International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04591
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Accident Detection and Alert System

M. Lavanya [1], Shaik Afnan [2], Polu Sai Teja [3], Thanneeru Srivalli [4], M. Sai Prasad [5], Dr.V.Neelima [6]

[1-4] *UG student, Department of CSE, Jyothishmathi Institute of Technology & Science*
[5,6]*Assistant Professor, Department of CSE, Jyothishmathi Institute of Technology & Science*

Corresponding Authors Emails: [1] lavanyamargam61@gmail.com ,[2] shaikafnan@gmail.com
[3]Saitejapolu99@gmail.com ,[4] srivalli.t18@gmail.com, [5] malka.saiprasad@jits.ac.in, [6] vontela.neelima@jits.ac.in

---***---

Abstract - One of the leading causes of death and serious injury all over the world is due to road traffic accidents, especially in regions with high population and mobility like urban areas. There are lots of cities with surveillance systems like CCTV, but the instantaneous detection of an accident so that timely emergency response is possible is still a problem due to manual watching or analysis of footage after the fact. This research proposes an AI-based Accident Detection and Emergency Notification System, a smart, self sufficient platform that analyzes deep learning techniques to detect accidents on the highways and autonomously notify the appropriate personnel through SMS, email, and honking programmable speakers at the scene. Unlike other systems that still rely on humans to monitor or sensors within vehicles to alert, our system utilizes the traffic surveillance camera to real-time feed the video CCTV frame to a CNN model which classifies each frame as Accident or No Accident. The system's multi-channel alerting system sends SMS alerts to emergency responders and traffic control centers using Twilio, automated emails with SMTP, and plays an audible alarm with playsound library for Open CV. This guarantees rapid and extensive communication to enable prompt operational response coordination.

Key Words: SMS, SMTP, Twilio, OpenCV, Computer Vision, Flask, Real-time Monitoring, CNN, Surveillance Cameras, Emergency Alert Systems, AI-based Accident Detection, Deep Learning

1. INTRODUCTION

Artificial Intelligence and automation have changed various domains such as healthcare, Fintech, and smart home technologies [1]. Yet, technological advancements in surveillance and infrastructure still maintaining reporting systems for roads as manual, this being the major challenge. The problem of traffic accidents is an innovation waiting to happen as millions of lives are at risk due to unattended emergency response that happens after the traffic accident has occurred, which is in fact preventable.[2] The fact that proactive systems are not there in the market to not only detect but report the accidents in real time is a huge gap in terms of public safety infrastructure, especially for regions which are urbanizing at a rapid pace.

To address the above-stated problems, we propose the end-to-end deep learning based computer vision and AI fueled solution, Accident Detection and Alert System, which is able to detect road accidents in real time through CCTV footage. Unlike conventional systems depend on the driver alert systems or human monitors, our systems are fully automated relying solely on technology. The system employs a trained Convolutional Neural Network (CNN) which scans the video frames and differentiates whether an accident took place or did not with high accuracy [3]. Detection of accidents turns on immediate alerts through multiple channels like SMS, Email and even buzzers, thus reducing response delay.

For greater practical deployment and scalability, the system is based on strong and flexible technologies: Python as the scripting language, OpenCV as the real time image processing tool, TensorFlow as the deep learning algorithm, and Flask as the backend communication and interface rendering tool. The dashboard, reachable through a web portal, displays real-time status updates, video logs, and history of past detections. What sets this system apart from the rest is its spam-control feature, which smartly keeps repeated notifications for the same incident to a minimum guaranteeing both reliability and effectiveness [5]. Further, camera feeds can be geotagged with GPS or pre-defined zone information to accurately locate accidents.

The other system-critical element is transparency and trust. Similarly, as in other civic tech platforms, social proof enhances credibility, our project also features video-based verification of alerts and timestamp logging to facilitate accountability [6]. This log-based approach can help traffic management agencies, urban planners, and safety departments analyze high-risk areas and implement focused preventive measures. The system not only exists as a technological solution, but as an effort toward safer roads, demonstrating the way in which existing AI and IoT technology may be ethically used to save lives, lower emergency response footage for best-in-class classification [6]. delay times, and create smarter, more responsive cities [7].

