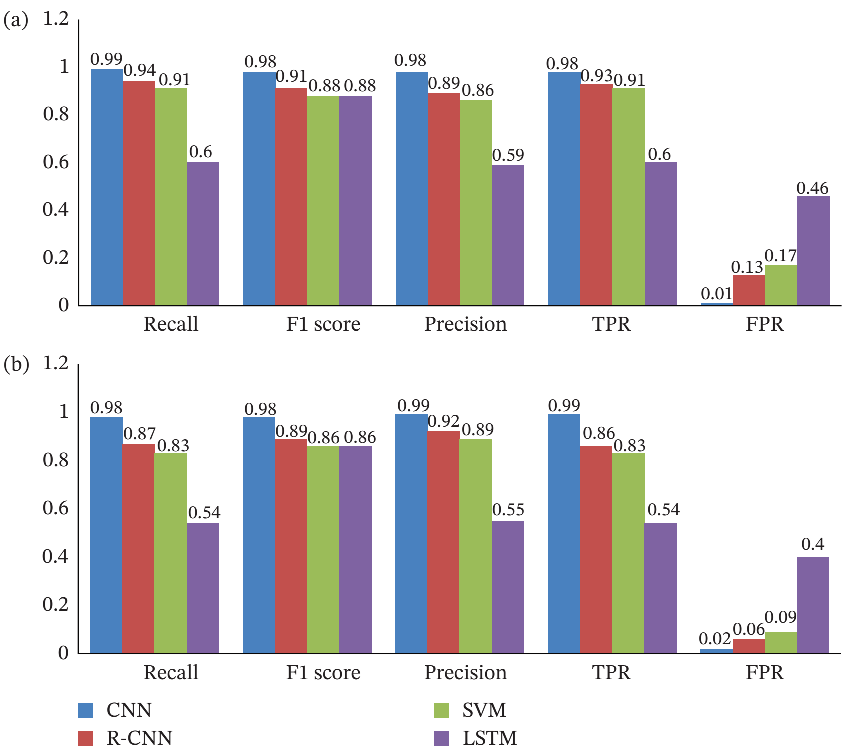

FIGURE 6 | (a) Analysis of classifiers measuring recall, F1-score, precision, TPR, and FPR for accident detection. (b) Analysis of classifiers measuring recall, F1-score, precision, TPR, and FPR for non-accident detection.

2D-CNN demonstrated the most favorable balance of accuracy and F1-score. Traditional machine learning models such as SVM and KNN rely on manually engineered features extracted from input images (e.g., histogram of oriented gradients [HOGs], edge features, or texture descriptors). These models do not inherently capture spatial hierarchies in raw image data.

In contrast, deep learning models such as CNN-based architectures automatically learn hierarchical feature representations directly from raw pixel data, making them more suitable for complex visual tasks such as accident detection. The inferior performance of SVM and KNN can be attributed to their dependance on hand-crafted features, which may fail to capture complex spatial and temporal patterns present in accident scenarios. Additionally, these models struggle with high-dimensional image data and lack the ability to learn deep feature representations. In contrast, CNN-based models effectively extract discriminative features through convolutional layers, leading to significantly higher accuracy.

Frame-based CCTV analysis focuses primarily on spatial cues, dashcam videos introduce temporal dynamics that require sequence modeling. To address this, hybrid architectures combining convolutional and recurrent components were evaluated. There are 4500 dashcam video occurrences in the test set, which are divided into crash (1500) and normal (3000) categories. Integrate VGG16, which is utilized for spatial feature extraction, with LSTM, designed for analyzing temporal sequences as depicted in Figure 8, because this combination demonstrated the highest accuracy when evaluated against dashcam datasets in comparison to other models.

Table 13 demonstrates how to ascertain which classifiers are effective in detecting traffic accidents from dashcam video

