

# DriveEYE: Intelligent Accident Detection and Reporting System

VIVEK KALE<sup>1</sup>, OMKAR SATHE<sup>2</sup>, VIRAJ JADHAVRAO<sup>3</sup>, RAJLAXMI PATIL<sup>4</sup>  
<sup>1,2,3,4</sup> *Computer Science Engineering, MIT Art, Design & Technology University*

*Abstract- The timely response to traffic accidents is critical for reducing casualties and mitigating property damage. Traditional dashcams, while valuable for recording events, are inherently passive devices, requiring manual review and intervention to initiate any form of response. This paper presents DriveEYE, an intelligent system designed to augment conventional dashcams with advanced AI-powered capabilities, transforming them from simple recording devices into proactive safety tools. DriveEYE actively monitors the video feed in real-time to detect collision events, automatically analyzes the scene to gather crucial evidence, and instantly sends detailed emergency alerts. The system's prototype is developed and demonstrated using standard smartphone and laptop cameras to replicate the video input of a dashcam, showcasing its potential for integration into existing hardware. The proposed solution leverages a sophisticated pipeline: a continuous 30-second pre-crash buffer ensures critical lead-up footage is retained; sudden spikes in motion, detected via optical flow, trigger the incident protocol; a multi-threaded analysis then employs a specialized YOLOv8 model for license plate detection, the Google Gemini API for contextual scene summarization, and Pytesseract for Optical Character Recognition. All collected data—including an AI-generated summary, vehicle screenshots, license plate details, and simulated GPS coordinates—is compiled into a comprehensive PDF report and automatically dispatched to emergency contacts via SMS and email using the Twilio API and smtplib. This low-cost, integrated approach, built in Python, significantly reduces the time from incident occurrence to emergency response, offering a powerful and accessible method for enhancing driver safety and streamlining post-accident procedures.*

*Index Terms- Accident Detection, Computer Vision, Artificial Intelligence, Deep Learning, Emergency Response, License Plate Recognition, Report Generation, Intelligent Transportation Systems, Dashcam Augmentation*

## I. INTRODUCTION

In the modern transportation landscape, road safety remains a significant and persistent global concern. Statistics from organizations like the World Health Organization (WHO) consistently show that road traffic accidents cause a massive number of fatalities and injuries, as well as enormous economic losses each year. A critical factor in the aftermath of such tragedies is the time that elapses from the moment an accident occurs to the arrival of professional emergency services. This period, often referred to as the "golden hour," is vital for maximizing survival chances and minimizing the severity of injuries. Traditional dashcams have become increasingly popular among drivers for their ability to record continuous video footage, providing an objective record of events that can be invaluable for insurance claims and legal disputes. However, these devices are fundamentally passive; they simply record data, leaving the tasks of incident detection, evidence analysis, and emergency notification entirely to human intervention. This reliance on manual action introduces significant delays, especially if the driver is incapacitated, disoriented, or unaware that an incident has been recorded. There is a clear and pressing need to evolve these passive recording tools into active, intelligent safety systems that can autonomously detect accidents and initiate a rapid response.

The advent of sophisticated artificial intelligence (AI) and computer vision technologies presents a transformative opportunity to bridge this gap. These advanced computational tools can now analyze visual data with a level of speed and nuance that rivals or exceeds human capability, enabling the development of systems that are not merely reactive but proactive. In the context of in-vehicle safety, AI can be employed to continuously monitor video feeds,

identify the complex visual signatures of a collision in real-time, and trigger a cascade of automated actions without any human input. The research community has made significant strides in this area, demonstrating the efficacy of deep learning models for tasks such as accident detection from video streams. For example, studies have shown that algorithms like YOLOv8 can be effectively trained to identify traffic accidents in real-time using CCTV footage, providing a foundation for automated alert systems. While such research is promising, its application has largely been confined to infrastructure-based surveillance. The potential to embed this intelligence directly into a vehicle, by augmenting a common device like a dashcam, remains a largely underexplored frontier with immense practical implications.

DriveEYE is conceived to address this very opportunity. It is not about creating a new recording device from scratch, but rather about adding a layer of intelligent "awareness" to existing dashcam technology. The core philosophy is to transform a standard, passive dashcam into an active safety sentinel. This paper details the development and implementation of a prototype system that demonstrates this capability. For the purposes of research and demonstration, the system utilizes the cameras from readily available devices like smartphones and laptops to simulate the video feed of a traditional dashcam. This approach allows for flexible and cost-effective development and testing of the underlying AI and software pipeline. The ultimate vision, however, is the direct integration of this intelligent software into the firmware or hardware of future dashcams, thereby democratizing access to advanced safety features. By automating the entire process from detection to reporting, DriveEYE aims to drastically cut down on critical response times, provide first responders with immediate, actionable information, and ultimately, contribute to saving lives and reducing the severity of road accident outcomes.

