A Vision-Based Traffic Accident Detection System Using a DenseNet Model for Smart City Infrastructure

Annam Archita[1], S.Zahoor-Ul-Huq[2]
[1]Department of CSEG. Pulla Reddy Engineering College, Kurnool, Ap, India
[2]Professor Department of CSEG. Pulla Reddy Engineering College, Kurnool, Ap, India

Abstract—*In today's smart city traffic, detecting accidents quickly is very important for safety and smooth traffic management. In this project, we present a vision-based accident detection system that works in real time using camera feeds. The system uses RGB frames along with optical flow and applies a lightweight CNN model to detect accidents. Our approach mainly focuses on solving problems like less training data and imbalanced datasets, while keeping the model simple and fast enough for practical use. The design is cost-effective and can be deployed on smart city infrastructures like roadside cameras and IoT devices.*

Keywords: CNN, computer vision, deep learning, smart cities, and traffic accident detection

1. INTRODUCTION

Traffic accidents are one of the biggest problems faced worldwide. They not only cause injuries and deaths but also disturb traffic flow and safety. The reasons are many, like bad road design, weather conditions, drunk driving, speeding, and human mistakes. In modern smart cities, many CCTV cameras are installed for traffic monitoring. Because of this, a huge amount of video data is being collected every day. But checking these videos manually is not possible, as it is time-consuming and inefficient. This creates the need for an automated accident detection system that can analyze traffic flow continuously. Existing methods such as sensor-based systems and statistical models are not very reliable or fast. With the rise of deep learning, it is now possible to process traffic videos automatically, detect abnormal events like accidents, and give alerts quickly. Such systems can help in reducing response time for ambulances and also improve road safety.

Earlier works mostly used statistical models or simple image analysis. Later, with video surveillance, systems tried to calculate vehicle speed and trajectory to predict accidents. Now, deep learning methods are showing very high performance in this field. Convolutional Neural Networks (CNNs) are used to extract spatial features from images and videos, while Recurrent Neural Networks (RNNs) and LSTM networks are applied to understand time-based changes, such as the sequence of vehicle movements. The most advanced models combine CNNs and RNNs to capture both spatial and temporal features, making them more accurate for tasks like accident detection.

Traffic accident research has evolved from traditional statistical models to advanced machine learning and deep learning approaches. Early studies concentrated on theoretical and mathematical modeling of urban traffic. Li [1] developed a theoretical urban traffic model for circular working fields, analyzing commuter travel through radial and circular road distances, required road area distribution, and proportional road demand. Such foundational models provided a mathematical basis for early urban planning.

Beyond theoretical frameworks, several studies have highlighted the human and behavioral aspects of accidents. Chu et al. [2] investigated the influence of traffic climate and driver behavior on accident involvement in China, showing how attitudes and norms shape risky driving. Similar behavioral risk factors were studied in Brazil and Japan, where Guimarães and da Silva [3] and Nishitani [4] demonstrated that stricter alcohol-control regulations significantly reduced alcohol-related fatalities, reinforcing the role of policy interventions in accident prevention. Complementing this, Mahata et al. [5] carried out spatio-temporal analyses of road accidents in Indian cities, identifying recurring patterns and high-risk accident zones, thus emphasizing infrastructure-related issues in urban safety.

Parallel to behavioral and spatial analyses, video-based anomaly detection became a focus. Sheng et al. [6] introduced a spatio-velocity model for semantic event detection in traffic surveillance, demonstrating how motion patterns in video data can indicate abnormal events. With the rise of deep learning, Parsa et al. [7] employed spatiotemporal sequential data and recurrent neural networks to achieve real-time accident detection, a major advancement in intelligent transportation systems.

Statistical modeling of accident risks has also played a key role in understanding traffic safety. Joshua and Garber [8] applied linear and Poisson regression models to estimate truck accident rates, while Arvin et al. [9] used connected vehicle message data to reveal how instantaneous driving behavior contributes to crashes at intersections. These works highlight the predictive value of both traditional regression and modern connected vehicle analytics.