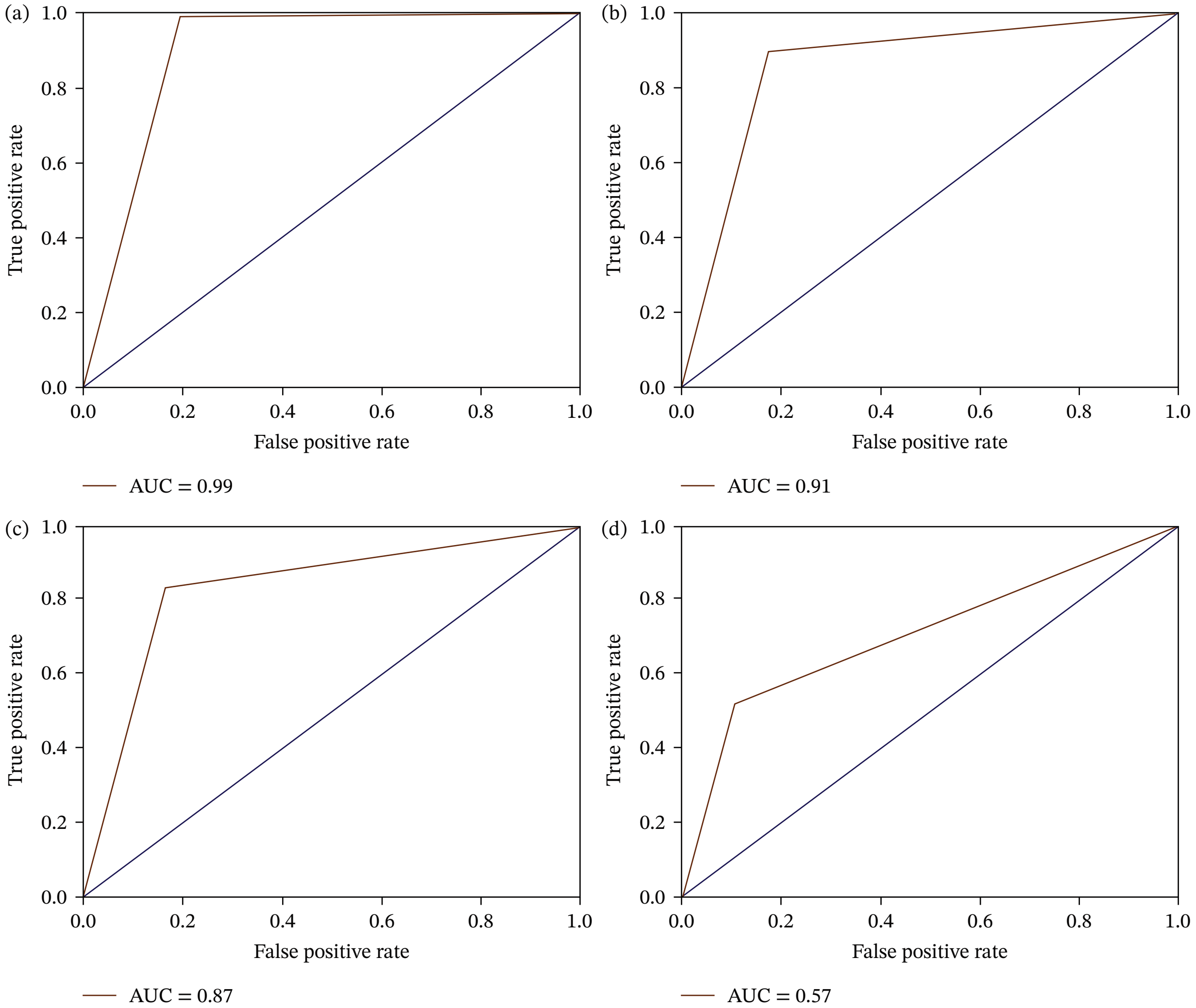

FIGURE 7 | Different classifiers' receiver operating characteristic (ROC) curves. (a) ROC curve for CNN. (b) ROC curve for R-CNN. (c) ROC curve for SVM. (d) ROC curve for LSTM.

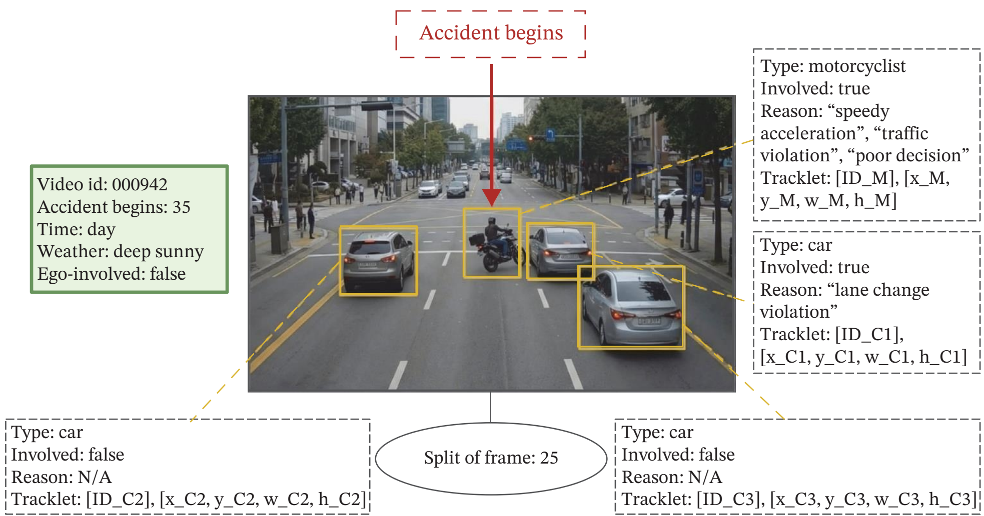

FIGURE 8 | An overview of accident annotations from dashcam video.

