Accident Detection & Alert System using Deep Learning Techniques

|  |  |  |
| --- | --- | --- |
| CH. Vamshi*B. Tech**School of Engineering* Computer Science-(AI&ML) Malla Reddy University, India | P. Vamshi*B. Tech**School of Engineering* Computer Science-(AI&ML) Malla Reddy University, India | D. Vamshi Krishna*B. Tech**School of Engineering* Computer Science-(AI&ML) Malla Reddy University, India |
| G. Vara Prasad*B. Tech**School of Engineering*Computer Science-(AI&ML) Malla Reddy University, India | P. VarshiniB. TechSchool of EngineeringComputer Science-(AI&ML) Malla Reddy University, India | Guide: Dr. Syed Saba Raoof Assistant ProfessorSchool of EngineeringComputer Science-(AI&ML) Malla Reddy University, India |

# ABSTRACT

The rising number of vehicle accidents in cities and highways has highlighted the necessity for an automated accident detection and alert system to enable quick emergency response. This research investigates the use of deep learning methods for CCTV footage-based real-time accident detection. In order to improve public safety and reduce response times, the goal is to create a system that can precisely identify incidents and notify emergency services right away. The system uses frame preprocessing methods such frame differencing, optical flow analysis, and background subtraction to analyze real-time video streams. To differentiate accidents from normal traffic flows, it makes use of anomaly detection algorithms and convolutional neural networks for object detection. In particular, Threshold- based motion intensity computation handles sequential anomaly detection, whereas Faster R-CNN is used for accident localization. To guarantee reliable performance in real-world situations, the system is tuned to operate in a variety of lighting settings, weather effects, and camera angles. The technology automatically sends out alerts to emergency services and the appropriate authorities as soon as an accident is identified. Metrics like detection accuracy, false positive rate, response time, and computing efficiency are used to assess the system's performance. Results from experiments show that deep learning-based accident detection methods can identify accidents accurately and quickly, which makes them ideal for public safety and traffic monitoring.

# INTRODUCTION

Road accidents are a serious worldwide issue because they result in a large number of fatalities, serious injuries, and property damage. An automated accident detection and alert system is urgently needed, as evidenced by the rising incidence of accidents on roads and in urban areas. Conventional accident reporting techniques depend on manual reporting or eyewitnesses, which frequently causes delays in emergency response. Since prompt medical attention is essential for preserving lives and lessening the severity of injuries, these delays may be fatal. A real-time accident detection system that can automatically detect accidents and alert emergency services without human intervention is therefore desperately needed. The goal of this project is to create an intelligent accident detection system that analyses live CCTV data using computer vision and deep learning algorithms. Several video processing

steps are included in the suggested approach. First, preprocessing methods such frame differencing, optical flow analysis, and background subtraction are applied to raw video frames. These methods aid in separating moving objects from their surroundings, spotting sudden shifts in motion patterns, and spotting odd occurrences. Convolutional neural networks are then used to analyse the pre-processed frames in order to identify and categorize items, including cars and pedestrians. The method uses Faster R-CNN for accident localization to increase accuracy even more. A strong object detection model that can recognize accident-prone regions in a frame is Faster R-CNN. Additionally, sequential anomaly detection uses a threshold-based motion intensity calculation. Making sure the system is resilient in a variety of environmental circumstances is one of the main obstacles in accident detection. Different lighting conditions, weather affects, and camera angles are all factors that affect how traffic surveillance cameras function. Through the use of computer vision and deep learning, this project seeks to create a real-time accident detection system that can instantaneously alert emergency services to accidents by analysing CCTV data. It enhances public safety by reducing response time and improving traffic monitoring efficiency.

In order to identify cars and other objects in the video frames, the project incorporates a Convolutional Neural Network (CNN)-based model. In order to help the system concentrate on regions where accidents happen, Faster R-CNN is also used for accident localization. The system can be deployed in the real world since it is designed to function well in a variety of lighting scenarios, weather situations, and camera angles.

