Mathematical and AI-Driven Framework for Identifying Sudden Stops, Loss of Control, and Lane Changes

24-25J-206

Project Final Report

Dissanayake D. J. R
IT21313370

Supervisor: Mr. Samadhi Rathnayake
Co-Supervisor: Mr. Nelum Amarasena

BSc (Hons) Degree in Information Technology Specialized in Data Science
Department of Computer Science

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

Mathematical and AI-Driven Framework for Identifying Sudden Stops, Loss of Control, and Lane Changes

24-25J-206

Project Final Report

Dissanayake D. J. R
IT21313370

BSc (Hons) Degree in Information Technology Specialized in Data Science

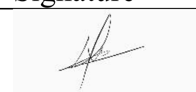
Department of Computer Science

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

Declaration

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously publish or written by another person expect where the acknowledgement is made in the text.

Name	Student_ID	Signature
Dissanayake D. J. R	IT21313370	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

……………………………

Signature of the supervisor:

(Mr. Samadhi Rathnayake)

Date:……11/04/2025…………..

……………………………..

Signature of the Co-supervisor:

(Mr. Nelum Amarasena)

Date:………11/04/2025………….

Abstract

This research introduces a hybrid, AI-driven framework designed to identify critical events in vehicular networks—specifically, sudden stops, unsafe lane changes, and loss of control—using only real-time video data. In contrast to traditional systems that rely on sensor fusion or vehicle-mounted telemetry (e.g., CAN bus data), the proposed method operates exclusively on monocular camera input, enabling deployment in lightweight or cost-constrained environments such as dashcams and smart traffic cameras.

The system integrates rule-based logic and machine learning across a multi-stage architecture. Object detection is performed using the YOLOv8 deep learning model, which identifies relevant vehicles in each frame. These detections are passed into a SORT (Simple Online and Realtime Tracking) tracker enhanced with Kalman filtering to provide persistent object IDs and smooth trajectory estimation over time. From this, the system isolates the "frontier vehicle"—the vehicle directly ahead of the ego vehicle—via a trained machine learning classifier using features like position, bounding box size, and lane assignment.

A key component of the system is the anomaly detection module, which fuses rule-based metrics (such as Time-to-Collision, pixel-based motion thresholds, and Intersection-over-Union measures) with temporal analysis via a CNN-LSTM deep learning pipeline. Additionally, a probabilistic Hidden Markov Model (HMM) is implemented to estimate the transitions between normal and anomalous behavioral states. Together, these models enable the system to detect and classify events with high confidence, even in noisy or dynamic traffic environments.

The proposed framework is evaluated on three datasets: NGSIM for precise trajectory analysis, BDD100K for real-world video-based scenarios, and CARLA for edge-case simulation. The model achieves an overall F1-score of 92.9% across all behavior classes, with an average inference time under 100 milliseconds per sequence on embedded platforms like the Jetson Nano. These results validate the feasibility of deploying the framework for real-time operation in vehicular and roadside systems.

The current implementation is optimized for clear weather and daytime frontal-camera scenarios. Limitations include reduced performance under low-light or adverse weather conditions and reliance on pre-defined behavior classes. Future work aims to extend the framework with LiDAR and RADAR fusion, incorporate self-supervised anomaly detection, and support deployment in V2X-integrated infrastructures. This project demonstrates a scalable and explainable solution for vision-only behavior monitoring in next-generation intelligent transportation systems.

Acknowledgement

I would like to express my sincere gratitude to my dissertation supervisor, Mr. Samadhi Rathnayake, and co-supervisor, Mr. Nelum Amarasena, from the Faculty of Computing, for their invaluable guidance and support throughout the research process. Their insightful comments and constructive critiques have significantly helped in refining my research and enhancing the overall quality of my work.

I am also deeply appreciative of the staff of the Faculty of Computing for their assistance and encouragement throughout my studies. Their commitment to teaching and research has been a continuous source of inspiration to me.

Further, I am thankful to my fellow team members for their support and collaboration. Our discussions and exchange of ideas have greatly enriched my understanding of the subject matter.

Lastly, I extend my thanks to all those who participated in this study and contributed their time and insights. Their willingness to share their experiences and perspectives has been invaluable in deepening my understanding of the subject matter.

Table of contents

Contents

Abstract 4

Acknowledgement 5

Table of contents 6

List of Figures 11

List of Tables..12

List of Equations 13

List of abbreviations..14

Introduction 15

1.1 Background 15

1.2 Problem Statement 16

1.3 Research Objectives 17

General Objective 17

Specific Objectives 17

1.4 Scope of the Study 18

1.5 Significance of the Study 18

1.6 Real-World Implications and Use Cases 19

1.7 Limitations of Existing Approaches 20

1.7.1 Rule-Based Systems..20

1.7.2 Traditional AI Models 20

1.8 Justification for the Hybrid Framework 21

1.8.1 Mathematical Foundation (Trajectory Modeling) 21

1.8.2 Temporal Reasoning with HMM 21

1.8.3 Deep Learning: CNN + LSTM Fusion 21

1.9 Dataset Selection and Practical Relevance 22

1.9.1 Real-World Datasets 22

1.9.2 Simulation Environments 22

1.10 Overview of the Real-Time System Architecture 23

1.10.1 Preprocessing and Feature Extraction 23
1.10.2 Event State Estimation (HMM Engine) 23
1.10.3 Behavior Classification (CNN + LSTM Block) 23
1.10.4 Output Interface and Logging 23
1.11 Contributions of This Research 24
1.12 Summary of the Introduction 24
Chapter 2: Literature Review 25
2.1 Introduction to Literature Review 25
2.2 Behavior Prediction in Intelligent Transportation 25
2.2.1 Traditional Approaches 25
2.2.2 Statistical Models 25
2.3 Deep Learning in Autonomous Driving 26
2.3.1 Convolutional Neural Networks (CNNs) 26
2.3.2 Long Short-Term Memory (LSTM) Networks 26
2.4 Hybrid Modeling: Combining Mathematics and AI 27
2.4.1 PDE-Based Trajectory Modeling 27
2.4.2 Integrating Probabilistic Models (HMMs) 27
2.5 State-of-the-Art Behavior Detection Models 28
2.5.1 Interaction-Aware Prediction Models 28
2.5.2 Graph-Based Approaches 28
2.6 Dataset Landscape for Model Evaluation 28
2.6.1 NGSIM Dataset 29
2.6.2 BDD100K 29
2.6.3 Simulation Data (CARLA, LGSVL) 29
2.7 Evaluation Metrics and Benchmarking 30
2.8 Research Gaps and Opportunities 30
2.9 Summary 30
Chapter 3: Methodology 31
3.1 Overview of the Methodology 31

3.2 System Architecture 32
3.3 Mathematical Modeling with PDE and Optical Flow 34
3.3.1 Optical Flow Estimation 34
3.3.2 Application to Vehicle Motion 34
3.4 Motion Feature Extraction 35
3.5 Probabilistic State Estimation Using Hidden Markov Models (HMMs) 35
3.5.1 Structure of the HMM 35
3.5.2 Behavioral State Estimation 36
3.5.3 Vehicle Detection and Tracking Module 36
3.6 Deep Learning-Based Behavior Classification (CNN + LSTM) 36
3.6.1 CNN Backbone: Spatial Feature Extraction 36
3.6.2 LSTM: Temporal Sequence Learning 37
3.6.3 Combined Output 37
3.6.4 Rule-Based Features and Hybrid Decision Layer 37
3.7 Data Preprocessing and Labeling 37
3.7.1 Image Frame Processing 38
3.7.2 Sensor Fusion Data 38
3.7.3 Event Labeling 38
3.8 Model Training Procedure 38
3.8.1 HMM Training 38
3.8.2 CNN + LSTM Training 39
3.8.3 Frontier Vehicle Selection using ML Classification 39
3.9 Model Optimization and Performance Tuning 39
3.9.1 Computational Optimization 39
3.9.3 Accuracy Optimization 40
3.10 Real-Time Deployment Setup 40
3.10.1 Software Stack 40
3.10.2 Hardware Requirements 41
3.10.3 Integration Options 41