datasets. It also demonstrates that standalone general classifiers, including SVM, RNN, KNN, R-CNN, and LSTM, are inferior to the hybrid model in terms of accuracy and performance. Standalone classifiers were included as baselines; however, their limited ability to model temporal dependencies restricts their effectiveness in video-based accident detection tasks. Thus, the evaluation focuses on the three hybrid models depicted in Table 14 through the use of confusion matrices.

TABLE 12 | Comparative analysis of F1-score and accuracy among different classifiers.

Classifiers	Datasets	F1-measure (%)	Accuracy (%)
CNN (2D)	CCTV footage	98	99
R-CNN	CCTV footage	90	91
SVM	CCTV footage	87	87
LSTM	CCTV footage	57	57

TABLE 13 | Comparisons among the classifiers/models.

Classifier/ model	Can handle video (temporal data)?	Typical role in accident detection	Overall suitability for dashcam accident detection
CNN + SVM	Yes	CNN extracts features, SVM classifies accident/normal	Good for dashcam video accident detection
R-CNN + LSTM	Yes	R-CNN detects vehicles, LSTM detects temporal crash	Good for dashcam video accident detection
VGG16 + LSTM	Yes	Spatial + temporal hybrid, simple and powerful	Most favorable trade-off accuracy, interpretability, and simplicity

TABLE 14 | Confusion metrics of the hybrid classifiers/models.

	Predicted	
Actual	Crash	Normal
(a) VGG16 + LSTM		
Accident	True positive (TP) = 2145	False negative (FN) = 13
Non-accident	False positive (FP) = 8	True negative (TN) = 2334
(b) R-CNN + LSTM		
Accident	True positive (TP) = 1844	False negative (FN) = 385
Non-accident	False positive (FP) = 218	True negative (TN) = 2053
(c) CNN + SVM		
Accident	True positive (TP) = 1673	False negative (FN) = 481
Non-accident	False positive (FP) = 388	True negative (TN) = 1958

Table 14 demonstrates that standalone general classifiers, including SVM, RNN, KNN, R-CNN, and LSTM, are inferior to the hybrid model in terms of accuracy and performance. Thus, the evaluation was done only on the three hybrid models depicted in Table 14 through the use of confusion matrices. The test results revealed the efficiencies of the hybrid models: VGG16 + LSTM, R-CNN + LSTM, and CNN + SVM, respectively as depicted in Table 14.

TABLE 15 | Confusion metrics parameters of VGG16 + LSTM.

Type		Recall	F1-measure	Precision	TPR	FPR
Accident	VGG16 + LSTM	1.00	0.99	0.99	1.00	0.006
Non-accident		0.99	0.99	1.00	0.99	0.003

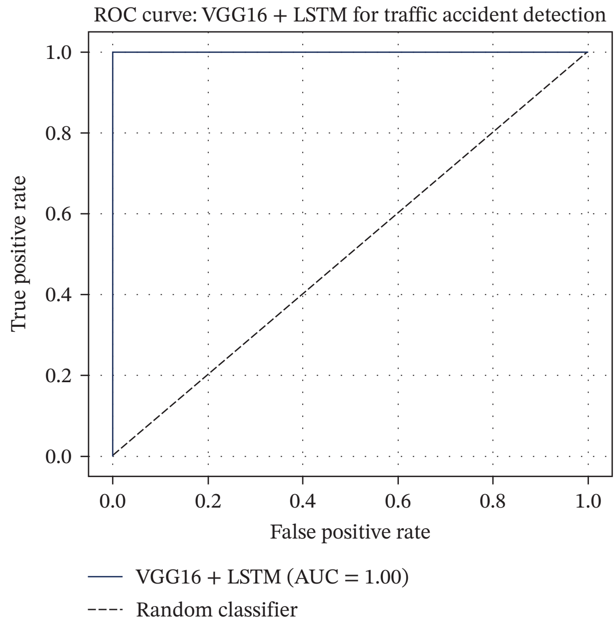

FIGURE 9 | Receiver operating characteristic (ROC) curve of VGG16 + LSTM.

