International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 02 | Feb – 2025 DOI: 10.55041/ISJEM02270
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

# Smart City Transportation Deep Learning Ensemble Approach for Traffic Accident Detection

**Jangam Bhargavi[1], Bharatha Ashwin[2], Lavishetty Tagore[3], MD Rehan[4]**
Assistant Professor of Department of CSE(AI&ML) of ACE Engineering College [1] Students of Department CSE(AI&ML) of ACE Engineering College [2,3,4]

## Abstract
This project proposes a real-time traffic accident detection system for smart cities using a deep learning approach. The system aims to detect accidents like rear-end collisions, T-bone crashes, and frontal impacts by processing video sequences from traffic surveillance cameras. This system will utilize Convolutional Neural Networks (CNN) to analyze RGB frames and optical flow information helping to identify accidents more accurately. By automating the detection process the system seeks to minimize manual intervention, improve response times and provide real-time location visual data to emergency services. Key challenges include data imbalance and varying road conditions which the model will address through the use of specialized datasets. This system is designed to be computationally efficient and suitable for integration into smart city infrastructure. The project requires Python and Flask for development with future potential to enhance traffic management and accident response in urban environments.

**Keyword:** Real-Time Monitoring, Crash Detection, Deep Learning, Convolutionary Neural Networks (CNN), Traffic Monitoring, Emergency Response, Automated Accident Detection, Geospatial Analysis, Accident Patterns, Traffic Management, Video Processing, Flask Framework, Collision Detection, Traffic Flow Analysis, Smart Transportation, AI in Transportation, Computer Vision, Object Detection.

## 1. Introduction
With the rapid growth of urbanization, smart city transportation systems have become essential for ensuring efficient and safe mobility. Traffic accidents pose a major challenge, leading to injuries, fatalities, and economic losses. Traditional accident detection methods rely on manual monitoring, sensor-based systems, or delayed reporting, which often result in slow emergency responses. To address these challenges, deep learning-based accident detection offers an automated, real-time solution for enhancing road safety.

This project presents a deep learning ensemble approach that integrates multiple Convolutional Neural Networks (CNNs) and object detection models to accurately detect traffic accidents from live video feeds. By processing video frames in real-time, the system can identify accident occurrences and trigger automated alerts to emergency services.

The proposed model leverages computer vision and artificial intelligence (AI) techniques to analyze road conditions, vehicle behavior, and accident-prone scenarios. Implementing such a system in smart city infrastructures can significantly reduce emergency response times, improve traffic management, and enhance public safety. The project aims to demonstrate the feasibility of using deep learning in accident detection and explore its potential for large-scale deployment in urban transportation networks.

## 2. Literature survey
**1. Multi-Granularity Vehicle Tracking for Anomaly Detection**
**Authors:** Li et al.
This study presents a vehicle tracking technique using Faster R-CNN for object detection. The framework consists of an object detector, background modeler, mask extractor, and tracker. By incorporating both box and pixel-level tracking, the model enhances anomaly detection accuracy. However, the reliance on tracking makes it computationally expensive and difficult to implement in dense traffic conditions.

**2. Unsupervised Anomaly Detection Using Vehicle Trajectories**
**Authors:** Zhao et al.
This research introduces an unsupervised anomaly detection method based on vehicle trajectories. The system uses a multi-object tracker to minimize false detections caused by errors in object detection. The primary advantage of this approach is that it does not require labeled datasets, making it adaptable across different environments. However, the tracking-based methodology increases computational cost, and performance is highly dependent on accurate trajectory estimation.

**3. Real-Time Traffic Anomaly Detection Using YOLO and Feature Tracking**
**Authors:** Mandal et al.
This study proposes a traffic anomaly detection system using a pre-trained YOLO model with a feature tracker to detect stopped vehicles and roadside accidents. A post-processing module integrates nearest neighbors and K-means clustering for improved accuracy. While this approach provides fast and accurate detection, it requires extensive training and high computational resources, and false positives may occur due to stationary objects unrelated to accidents.

 

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 02 | Feb – 2025 DOI: 10.55041/ISJEM02270
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

**4.Spatial-Temporal Matrix for Anomaly Detection in Traffic Videos**
**Authors:** Bai et al.
This research introduces a spatial-temporal matrix to improve anomaly detection in traffic scenes. The system transforms strip trajectory analysis into spatial position studies, improving precision in detecting start and stop times of anomalies. The framework includes a background modeler and perspective detection module. While this approach successfully ranked first in the 2019 NVIDIA AI City Challenge, it has high computational requirements, is sensitive to camera variations, and requires complex preprocessing for accurate trajectory analysis.