## II. SYSTEM ARCHITECTURE AND IMPLEMENTATION

DriveEYE is designed as a modular, Python-based software engine designed to process a continuous video stream, emulate the function of an advanced

dashcam, and execute a fully automated incident response workflow. The system is built with a clear separation of concerns, where distinct modules handle specific tasks such as video ingestion, accident detection, intelligent scene analysis, and alert dissemination. This modular design not only simplifies development and debugging but also ensures that the system can be easily adapted or extended in the future. For prototyping and validation, the video input is sourced from the built-in cameras of standard laptops or smartphones, streamed over a local network. This method effectively replicates the primary function of a traditional dashcam—providing a continuous view of the road ahead—allowing the core AI logic to be developed and tested without the need for specialized hardware. The overarching goal is to create a software core that can eventually be embedded into dedicated dashcam hardware, transforming it from a passive recorder into an intelligent safety system. The complete workflow, from video input to emergency notification, is illustrated in Fig. 1.

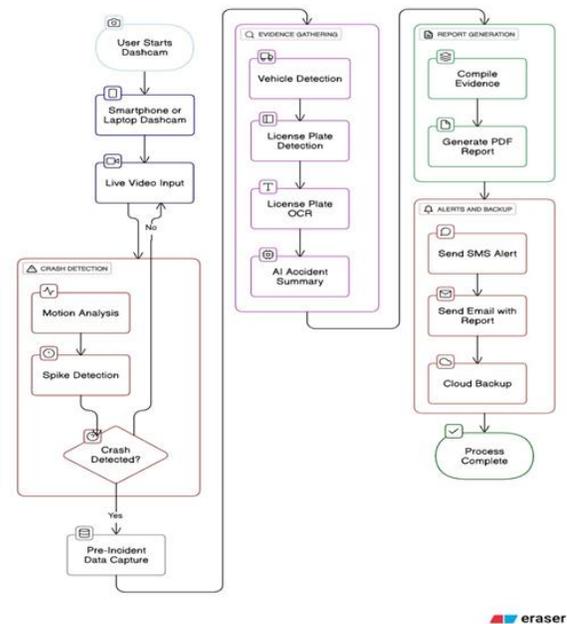


Fig. 1. System architecture of the DriveEYE system, demonstrating how a standard video feed is processed to detect accidents and generate automated reports, thereby augmenting a traditional dashcam's capabilities.

The first component in the pipeline is the Video Input and Buffering module, which is responsible for acquiring and managing the video data. Utilizing the OpenCV library, the system can connect to various video sources, including local video files for testing or live network streams from a smartphone camera acting as an IP camera. A critical and innovative feature of this module is the implementation of a continuous, in-memory, 30-second rolling buffer. As video frames are captured, they are added to this buffer, which automatically discards frames older than 30 seconds. This ensures that the system always has access to the most recent half-minute of footage, capturing the crucial moments immediately preceding an incident. This pre-crash video is invaluable for understanding the context and cause of an accident, information that is typically lost with simple, motion-activated recording.

The core of the system's intelligence lies in the Accident Detection module. This module operates in real-time, analyzing the incoming video stream for the characteristic signatures of a collision. The detection algorithm is based on dense optical flow, a computer vision technique for estimating the motion of objects between consecutive frames. By calculating the magnitude of the overall motion vectors across the entire frame, the system can detect sudden, violent jolts that are indicative of a crash. When the calculated motion magnitude exceeds a predefined, empirically determined threshold, the system flags the event as a potential accident. This method is computationally efficient and does not require a large, labeled dataset of accident videos for training, making it ideal for a robust, real-time detection system.

Once an accident is detected, the system immediately transitions to the Intelligent Scene Analysis module. This is a multi-threaded process designed to rapidly extract and compile a comprehensive set of evidence. The first action is to permanently save the 30-second pre-crash video buffer to the disk, ensuring this critical footage is preserved. Concurrently, several AI-powered sub-tasks are initiated:

- **AI-Powered Scene Summarization:** Key frames from the incident are sent to the Google Gemini API, a multimodal large language model. Gemini

analyzes these frames and generates a concise, natural-language description of the accident (e.g., "Vehicle A abruptly changed lanes and struck Vehicle B").

- **License Plate Detection and Recognition:** A specialized YOLOv8 model, fine-tuned specifically for license plate detection, is applied to the frames to locate the license plate of any other involved vehicle. The detected plate region is then processed by Pytesseract, an Optical Character Recognition (OCR) engine, to extract the alphanumeric characters. The effectiveness of combining YOLO for detection and OCR for text extraction is a well-established approach in automated license plate recognition systems
- **Vehicle Identification:** A general-purpose YOLOv8 model, trained on a large dataset of common objects, is used to identify and capture a clean screenshot of the other vehicle involved in the incident.