4.5 | Assessment of the VGG16 + LSTM Model

VGG16 extracts spatial features (what is seen in each frame, e.g., cars, roads, and collisions). LSTM learns temporal patterns (how motion evolves over time, e.g., sudden impact, braking, or collision), as shown in Table 15 and the ROC curve in Figure 9.

Although the VGG16 + LSTM model achieved near-perfect performance metrics, these results should be interpreted cautiously given the controlled dataset conditions and the absence of cross-dataset validation. Among the hybrid models, the highest accuracy attained was 99.53, and it also showed an AUC of 1.00 in the ROC curve, as presented in Figure 9. Likewise, the F1-score demonstrates a value of 99%, where the FPR rates are minimal, recorded at 0.006 for crashes and 0.003 for normal situations as illustrated in Table 15.

4.6 | Assessment of the R-CNN + LSTM Model

A notable performance difference is that its accuracy performance is considerably lower than that of the VGG16 + LSTM model, as demonstrated in Table 16.

TABLE 16 | Confusion metrics parameters of R-CNN + LSTM.

Type		Recall	F1-measure	Precision	TPR	FPR
Accident	R-CNN + LSTM	0.90	0.87	0.84	0.90	0.17
Non-accident		0.83	0.87	0.91	0.83	0.10

4.7 | Assessment of the CNN + SVM Model

Regarding accuracy and F1-measure, this hybrid model is significantly less effective compared to the VGG16 + LSTM and R-CNN + LSTM models as depicted in Table 17.

Through the analysis of confusion metrics, it can be determined the classifier that achieved the highest accuracy utilizing datasets derived from dashcam videos. The hybrid model VGG16 + LSTM demonstrates a notable disparity in recall, F1-score, precision, TPR, and FPR when compared to other classifiers/models, which is indicative of its superior efficiency as depicted in Table 18.

The results which encompass F1-score, TPR, FPR, recall, and precision, are illustrated in the flat bar diagrams presented in Figure 10a,b.

As depicted in Table 19, it is evident that the VGG16 + LSTM is the most capable and appropriate classifier/model, achieving the highest F1-score and accuracy of 99% and 99.53%, respectively, in contrast to other hybrid models analyzed in this research study utilizing a dashcam videos dataset.

TABLE 17 | Confusion metrics parameters of CNN + SVM.

Type		Recall	F1-measure	Precision	TPR	FPR
Accident	CNN + SVM	0.83	0.81	0.80	0.83	0.22
Non-accident		0.78	0.79	0.81	0.78	0.17

TABLE 18 | Comparisons of F1-score, recall, precision, TPR, and FPR among the hybrid models.

Type		Recall	F1-score	Precision	TPR	FPR
Accident	VGG16 + LSTM	1.00	0.99	0.99	1.00	0.006
	R-CNN + LSTM	0.90	0.87	0.84	0.90	0.17
	CNN + SVM	0.83	0.81	0.80	0.83	0.22
Non-accident	VGG16 + LSTM	0.99	0.99	1.00	0.99	0.003
	R-CNN + LSTM	0.83	0.87	0.91	0.83	0.10
	CNN + SVM	0.78	0.79	0.81	0.78	0.17

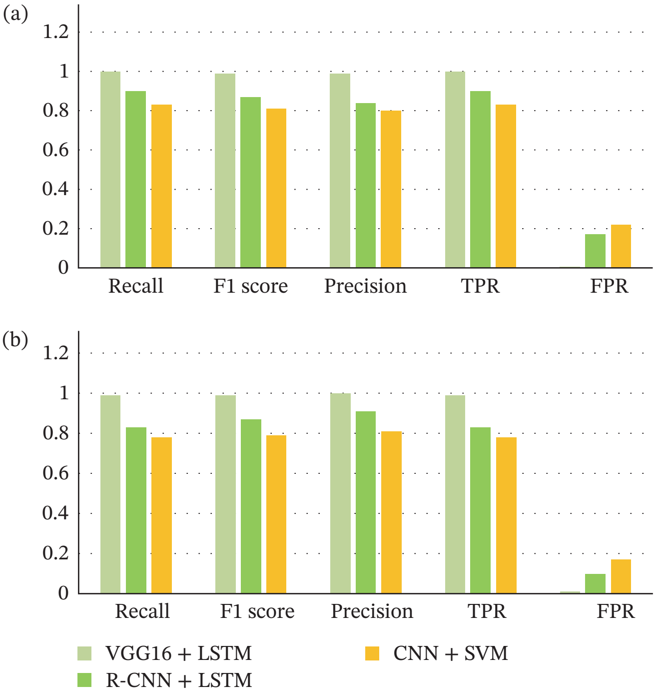

FIGURE 10 | (a) Analysis of hybrid models measuring recall, F1-score, precision, TPR, and FPR for crash detection. (b) Analysis of hybrid models measuring recall, F1-score, precision, TPR, and FPR for normal detection.