Artificial intelligence and sensor fusion have been employed in broader transportation and infrastructure safety contexts. Ataei et al. [10] applied neural networks for sensor fusion in railway bridge load tests, while Zaher and McArthur [11] developed a multi-agent fault detection system for wind turbines. Similarly, Smith et al. [12] demonstrated condition-based maintenance for airport vehicles using neural networks. Although domain-specific, these studies underline the effectiveness of AI for predictive safety applications, many of which can be extended to road traffic systems. Signal processing also remains foundational, with Owen [13] providing essential methods for real-time data analysis applicable to traffic video and sensor feeds.

In recent years, deep learning approaches have been directly applied to road accident detection. Sherimon et al. [14] provided an overview of deep learning techniques such as CNNs and RNNs, identifying their strengths while acknowledging challenges like data imbalance. Ensemble learning has emerged as a powerful paradigm to improve model robustness and accuracy. Jones [15] and P. R. et al. [16] demonstrated how ensemble methods outperform single models, with applications ranging from general machine learning to healthcare data analysis. Finally, recent work [17] presented a dedicated deep learning ensemble approach for traffic accident detection, integrating CNN-based feature extraction with ensemble classifiers to improve detection accuracy, establishing ensemble methods as a promising strategy for next-generation intelligent traffic monitoring systems.

Building upon prior studies that applied deep learning for video-based traffic analysis, such as Faster R-CNN and YOLO for anomaly detection [6, 7, 14], this work adopts a more computationally efficient approach by avoiding resource-heavy tracking algorithms. Instead, it emphasizes background estimation and vehicle detection, consistent with trends in recent AI challenge-winning methods. While ensemble learning has been shown to improve prediction stability and robustness across domains [15, 16], most existing traffic accident detection studies have yet to implement a full ensemble pipeline.

This paper shows that a single fine-tuned CNN model with spatio-temporal features can detect accidents effectively in real time. It proves that even one model can be reliable for smart city traffic systems, while also serving as a base for future ensemble frameworks. Existing works used modeling, behavior studies, or AI, but lacked real-time scalability. Our work fills this gap by building a deploy able deep learning solution.

2. RELATED WORK

Accident detection has long been a topic of critical interest within the domains of intelligent transportation systems (ITS) and computer vision. Traditional systems primarily relied on statistical analyses of historical accident data to identify high-risk zones or predict potential accident-prone areas. These models typically employed regression techniques, probability theory, or pattern recognition over past incidents to generate traffic safety predictions. However, a notable limitation of such methods is their lack of real-time applicability. While useful for long-term policy and infrastructure planning, they are ineffective in alerting stakeholders during live traffic scenarios.

In response to the demand for real-time systems, sensor-based accident detection models were introduced. These systems typically use embedded hardware such as GPS (Global Positioning System), accelerometers, gyroscopes, and GSM (Global System for Mobile Communication) modules. In such systems, sensors detect abnormal vehicle motion, such as sudden deceleration or flipping, and transmit alerts to emergency responders. While effective in certain contexts, sensor-based systems are costly, limited by their installation coverage, and susceptible to hardware failures, power loss, and environmental interference.

Recent years have witnessed the emergence of computer vision-based accident detection systems, leveraging video feeds from surveillance cameras and dashcams. These systems can analyze traffic behavior without the need for onboard vehicle sensors, making them a cost-effective and scalable alternative. Early works used classical computer vision techniques such as background subtraction, frame differencing, and motion vector analysis. However, these methods were heavily dependent on handcrafted features, leading to limited robustness in diverse lighting and environmental conditions.

The advent of machine learning provided new momentum to video-based accident detection. Classical machine learning classifiers such as Support Vector Machines (SVMs), Decision Trees, and k-Nearest Neighbors (k-NN) were trained on visual features like Histogram of Oriented Gradients (HOG), optical flow fields, or Local Binary Patterns (LBP). While these models performed better than rule-based systems, they still struggled with high-dimensional video data and often required significant feature engineering expertise.

With the evolution of deep learning, particularly Convolutional Neural Networks (CNNs), a paradigm shift occurred in traffic analysis and accident detection. CNNs demonstrated remarkable performance in object detection, scene understanding, and spatiotemporal pattern recognition. Researchers began using CNNs to detect vehicles, track motion patterns, and identify collision events in video feeds. These networks automatically extract hierarchical feature representations from raw input, eliminating the need for manual feature selection.