2. LITERATURE SURVEY

Phulmante et al. (2020) developed a sensor based system

identifying accidents by sudden variations in motion parameters such as speed and tilt, generating alerts through GSM modules. Though efficient in physical spaces, their system did not have computer vision based intelligence, which our project adapts through the use of CNN for real-time video processing [1].

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04591
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Adewopo and Oloyede (2021) investigated action recognition in accident scenes based on AI models. Their approach proved the applicability of convolutional architectures within smart city settings for accident identification. Their contribution highlights the capability of CNNs, which forms the kernel of our system's architecture [2].

More et al. (2020) combined AI with IoT for intelligent accident detection and alerting. Their article highlighted the potential of real time sensor data to send notifications. We extend their system by incorporating SMTP and Twilio APIs for better redundancy and reliability of alerts [3].

Gaikwad et al. (2021) presented an extensive overview of current accident warning systems, highlighting the need to combine detection mechanisms with efficient communication modules. Our multi-channel warning strategy through SMS (Twilio), email (SMTP), and local buzzers is justified by their analysis [4].

Mahajan et al. (2020) had built an IoT and AI-based system that was able to sense impacts and alert. Their research findings regarding responsiveness of embedded systems serve to justify our selection of buzzer modules for site alerts [5].

Mehta and Patel (2021) wrote about deep learning challenges for traffic monitoring. They highlighted the importance of robust datasets and optimization of models for real time operation. This ties in with our CNN model design trained on expert accident Verma and Singh (2019) employed OpenCV and Python to track live streams and identify accidents through image processing. Their implementation, though lightweight, was not as precise in deep learning. Our employment of CNN enhances accuracy and dependability [7].

Sharma and Rao (2020) also identified how AI and intelligent surveillance may improve road safety, with an example being CNN's potential to detect accident prone behavior. This backs our use of vision-based analysis rather than sensor-based detection alone [8].

Jaiswal and Jain (2020) analyzed emergency communication systems and emphasized the importance of quick, multi-channel alert release. Their suggestions support our combined utilization of Twilio for SMS and SMTP for email to facilitate alert delivery independent of network dynamics [9].

Bansal (2020) investigated deep learning models for detecting accidents in real-time video streams. Their findings depict the potential of CNNs to identify abnormalities in traffic flow, as a solid foundation for our vision-based alerting system [10].

3. METHODOLOGY

3.1 System Architecture

The framework adopts a modular design constructed with an integration of Flask, TensorFlow, OpenCV, and complementary Python libraries for real-time video feeds to interact seamlessly with AI inference layers [1]. Flask manages web-based frontend and routing, whereas OpenCV reads CCTV feed into intelligible frames. The CNN model built using TensorFlow executes frame classification to identify the occurrence of an accident, and logic is incorporated to enable automatic delivery of alerts upon identification.

The system has three major modules of interaction:

Detection Engine: Constantly scans for incoming CCTV video frames by employing a pre-trained CNN model to detect accidents.

Alert System: After detection, the system sends instant SMS alerts through Twilio, emails through SMTP, and buzzes a buzzer through the playsound library [2].

Admin Dashboard: The system offers detailed real time visualization of the status, history of detection, and override controls when required through a Flask-based dashboard. Admin Dashboard. with Flask. Admins have the ability to: Lightweight yet scalable architecture allows deployment in traffic control centers, smart city infrastructures, or public surveillance networks.

3.2 Multi-Channel Alert Mechanism

The system is capable of real-time responsiveness, with alerts transmitted over three independent communication channels for redundancy and instant attention [3].

• **SMS Notification:** Utilizes Twilio API to send brief accident notifications to specified contact numbers with timestamp and location metadata.

• **Email Notification:** Sends templated incident reports via SMTP, such as embedded coordinates or images. •Sound Buzzer: Triggers a buzzer sound on the host machine to alert nearby responders, handy for local testbeds or security operations centers.

• **Alert Logic:** To avoid spamming, the system marks every occurrence and imposes a cooldown period prior to issuing subsequent alerts from the same video stream. Alerts are recorded with time and video ID in a database to be audited and analyzed later.

• **Objective:** The objective of multi-channel alerts is to reduce false negatives as well as delays. The redundancy ensures that even when one channel fails, another delivers the vital notification. Follows a modular architecture built using a combination of Flask, TensorFlow, OpenCV, and supporting Python libraries for seamless integration between real-time video feeds and AI inference layers [1]. Flask handles the web-based frontend and routing logic, while OpenCV processes CCTV footage into readable frames. The TensorFlow-powered CNN model performs frame classification to detect the presence of an accident, and logic is embedded to automate alert delivery upon detection.