# LITERATURE REVIEW

Road accidents continue to be a major cause of property damage, fatalities, and serious injuries worldwide. Conventional accident detection techniques that depend on physical intervention and eyewitness accounts frequently cause delays in emergency responses, which might have a fatal consequence. By examining real-time CCTV footage, automated accident detection systems that make use of AI and computer vision have surfaced to improve reaction times. Early approaches had trouble with environmental elements like weather and camera angles and relied on motion detection techniques like background subtraction and optical flow. Accuracy is greatly increased by modern systems using deep learning methods like CNNs and Faster R-CNN. Reducing false positives and attaining real-time performance are still difficult tasks, nevertheless.

The use of surveillance film for accident detection has been extensively studied. Motion tracking and rule-based techniques were the mainstays of early techniques, but they were ineffective in dynamic settings. SVMs and other machine learning models enhanced performance but lacked practical applicability. Although they have computational limits, deep learning techniques like CNNs, YOLO, and Transformers improved detection accuracy. Reliability was further enhanced by sophisticated methods like autoencoder anomaly detection and CNN-GNN combinations. High false positive rates and difficulties differentiating between accidents and typical traffic patterns persist, nevertheless. These gaps are still being filled by IoT-enabled systems and hybrid models, which aim to improve accident detection.

#  METHODOLOGY

## Existing System

Significant flaws in the current accident detection systems cause delays in emergency responses and jeopardize public safety. Due to human error and slowness, manual CCTV footage monitoring frequently misses situations because of distractions or weariness. Helpline numbers for eyewitness reporting are unreliable, particularly in rural locations or during off-peak hours. GPS-based systems are useless for general traffic surveillance and pedestrian occurrences because they can only identify accidents in vehicles that are equipped with them. High false positive rates result from traditional computer vision techniques that rely on rule-based algorithms and struggle with real-world issues including uneven camera angles, bad illumination, and changing weather. These systems are also unsuitable for large-scale deployment due to their inability to interpret data in real-time and their inability to adapt to various contexts. These shortcomings highlight the need for a sophisticated, automated system that uses computer vision and deep learning to detect accidents accurately in real time, guaranteeing quicker emergency responses and increased road safety.

## Proposed System

To overcome the drawbacks of conventional techniques, the suggested Real-Time Accident Detection and Alert System makes use of deep learning. It analyzes real-time CCTV data using CNNs, using methods like optical flow and background subtraction to identify abrupt motion changes. Motion intensity analysis detects anomalous patterns, and faster R- CNN guarantees accurate accident location. The solution reduces response times by using the Twilio API to trigger automatic alerts and functions effectively in a variety of scenarios. It smoothly analyzes video data on common hardware thanks to TensorFlow's AUTOTUNE feature, which optimizes performance in real-time. System monitoring is improved with a user-friendly GUI that shows logs and detection results. Response time, accuracy, and false positive rate are used to assess performance. By doing away with the need for human reporting, the system increases detection reliability. Road safety and response efficiency are greatly increased by automating accident detection and emergency warnings, providing a scalable and flexible solution for a variety of traffic situations.

##  Modules

* + 1. **Accident Detection & Alert System Modules 1 Video Processing Module(Numpy and OpenCV):**

In order to detect accidents, prepare video frames for analysis, and make sure that only pertinent data is given to the deep learning model, the video processing module is crucial. While NumPy manages numerical tasks like frame normalization and motion intensity calculation, OpenCV is used for frame extraction, grayscale conversion, and motion detection. Detecting motion changes to indicate potential accidents, extracting frames, and converting them to grayscale to decrease complexity are important functions. By enabling real-time processing and lowering computing burden, the module improves efficiency. It improves accident identification by precisely detecting motion irregularities.

## Deep Learning Model (TensorFlow & Keras):

By identifying accident patterns in video frames, the deep learning model drives the accident detection system. Developed with TensorFlow and Keras, it uses dense layers for classification, more convolutional layers for fine-tuning processing, and MobileNetV2 for feature extraction. Feature refinement, likelihood scoring, and transfer learning-based accident detection are among the fundamental features. It effectively detects mishaps in real time, guaranteeing precise forecasts. Although MobileNetV2 improves speed and accuracy, a variety of training datasets are necessary for the model to function well. Despite challenges such as potential misgeneralization and data dependency, this module remains critical to automating accident detection.