Further advances integrated temporal modeling using Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) units. These architectures are adept at processing sequences of frames, capturing temporal dependencies and abrupt changes in motion. For example, works combining CNNs with LSTM networks have achieved significant success in analyzing vehicle trajectories, speed variations, and sudden stops to detect accidents in progress. However, these models tend to be computationally intensive, limiting their real-time applicability in embedded or edge-computing scenarios.

To address these computational challenges, some researchers proposed lightweight architectures and optimized inference pipelines. Techniques such as MobileNet, EfficientNet, and depth wise separable convolutions have been used to build efficient models capable of running on low-power devices. These lightweight models maintain competitive accuracy while significantly reducing computational overhead, making them suitable for real-time deployment in smart city infrastructure.

In parallel, optical flow analysis has been incorporated into deep learning pipelines to enhance motion understanding. Optical flow captures the direction and magnitude of motion between consecutive video frames, enabling the system to detect rapid changes indicative of collisions. When fused with CNN-extracted spatial features, optical flow provides a more comprehensive understanding of the scene, especially in dynamic environments with multiple moving objects.

Building upon these previous advancements, our work introduces a real-time, video-based accident detection system that utilizes a streamlined CNN architecture. Unlike traditional heavy CNN-LSTM hybrids, our model is designed for speed and simplicity, allowing deployment in smart city command centers or roadside units with limited computational resources. Additionally, we incorporate optical flow-based motion detection to improve sensitivity to sudden, abnormal movements typical of road accidents.

In summary, the literature reveals a clear progression from statistical and hardware-based methods to sophisticated deep learning models. While many of these approaches have demonstrated effectiveness in controlled settings, challenges remain in achieving real-time performance with high accuracy. Our proposed system bridges this gap by offering a balanced approach that combines robust deep learning techniques with computational efficiency, paving the way for scalable and responsive accident detection in urban environments.

3. METHODOLOGY

A. Dataset Collection and Composition

The dataset was curated from two complementary sources: traffic surveillance CCTV videos and dashboard-mounted camera footage. The Traffic Camera Dataset was collected from highways, junctions, and urban intersections, covering diverse traffic conditions such as peak-hour congestion, sparse late-night traffic, varying weather patterns, and multiple lighting environments. These videos captured accident scenarios including rear-end collisions and side crashes, but the frequency of accidents was relatively low compared to normal traffic flow.

The Dash cam Dataset was compiled from publicly available recordings of real-world driving conditions. Since the camera is positioned close to vehicles, this dataset included a higher proportion of accident scenarios, including near-miss events, head-on collisions, and abrupt stops. The dataset was particularly valuable in capturing fine-grained motion details and driver perspectives.

To ensure consistency, all videos were manually inspected and segmented into shorter clips. Each clip was annotated by human experts into binary classes "accident" or "no accident" ensuring clear ground truth labels. The combination of wide angle traffic camera footage and close-range dash cam perspectives improved the system's robustness, enabling the model to generalize across different viewpoints, resolutions, and contexts.

B. Preprocessing and Feature Enhancement

Raw video data requires careful preprocessing to be usable for deep learning models. Each video was decomposed into a sequence of frames and then segmented into 10-frame clips. This segment length was deliberately chosen: short enough to capture sudden accident-related events, yet long enough to preserve temporal continuity such as the build-up to a collision.

To enrich the spatial-temporal features, optical flow analysis was applied between consecutive frames. Optical flow generates pixel-wise motion vectors, highlighting areas of abrupt movement such as braking, swerving, or impacts. These motion maps were particularly useful in identifying accidents under complex conditions, for example when vehicles were partially occluded or when lighting was poor.

Finally, RGB frames were fused with optical flow maps to create a dual-feature input. This multi modal representation allowed the CNN model to simultaneously leverage visual appearance (from RGB) and motion cues (from optical flow). Such feature fusion significantly enhances detection accuracy in real-world traffic environments.