TABLE 19 | Comparative analysis of F1-score and accuracy among different hybrid models.

Hybrid models	Datasets	F1-measure (%)	Accuracy (%)
VGG16 + LSTM	Dashcam videos	99	99.53
R-CNN + LSTM	Dashcam videos	87	87
CNN + SVM	Dashcam videos	80	81

5 | Discussion

The discussion of the results relative to prior work that used publicly available CCTV accident datasets from Kaggle. While these studies share a similar data source, differences in data curation, frame sampling, and train-test splits can affect reported performance; therefore, comparisons are indicative rather than strictly equivalent. The two studies have identified that are nearly relevant to our research. Firstly, CNN-based image classifiers have achieved an accuracy of 92.38% when applied to relatively smaller datasets including 300 instances from CCTV footage at a time [32]. On the other hand, utilizing the same dataset of 300 instances in our developed 2D-CNN model, that attained an accuracy of 99%, which was assessed using confusion matrices as illustrated in Table 20.

Secondly, they used 1000 frames from CCTV of Kaggle and got the accuracy of 86% [18]. Conversely, by employing the identical dataset of 1000 frames from Kaggle's CCTV within our established 2D-CNN model, achieved an accuracy of 99.6%, which was evaluated using confusion matrices as demonstrated in Table 21 and ROC curve in Figure 11. The near-ceiling

TABLE 20 | (a), (b): Assessment of the 2D-CNN model using confusion matrices.

Type		Predicted					
		Accident			Non-accident		
(a)							
Actual	Accident	True positive (TP) = 122			False negative (FN) = 3		
	Non-accident	False positive (FP) = 1			True negative (TN) = 174		
(b)							
		Recall	F1-measure	Precision	TPR	FPR	Accuracy
Accident	2D-CNN	0.99	0.98	0.98	0.99	0.02	**99%**
Non-accident		0.98	0.98	0.99	0.98	0.01	

Note: Bold text is used to draw attention to important values, categories, and headers, helping distinguish them clearly from data values.

TABLE 21 | (a), (b): Evaluation of the 2D-CNN model through confusion matrices.

Type		Predicted					
		Accident			Non-accident		
(a)							
Actual	Accident	True positive (TP) = 419			False negative (FN) = 3		
	Non-accident	False positive (FP) = 1			True negative (TN) = 577		
(b)							
		Recall	F1-measure	Precision	TPR	FPR	Accuracy
Accident	2D-CNN	1.0	0.99	0.99	1.0	0.01	**99.6%**
Non-accident		0.99	0.99	1.0	0.99	0.001	

Note: Bold text is used to draw attention to important values, categories, and headers, helping distinguish them clearly from data values.

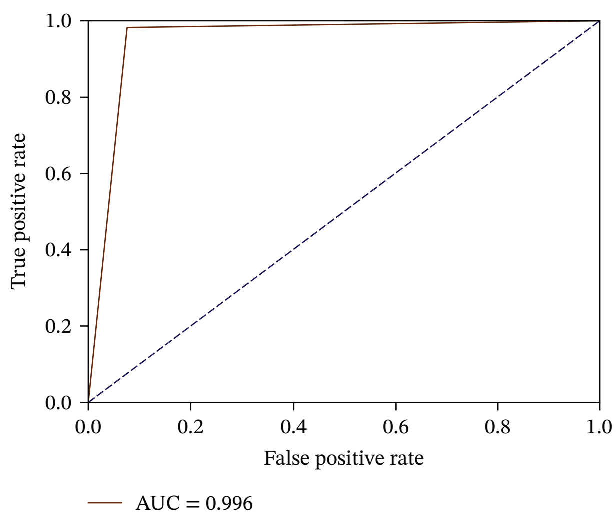

FIGURE 11 | Scrutiny through the ROC curve.

performance observed in some settings may be influenced by dataset characteristics such as limited scene diversity, consistent camera viewpoints, and strong visual cues distinguishing accident versus non-accident frames. These results should therefore be interpreted as performance under the evaluated controlled conditions rather than guaranteed generalization across diverse real-world deployments.