## Dataset Management & Training (TensorFlow, Pandas, Matplotlib):

Using Matplotlib, Pandas, and TensorFlow's ImageDataset, the Dataset Management & Training module arranges, preprocesses, and displays the accident detection model's training. Although it demands enormous datasets and processing resources, it improves efficiency, tracks model performance, and optimizes data handling.

## SMS Alert System (Twilio API):

The SMS Alert System notifies emergency services and pre- designated contacts in real-time during accidents via Twilio's REST API. Although it requires internet connectivity and is expensive for widespread use, it guarantees dependable message delivery, provides multi-recipient alerts, and improves reaction efficiency.

## Graphical User Interface (Tkinter and PIL):

Non-technical users may simply interact with the accident detection system thanks to the GUI module, which was constructed using Tkinter and PIL. It enables frame display, accident detection initiation, and video selection. Despite being easy to use and straightforward, it has limited customization possibilities and performance issues with large files.

# *5* ARCHITECTURE

To identify accidents from video feeds and alert emergency services, the technology combines deep learning and computer vision.

**Input Frames:** Surveillance camera input frames are preprocessed for precise analysis. **Convolutional Layers:** Gather visual information to identify trends.

**MobileNetV2:** A lightweight backbone for effective feature extraction is MobileNetV2. **Conv2D + Flatten:** Enhances characteristics and gets data ready for categorization. **Dense Layer:** identifies patterns unique to accidents to classify frames.

**Prediction layer:** The prediction layer generates final accident forecasts and, if required, sounds an alarm.

## 5.1 Methods & Algorithms:

**Algorithm 1: Working of CNNs:**

Step 1. Data Collection and Preprocessing

 Gather video data from CCTV, dashcams, or roadside cameras.

 Extract frames, resize, normalize, and reduce noise for model input.

Step 2. Input Layer

 Receives preprocessed frames as input.

 Converts images into numerical arrays for processing.

Step 3. Convolutional Layers (Feature Extraction)

 Applies convolution filters to detect edges, shapes, and motion patterns.

 Extracts low-level and high-level features, identifying potential crash indicators.

Step 4. Activation Function (ReLU)

 Introduces non-linearity to the feature maps.

 Helps detect complex patterns in accident scenarios. Step 5. Pooling Layer (Dimensionality Reduction)

 Downsamples feature maps, reducing computational load.

 Retains essential accident-related features. Step 6. Additional Convolution + Pooling Layers

 Refines extracted features with deeper convolutional operations.

 Enhances ability to recognize intricate accident patterns. Step 7. Flattening Layer

 Converts 2D feature maps into a 1D vector.

 Prepares data for classification by dense layers. Step 8. Dense (Fully Connected) Layers

 Analyzes flattened features to identify accident-specific patterns.

 Combines extracted information for final decision- making.

Step 9. Output/Prediction Layer

 Classifies frame as “Accident Detected” or “No Accident.”

 Assigns confidence scores to predictions. Step 10. Decision and Alert Trigger

 If accident detected, saves frame and triggers alert.

 Sends notification to relevant authorities for immediate response.

## Algorithm 2: Working of MobileNetV2

Step 1. Initialize MobileNetV2 Model

 Load pre-trained MobileNetV2 model with ImageNet weights.

 Fine-tune model for accident detection by modifying output layers.

Step 2. Input Frame Preprocessing

 Resize frames to required dimensions (e.g., 224x224 pixels).

 Normalize pixel values for consistency and better model performance.

Step 3. Feature Extraction Using Depth wise Convolutions

 Apply depthwise separable convolutions to extract low and high-level features.

 Detect patterns like vehicle orientation, sudden movements, or debris.

Step 4. Inverted Residuals and Linear Bottlenecks

 Compress extracted features efficiently while preserving essential information.

 Ensure lightweight architecture to enable real-time processing.

Step 5. Pass Features to Additional Conv2D Layers

 Refine extracted features by identifying subtle accident- specific patterns.

 Enhance detection accuracy before flattening feature maps.

Step 6. Flatten and Classify Frames

 Flatten feature maps into a 1D vector.

 Use dense layers to classify frames as either **Accident** or

## No Accident.

Step 7. Generate Prediction with Confidence Score

 Assign a confidence score to each frame indicating accident likelihood.

 Trigger alert if confidence exceeds predefined threshold.