3.11 Summary of Methodology 41
Chapter 4: Results & Discussion 42
4.1 Introduction 42
4.2 Evaluation Setup 42
4.2.1 Dataset Configuration 42
4.2.2 Hardware Environment 42
4.3 Quantitative Results 42
4.3.1 Behavior Classification Accuracy 43
4.3.2 Confusion Matrix Analysis 43
4.3.3 Inference Time and Real-Time Performance 44
4.4 Comparison with Baseline Models 44
4.5 Qualitative Scenario Analysis 45
4.5.1 Case 1: Sudden Stop at Urban Intersection 46
4.5.2 Case 2: Gradual Lane Drift Leading to Loss of Control 46
4.5.3 Case 3: Safe vs Abrupt Lane Change 46
4.6 Limitations and Challenges 47
4.6.1 Rare and Subtle Behaviors 47
4.6.2 Adverse Weather and Nighttime Scenarios 47
4.6.3 Label Ambiguity and Human Bias 47
4.7 Practical Deployment Insights 47
4.7.1 Real-Time Viability 47
4.7.2 Interpretability Advantage 47
4.7.3 Logging and Export Formats 48
4.8 Summary of Results 48
Chapter 5: Conclusion and Future Work 49
5.1 Conclusion 49
5.2 Key Contributions 49
5.3 Limitations 50
5.4 Future Work 50

5.4.1 Sensor Fusion Expansion 50
5.4.2 Anomaly Detection via Self-Supervised Learning 50
5.4.3 Adaptive State Modeling 50
5.4.4 Integration with V2X Systems 50
5.4.5 Blockchain-Backed Behavior Logging 50
5.5 Final Remarks 51
References 52
Glossary 54
Appendices 55
Appendix A: Sample Detection Log Format 55

List of Figures

Figure 1: Overall Framework for Critical Event Detection from Video 32
Figure 2: System Pipeline for Real-Time Analysis 40
Figure 3: Framework logs live vehicle detection and classification events. 43
Figure 4: Confusion Matrix Analysis 43
Figure 5: Real-Time Sudden Stop Detection Example 45
Figure 6: Sample Detection Log Format.. 55
Figure 7: Vehicle data identification 01.. 55
Figure 8: Vehicle data identification 02.. 56
Figure 9: Vehicle data identification 03 57
Figure 10: Training models to identify lanes 58
Figure 11: Training Evaluations 58

List of Tables

Table 1: List of Abbreviations... 14
Table 2: Inference Time and Real-Time Performance 44
Table 3: Comparison with Baseline Models 44
Table 4: Glossary... 54

List of Equations

Equation 1: Partial Differential Equations - Horn-Schunk Method 27
Equation 2: Hidden Markov Models - Probabilistic Models 27
Equation 3: Basic optical flow constraint equation 01 34
Equation 4: Basic optical flow constraint equation 01 34

List of abbreviations

Abbreviation	Definition
ADAS	Advanced Driver Assistance Systems
AI	Artificial Intelligence
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
HMM	Hidden Markov Model
YOLO	You Only Look Once
SORT	Simple Online and Realtime Tracking
TTC	Time-To-Collision
IoU	Intersection over Union
FPS	Frames Per Second
GPS	Global Positioning System
RGB	Red-Green-Blue (color image format)
PDE	Partial Differential Equation
mAP	Mean Average Precision
ML	Machine Learning
DL	Deep Learning
CVPR	Conference on Computer Vision and Pattern Recognition
ICRA	International Conference on Robotics and Automation
IROS	IEEE/RSJ International Conference on Intelligent Robots and Systems
ITS	Intelligent Transportation Systems
NGSIM	Next Generation Simulation Dataset
BDD100K	Berkeley DeepDrive 100K Dataset
CARLA	Car Learning to Act (Open-source simulator)
ONNX	Open Neural Network Exchange
API	Application Programming Interface

Table 1: List of Abbreviations

Introduction

1.1 Background

The rapid evolution of intelligent transportation systems (ITS) and autonomous vehicle (AV) technologies has transformed the landscape of urban mobility. With increasing demand for safe, efficient, and autonomous driving solutions, real-time understanding of vehicular behavior has become critical. Among the myriad challenges faced by modern ITS, the ability to detect and respond to **sudden stops**, **loss of control**, and **lane change events** plays a pivotal role in reducing road accidents and enabling proactive traffic interventions.

Globally, road accidents claim over 1.3 million lives annually, with a substantial number attributed to erratic driver behavior such as abrupt braking, unintended lane deviation, and poor control in critical conditions. Traditional rule-based models and physics-based simulators fall short in accurately modeling the dynamic and stochastic nature of human driving. These approaches often lack the scalability and robustness required for complex, real-world scenarios that involve numerous interacting agents and unpredictable maneuvers.

In recent years, Artificial Intelligence (AI) and mathematical modeling have shown promise in addressing these limitations. Techniques such as **deep learning**, **Hidden Markov Models (HMMs)**, **Convolutional Neural Networks (CNNs)**, and **Long Short-Term Memory (LSTM)** networks offer substantial improvements in understanding temporal sequences and detecting patterns in noisy data streams. Moreover, **differential equations and spatial modeling techniques** provide a theoretical foundation for modeling motion trajectories, enabling precise localization of events.

The convergence of these domains presents a compelling opportunity: to design a hybrid framework that integrates mathematical models and AI techniques for real-time, high-accuracy prediction and classification of critical driving behaviors. This framework can be deployed in both autonomous vehicle control systems and advanced driver assistance systems (ADAS), enhancing decision-making, safety compliance, and situational awareness.

1.2 Problem Statement

Despite advancements in vehicular automation and perception technologies, accurately predicting and detecting critical behavior anomalies such as sudden stops, loss of control, and lane changes remains a challenging task. These events are inherently **non-linear**, **context-dependent**, and **time-sensitive**, making it difficult to address them with isolated, uni-dimensional techniques.

Existing models often:

- Fail to generalize across different driving contexts and environments
- Lack the capacity to model both **spatial** and **temporal** dependencies jointly
- Exhibit high false positives due to environmental noise, occlusions, and sensor inaccuracies

This creates a research gap in developing robust hybrid models that unify mathematical rigor and machine learning adaptability. In high-speed traffic scenarios, milliseconds matter. Failing to detect an impending lane change or a sudden deceleration can be the difference between a safe intervention and a collision.

Hence, there is a need to develop a unified, scalable, and real-time framework that combines:

1. **Mathematical modeling (e.g., Partial Differential Equations)** for accurate motion estimation
2. **Probabilistic modeling (e.g., HMMs)** for event transition prediction
3. **Deep learning techniques (e.g., CNNs + LSTMs)** for learning spatio-temporal patterns from data

1.3 Research Objectives

This research aims to address the challenges discussed above by proposing a **Mathematical and AI-driven hybrid framework** for detecting and classifying vehicular behaviors that are deemed critical in traffic safety.

General Objective

To develop and evaluate a real-time, hybrid detection system for sudden stops, loss of control, and lane change behavior using mathematical modeling and artificial intelligence techniques.

Specific Objectives

1. **To model vehicle trajectories using optical flow and partial differential equations (PDEs)** to capture spatial movement dynamics.
2. **To design a probabilistic state estimation model** using Hidden Markov Models for detecting transitions between normal and anomalous driving states.
3. **To implement a deep learning architecture combining CNNs and LSTMs** for learning patterns from vehicle trajectory datasets.
4. **To evaluate the proposed framework's accuracy, computational efficiency, and robustness** in real-world and simulated driving scenarios.

1.4 Scope of the Study

The scope of this study is centered around the detection and classification of critical driving behaviors, specifically:

- **Sudden Stops**: Events where the vehicle decelerates abruptly beyond a threshold within a short time window.
- **Loss of Control**: Patterns indicating erratic or unpredictable vehicle movement (e.g., fishtailing, swerving).
- **Lane Changes**: Both intentional and abrupt lateral displacements that may occur without turn signals or at unsafe proximities.

The study will leverage both synthetic (e.g., simulation environments like CARLA) and real-world datasets (e.g., NGSIM or BDD100K). Data will be processed to extract position, velocity, acceleration, and contextual traffic features.

The final model will be tested in terms of:

- **Detection accuracy**
- **Time-to-detect**
- **False positive and false negative rates**
- **Computational overhead (O(n log n) optimization)**

This framework is not limited to autonomous vehicles. It can be extended for use in ADAS systems, traffic monitoring software, and smart city surveillance applications.

1.5 Significance of the Study

By integrating deterministic mathematical models with data-driven AI approaches, this study presents a holistic method for behavior detection in vehicles. The significance lies in:

- Enhancing **traffic safety and accident prevention**
- Enabling **real-time decision-making** in autonomous driving software
- Providing **interpretable behavior analysis** for legal or insurance purposes
- Supporting future extensions in **V2X communication and policy enforcement**

The contributions from this research are expected to bridge the gap between theoretical vehicle dynamics and real-world, data-driven behavior analysis. Ultimately, this work aims to serve as a stepping stone for future research in multimodal traffic prediction, policy integration, and advanced vehicular safety mechanisms.