To assess the effectiveness of the proposed VGG16-LSTM hybrid model, its performance was compared with closely related studies that employed dashcam video datasets for accident detection or prediction. One study has been pinpointed that is closely related to our research. In that study, they assessed 1500 real-world driving scenarios of collision prediction dataset from NEXAR dashcam [33], achieved a ROC-AUC of 0.79 and an accuracy of 72% [25]. Although they evaluated accident prediction on dashcam data under a different protocol and model design, their reported AUC provides a useful reference point; however, direct numerical comparison should be interpreted cautiously due to differences in task definition (prediction vs. detection), sampling strategy, and evaluation setup. On the flip side, by leveraging the same dataset of 1500 videos from dashcam in our designed VGG16 + LSTM hybrid model, obtained an accuracy of 98% and ROC-AUC of 1.00, as detailed in Table 22 and the ROC curve represented in Figure 12.

Table 23 summarizes the efficiency and performance of the developed models, 2D-CNN and VGG16 + LSTM, in comparison to other pertinent studies, utilizing datasets derived from CCTV footage and dashcam videos, respectively.

TABLE 22 | (a), (b): Assessment of the VGG16 + LSTM hybrid model.

Type		Predicted	
		Crash	Normal
(a)			
Actual	Accident	True positive (TP) = 562	False negative (FN) = 26
	Non-accident	False positive (FP) = 4	True negative (TN) = 908

(b)		Recall	F1-measure	Precision	TPR	FPR	Accuracy
Accident	VGG16 + LSTM	1.0	0.98	0.97	1.0	0.04	**98%**
Non-accident		0.96	0.97	0.99	0.96	0.004	

Note: Bold text is used to draw attention to important values, categories, and headers, helping distinguish them clearly from data values.

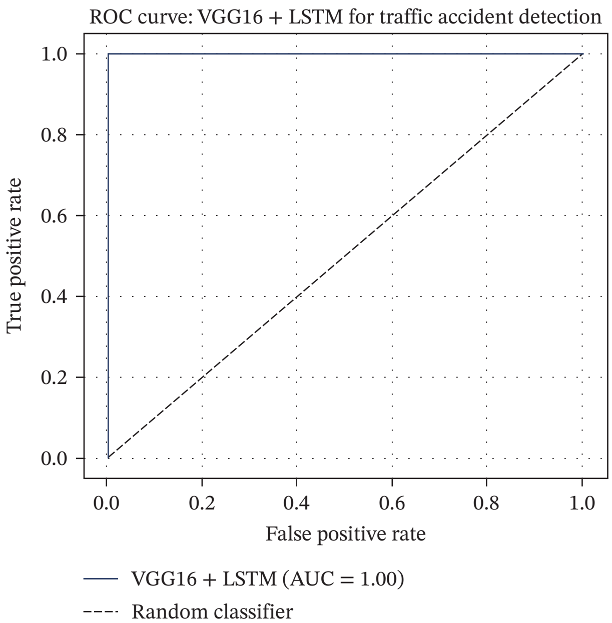

FIGURE 12 | Inspection using the AUC–ROC curve.

The results of the experiment indicate that the 2D-CNN method attained the highest accuracy of 99% when evaluated against datasets derived from CCTV footage, surpassing all other classifiers. Likewise, the hybrid model combining VGG16 + LSTM achieved an accuracy of 99.53% based on datasets sourced from dashcam videos. Despite the high accuracy achieved by the evaluated models (CNN: 99% and VGG16 + LSTM: 99.53%), several important limitations must be considered when interpreting these results. First, the dataset used in this study is relatively limited in size, which may restrict the generalizability of the findings to more diverse real-world scenarios. Small datasets can also lead to inflated performance metrics, particularly when models capture dataset-specific patterns rather than generalizable features.

Second, there is an inherent risk of overfitting, especially for deep learning models trained on limited data. Although regularization techniques such as data augmentation, dropout, and early stopping were employed to mitigate this issue, the possibility of overfitting cannot be entirely eliminated. Third, the study relies on a fixed train–validation–test split rather than a more robust validation strategy such as k-fold cross-validation. While care was taken to prevent data leakage by ensuring no overlap between training, validation, and test samples, k-fold cross-validation would provide a more reliable estimate of model performance across different data partitions.

Hybrid models such as VGG16–LSTM, while achieving higher accuracy in video-based detection, incur increased computational cost and latency due to sequential temporal processing. 2D-CNN models can be efficiently deployed on edge devices with real-time performance, whereas the VGG16–LSTM hybrid, despite its superior accuracy, incurs higher computational cost and latency, making it less suitable for strict real-time deployment without hardware acceleration or model optimization.

TABLE 23 | Outline of the efficiency and performance metrics of the developed models.