The final stage is the Automated Reporting and Alerting module. All the collected data—the AI-generated summary, the precise timestamp, the extracted license plate text, the vehicle screenshot, and simulated GPS coordinates (obtained via a geocoding library for prototyping)—is aggregated. The fpdf2 library is then used to programmatically generate a detailed and well-formatted PDF report that serves as a comprehensive evidence package. To ensure immediate notification, the system employs a dual-channel alerting strategy. Using the Twilio API, a concise SMS message with critical details is sent to a pre-defined emergency contact number. Simultaneously, an email, with the complete PDF report attached, is dispatched using Python's built-in smtplib. This automated workflow ensures that relevant parties are alerted within minutes of an incident, providing them with crucial information that can expedite emergency response and post-accident procedures.

### III. KEY MODULES AND TECHNOLOGIES

The robust functionality of the DriveEYE prototype is achieved through the strategic integration of several key technological modules. Each component plays a vital role in the system's ability to accurately

detect accidents, analyze complex scenes, and communicate critical information. The choice of technology for each module was driven by the need for real-time performance, accuracy, and ease of integration within a Python-based environment.

**Core Language and Environment:** The entire DriveEYE system is developed in Python 3, leveraging its extensive and mature ecosystem of libraries for scientific computing, artificial intelligence, and system automation. This choice allows for rapid prototyping and a high level of abstraction. To manage sensitive credentials such as API keys for external services like Twilio and Google Gemini, the `python-dotenv` library is utilized. This library loads configuration variables from a `.env` file into the environment, ensuring that secrets are not hard-coded into the source code, which is a critical security best practice.

**Computer Vision and Video Processing:** OpenCV (Open Source Computer Vision Library) is the cornerstone of all video and image manipulation tasks within DriveEYE. Its highly optimized C++ core, accessed through Python bindings, provides efficient functions for capturing video streams from various sources (including IP cameras and local files), decoding video frames, and performing a wide array of image processing operations. Crucially, OpenCV is used to implement the dense optical flow algorithm for motion-based accident detection and to manage the 30-second rolling video buffer. The Pillow library (a fork of PIL) is also employed for specific image manipulation tasks, such as converting image formats and adjusting properties before they are sent to AI models or APIs.

**Artificial Intelligence and Machine Learning:** The AI capabilities of DriveEYE are powered by a combination of locally-run, custom-trained models and cloud-based API services.

- **Ultralytics YOLOv8:** You Only Look Once version 8 (YOLOv8) is a state-of-the-art, real-time object detection framework. DriveEYE leverages two instances of YOLOv8 models. The first is a general-purpose model, pre-trained on the extensive COCO dataset, used for detecting common objects like vehicles, pedestrians, and

traffic signs. The second is a specialized model that has been fine-tuned on a custom dataset of license plate images to accurately localize license plates in various conditions. YOLOv8 was chosen for its exceptional balance of speed and accuracy, making it ideal for real-time applications on resource-constrained hardware.

- **Google Gemini API:** The multimodal capabilities of Google's Gemini large language model are harnessed for advanced scene understanding. By providing the API with key image frames from the accident, DriveEYE can generate a human-readable, contextual summary of the event. This moves beyond simple object detection to provide semantic meaning, which is invaluable for creating an immediate narrative of the incident for emergency responders.
- **Pytesseract:** This Python wrapper for Google's Tesseract-OCR engine is used for the critical task of extracting text from the license plate images identified by the YOLOv8 model. While its accuracy can be influenced by image quality, it provides a robust, open-source solution for OCR, which is a fundamental component of any automated license plate recognition system.

#### Reporting and Notifications:

- **fpdf2:** This library is used for the programmatic generation of the final accident report in PDF format. It allows the system to dynamically create a professional-looking document that seamlessly integrates text, images (such as the vehicle screenshot), and structured data, ensuring all collected evidence is presented in a clear and organized manner.
- **Twilio API:** For immediate, near-instantaneous alerting, DriveEYE utilizes the Twilio API to send SMS messages. SMS is chosen for its high reliability and ubiquity, ensuring that the emergency alert is likely to be seen quickly, even if the recipient has limited internet connectivity.
- **smtplib:** Python's standard library for sending emails (`smtplib`) is used to dispatch the comprehensive PDF report. Email serves as the channel for delivering the detailed documentation, ensuring that all evidence is securely and reliably transmitted to the intended recipients for further action.

- Geocoder: To provide location context in the prototype, this library is used to simulate GPS coordinates based on the system's public IP address. In a production environment integrated into a vehicle or mobile device, this module would be replaced with a direct interface to the device's GPS sensor for accurate location data.