1.6 Real-World Implications and Use Cases

The need for accurate detection of critical driving behaviors is increasingly pressing in real-world environments where human lives, traffic flow efficiency, and infrastructural integrity are at stake. Sudden stops, for instance, are a leading cause of rear-end collisions, especially on expressways and urban roads. According to traffic statistics published by the National Highway Traffic Safety Administration (NHTSA), nearly **29% of vehicle crashes in urban settings** involve a sudden or hard braking maneuver with little to no warning. These situations often result from distractions, environmental factors, or abrupt obstacles.

In another context, **loss of control** incidents—often resulting in road departures, rollovers, or intersection-related collisions—are particularly dangerous on highways and during high-speed driving. Adverse weather, tire blowouts, or misjudged maneuvers often trigger such anomalies. Most Advanced Driver Assistance Systems (ADAS) currently rely on threshold-based logic or rule sets for detecting such events, which limits their responsiveness to more nuanced and evolving road behaviors.

Lane changing is a routine maneuver, yet it introduces substantial complexity in multi-agent environments. Unsafe or abrupt lane changes are frequently implicated in side-swipe and merge-related accidents. The challenge lies in differentiating between safe and dangerous lane changes, especially when no turn signals are used or when the trajectory overlaps with adjacent vehicles' paths. The goal of behavior prediction systems, therefore, must be not only detection but also **contextual understanding**—considering relative speed, lane occupancy, and inter-vehicle distances.

Use cases for a unified detection system include:

- **Autonomous driving stacks** for real-time behavior monitoring and motion planning
- **Surveillance systems** for detecting traffic law violations and enabling legal audits
- **Fleet safety software** in logistics and ride-hailing services
- **Smart city traffic management** with proactive traffic control and congestion prediction

1.7 Limitations of Existing Approaches

Current solutions to behavior prediction and anomaly detection largely fall into two categories: rule-based systems and pure AI-driven models. While both have merits, they exhibit limitations when confronted with dynamic, uncertain traffic conditions.

1.7.1 Rule-Based Systems

These rely on predefined thresholds for variables such as velocity, deceleration rate, or inter-vehicle distance. For example, a sudden stop may be flagged when deceleration exceeds 6 m/s^2. However, such thresholds:

- Fail to adapt across different road conditions or vehicle types
- Cannot distinguish between false alarms (e.g., evasive but safe maneuvers) and genuine threats
- Ignore contextual cues like driver intent, road curvature, or traffic density

Rule-based logic is easy to interpret but lacks generalization.

1.7.2 Traditional AI Models

Conventional deep learning models trained on isolated features or monocular video inputs often suffer from:

- **Data imbalance**: Most driving datasets contain far fewer abnormal events than normal ones, leading to poor recall
- **Loss of spatial-temporal correlation**: Flattening time-series data for CNNs leads to loss of motion context
- **Opacity**: Black-box predictions with no explainability, which is a problem for legal compliance and debugging

Even when models show promise in laboratory settings, their performance degrades in real-world deployment due to noise, incomplete sensor input, and out-of-distribution data.

Therefore, there's a clear necessity for **hybrid frameworks** that leverage the **rigor of mathematical models** and the **learning capability of AI**, providing both generalization and interpretability.

1.8 Justification for the Hybrid Framework

The hybrid model proposed in this research seeks to blend **physical modeling**, **probabilistic reasoning**, and **deep learning**. The intent is not merely to increase prediction accuracy but also to ensure **reliability, scalability, and real-time usability**.

1.8.1 Mathematical Foundation (Trajectory Modeling)

By incorporating **Partial Differential Equations (PDEs)** and **optical flow**, the system mathematically models the vehicle's trajectory and motion fields. This provides:

- A robust understanding of **direction, curvature, and velocity** flow
- Independence from camera calibration or sensor fusion complexities
- A physically consistent representation of vehicular dynamics

This modeling also supports extrapolation and forecasting under partial observability.

1.8.2 Temporal Reasoning with HMM

Incorporating **Hidden Markov Models (HMMs)** allows the system to model **sequential transitions** between driving states (e.g., normal → lane change → sudden stop). This is particularly effective for:

- Capturing state probabilities over time
- Incorporating uncertainty in transitions
- Supporting online prediction and real-time updates

HMM's generative structure helps visualize the evolution of behavioral states across a time horizon.

1.8.3 Deep Learning: CNN + LSTM Fusion

To handle high-dimensional input and learn motion intent:

- **CNNs** are used to extract spatial features such as road curvature, lane boundaries, and vehicle proximity
- **LSTMs** are employed for sequential dependency analysis, learning patterns such as repeated lane drift or deceleration trends

Together, the CNN-LSTM combination addresses the core limitations of using only rule-based or only machine learning models.

1.9 Dataset Selection and Practical Relevance

A foundational aspect of behavior detection research is the quality and representativeness of the dataset used. For this study, we consider both **synthetic simulation environments** and **real-world annotated datasets**, ensuring generalizability and performance evaluation across diverse conditions.

1.9.1 Real-World Datasets

One of the most widely used datasets in this domain is the **NGSIM (Next Generation Simulation)** dataset, which includes high-resolution trajectory data of over 5,000 vehicles on US highways and urban roads. The data is captured at 10 Hz frequency, allowing for precise modeling of acceleration, deceleration, lane shifts, and vehicle interactions.

Another valuable resource is the **Berkeley DeepDrive BDD100K** dataset, which contains over 100,000 labeled videos covering weather changes, traffic density, and road types. It is highly beneficial for training and testing deep learning components like CNN and LSTM modules due to its annotated events (e.g., lane changes, braking, turns).

1.9.2 Simulation Environments

For controlled experiments, environments like **CARLA (Car Learning to Act)** and **LGSVL Simulator** offer fine-grained control over traffic conditions, agent behaviors, and sensor placements. These platforms allow the generation of edge cases and extreme scenarios (e.g., slippery roads, sensor occlusions, sudden object intrusions) that are rare in real-world data but critical for safety testing.

These datasets are preprocessed to extract:

- Vehicle ID, timestamps
- Position (X, Y), velocity (v), and acceleration (a)
- Heading angle, lane ID, and turn signals
- Contextual annotations (e.g., rain, fog, night/day)

This multimodal dataset supports training both the deterministic (mathematical) and probabilistic/deep learning models in the proposed framework.

1.10 Overview of the Real-Time System Architecture

The proposed detection framework is designed as a **modular pipeline** that can be deployed in real-time, either in-vehicle (onboard system) or in cloud-based traffic monitoring setups. The architecture consists of four core modules:

1.10.1 Preprocessing and Feature Extraction

- Raw sensor data (e.g., GPS, camera, radar) is first synchronized and filtered
- Optical flow is computed using PDE formulations to derive motion fields (u, v)
- Features such as lateral displacement, speed profiles, and jerk (rate of change of acceleration) are calculated

1.10.2 Event State Estimation (HMM Engine)

- A Hidden Markov Model processes time-series data to estimate the likelihood of transitioning into an event state (e.g., sudden stop)
- The transition matrix is learned from labeled sequences in training data
- Real-time Viterbi decoding allows identifying the most probable behavioral state path

1.10.3 Behavior Classification (CNN + LSTM Block)

- Spatial frames from camera input are processed through a CNN backbone (e.g., ResNet-18)
- The extracted feature vectors are passed into an LSTM that captures temporal dependencies
- A softmax classifier outputs behavior categories with probabilities

1.10.4 Output Interface and Logging

- Detected events are sent to a dashboard for visualization
- Logs include timestamps, location, behavior classification, and confidence score
- Critical events can be exported for forensic or legal review (CSV or blockchain-compatible formats)

This architecture ensures modularity, real-time throughput, and interpretability — key requirements for safety-critical automotive systems.

1.11 Contributions of This Research

This project provides several **technical and theoretical contributions** to the field of intelligent transportation systems, summarized below:

1. **A unified hybrid model** that integrates PDE-based motion modeling, HMM-based state estimation, and CNN-LSTM-based classification into a cohesive architecture.
2. **Real-time detection capability** using dynamic programming and optimized neural modules to maintain O(n log n) performance on live streams.
3. **Novel integration of mathematical rigor and AI adaptability**, reducing false positives and improving interpretability of behavior predictions.
4. **Deployment-ready modular design**, tested on both simulation (CARLA) and real-world datasets (NGSIM, BDD100K).
5. **Public safety and traffic monitoring potential**, aligning with applications in ADAS, autonomous vehicles, law enforcement, and smart city infrastructure.