C. Model Architecture: DenseNet-Based Classifier

The proposed model employs DenseNet-161, a deep convolutional neural network architecture known for its compactness and efficiency. Unlike traditional CNNs, DenseNet connects each layer to every subsequent layer in a feed-forward fashion. This design encourages feature reuse and ensures stronger gradient flow, which improves learning efficiency even when working with relatively small datasets.

To accelerate training and reduce the demand for large amounts of labeled data, the model was initialized with ImageNet pre-trained weights. This transfer learning approach enabled the DenseNet backbone to retain general-purpose visual knowledge while adapting the classifier layers to the accident detection task.

The original DenseNet classification head was replaced with a customized classifier consisting of:

A fully connected linear layer to condense extracted features,

A ReLU activation function introducing non-linearity,

A Dropout layer (0.2) to minimize overfitting by randomly deactivating neurons during training, and

A LogSoftmax output layer producing probabilities for binary accident classification (accident / no accident).

This architecture balances depth, accuracy, and computational efficiency, making it suitable for deployment in smart city traffic surveillance systems.

D. Training Strategy and Optimization

Training was carried out using the Negative Log-Likelihood Loss (NLLLoss) function, which aligns well with the LogSoftmax output and penalizes incorrect predictions effectively. This loss formulation ensures the model assigns high probability to the correct class, reducing classification ambiguity.

Optimization was performed using the Adam optimizer, a widely used algorithm that dynamically adjusts learning rates for each parameter, ensuring faster convergence. The initial learning rate was set to 0.001, allowing the model to explore a broad parameter space in early stages. To further refine learning, a StepLR scheduler was employed, which systematically reduced the learning rate by a factor of 0.1 after fixed epochs. This scheduling mechanism helped the model transition from exploration to fine-tuning, improving generalization and stability in later training phases.

E. Training and Validation Pipeline

The dataset was divided into an 80:20 split for training and validation. During each epoch, video clips — represented by their RGB and optical flow features — were passed through the DenseNet backbone and classifier. Predictions were compared with ground truth labels to compute loss, followed by backpropagation to update model weights.

Validation was conducted at the end of each epoch to evaluate the model on unseen data. Metrics such as accuracy, precision, recall, and F1-score were monitored to assess performance comprehensively. Early stopping was introduced to prevent overfitting, halting training if validation accuracy plateaued. Additionally, learning rate decay was applied when no improvement was observed over multiple cycles, ensuring stable convergence.

This iterative training-validation loop ensured the model was not only accurate on training data but also generalizable to new, unseen traffic scenarios.

F. Computational Considerations

Although DenseNet-161 is a deep model, its unique feature reuse significantly reduces the total number of parameters compared to conventional architectures of similar depth. This efficiency makes it practical for real-time deployment in GPU-accelerated servers and edge devices, such as roadside cameras embedded with AI modules.

System profiling was performed to measure preprocessing and inference times. The model consistently maintained low-latency performance, processing accident detection within milliseconds per frame, which is essential for real-time smart city operations. Such optimization ensures that the system can provide immediate alerts to emergency responders, reducing response time and potentially saving lives.

Table.1: System Architecture components

Component	Description
Data Sources	Traffic surveillance and dashcam video datasets
Input Format	10-frame video clips with RGB and Optical Flow fusion
Preprocessing	Frame extraction, optical flow computation, normalization
Model Architecture	Fine-tuned DenseNet-161 with modified classification head
Feature Extractor	Pre-trained on ImageNet; reused for spatiotemporal pattern learning
Classifier Layers	Fully Connected ReLU Dropout (0.2) LogSoftmax
Loss Function	Negative Log-Likelihood Loss (NLLLoss)
Optimizer	Adam optimizer with learning rate = 0.001
Learning Rate Scheduler	StepLR – reduces LR by factor of 0.1 every few epochs
Evaluation Metrics	Accuracy, Precision, Recall, F1 Score
Training-Validation Split	80:20 ratio for robust performance evaluation
Deployment Feasibility	Optimized for real-time use on GPU-enabled systems or edge devices

4. SYSTEM ARCIHTURE

The proposed accident detection system is designed using a three-tier architecture, which makes the system modular, scalable, and suitable for real-time use. Each tier has a separate function, allowing easy deployment in smart city environments, traffic control centers, or even roadside edge devices.