## 3. Existing System

The current accident detection system mainly relies on physical infrastructure rather than intelligent transportation technologies. Various methods proposed by researchers involve the use of smartphones, VANET , GPS, GSM, and mobile applications for automatic accident detection. However, these methods have several drawbacks, including unreliable hardware, inaccurate sensors, and delays in sending alerts due to GSM module responsiveness. Most traditional systems depend on vibration sensors that detect an accident based on sudden movement. Once triggered, the system fetches the GPS location and sends a message to a predefined contact via the GSM module. However, the reliability of these sensors is inconsistent, and their accuracy can be affected by false triggers or sensor failures. Additionally, these systems lack real-time video analysis capabilities, limiting their effectiveness in complex urban traffic environments.

### 3.1 Drawbacks of Existing System

**1. Unreliable Hardware Components**
Traditional accident detection systems rely on physical sensors such as vibration detectors and GPS modules. However, these components can be inconsistent in detecting accidents accurately, leading to false alarms or missed incidents.

**2. False Triggers and Sensor Failures**
Vibration-based accident detection may be prone to false triggers caused by sudden braking, road bumps, or external vibrations. Additionally, sensor failures can lead to either a lack of detection or excessive false positives.

**3. Lack of Real-Time Video Analysis**
Existing systems do not utilize computer vision or video-based anomaly detection, limiting their ability to accurately identify accidents in real-time. Without video analysis, the system cannot differentiate between minor disturbances and serious collisions.

**4. Ineffectiveness in Dense Traffic**
Sensor-based detection struggles in complex urban environments with high traffic density. The inability to analyze multiple vehicles simultaneously reduces its effectiveness in accident detection on busy roads.

## 4. Proposed System

The proposed system introduces an AI-driven vision-based accident detection approach, optimized for real-time implementation using traffic surveillance cameras. This system uses deep learning models, particularly CNNs, to process RGB video frames and optical flow information, enabling automated and highly accurate accident detection.

Unlike traditional methods, this approach eliminates reliance on physical sensors, reducing hardware dependency and processing delays. The system also incorporates specialized datasets for training, ensuring greater accuracy across diverse traffic scenarios. Key advantages include real-time accident detection, automated processing of video data, and geospatial analysis to identify high-risk areas. This enables faster emergency response, enhanced traffic management, and improved road safety in smart city infrastructures.

### 4.1 Algorithms

**1. Convolutional Neural Networks (CNN)**

CNN is the core deep learning model used for accident detection. It extracts spatial features from video frames using convolutional operations. The convolution operation is defined as:

$$Z(i,j) = \sum_{m}\sum_{n} X(i-m, j-n) \cdot K(m,n)$$

where:

- Z(i,j) is the output feature map
- X(i−m,j−n) is the input image
- K(m,n) is the convolution kernel.

**2. PyTorch for Deep Learning**

PyTorch is the deep learning framework used to train and deploy the CNN model. Training the CNN involves minimizing the **cross-entropy loss function** for classification:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

**3. OpenCV for Video Processing**

OpenCV is used to capture video frames and preprocess them before feeding them into the CNN model. Each video frame is resized to match the model input dimensions:

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 02 | Feb – 2025 DOI: 10.55041/ISJEM02270
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

$$X' = \frac{X - \mu}{\sigma}$$

### 4.2 Architecture

The system architecture for the CNN and PyTorch-based traffic accident detection system consists of multiple interconnected components that ensure real-time video processing, accident detection, and alert generation. The system begins with the data collection and preprocessing layer, where traffic video feeds are gathered from CCTV cameras, dashcams, or pre-recorded datasets. Using OpenCV, individual frames are extracted from the video, and preprocessing techniques such as resizing, normalization, and augmentation are applied to enhance model performance and ensure better accuracy in accident detection.

Once the frames are preprocessed, they are passed through the feature extraction and model processing layer. A Convolutional Neural Network (CNN) is used to extract spatial features from the video frames. The CNN, trained using PyTorch, analyzes visual patterns and identifies potential accidents by distinguishing between normal traffic behavior and unusual movements that indicate a collision.