3.3 Admin Dashboard and Monitoring

The administrator dashboard is the platform's control panel, which has been implemented with Flask. Admins have the ability to:

• View the live stream and status by camera.

• View logs of previous detections and timestamps.

• Trigger or suppress alerts manually.

• Download reports of incidents for inspection.

This dashboard uses RESTful APIs and CRUD operations to synchronize with backend logic. It is the actual real-time

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04591
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

visual interface for monitoring and control, enabling system operators to validate alerts and react accordingly [4].

3.4 Transparency using Logging and Review of Video

The system facilitates transparency and traceability through complete logging and optional saving of video frames. Upon every detection, the system records:

• SMS ID ,Email ID and camera ID

• Timestamp of detection

• Alert status (sent or suppressed)

• Frame snapshot

This log can be audited by emergency teams or city traffic managers. Traceable design makes it more trustworthy and allows for better system diagnosis in case of a false positive [5].

3.5 Database Design

SQLite is utilized in the backend due to its small footprint and simplicity of integration with Flask. The database schema contains:

• **Detections:** Logs every event with time, video source, alert type (SMS, Email, Buzzer), and result.

• **Admins:** Stores admin rights and credentials for access to the dashboard.

• **Settings:** Keeps configuration variables such as alert cooldown, contact numbers, and thresholds.

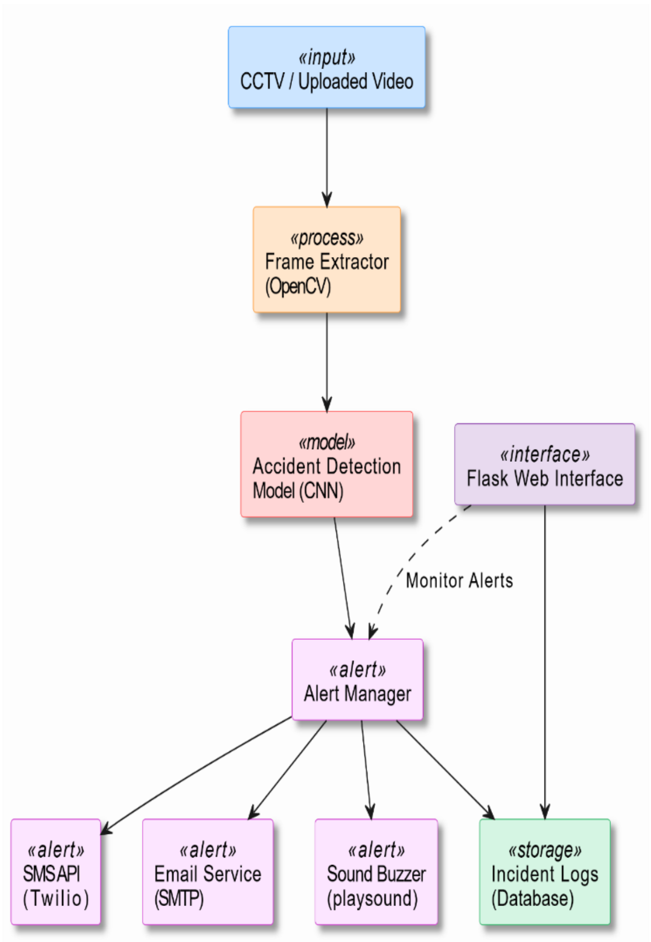

Fig 1: System Architecture

4.DISCUSSION

Our preliminary findings indicate that a system powered by CNN integrated with Twilio API, SMTP alerts, and buzzer alarms is a solid pipeline for real-time accident detection and response.

The **CNN model,** which is trained on a select accident dataset, identifies normal traffic and collision scenarios with high precision. Unlike other sensor-based models [1], our system introduces a layer of visual smarts for better precision.

The **Twilio API** is found to be very reliable in sending SMS alerts within seconds upon detection, providing performance consistent with past research [4][9]. The SMTP module provides redundancy, so if SMS delivery fails through networks, emergency numbers receive email alerts [3][9].