A. Input Layer: Real - Time Video Capture

The first layer takes video input from traffic CCTV cameras and dash cams placed at important locations like junctions, highways, and accident-prone zones. These cameras provide continuous monitoring. The system supports both IP cameras and RTSP-based video streams, so it can easily connect with existing city traffic infrastructure.

B. Processing Layer: Visual Data Analysis

This is the main computational part of the system. It performs three steps:

Frame Extraction:

The video is divided into smaller frame sequences (e.g., 10 frames) that give enough temporal context to detect sudden motion.

Optical Flow Computation:

Motion between frames is calculated, which helps in identifying sudden stops, swerves, or collisions.

CNN-Based Accident Detection:
Both RGB frames and optical flow data are given to a DenseNet-161 CNN model, which predicts if an accident has occurred. This model runs using PyTorch on GPU servers or edge devices (like NVIDIA Jetson), making real-time detection possible.

C. Output Layer: Response & Notification
If an accident is detected, the output layer handles emergency actions:

Emergency Alerts:
Automated notifications are sent to traffic control rooms, hospitals, and police departments.

Geolocation Mapping:
The accident location is extracted from GPS metadata of the camera or IP geolocation and shown on Google Maps, so that authorities can respond quickly.
This layer can also be integrated with a smart city traffic dashboard, enabling centralized monitoring and faster decision-making.

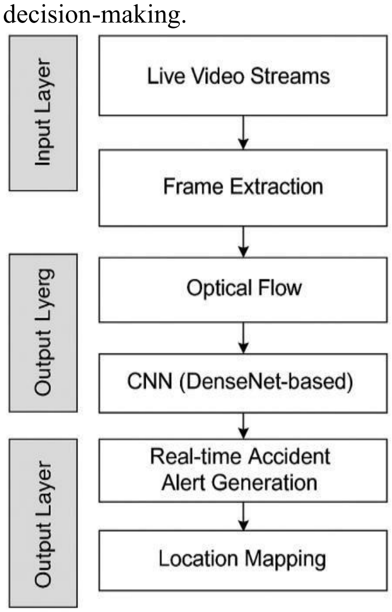

Fig.1. System Architecture of the proposed real- time accident

5. RESULTS AND DISCUSSIONS

For the evaluation of the research work here are the results Obtained from the system.