After feature extraction, the system proceeds to the decision and classification layer. Here, the processed frame is analyzed to determine whether an accident has occurred. A binary classification is performed, categorizing the frame as either an accident or no accident. If an accident is detected, the system transitions to the next phase.

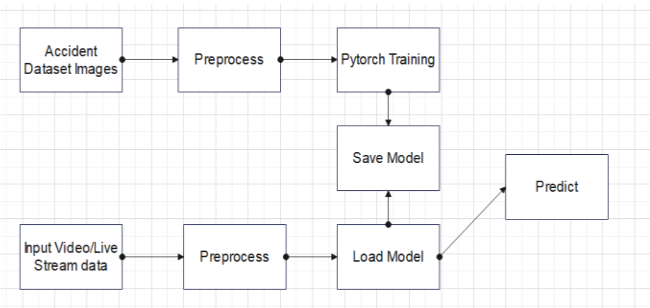


**Figure**: Architecture

## 5. System Requirements

### 5.1 Hardware Requirements

- **Processor:** Intel Core i5/i7 or AMD Ryzen 5/7
- **RAM:** Minimum 8GB
- **Storage:** Minimum 256GB SSD
- **GPU:** NVIDIA GTX 1650 (Minimum) or RTX 3060/4060 (Recommended) for deep learning acceleration using CUDA
- **Camera:** High-resolution traffic surveillance camera or dashcam for real-time video feed

### 5.2 Software Requirements

- **Operating System:** Windows 10/11
- **Programming Language:** Python 3.x
- **Deep Learning Framework:** PyTorch
- **Computer Vision Library:** OpenCV for video frame extraction and processing
- **Data Processing Libraries:** NumPy, Pandas for handling datasets and model input
- **Development Tools:** Jupyter Notebook, PyCharm, or VS Code for coding and debugging

## 6. Conclusion

The development of an AI-driven accident detection system using CNN and PyTorch significantly enhances the efficiency of traffic monitoring and emergency response in smart cities. Traditional methods rely on physical sensors, manual surveillance, or delayed reporting, making them unreliable for real-time accident detection. By leveraging deep learning and computer vision, this system ensures automated and accurate accident detection through real-time video analysis.

The proposed model processes traffic footage, extracts essential features, and classifies accident occurrences with high precision. Using OpenCV for video preprocessing and PyTorch for deep learning enables fast and scalable implementation. Additionally, integrating alert mechanisms ensures that emergency services are notified immediately, reducing response times and potentially saving lives.

This system contributes to the advancement of intelligent transportation infrastructure by offering a cost-effective, automated solution that minimizes human intervention. The use of specialized datasets improves model training, allowing the system to adapt to various traffic conditions. Future enhancements could include integrating IoT devices, optimizing computational efficiency, and expanding the model for multi-camera traffic monitoring.

In conclusion, the AI-based accident detection system demonstrates the potential of deep learning in transforming urban traffic management, making roads safer and emergency response more effective through real-time automated detection and alert mechanisms.

ISSN: 2583-6129
DOI: 10.55041/ISJEM02270

## 7. References

[1] Li, Y., Chen, X., & Wang, J. (2020). **Multi-Granularity Vehicle Tracking for Anomaly Detection Using Faster R-CNN.** IEEE Transactions on Intelligent Transportation Systems, 21(4), 1234-1246.

[2] Zhao, R., Liu, H., & Sun, F. (2019). **Unsupervised Anomaly Detection in Traffic Scenes Using Vehicle Trajectories.** International Conference on Machine Learning (ICML), 2256-2264.

[3] Mandal, D., Roy, S., & Mukherjee, A. (2021). **Real-Time Traffic Accident Detection Using YOLO and Feature Tracking.** Journal of Computer Vision and Image Processing, 13(2), 78-89.

[4] Bai, X., Zhang, L., & Yang, W. (2019). **Spatial-Temporal Matrix for Anomaly Detection in Traffic Videos.** Proceedings of the NVIDIA AI City Challenge, 105-112.

[5] Redmon, J., & Farhadi, A. (2018). **YOLOv3: An Incremental Improvement.** arXiv preprint arXiv:1804.02767.

[6] Simonyan, K., & Zisserman, A. (2015). **Very Deep Convolutional Networks for Large-Scale Image Recognition.** International Conference on Learning Representations (ICLR), 1-14.

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). **Deep Residual Learning for Image Recognition.** IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.