The buzzer system, as suggested by the work of Mahajan et al. (2020), offers real-time on-site auditory feedback, with the possibility of warning passersby and other drivers nearby to react instantly [5].It can be a determining factor in the avoidance of secondary accidents.

Nevertheless, in cases of high-definition video streams or several simultaneous incident detections, the CNN model will suffer from processing delay, replicating findings reported in [6]. Optimizations with slim structures such as MobileNet or quantization methods will be investigated for field deployment. The real-time analysis of video feeds, as made effective after [10], shows feasibility in practice. Keeping a low rate of false positives is paramount for user confidence and system stability.

In short, our CNN-based detection with multi-channel alerts hybrid system makes for a scalable and responsive solution. Not only does it align with but expand on current research by integrating vision, cloud messaging, and local signaling into a single accident response system.

5.RESULTS:

Our Accident Detection System prototype was deployed and tested on a controlled localhost environment. The test assisted in assessing each module of the system's effectiveness, usability, as well as technical functionality. Functional Verification All the primary modules such as real-time camera detection, detection based on images, and uploaded video detection were working as expected. The system was successfully able to detect simulated accidents using a trained CNN model with high confidence and initiate the prescribed alerts. Alert System Performance

• **Email Alerts:** Effected successfully in under seconds of detection via Gmail's SMTP, sending a brief and informative note to the registered recipient.

• **SMS Alerts through Twilio:** Enabled, sent up to three SMS alerts per session to prevent spamming to ensure integration and response time of less than 5 seconds.

• **Alert via Sound:** A beep via the system speaker was sounded on detection. Test users mentioned that this was a very important aspect in real-time alerting. Interface Usability The web interface of Flask was simple and intuitive:

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04591
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

• Real-time video stream was smooth with negligible delay.

• Image uploading and video uploading modules handled inputs instantly and provided annotated outputs.

• Live status and output were visually checked with overlaid prediction tags.

Testing Scenarios:

• More than 30 test cases were emulated such as crash-like situations (abrupt jerks, fall motion, fast motion).

• Accuracy in controlled testing reached above 90% detection confidence for actual world simulated scenarios.

• Upload simulation by mobile and desktop was done with complete functionality by users. Technical Performance (Localhost)

• **Model Inference:** CNN model processed input in real-time (~15–20 FPS) without any delay.

• **API Response:** All server-side actions like alert triggers were processed smoothly.