#### IV. RESULT AND DISCUSSION

The primary outcome of this research is a fully functional prototype of the DriveEYE system, which successfully demonstrates a complete, end-to-end automated workflow for accident detection and reporting. The system was rigorously tested using a diverse dataset of pre-recorded dashcam videos featuring various collision scenarios, including rear-end collisions, side-impacts, and multi-vehicle accidents. In all test cases where a significant motion spike characteristic of a crash occurred, the system's detection module reliably triggered the incident response protocol. The subsequent multi-threaded analysis pipeline consistently executed its designated tasks: the pre-crash video buffer was successfully saved, the Google Gemini API generated coherent and contextually relevant summaries of the accidents, the YOLOv8 models accurately identified vehicles and localized license plates, Pytesseract extracted readable text from the detected plates, and the fpdf2 library compiled a well-structured PDF report. Finally, the notification modules reliably dispatched SMS alerts and emails with the attached reports.

The workflow efficiency of DriveEYE represents a substantial leap forward compared to traditional, manual processes. The entire cycle—from the moment of impact to the delivery of a detailed report to an emergency contact—is fully automated and completes within a matter of minutes. This rapid turnaround is critical in emergency scenarios, where reducing response time by even a few minutes can have a profound impact on casualty severity and long-term outcomes. The system's ability to be prototyped using standard consumer hardware like laptops and smartphones underscores its potential for cost-effective development and its adaptability. This approach demonstrates that advanced safety features need not be confined to premium, high-end vehicles

but can be made accessible to a much wider audience through software augmentation of existing devices.

The quality and comprehensiveness of the automated reports are particularly promising. The AI-generated summaries from the Google Gemini API provide an immediate, human-readable narrative of the incident, which can offer first responders crucial situational awareness before they even arrive on the scene. The combination of YOLOv8 for object detection and Pytesseract for OCR provides a robust mechanism for gathering key evidence, such as the other vehicle's license plate, which is essential for post-accident procedures. The final PDF report serves as a self-contained, time-stamped evidence package, consolidating all critical visual and textual information into a single, easily shareable document. This automated evidence collection directly addresses a major challenge in accident investigations, which often rely on fragmented, manually gathered, and sometimes unreliable information.

However, as a proof-of-concept prototype, DriveEYE has several limitations that define important avenues for future work. The most significant limitation is the use of simulated GPS data via IP geocoding, which lacks the precision and reliability required for real-world emergency response. A production system must integrate with a dedicated GPS sensor to provide accurate location coordinates. The accuracy of the OCR process is another area for improvement; while generally effective, its performance can degrade under challenging conditions such as poor lighting, motion blur, or unusual license plate fonts. Future versions could incorporate more advanced, deep learning-based OCR models or employ post-processing techniques to improve text recognition accuracy. Furthermore, the current accident detection algorithm, based on a global motion threshold, could be refined to better distinguish between actual collisions and other high-motion events like aggressive driving maneuvers or driving on rough roads, thereby reducing the potential for false positives. Finally, while the current software prototype demonstrates the core logic, the ultimate goal is hardware integration. Future development will focus on optimizing the software for embedded systems and exploring partnerships with dashcam

manufacturers to integrate DriveEYE's intelligent capabilities directly into consumer products.

## V. CONCLUSION

This paper has detailed the design and implementation of DriveEYE, an intelligent system engineered to augment traditional, passive dashcams with advanced, AI-powered safety features. By actively monitoring video feeds for collision events, automatically analyzing scenes to extract critical evidence, and generating instant, comprehensive emergency alerts, DriveEYE transforms a simple recording device into a proactive safety tool. The successful development of this prototype demonstrates a significant step toward automating the critical first steps in post-accident response, thereby addressing the crucial need for speed and efficiency in emergency situations.

The primary contribution of this work lies in the creation of a functional, end-to-end pipeline that seamlessly integrates diverse technologies—from computer vision and deep learning to automated communication—to solve a real-world, high-impact problem. The system's ability to detect accidents, generate contextual summaries, identify license plates, and dispatch detailed reports within minutes showcases the transformative potential of AI in enhancing road safety. While the current prototype has limitations, particularly regarding GPS accuracy and the robustness of OCR under all conditions, its modular architecture provides a solid foundation for future development and refinement. DriveEYE represents a promising advancement toward a future where intelligent, automated systems are an integral part of every vehicle, working tirelessly to protect drivers and passengers and to provide invaluable support to emergency services when it matters most.

## REFERENCES

- [1] Real-time traffic accident detection using yolov8.  
<https://www.sciencedirect.com/science/article/pii/S2352146525001942>.
- [2] Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for

high-precision license plate detection and recognition.

<https://www.nature.com/articles/s41598-024-65272-1>.

- [3] EmergEye: AI-Based Car Accident Detection and Notification System.  
<https://www.ischool.berkeley.edu/projects/2024/emergeye-ai-based-car-accident-detection-and-notification-system>.