Step 8. Store Frame and Trigger Alert

 Save detected accident frame and metadata.Send alert to emergency services and relevant stakeholders.

## 5.2 UML Diagram

An UML diagram representing high level system design of the product



## 5.3 Dataset Selection:

The dataset includes surveillance films and CCTV clips that are used to train and assess the accident detection model. The model can differentiate between typical and accident scenarios since it incorporates a variety of traffic circumstances. To improve motion-based recognition, frames are pre-processed utilizing optical flow analysis, frame differencing, and background subtraction. To differentiate accident-related events like collisions and odd car behavior from regular traffic flow, each frame is labeled. By using structured labeling, false positives are decreased and model precision is increased. Reliable performance under a variety of circumstances is guaranteed by the balanced dataset. Future developments could integrate synthetic simulations, dashcam footage, and drone monitoring to improve adaptation to various traffic situations and, eventually, the accuracy of real-world accident detection.

# *6* EXPERIMENTAL RESULTS

## 6.1 Dataset Description

The dataset utilized in our study comprises CCTV footage and surveillance video clips specifically selected to train and evaluate our accident detection model. These videos contain various traffic conditions, including normal traffic flow and accident scenarios, enabling the deep learning model to learn distinguishing patterns effectively. The dataset is processed into video frames and further pre-processed using frame differencing, background subtraction, and optical flow analysis to enhance motion-based accident recognition.

Each frame is labeled based on whether it contains an accident or not, ensuring a clear distinction between normal and critical events. The labeling process follows a structured approach where frames depicting collisions, abrupt stops, unusual vehicle behavior, and crowd formation are marked as accident-related, while regular traffic flow is categorized as non-accident. This structured annotation helps improve the model’s ability to detect accidents with high precision and minimal false positives.

The dataset serves as a fundamental resource for developing deep learning-based accident detection models. It allows for robust model training and validation, ensuring high accuracy in real-world traffic monitoring applications. Additionally, the dataset is balanced to prevent bias toward any particular class, promoting reliable performance across varied lighting conditions, weather scenarios, and traffic densities. Future extensions of the dataset may include dashcam footage, drone surveillance, and synthetic accident simulations to enhance the model’s adaptability to diverse traffic environments.

## 6.2. Experimental Design

The experimental design for the Accident Detection and Alert System using Deep Learning involves data collection, model training, system implementation, and performance evaluation. Accident and non-accident videos are preprocessed using OpenCV, and frames are fed into a MobileNetV2-based deep learning model for classification. The trained model detects accidents in real-time from video feeds. If an accident is detected, an SMS alert is sent via Twilio API, and results are logged in an SQLite database. A Tkinter GUI provides user interaction. System performance is evaluated using accuracy, precision, recall, and F1-score to ensure reliable accident detection and alerts.

## Data Collection & Preprocessing

* + - Accident and non-accident videos are collected from publicly available sources.
		- Frames are extracted using OpenCV, resized, normalized, and augmented for better model generalization.

## Model Training & Selection

* + - A MobileNetV2-based deep learning model is used for accident detection.
		- The dataset is split into training (80%) and testing (20%), and the model is trained using Adam optimizer with cross-entropy loss.
		- Performance is evaluated using accuracy, precision, recall, and F1-score.

## System Implementation

* + - **Video Processing Module**: Captures video frames in real-time.
		- **Accident Detection Module**: Processes frames and classifies them as accident or non-accident.
		- **Alert System (Twilio API)**: Sends SMS notifications when an accident is detected.
		- **Database (SQLite)**: Logs accident events with timestamps and images.
		- **GUI (Tkinter)**: Provides a user-friendly interface for uploading videos and displaying results.



**Figure 6.5.1 Model Accuracy**

****

**Figure 6.5.2 Model Loss**

* 1. **Performance Evaluation**
		+ The system is tested on real-world accident footage.
		+ Accuracy metrics are used to assess reliability.