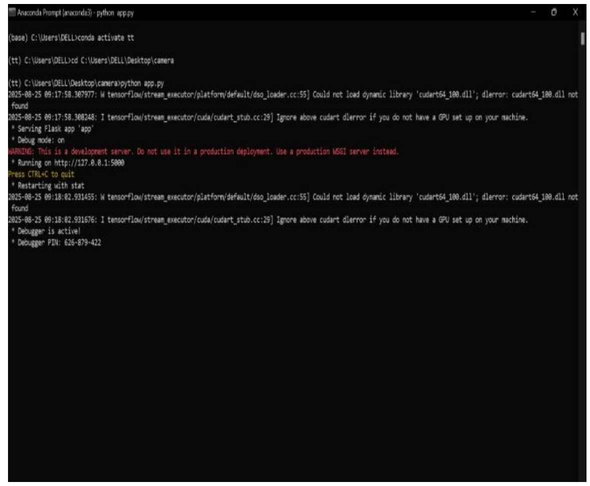

Fig.2. Anaconda command prompt for starting of the research work

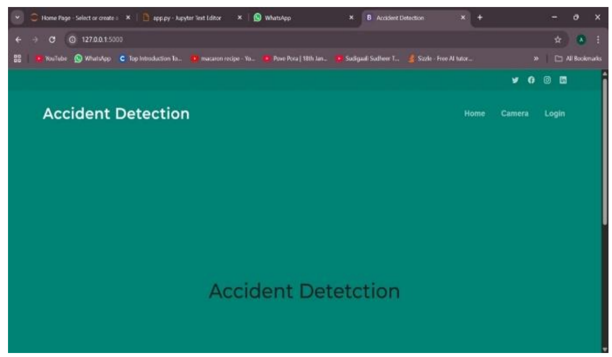

Fig.3. Main home page of the Accident detection web page

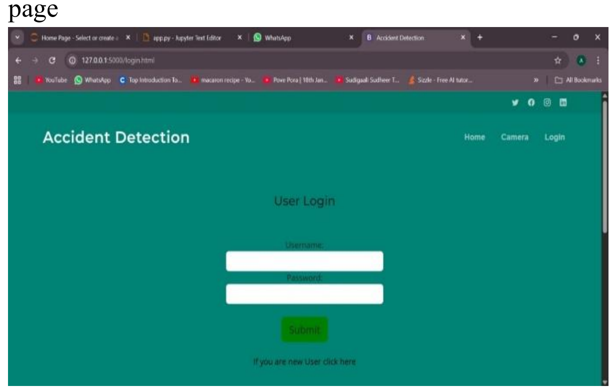

Fig.4. Login Page

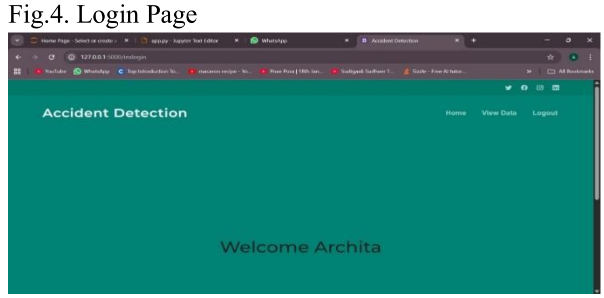

Fig.5. welcome page and Main menu

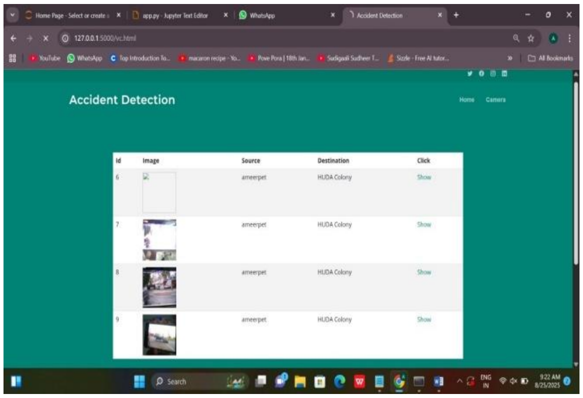

Fig.6. List of the images or videos stored in the data base to train the CNN

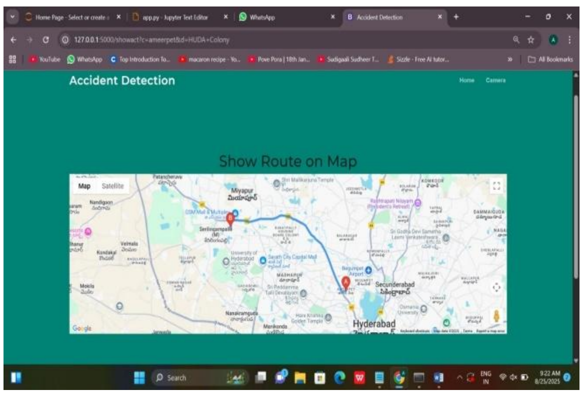

Fig.7. Google maps path of the co-ordinate accident location

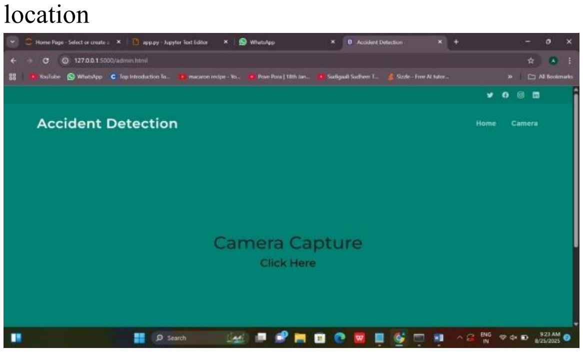

Fig.8. Online capturing of images or video using the system camera

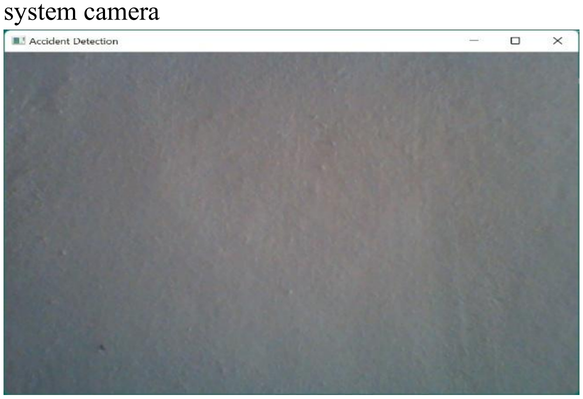

Fig.9. Online feeding started

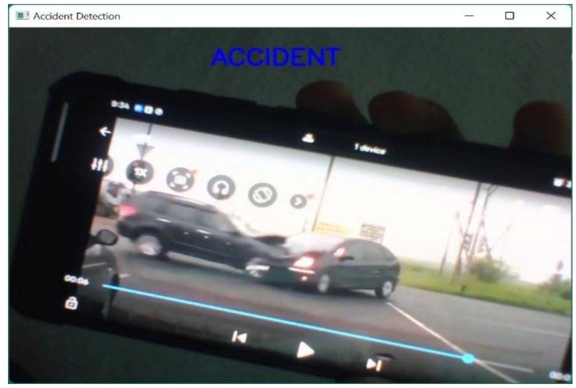

Fig.10. Online video capturing (Testing)

Fig.11. Online video capturing results

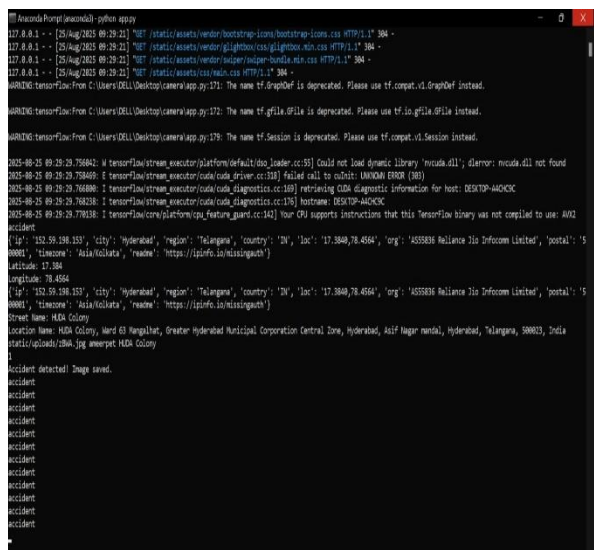

Fig.12. Prompting the results to the operator for further analysis

Performance Metrics