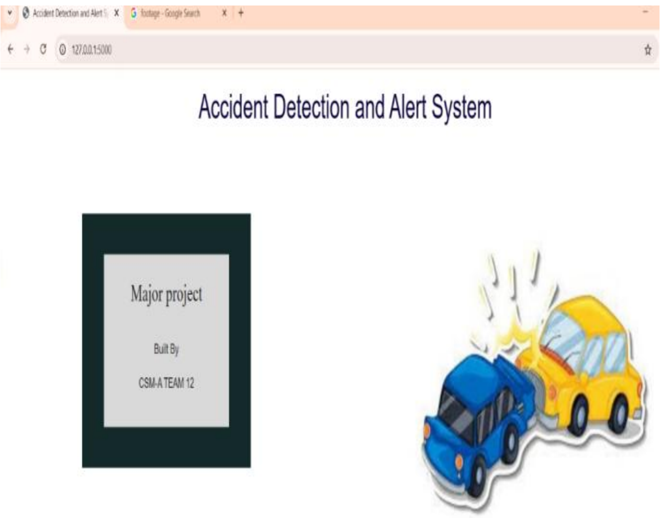

Fig 2: User Interface

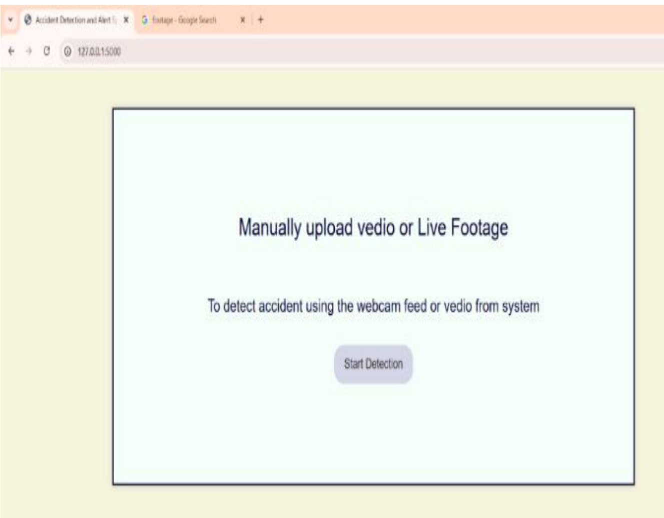

Fig3: Upload CCTV

Fig 4: Accident Detection Result

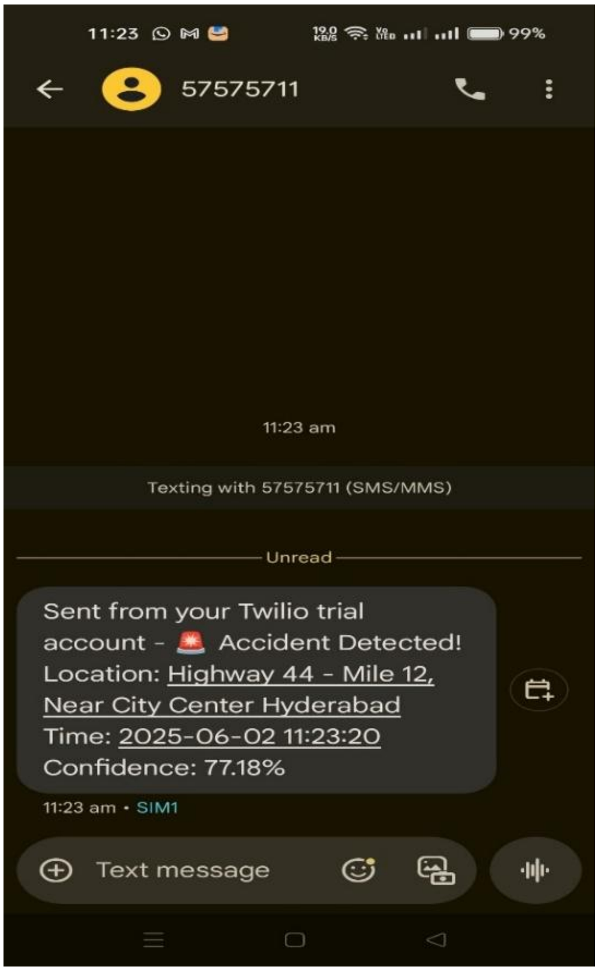

Fig 5: SMS Alert

ISSN: 2583-6129
DOI: 10.55041/ISJEM04591

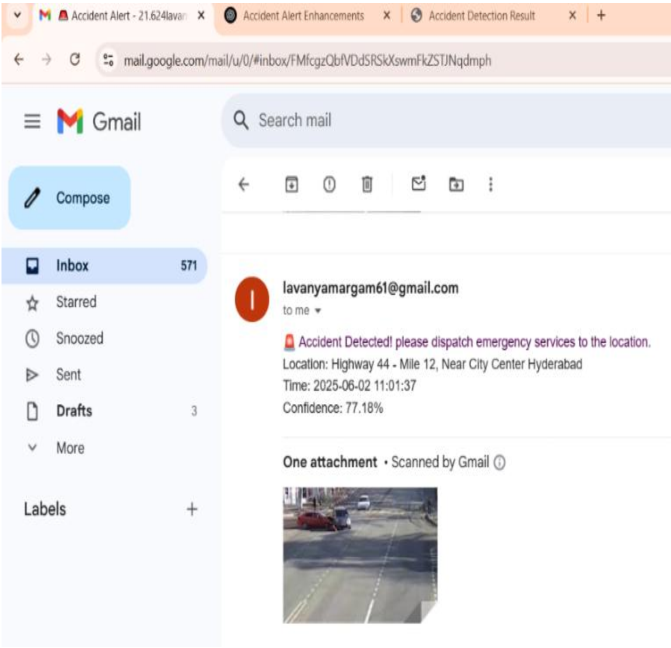

Fig 6: Email Alert

6.CONCLUSION

The Accident Detection System showcases a deliberate combination of artificial intelligence and emergency response that advances real-time safety monitoring towards widespread application. Through computer vision using a trained Convolutional Neural Network (CNN), the system accurately detects possible accidents in real-time, image, and video streams. The incorporation of real time SMS and email notifications, as well as sound-based alerts, guarantees not just visualization of the detection but active notification to pertinent stakeholders.