**Figure 6.6.1 Training Loss & Accuracy**

****

**Figure 6.6.2 Validation Loss & Accuracy**

**6.7. Evaluation Metrices**

* **Precision :** Measures the proportion of correctly predicted accident cases out of all predicted accidents. A higher precision means fewer false positives, ensuring the system does not generate unnecessary accident alerts.
* **Recall :** Quantifies the proportion of actual accidents correctly identified. A higher recall ensures fewer missed accident detections, which is critical for emergency response.
* **F1-Score :** Provides a harmonic mean of precision and recall, especially useful for imbalanced datasets. Ensures a balance between false positives and false negatives.
* **ROC-AUC :** Measures the model’s ability to distinguish between accident and non-accident cases. A higher AUC means the model has a strong ability to differentiate between accident and non- accident cases, making it highly reliable for real- time detection

# *7* CONCLUSION *9.* REFERENCES

Our work, Real-Time Accident Detection and Alert System Using Advanced Deep Learning Techniques, effectively applies deep learning models to video footage to create an automated accident detection system. The system processes video using OpenCV (cv2), which includes motion detection using frame differencing, frame extraction, and grayscale conversion. Pandas helps with handling and processing detection data, while NumPy facilitates numerical calculations and frame differencing. TensorFlow and Keras are used to build the model for accident detection, which uses CNN and MobileNetV2 layers for feature extraction and classification. A user-friendly interface for uploading and analyzing video files is offered by the Tkinter-based GUI, which also shows processed frames in real time along with the outcomes of accident detection. Accuracy and loss curves are two more model training measures that are visualized using Matplotlib.

Twilio API connection allows automated SMS alerts to emergency contacts upon accident detection, guaranteeing prompt action. To help with performance evaluation, the system records response times, false positive rates, and detection accuracy. Although the model is capable of accurately identifying accidents under a variety of circumstances, issues like false positives and computational efficiency still need to be addressed. To further maximize response efficiency, future improvements might incorporate GPS-based accident location monitoring, real-time CCTV integration, and adjustable sensitivity settings for detection. All things considered, our technology offers a workable and expandable solution for emergency warning and real-time accident detection, enhancing traffic safety. All things considered, our technology offers a workable, scalable, and effective way to detect accidents and send out emergency alerts in real time, which helps to improve traffic management, road safety, and emergency response times.

# *8* FUTURE WORK

Future developments appear potential with the Real-Time Accident Detection and Alert System. Real-time notifications from integration with traffic monitoring and smart city infrastructure can speed up emergency response times. Remote places could benefit from the addition of GPS-based accident reporting, which could identify accident locations. In order to avoid secondary collisions, alerts might also warn drivers in the vicinity and provide alternate routes. In complicated situations, improving machine learning models with transformers and hybrid architectures can increase detection accuracy. By integrating edge computing, local video feed analysis would allow for speedier processing. Communication efficiency can be increased by extending alert systems beyond SMS to include voice calls and app notifications. The system's reach might include public surveillance and industrial safety, making it an essential part of global intelligent transportation systems.

1. Ren, S., He, K., Girshick, R., & Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv preprint arXiv:1506.01497.
2. Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25.
3. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.
4. He, K., Zhang, X., Ren, S., & Sun, J., 2016. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

pp. 770-778.

1. OpenCV Developers, 2023. OpenCV: Open Source Computer Vision Library Documentation. Retrieved from https://docs.opencv.org/
2. Chollet, F., et al., 2015. Keras: Deep Learning Library for TensorFlow. Retrieved from <https://keras.io/>
3. Abadi, M., et al., 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv preprint arXiv:1603.04467.
4. Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp. 2825-2830.
5. Twilio API Documentation, 2024. Automated SMS Alerts for Emergency Notifications. Retrieved from https:/[/www.twilio.com/docs/sms](http://www.twilio.com/docs/sms)
6. Bishop, C. M., 2006. Pattern Recognition and Machine Learning. Springer.
7. Chandrasekaran, R., et al., 2021. Accident Detection and Alert System Using Deep Learning and IoT. IEEE Access.
8. Kumar, S., et al., 2020. Real-Time Traffic Accident Detection Using Convolutional Neural Networks. Journal of Intelligent Transportation Systems.
9. Rosenfeld, A., Zemel, R., & Tsotsos, J. K., 2018. Object Tracking and Scene Understanding with Deep Learning. arXiv preprint arXiv:1806.08728.
10. Kingma, D. P., & Ba, J., 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
11. Girshick, R., 2015. Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1440- 1448.