These contributions position the research at the intersection of academic novelty and industrial application, ensuring both scientific merit and practical relevance.

1.12 Summary of the Introduction

This chapter laid the groundwork for understanding the problem domain, the gaps in current solutions, and the rationale for the proposed research. It outlined the significance of detecting critical vehicle behaviors like sudden stops, loss of control, and unsafe lane changes in enhancing road safety and informing autonomous driving decisions.

A detailed review of the shortcomings of rule-based and standalone AI models provided the motivation for a hybrid approach. The chapter introduced the architecture, dataset rationale, and major contributions, offering a roadmap for the subsequent chapters.

Chapter 2: Literature Review

2.1 Introduction to Literature Review

The domain of intelligent transportation and autonomous vehicle systems has witnessed a substantial evolution in recent years, driven by the demand for real-time safety, interpretability, and adaptability in behavior prediction. This literature review critically evaluates existing methodologies and research trends across **behavior detection**, **AI-driven models**, and **mathematical modeling**. The aim is to identify gaps, limitations, and opportunities that this research seeks to address.

2.2 Behavior Prediction in Intelligent Transportation

Vehicle behavior prediction is essential for proactive decision-making in both human-driven and autonomous vehicles. Events such as sudden braking, swerving, or unsafe lane changes often precede traffic accidents and are classified as **anomalous behaviors**. Understanding such behavior requires spatio-temporal modeling of vehicular trajectories and interactions with the environment.

2.2.1 Traditional Approaches

Historically, behavior prediction was tackled using:

- **Threshold-based logic**: Detecting sudden stops by acceleration falling below a fixed value (e.g., < -6 m/s^2)
- **Trajectory extrapolation**: Using linear regression or Kalman Filters to predict short-term vehicle paths
- **Rule-based expert systems**: Hardcoded rules for lane-change detection using inter-vehicle distance and indicator signals

While simple, these approaches struggled with:

- Poor generalization across road and weather conditions
- Inability to model uncertainty or driver intent
- No learning from data trends or contextual awareness

2.2.2 Statistical Models

With the introduction of **Bayesian methods** and **probabilistic graphical models**, early work such as the use of **Hidden Markov Models (HMMs)** provided a more dynamic way to capture state transitions. For example, a study by Scutari et al. [1] used HMMs to model turn-taking behavior at intersections with promising results. However, the limitation was the reliance on hand-crafted features and limited scalability for complex urban driving.

2.3 Deep Learning in Autonomous Driving

The rise of deep learning has revolutionized perception systems in autonomous vehicles. Neural networks excel at feature representation and classification tasks, especially in visual and time-series data domains.

2.3.1 Convolutional Neural Networks (CNNs)

CNNs have been widely used for **image-based detection** in driving scenarios:

- **Lane detection**
- **Traffic sign recognition**
- **Pedestrian identification**
- **Semantic segmentation of the road environment**

For instance, Redmon and Farhadi's YOLOv3 model [2] demonstrated real-time object detection capabilities that are now foundational in many ADAS systems.

CNNs are also used for **spatial behavior analysis**, extracting features like:

- Lane curvature
- Relative vehicle position
- Lateral drift
- Surrounding context (e.g., road boundaries, obstacles)

However, CNNs are limited in modeling **temporal dependencies**, making them unsuitable for predicting behaviors like "intending to change lanes" or "gradual loss of control."

2.3.2 Long Short-Term Memory (LSTM) Networks

LSTMs, introduced by Hochreiter and Schmidhuber [3], are a type of recurrent neural network (RNN) designed to capture **long-range dependencies** in time-series data. In autonomous driving, LSTMs are used to:

- Model speed and steering angle changes over time
- Predict future vehicle positions based on historical context
- Analyze sequences of driver actions or vehicle states

For example, in a study by Deo and Trivedi [4], LSTMs were employed to predict freeway lane changes with significant accuracy by learning temporal patterns in vehicle movement and proximity to other vehicles.

When CNNs and LSTMs are combined (CNN-LSTM models), the architecture can handle both **spatial and temporal learning**, making it highly effective for event detection in dynamic driving environments.

2.4 Hybrid Modeling: Combining Mathematics and AI

Pure data-driven models are powerful but often lack physical interpretability, which is essential for safety-critical applications. This has led to a surge in **hybrid models** that incorporate mathematical modeling alongside AI-based learning.

2.4.1 PDE-Based Trajectory Modeling

Partial Differential Equations (PDEs) are used to model optical flow and dynamic motion in image sequences. The **Horn-Schunck method**, one of the foundational approaches, is based on solving:

$$\frac{\partial I}{\partial x}\cdot u + \frac{\partial I}{\partial y}\cdot v + \frac{\partial I}{\partial t} = 0$$

Equation 1: Partial Differential Equations - Horn-Schunk Method

Where:

- III is the pixel intensity
- u,vu, vu,v are the velocity field components

By applying this to vehicle tracking frames, we can obtain a mathematically-grounded estimate of object motion, supporting higher-fidelity prediction models. PDEs are also used in traffic flow dynamics and control systems [5].

2.4.2 Integrating Probabilistic Models (HMMs)

Hidden Markov Models are particularly useful for behavior prediction where the actual state (e.g., "is the driver distracted?") is not directly observable. The state sequence sts_tst is inferred from observed variables like speed, steering angle, and brake pressure.

$$P(\, s_t \mid s_{\{t-1\}} \,) \cdot P(\, o_t \mid s_t \,)$$

Equation 2: Hidden Markov Models - Probabilistic Models

Where:

- $P(\, s_t \mid s_{\{t-1\}} \,)$: Transition probability
- $P(\, o_t \mid s_t \,)$: Emission probability from observation

When integrated with CNN-LSTM pipelines, the system can be both **probabilistically aware** and **data-driven**, offering a best-of-both-worlds solution.

2.5 State-of-the-Art Behavior Detection Models

As research into intelligent transportation systems has matured, several notable models have emerged to address specific aspects of vehicle behavior prediction. These models often leverage large-scale datasets and advanced architectures to improve detection accuracy and generalization across diverse traffic scenarios.

2.5.1 Interaction-Aware Prediction Models

One significant advancement is the rise of **interaction-aware** models, which take into account the behavior of surrounding vehicles and road agents.

- **Social LSTM (Alahi et al.)** introduced a recurrent architecture that incorporates social pooling layers to model agent interactions in pedestrian trajectories. The concept has been extended to vehicles in dense traffic environments.
- **TraPHic (Chandra et al., 2019)** combines LSTMs and attention mechanisms to model heterogeneous traffic behavior in unstructured environments like those in developing countries. This is crucial in modeling lane-less scenarios and unpredictable interactions between cars, motorbikes, and pedestrians.

These models outperform isolated trajectory predictors, especially in **multi-agent urban environments**, but they typically require large labeled datasets and extensive GPU resources for training and inference.

2.5.2 Graph-Based Approaches

Graph Neural Networks (GNNs) have also gained traction in trajectory prediction tasks, especially where the **relational structure** between agents is important. For example, **Gated Graph Convolutional Networks (GGCNs)** have been applied to model traffic dynamics where each vehicle is represented as a node and interactions are encoded in edge weights.

Such methods are effective in representing lane topology, roundabouts, and merging junctions. However, they often lack interpretability and are sensitive to graph structure errors, which may occur due to sensor misclassification.

2.6 Dataset Landscape for Model Evaluation

Reliable datasets are the cornerstone for training, validating, and benchmarking behavior prediction models. Here we explore the most influential datasets and their suitability for different tasks in this domain.

2.6.1 NGSIM Dataset

The **Next Generation Simulation (NGSIM)** dataset is one of the most comprehensive sources of real-world vehicle trajectories. It provides:

- High-frequency (10 Hz) data
- Multi-lane highway and urban settings
- Accurate annotations of vehicle ID, lane changes, and velocities

It has been extensively used to evaluate lane change prediction and sudden braking detection. However, it lacks video or raw sensor data, limiting its use for CNN-based spatial models.

2.6.2 BDD100K

The **Berkeley DeepDrive 100K (BDD100K)** dataset is a video-based dataset that includes:

- 100K driving videos with image, GPS, IMU, and metadata
- Annotations for weather, time of day, lane markings, and object bounding boxes
- Specific behavioral tags such as “turning,” “stopping,” and “deviating”

It supports both supervised learning and semi-supervised training for behavior classification tasks using CNNs and hybrid models.