Existing works	Dataset used	Classifier/model	F1-measure (%)	Accuracy (%)
[18]	CCTV footages	1D-CNN	84	86
[33]	CCTV Footages	CNN with LSTM	89.35	92.38
[25]	Dashcam videos	CNN with Efficient-Net	71	72
[34]	4677 videos from You-Tube	LSTM autoencoder	78.58	76
Our developed model	CCTV footages	2D-CNN	98	99
Our developed hybrid Model	Dashcam videos	VGG16 + LSTM	99	99.53

Note: Reported results across studies may not be directly comparable due to differences in datasets, annotation criteria, splits, and evaluation protocols.

Future work should incorporate larger and more diverse datasets, along with cross-validation techniques, to further validate the robustness and generalizability of the proposed models. Specifically, we now analyze the inference efficiency of the evaluated models, highlighting that lightweight architectures such as the 2D-CNN are more suitable for real-time CCTV-based deployment due to lower computational overhead and faster processing times.

6 | Conclusion

This study demonstrates that model effectiveness in accident detection is closely aligned with the nature of the input data. The 2D-CNN model performs strongly on CCTV imagery, indicating that spatial features alone can be sufficient for detecting static accident patterns. In contrast, the superior performance of the hybrid VGG16 + LSTM model on dashcam videos highlights the importance of temporal modeling for capturing dynamic accident events. These findings suggest that selecting an appropriate architecture should depend on whether the task involves static or sequential visual data.

However, the study is limited by the relatively small dataset size and the use of controlled evaluation settings, which may not fully reflect real-world variability. The absence of cross-dataset validation also restricts the generalisability of the results. Future work will focus on validating the models across diverse datasets, incorporating event-level detection to better capture temporal accident dynamics, and improving robustness under varying environmental conditions such as lighting, weather, and camera viewpoints. Additionally, exploring lightweight architectures for real-time deployment and evaluating computational efficiency will be important for practical intelligent transportation system applications.

Author Contributions

Syed Jamaluddin Ahmad: conceptualization, methodology, validation, writing. **Hamimah Ujir and Irwandi Hipiny:** methodology, supervision, review and editing. **Syahrul Nizam Junaini:** supervision, review and editing, visualization.

Acknowledgments

The authors gratefully acknowledge the support of the University Malaysia Sarawak in the preparation of this manuscript.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. H. Liu, "Road Traffic Accident Risk Prediction Based on Random Forest Model," *Theoretical and Natural Science* 134, no. 1 (2026): 62–72.

2. Z. Zhou, X. Dong, Z. Li, K. Yu, C. Ding, and Y. Yang, "Spatio-Temporal Feature Encoding for Traffic Accident Detection in VANET Environment," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 10 (2022): 19772–19781.

3. Z. Xi, X. Liu, Y. Liu, Y. Cai, and Y. Zheng, "Integrating Generative Adversarial Networks and Convolutional Neural Networks for Enhanced Traffic Accidents Detection and Analysis," *IEEE Transactions on Intelligent Transportation Systems* (2025).

4. P. Chen, Z. Zhang, Y. Huang, L. Dai, and H. Hu, "Risk Assessment of Marine Accidents With Fuzzy Bayesian Networks and Causal Analysis," 228 (2022).

5. V. V. Anusuy, B. Jasmith, P. Sugasini, and M. J. Dharshini, "An IoT Based Accident Detection and Rescue System," *Iternational Research Journal on Advanced Engineering and Management (IRJAEM* 1, no. 1 (2023): 01–12.

6. B. Hildebrand, M. Baza, T. Salaman, et al., "A Comprehensive Review on Blockchains for Internet of Vehicles: Challenges and Directions," *Computer Science Review* 48 (2023): 100547.

7. S. P. Shubham, M. Kumar, R. Kishore, and S. Jain, "A Survey on IoT Based Automatic Road Accident Detection," in *5th International Conference* (2021): 1–7

8. G. Ahmad and M. Fareed, "Smart Contract Implementation Using Blockchain IoV for Vehicle Accident Investigation," *Azerbaijan Journal of High Performance Computing* 6, no. 1 (2023): 77–182.

9. M. I. B. Ahmed, R. Zaghdoud, M. S. Ahmed, et al., "A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents," *Big Data and Cognitive Computing* 7, no. 1 (2023): 22.

10. R. Phursule, D. Lal, S. Waghere, M. A. Mughini, S. Ransubhe, and C. Shiralkar, "Enhanching Traffic Flow U = using Computer Vision Based-Dynamic Traffic Light Control and Lane Management," *International Journal on Recent and Innovation Trends in Computing and Communication* 11 (2023).

11. M. Hasan, A. A. Nafees, and H. D. Mahin, "Application of Machine Learning in Intelligent Transport Systems: A Comprehensive Review and Bibliometric Analysis," *Discover Civil Engineering* 2 (2025): 98.