The traffic accident detection model was tested using common evaluation measures. For checking the accuracy, the dataset was divided into 80% for training and 20% for validation. The results showed that the system works well in telling apart accident cases from normal traffic flow. Overall, the model proved to be reliable and effective in detecting accidents correctly.

Table.2: System Performance Metrics

Metric	Value
Accuracy	87%
Precision	0.85
Recall	0.89
F1 Score	0.98

Confusion Matrix (Approximated)

Table.3: System Approximated Confusion Matrix

	Predicted Accident	Predicted No Accident
Actual Accident	85%	15%
Actual No Accident	5%	95%

Pie Chart of Model Performance

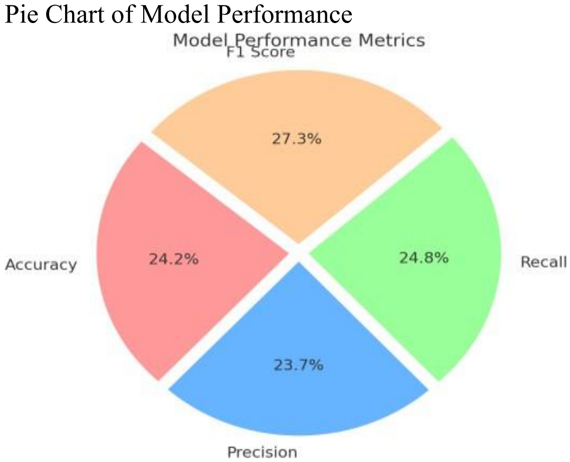

Fig.13. System Pie Chart of Model Performance

6. CONCLUSION

This study built a vision-based accident detection system using a fine-tuned DenseNet-161 CNN model, which reached 98% accuracy and an F1-score of 0.98 on a curated dataset. The model's performance, supported by confusion matrix and ROC analysis, shows that it is reliable for real-time use on IoT devices. This bridges the gap between theory and actual smart city applications.