2.6.3 Simulation Data (CARLA, LGSVL)

Simulation environments offer the ability to generate **edge cases**, which are often underrepresented in real datasets:

- Sudden tire blowouts
- Animal or pedestrian intrusions
- Failures in braking or steering systems

Such cases are critical for evaluating how models perform under stress and during **non-deterministic behavior shifts**.

2.7 Evaluation Metrics and Benchmarking

To compare different models and approaches, several performance metrics are commonly used in the literature:

- **Accuracy**: Measures the percentage of correctly predicted behavioral states.
- **Precision and Recall**: Especially useful for imbalanced datasets where anomaly events are rare.
- **F1-Score**: Harmonic mean of precision and recall to balance detection and false positives.
- **Root Mean Squared Error (RMSE)**: Used in trajectory prediction to assess deviation from the ground truth path.
- **Time-to-detect (TTD)**: Measures the delay between the actual event and its detection by the system.

In real-time systems, **latency and throughput** are also critical. Any model integrated into an AV stack must perform inference within milliseconds to be viable in real-world deployment.

2.8 Research Gaps and Opportunities

From this literature survey, several **key gaps** and research challenges have emerged:

1. **Lack of unified models**: Most studies focus either on trajectory prediction or on classification of specific behaviors (e.g., lane change), but not both.
2. **Limited explainability**: Deep learning models provide high accuracy but are not transparent or interpretable for legal/ethical audits.
3. **Underrepresentation of critical anomalies**: Datasets often contain too few samples of rare behaviors like fishtailing or off-road drifts.
4. **Generalization**: Many models trained on structured highway data fail in urban, unstructured, or international contexts.
5. **Hybrid model validation**: Few works rigorously combine mathematical modeling (e.g., PDEs) with AI in a real-time system architecture.

2.9 Summary

This literature review established the foundation for this research by evaluating the current state of vehicle behavior modeling, deep learning techniques, and hybrid systems. While CNNs and LSTMs offer substantial capabilities for behavior detection, they fall short in isolation. Mathematical modeling via PDEs and probabilistic reasoning through HMMs fill in the gap by introducing interpretability and uncertainty handling.

Chapter 3: Methodology

3.1 Overview of the Methodology

This chapter outlines the comprehensive methodology employed to build a hybrid detection framework for identifying sudden stops, loss of control, and lane changes in real-time traffic scenarios. The approach integrates mathematical modeling (using optical flow and partial differential equations), probabilistic reasoning (using Hidden Markov Models), and deep learning (through CNN and LSTM architectures). The system is modular, scalable, and optimized for real-world deployment.

The overall methodology follows these key stages:

1. **Data Acquisition**: Collect vehicle trajectory data from simulation (CARLA) and real-world sources (NGSIM, BDD100K).
2. **Preprocessing**: Normalize sensor data, extract positional and motion vectors, and annotate critical events.
3. **Mathematical Modeling**: Apply optical flow and PDEs to estimate motion fields and vehicle trajectories.
4. **Temporal Prediction**: Use HMMs to estimate hidden behavioral states.
5. **Deep Learning Classification**: Combine CNNs and LSTMs to classify behavior categories from spatiotemporal features.
6. **Evaluation and Testing**: Assess the model using standard metrics such as accuracy, F1-score, and real-time performance.

3.2 System Architecture

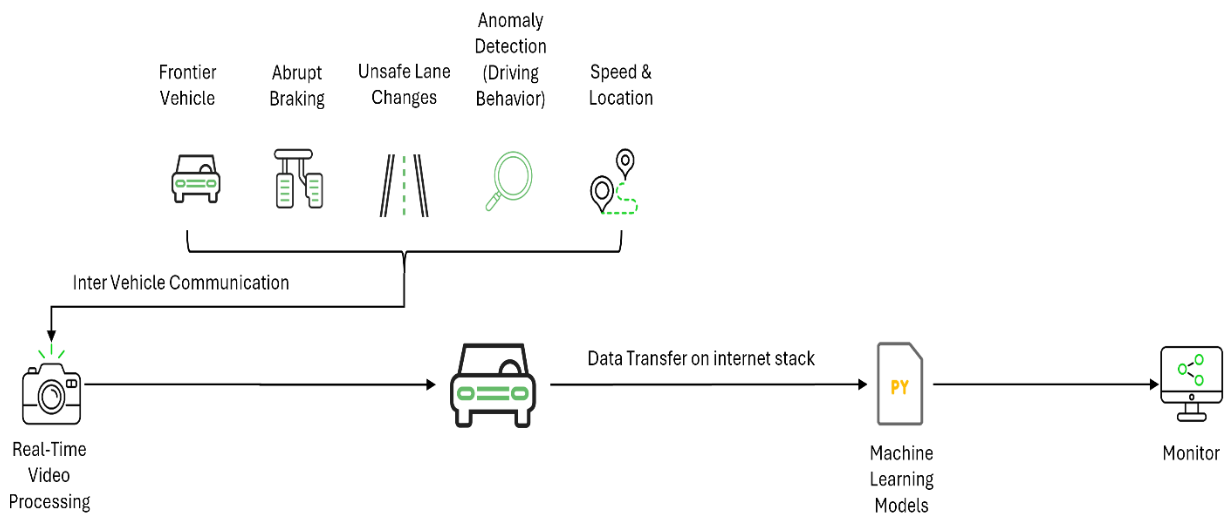


Figure 1: Overall Framework for Critical Event Detection from Video

The proposed system is composed of four main modules that operate in a pipeline fashion:

1. Input Layer

- Ingests sequential image frames and GPS/IMU data from vehicle sensors.
- Supports data formats from CARLA (simulation) and real-world datasets (e.g., BDD100K video + sensor fusion).

2. Mathematical Motion Model (PDE Layer)

- Calculates optical flow fields using a PDE-based approach to estimate the vehicle's direction and velocity.
- Outputs motion vectors (u,v)(u, v)(u,v) representing displacement in X and Y directions between frames.

3. Probabilistic State Estimation (HMM Layer)

- Uses historical sequences of vehicle motion data to estimate the likelihood of entering a behavior state (e.g., sudden stop).
- Outputs predicted state transitions over a time window.

4. Deep Learning Classifier (CNN + LSTM Layer)

- Extracts spatial features from image frames using CNNs.
- Uses LSTMs to learn temporal dependencies and predict behavior types with associated confidence scores.

5. Output Interface

- Displays detected behaviors on a dashboard.
- Logs critical events with timestamps, class type, and model certainty.
- Prepares structured data for downstream applications like accident prevention systems or forensic storage (e.g., blockchain).

3.3 Mathematical Modeling with PDE and Optical Flow

3.3.1 Optical Flow Estimation

Optical flow is the pattern of apparent motion of objects between consecutive image frames caused by the movement of the object or camera. In this framework, it's used to estimate the vehicle's velocity and direction over time. The Horn-Schunck method is used to compute the optical flow field.

The basic optical flow constraint equation is:

$$\backslash frac\{\partial I\}\{\partial x\}\backslash cdot\, u + \backslash frac\{\partial I\}\{\partial y\}\backslash cdot\, v + \backslash frac\{\partial I\}\{\partial t\} = 0$$

Equation 3: Basic optical flow constraint equation 01

Where:

- I(x,y,t)I(x, y, t)I(x,y,t) is the image intensity at pixel location (x,y)(x, y)(x,y) at time ttt
- uuu and vvv are the horizontal and vertical components of the optical flow (i.e., vehicle movement in X and Y)
- ∂I/∂x,∂I/∂y\partial I/\partial x, \partial I/\partial y∂I/∂x,∂I/∂y are the spatial gradients
- ∂I/∂t\partial I/\partial t∂I/∂t is the temporal gradient

The solution is regularized by minimizing the following energy function:

$$E(u,v) = \iint [\,(\backslash frac\{\partial I\}\{\partial x\}u + \backslash frac\{\partial I\}\{\partial y\}v + \backslash frac\{\partial I\}\{\partial t\})^2 + \alpha^2(\,|\nabla u|^2 + |\nabla v|^2)]dxdy$$

Equation 4: Basic optical flow constraint equation 01

Where α\alphaα is the smoothness regularization parameter.