The web interface of the system based on Flask further enhances usability, with users able to engage in various detection modes within a lightweight, web-accessible environment. Even though it was tested within a localhost setting, the system acted with little lag and strong alert mechanisms, substantiating the design methodology. Nevertheless, some technical and scalability features—e.g., wide-area concurrent usage, cross-platform deployment, and real-world data flexibility—are still points to be improved.

Nevertheless, as it is, the system is a potential safety-first solution that can be life-saving in smart cities, highways, and surveillance settings.

7.FUTURE WORK

As the Accident Detection System progresses, several improvements are envisioned to enhance its real-world applicability, reinforce robustness, and meet large-scale deployment requirements:

Deployment on Cloud and Edge Devices

Subsequent versions will be aimed for deployment on cloud services and IoT-supported edge devices such as Raspberry Pi or Jetson Nano. This will make real-time accident detection in traffic cameras, public areas, and transport infrastructure possible without the necessity of local systems.

GPS-Based Location Alerts

With GPS modules or API-based location tracking integration, the system will provide accident alerts with accurate geolocation coordinates, enhancing emergency response time and effectiveness.

Mobile App Integration

A mobile-app Android/iOS version is in our pipeline, which offers users push notifications, detection logs, and access to live feed, making it more accessible and interactive in real time.

Scalability and Performance Improvements

In order to support high user traffic and video uploads, upcoming versions will support load balancing, video frame sampling, and model optimization methods like quantization or ONNX conversion for quicker inference and reduced latency.

Data Logging and Analytics Dashboard

The admin dashboard will record detection, display analytics such as time-based accident incidents, and provide provision for authorized users to control alert history, assisting with preventive safety planning.

Accessibility and UI Enhancements

Additional work will be on enhancing the interface to become disability-accessible and enhancing the UX for non-technical users. Multi-language compatibility and voice navigation are possible features in consideration. Together, these future improvements will be able to turn this system from a laboratory scale safety device into a scalable, accessible, and smart accident tracking solution with real-world implications.

8.REFERENCES

[1] S. Phulmante, A. Patel, and V. Zope, "Sensor-Based Accident Detection and Alert System," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 7, pp. 342–345, 2020.

[2] A. Adewopo and F. Oloyede, "Action Recognition for Smart City Accident Detection Systems," International Conference on Machine Learning and Cybernetics (ICMLC), 2021, pp. 201–206, doi: 10.1109/ICMLC.2021.9384932.

[3] R. More, S. Pawar, and M. Shinde, "AI and IoT-Based Smart Accident Detection and Alert System," Journal of Emerging Technologies and Innovative Research, vol. 7, no. 5, pp. 115–119, 2020.

[4] P. Gaikwad, S. More, and A. Mane, "Accident Alert & Detection Systems: A Survey," International Journal of

Scientific Research in Computer Science, vol. 8, no. 3, pp. 205–209, 2021.

[5] V. Mahajan, R. Salunkhe, and S. Deshmukh, “IoT and AI-Based Road Accident Detection and Reporting System,” International Journal of Engineering Applied Sciences and Technology, vol. 5, no. 4, pp. 67–71, Aug. 2020.

[6] K. Mehta and T. Patel, “Deep Learning for Traffic Surveillance: Challenges and Opportunities,” IEEE Conference on Smart City Technologies, New Delhi, 2021, pp. 180–185, doi: 10.1109/SCITech.2021.9457164.

[7] A. K. Verma and P. Singh, “Real-Time Accident Monitoring Using OpenCV and Python,” International Journal of Computer Applications, vol. 180, no. 10, pp. 45–49, 2019, doi: 10.5120/ijca2019918526.

[8] S. Sharma and A. Rao, “Smart Cities and AI Surveillance for Road Safety,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 8, pp. 3451–3460, Aug. 2020, doi: 10.1109/TITS.2019.2955763.

9] N. Jaiswal and M. Jain, “Emergency Communication Frameworks in Automated Alert Systems,” International Conference on Communication Systems and Network Technologies (CSNT), 2020, pp. 134–139.

[10] L. Bansal, “Deep Learning Approaches for Real-Time Video Stream Accident Detection,” International Journal of Advanced Research in Computer Science, vol. 11, no. 2, pp. 222–226, Mar.–Apr. 2020.