12. S. Robles-Serrano, G. Sanchez-Torres, and J. Branch-Bedoya, "Automatic Detection of Traffic Accidents From Video Using Deep Learning Techniques," *Computers* 10, no. 11 (2021): 148.

13. D. K. Yadav, Renu, Ankita, and I. Anjum, "Accident Detection Using Deep Learning," *2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (2020): 232–235.

14. S. Alkheder, M. Taamneh, and S. Taamneh, "Deep Learning Approach For Severity Prediction of Traffic Accident Using an Artificial Neural Network," 36, no. 1 (2017).

15. E. I. Liu, W. Mogh, and Q. Mesh, "Drone-Assisted Intelligent Transportation System Based On Edge Computing and Deep Learning Techniques," 2015, https://scholar.google.com.my/scholar/Liu/543201.

16. V. Adewopo, N. Elsayed, Z. ElSayed, M. Ozer, V. Wangia-Anderson, and A. Abdelgawad, "AI on the Road: A Comprehensive Analysis of Traffic Accidents and Accident Detection System in Smart Cities," arXiv: 2307.12128v1 [cs.CV] (2023).

17. B. K. M, A. Basit, K. MB, G. R, and K. SM, "Road Accident Detection Using Machine Learning," in *International Conference on System, Computation, Automation and Networking (ICSCAN)* (2021): 1–5

18. S. Roy, "Road Accident Detection Using Machine Learning," *International Journal for Research in Applied Science and Engineering Technology* 12, no. 6 (2024): 1277–1281.

19. S. Sharma, S. Harit, and J. Kaur, "Traffic Accident Detection Using Machine Learning Algorithms," in *Proceedings of Third International Conference on Sustainable Computing*, (Springer Nature, 2022): 501–507.

20. P. Mahendhar, R. Nihal, and P. Vikas, "Artificial Intelligence for Accident Detection and Response," *Journal of Emerging Technologies and Innovative Research (JETIR)* 6, no. 3 (2019).

21. R. Desai, A. Jadhav, S. Sawant, and N. Thakur, "Accident Detection Using ML and AI Techniques," Semantic Scholar. Computer Science, Engineering. Corpus ID: 252277099 (2022).

22. T. Tamagusko, M. G. Correia, M. A. Huynh, and A. Ferreira, "Deep Learning Applied to Road Accident Detection With Transfer Learning and Synthetic Images," *Transportation Research Procedia* 64 (2022).

23. F. Ali, A. Ali, M. Imran, R. A. Naqvi, M. H. Siddiqi, and K.-S. Kwak, "Traffic Accident Detection and Condition Analysis Based on Social Networking Data," *Accident Analysis & Prevention* 151 (2021).

24. D. Hadi Nassar and J. Mustafa Al-Tuwaijari, "A Review of Vehicle Accident Detection and Notification Systems Based on Machine Learning Techniques," *Academic Science Journal* 2, no. 2 (2024): 105–126.

25. L. Mondonico and K. Yang, "DashGuard: Hierarchical Attention for Dashcam Video Accident Detection," "2022," (Stanford University

26. "Data Arrays Pertaining to Traffic Crash," https://www.kaggle.com: cctv recordings traffic crash data arrays.

27. Road Accident dataset from DAD, https://drive.google.com/drive/folders/1Rx4LCo-9AbAdPw5Zh7wpKhMKLabZ5oA8.

28. "Keras; High-level Api: TensorFlow Platform," 2015, Repository. github.com/keras-team/keras. Python Platform Cross-platform https://www.tensorflow.org/guide/keras.

29. "TensorFlow; Software Library: -Machine Learning and Artificial Intelligence," Python, C++, CUDA. Platform: Linux, macOS, Windows, Android, JavaScript. Machine learning library: Apache 2.0 https://www.tensorflow.org.

30. F. Qanouni, H. E. Massari, N. Gherabi, and M. E. Badaoui, "Road Accident Detection Using SVM and Learning: A Comparative Study," *International Journal of Advanced Computer Science and Applications* 15, no. 5 (2024).

31. https://www.researchgate.net/figure/The-proposed-CNN-model-for-road-accidents-detection_fig1_381100701.

32. S. Ghosh, S. J. Sunny, and R. Roney, "Accident Detection Using Convolutional Neural Networks," in *2019 International Conference on Data Science and Communication (IconDSC)* (2019): 1–6

33. NEXAR datasets collisions view, https://data.getnexar.com/solution/autonomous-vehicles/.

34. K. Pawar and V. Attar, "Deep Learning-Based Detection and Localization of Road Accidents From Traffic Surveillance," *ICT Express* 8, no. 3 (2022): 379–387.