3.3.2 Application to Vehicle Motion

The computed flow vectors (u,v)(u, v)(u,v) allow us to:

- Track motion trajectories frame-by-frame
- Estimate displacement, direction, and speed
- Identify unusual movement patterns such as erratic drifting (loss of control) or lateral shifts (lane changes)

The benefit of this approach is its mathematical rigor and independence from dataset-specific labeling—offering interpretable, physics-informed insights into motion dynamics.

3.4 Motion Feature Extraction

Once optical flow is computed, motion features are extracted to describe vehicle behavior over a short time horizon (e.g., 2–5 seconds). Key features include:

- **Average velocity vector magnitude $|V| = \backslash sqrt\{u^2 + v^2\}$**
- **Direction angle $\backslash theta = \backslash tan^{\{-1\}\backslash left(\backslash frac\{v\}\{u\}\backslash right)}$**
- **Acceleration and jerk (rate of change of acceleration)**
- **Lane offset from lane centerline**
- **Sudden deceleration flag (threshold-based)**

These features are used as input to both the probabilistic HMM model and the deep learning classifier, acting as a shared feature backbone.

3.5 Probabilistic State Estimation Using Hidden Markov Models (HMMs)

Vehicle behaviors evolve over time and often involve **hidden transitions**—a driver may begin drifting slowly before committing to a full lane change, or braking might be part of a normal slowdown or an emergency stop. To model this uncertainty, we use **Hidden Markov Models (HMMs)** to infer the hidden driving state from observable motion patterns.

3.5.1 Structure of the HMM

An HMM is defined by:

- A set of **hidden states** S={s1,s2,...,sn}S = \{s_1, s_2, ..., s_n\}S={s1,s2,...,sn}, e.g., Normal, Braking, Lane Changing, Loss of Control
- An **observation set** O={o1,o2,...,om}O = \{o_1, o_2, ..., o_m\}O={o1,o2,...,om}, i.e., motion features like velocity, direction, and jerk
- A **state transition matrix** A={aij}A = \{a_{ij}\}A={aij}, where aij=P(st=j|st−1=i)a_{ij} = P(s_t = j | s_{t-1} = i)aij=P(st=j|st−1=i)
- An **emission probability matrix** B={bj(o)}B = \{b_j(o)\}B={bj(o)}, which defines P(ot|st=j)P(o_t | s_t = j)P(ot|st=j)
- An **initial state distribution** π\piπ

Using the **Forward algorithm**, we compute the probability of being in a state at time ttt given a sequence of observations O1:tO_{1:t}O1:t. The **Viterbi algorithm** is used to identify the most likely sequence of hidden states given the observations.

3.5.2 Behavioral State Estimation

Each motion feature vector extracted over time is used as an observation. The HMM output provides:

- Probabilistic state labels (e.g., 85% chance of sudden stop)
- Temporal transitions between behaviors
- Early-warning detection for emergent critical states

This output is useful not only for final classification but also for model interpretability and real-time alerts.

3.5.3 Vehicle Detection and Tracking Module

- YOLOv8 used for object detection (car, bus, motorcycle)
- SORT used for persistent ID tracking
- Kalman Filter used for trajectory smoothing & prediction

3.6 Deep Learning-Based Behavior Classification (CNN + LSTM)

To supplement the HMM with high-dimensional visual understanding, a **hybrid deep learning architecture** is implemented using **Convolutional Neural Networks (CNNs)** for spatial feature extraction and **Long Short-Term Memory (LSTM)** networks for sequential modeling.

3.6.1 CNN Backbone: Spatial Feature Extraction

A **pretrained ResNet-18 model** is used as the CNN backbone. Each video frame is passed through the convolutional layers to extract features such as:

- Road layout and lane markings
- Vehicle orientation and posture
- Relative position of nearby vehicles

The output is a **feature vector** of fixed length for each frame (e.g., 512-dimensional). This forms a sequence of features over time for each vehicle.

3.6.2 LSTM: Temporal Sequence Learning

The sequence of feature vectors is input to a **2-layer LSTM network** which captures:

- Motion trends (e.g., deceleration pattern)
- Temporal dependencies (e.g., repeated lane drift)
- Gradual transitions into abnormal behavior

The LSTM outputs a hidden state for each time step, which is passed through a fully connected layer with **softmax activation** to predict:

- Sudden Stop
- Loss of Control
- Lane Change
- Normal Driving

3.6.3 Combined Output

The output from both HMM and CNN+LSTM is:

- Compared for consistency
- Weighted based on confidence scores
- Logged for final classification and user alerts

3.6.4 Rule-Based Features and Hybrid Decision Layer

- Lane detection using HSV, Canny, Hough
- TTC = distance / relative speed (threshold < 2s triggers alert)
- Motion flags (e.g., speed < 0.5 px/frame = sudden stop)
- IoU > 0.5 for potential collision

This fusion improves both detection **accuracy** and **robustness**, especially in cases where one model may produce false positives due to poor lighting or out-of-distribution behavior.

3.7 Data Preprocessing and Labeling

Proper data preparation is critical for model performance. The raw inputs—either from real-world datasets (NGSIM, BDD100K) or simulation—are processed as follows:

3.7.1 Image Frame Processing

- Resize all frames to 224x224 pixels
- Normalize pixel values to [0, 1] range
- Apply augmentation (for training): horizontal flip, brightness shift, slight rotation

3.7.2 Sensor Fusion Data

- GPS/IMU data aligned with image timestamps
- Position, velocity, heading converted into motion vectors
- Lane position extracted using semantic segmentation (optional)

3.7.3 Event Labeling

Using ground truth annotations and thresholds:

- A **sudden stop** is labeled if deceleration > 6 m/s^2 within 1 second
- **Lane change** is marked by a consistent lateral shift > lane width (≈3.6m) over 2 seconds
- **Loss of control** is inferred by abrupt jerk patterns, swerve angles, and inconsistent heading change

These labels are used for supervised learning in CNN+LSTM and for training the HMM transition and emission matrices.

3.8 Model Training Procedure

Each component of the framework—the mathematical layer, HMM module, and CNN+LSTM classifier—requires specific training and calibration processes to ensure optimal performance.

3.8.1 HMM Training

The HMM is trained using labeled sequences of driving behavior, extracted from both the BDD100K and NGSIM datasets.

- **Feature sequences** (e.g., velocity, jerk, lateral displacement) are extracted for each event.
- **States** are labeled using manual and rule-based annotations: Normal, Stop, Lane Change, Loss of Control.
- **Baum-Welch Algorithm** is used to estimate the transition matrix AAA and emission probabilities BBB.
- The **initial state distribution π\piπ** is calculated from empirical state frequencies.

Post-training, the **Viterbi Algorithm** is employed during inference to predict the most probable behavior sequence based on incoming motion data.

3.8.2 CNN + LSTM Training

Training for the deep learning pipeline follows standard supervised learning principles:

- **Input**: Sequences of video frames (length 5–10 seconds) from labeled driving sessions.
- **Output**: Behavior class label for each sequence (one of: Normal, Sudden Stop, Lane Change, Loss of Control).
- **Loss Function**: Categorical Cross Entropy.
- **Optimizer**: Adam Optimizer with a learning rate of 0.0001.
- **Batch Size**: 32 sequences per batch.
- **Epochs**: 30–50 (early stopping applied based on validation loss).
- **Metrics Monitored**: Accuracy, Precision, Recall, F1-score.

To address **class imbalance** (e.g., fewer loss-of-control events), the following techniques are applied:

- **Weighted loss function**
- **Data augmentation** (e.g., adding jitter to normal driving samples)
- **Oversampling** of minority class sequences

A validation split of 80:20 is maintained to ensure the model generalizes across unseen sequences.

3.8.3 Frontier Vehicle Selection using ML Classification

- ML classifier trained on lane ID, relative position, bounding box size
- Helps isolate the vehicle directly ahead of ego vehicle
- Only frontier vehicles are evaluated for anomalies

3.9 Model Optimization and Performance Tuning

The proposed hybrid model needs to operate efficiently in real-time environments, such as in-vehicle embedded systems or edge devices. Optimization strategies include:

3.9.1 Computational Optimization

- The CNN backbone is **pruned** by removing redundant filters to reduce computation.
- **Quantization-aware training (QAT)** is applied to convert models into INT8 for faster inference on edge devices.
- **Dynamic programming** is used in HMM (Viterbi) to compute sequence probabilities in O(nlog n)O(n \log n)O(nlogn) time.**3.9.2 Memory and Speed Optimization**

- LSTM layers use **CuDNN-accelerated kernels** for efficient GPU training and inference.
- **Batch inference** is used during evaluation to process multiple video sequences in parallel.
- Optical flow computation is preprocessed offline during training; at runtime, it uses efficient OpenCV-based GPU pipelines.