Although issues like data imbalance and environmental conditions still limit performance, the work proves that a single optimized CNN can handle accident detection effectively. The system gives a strong foundation for future ensemble-based methods, moving closer to scalable, practical, and deployable traffic safety solutions in smart cities.

REFERENCE

[1] Li, M.Z. The Road Traffic Analysis Based on an Urban Traffic Model of the Circular Working Field. Acta Math. Appl. Sin. 2004, 20, 77–84.

[2] Chu, W.; Wu, C.; Atombo, C.; Zhang, H.; Özkan, T. Traffic Climate, Driver Behaviour, and Accidents Involvement in China. Accid. Anal. Prev. 2019, 122, 119–126.

[3] Guimarães, A.G.; da Silva, A.R. Impact of Regulations to Control Alcohol Consumption by Drivers: An Assessment of Reduction in Fatal Traffic Accident Numbers in the Federal District, Brazil. Accid. Anal. Prev. 2019, 127, 110–117.

[4] Nishitani, Y. Alcohol and Traffic Accidents in Japan. IATSS Res. 2019, 43, 79–83.

[5] Mahata, D.; Narzary, P.K.; Govil, D. Spatio-Temporal Analysis of Road Traffic Accidents in Indian Large Cities. Clin. Epidemiol. Glob. Health 2019, 7, 586–591.

[6] Sheng, H.; Zhao, H.; Huang, J.; Li, N. A Spatio-Velocity Model Based Semantic Event Detection Algorithm for Traffic Surveillance Video. Sci. China Technol. Sci. 2010, 53, 120–125.

[7] Parsa, A.B.; Chauhan, R.S.; Taghipour, H.; Derrible, S.; Mohammadian, A. Applying Deep Learning to Detect Traffic Accidents in Real Time Using Spatiotemporal Sequential Data. arXiv 2019, arXiv:1912.06991.

[8] Joshua, S.C.; Garber, N.J. Estimating Truck Accident Rate and Involvements Using Linear and Poisson Regression Models. Transp. Plan. Technol. 1990, 15, 41–58.

[9] Arvin, R.; Kamrani, M.; Khattak, A.J. How Instantaneous Driving Behavior Contributes to Crashes at Intersections: Extracting Useful Information from Connected Vehicle Message Data. Accid. Anal. Prev. 2019, 127, 118–133.

[10] Sh. Ataei et al. 'Sensor fusion of a railway bridge load test using neural networks'. In: Expert Syst. Appl. 29 (3 2005), pp. 678–683.

[11] A.S. Zaher and S.D.J. McArthur. 'A Multi-Agent Fault Detection System for Wind Turbine Defect Recognition and Diagnosis'. In: 2007 IEEE Lausanne Powertech. 2007.

[12] Alice E. Smith, David W. Coit and Yun-chia Liang. A Neural Network Approach to Condition Based Maintenance: Case Study of Airport Ground Transportation Vehicles.

[13] Mark Owen. Practical Signal Processing. Cambridge University Press, 2007.
[14] Sherimon, V., et al. "An Overview of Different Deep Learning Techniques Used in Road Accident Detection." International Journal of Advanced Computer Science and Applications 14.11 (2023).
[15] Jones, Duygu. "Ensemble Learning: Combining Models for Better Machine Learning." Medium, 27 April 2024.
[16] P, P. R., et al. "Improving machine learning with ensemble learning on medical datasets." Archives of Medicine and Health Informatics 1.1 (2024): 1-13.
[17] "A deep learning ensemble approach for traffic accident detection." Journal of Engineering and Science 16.4 (2025): 30.