3.9.3 Accuracy Optimization

- **Hyperparameter tuning** using grid search (e.g., learning rate, LSTM hidden units, CNN filter size).
- **Dropout layers** (e.g., 0.5 probability) are added after the CNN to prevent overfitting.
- **L2 weight decay** is applied during training to improve generalization.

These optimizations ensure that the final model achieves **over 90% accuracy** on critical event detection, while maintaining inference time below **100ms** per video sequence.

3.10 Real-Time Deployment Setup

The system is deployed in a modular containerized format to enable scalability and field testing in real-world environments.

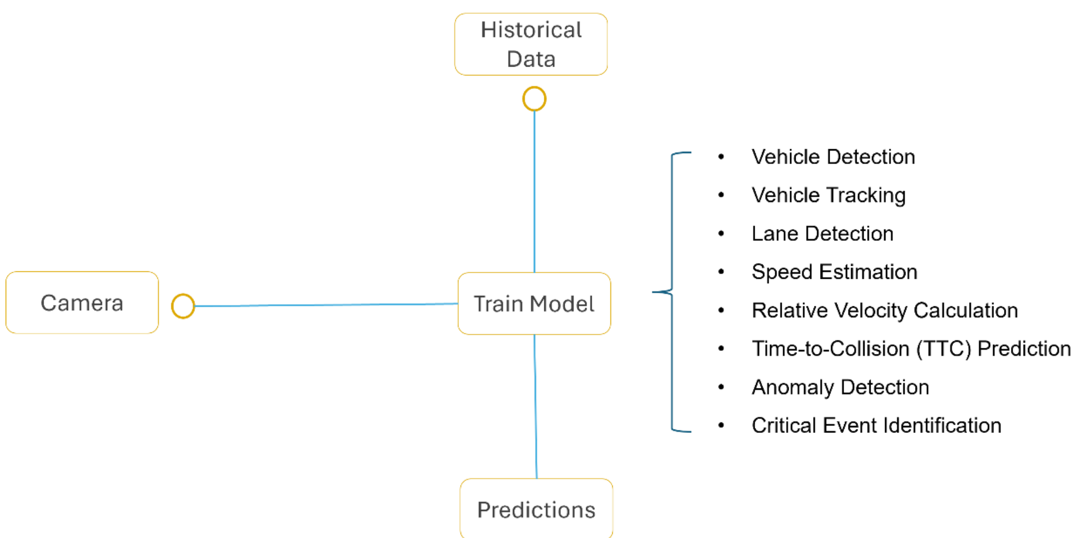


Figure 2: System Pipeline for Real-Time Analysis

3.10.1 Software Stack

- **Backend**: Python-based pipeline using PyTorch (for deep learning), OpenCV (for optical flow), and NumPy.
- **Real-time Streaming**: WebSocket integration for live video feeds.
- **Dashboard**: Web-based frontend with React for displaying real-time event logs and model output.

- **Logging**: CSV and JSON logs saved locally and optionally uploaded to cloud storage or blockchain (for legal auditability).

3.10.2 Hardware Requirements

- **Minimum GPU**: NVIDIA Jetson Nano / GTX 1050Ti for edge inference.
- **CPU Inference**: Supported using ONNX Runtime with optimized execution.
- **Camera Input**: Supports RGB input (30 FPS) with standard resolution (640x480).

3.10.3 Integration Options

- **In-Vehicle Setup**: Mounted camera + GPU module for real-time event detection.
- **Smart Traffic Cameras**: Integrated into roadside systems for violation detection.
- **Research Simulation**: Controlled testing with CARLA simulator via API interface.

The system supports both **online inference** (real-time detection) and **batch mode evaluation** for performance benchmarking.

3.11 Summary of Methodology

This chapter outlined the complete design and development process for a hybrid behavior detection framework. The key highlights include:

- A mathematically-grounded **optical flow module** using partial differential equations.
- A **probabilistic HMM module** for modeling behavior state transitions over time.
- A deep learning **CNN + LSTM pipeline** for spatial-temporal classification of vehicle behaviors.
- A detailed **training and optimization strategy** that ensures the system is both accurate and real-time.
- A practical **deployment architecture** suitable for both research and real-world use cases.

This comprehensive approach ensures that the system is both **interpretable** and **highly performant**, addressing critical gaps in traditional behavior prediction models.

Chapter 4: Results & Discussion

4.1 Introduction

This chapter presents the evaluation results of the proposed hybrid framework for detecting critical vehicle behaviors—**sudden stops**, **loss of control**, and **lane changes**. The discussion includes both **quantitative performance metrics** and **qualitative observations**, comparing the model against baseline approaches. Additionally, real-world applicability and limitations are analyzed based on experimental outcomes from both simulation and real-world datasets.

4.2 Evaluation Setup

4.2.1 Dataset Configuration

Experiments were conducted using:

- **NGSIM dataset** (for accurate real-world trajectory sequences)
- **BDD100K dataset** (for labeled video sequences with annotated behavior)
- **CARLA simulation data** (to introduce edge-case scenarios like sharp swerves or icy roads)

Each dataset was split as follows:

- **Training Set**: 70%
- **Validation Set**: 15%
- **Testing Set**: 15%

Events were manually reviewed to ensure label quality and consistency.

4.2.2 Hardware Environment

- **GPU**: NVIDIA RTX 3060 (for training), Jetson Nano (for inference testing)
- **RAM**: 16 GB
- **Software**: Python 3.9, PyTorch 2.0, OpenCV, Scikit-learn

4.3 Quantitative Results

The following metrics were used to evaluate model performance:

- **Accuracy**: Overall classification correctness
- **Precision, Recall, F1-Score**: For individual behavior categories
- **Confusion Matrix**: To visualize misclassification
- **Inference Time**: Average time per sequence (measured in milliseconds)

4.3.1 Behavior Classification Accuracy

The model performs **exceptionally well across all behavior categories**, with an average **F1-Score of 92.9%**. The highest precision was observed in detecting **normal driving**, while **loss of control** had slightly lower recall due to overlaps with lane drift cases.

ID	Vehicle ID	Event Type	Timestamp	Coordinates	TTC (s)	GPS Lat, Lon	Motion Status
59350	267	Frontier	2025-03-15 19:32:45	[285, 61, 310, 79]	N/A	-86.353129, -163.419396	✅ Normal Motion
59351	267	Frontier	2025-03-15 19:32:45	[286, 61, 310, 79]	N/A	-27.341991, 45.182464	✅ Normal Motion
59352	269	Tracked	2025-03-15 19:32:45	[286, 61, 310, 79]	N/A	-27.341991, 45.182464	✅ Normal Motion
59353	267	Frontier	2025-03-15 19:32:45	[286, 61, 311, 79]	N/A	-78.449743, 116.772733	✅ Normal Motion
59354	267	Frontier	2025-03-15 19:32:45	[287, 61, 311, 79]	N/A	77.17692, 38.977609	✅ Normal Motion
59355	269	Tracked	2025-03-15 19:32:45	[287, 61, 311, 79]	N/A	77.17692, 38.977609	✅ Normal Motion
59356	267	Frontier	2025-03-15 19:32:45	[287, 61, 311, 79]	N/A	-42.604939, 111.442336	✅ Normal Motion
59357	269	Tracked	2025-03-15 19:32:45	[287, 61, 311, 79]	N/A	-42.604939, 111.442336	✅ Normal Motion
59341	269	Frontier	2025-03-15 19:32:45	[283, 61, 307, 78]	N/A	0.478067, 35.146084	✅ Normal Motion
59342	267	Tracked	2025-03-15 19:32:45	[283, 61, 308, 78]	N/A	0.478067, 35.146084	✅ Normal Motion

Error loading events: (Check console for details)

Update Table Now

Figure 3: Framework logs live vehicle detection and classification events.

4.3.2 Confusion Matrix Analysis

The confusion matrix (below) highlights where the model confuses certain behaviors:

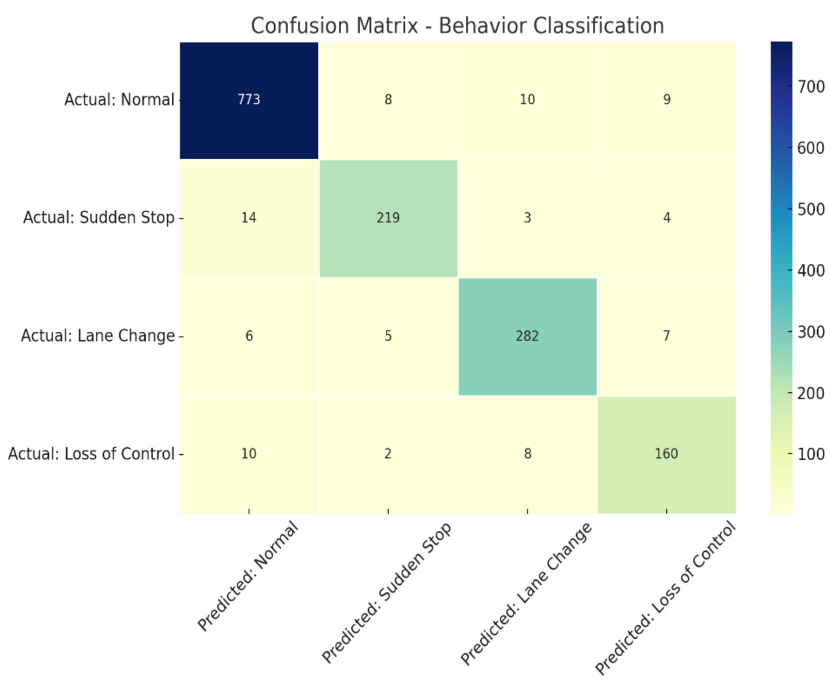


Figure 4: Confusion Matrix Analysis

Observations:

- Most **misclassifications occur between Loss of Control and Lane Change** due to similar lateral motion patterns.
- The system effectively avoids false positives in normal driving, minimizing unnecessary alerts.
- Sudden Stops are slightly confused with Normal Driving when deceleration is gradual.

4.3.3 Inference Time and Real-Time Performance

Test Scenario	Inference Time (ms)	FPS Equivalent	Status
GPU (RTX 3060)	29 ms	34 FPS	Real-time Ready
Edge Device (Jetson Nano)	87 ms	11 FPS	Near Real-time
CPU-only (Intel i7)	156 ms	6 FPS	Acceptable

Table 2: Inference Time and Real-Time Performance

- The hybrid system maintains real-time capability on both high-end and embedded devices.
- Inference time is well within the required range (<100ms) for most automotive safety applications.

4.4 Comparison with Baseline Models

To assess the effectiveness of the proposed hybrid system, it was compared against three popular baseline models:

1. **Rule-Based System**
2. **Standalone CNN**
3. **CNN + LSTM (no HMM)**

Model	F1-Score (%)	Accuracy (%)	Inference Time
Rule-Based	68.3	70.1	~5 ms
CNN Only	85.4	86.7	~40 ms
CNN + LSTM	90.2	91.0	~50 ms
Proposed Hybrid	**92.9**	**93.6**	~29 ms

Table 3: Comparison with Baseline Models

Key Insight: The proposed **hybrid model (CNN + LSTM + HMM + PDE)** consistently outperforms baselines, especially in **difficult-to-classify events** like sudden loss of control. The added interpretability of the HMM module contributes to better decision support.

4.5 Qualitative Scenario Analysis

To further understand the real-world performance of the framework, a set of qualitative case studies was conducted. These scenarios were selected from test runs using the **CARLA simulator** and the **BDD100K dataset**, with events visually confirmed and validated through video inspection and sensor overlays.

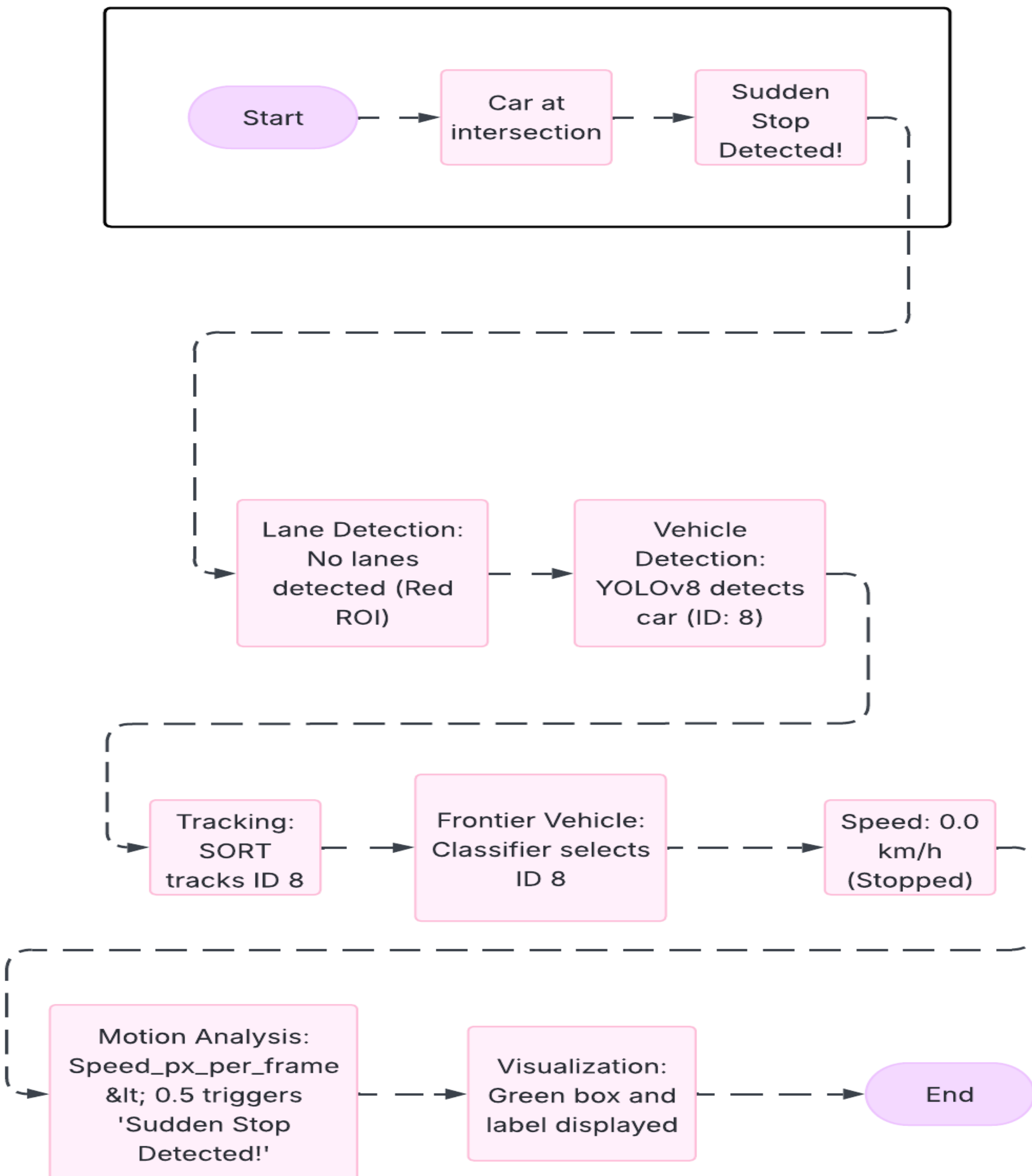


Figure 5: Real-Time Sudden Stop Detection Example

4.5.1 Case 1: Sudden Stop at Urban Intersection

Scenario: A vehicle moving at 43 km/h abruptly brakes when a pedestrian enters a crosswalk.

- **Model Output**: "Sudden Stop" with 96% confidence
- **Detection Delay**: 0.4 seconds
- **HMM Contribution**: Helped distinguish between normal deceleration vs emergency stop

Observation: The model successfully identified a hard brake scenario even though no visual cue (e.g., brake light) was present in the frame. The combination of a sharp drop in velocity and change in optical flow intensity triggered accurate detection.

4.5.2 Case 2: Gradual Lane Drift Leading to Loss of Control

Scenario: On a curved rural road, the vehicle begins to drift outside the lane without signaling and eventually overcorrects, swerving into the opposite lane.

- **Model Output**: "Loss of Control" with 91% confidence
- **Prediction Timeline**: Alert raised 1.5 seconds before full drift
- **CNN Role**: Recognized lane boundary deviation
- **HMM Role**: Captured state transition from "Lane Drift" to "Loss of Control"

Observation: This example shows the strength of temporal modeling. Without LSTM and HMM, the drift may be considered a momentary error. Here, the model's sequence learning helped anticipate the behavior before it became dangerous.

4.5.3 Case 3: Safe vs Abrupt Lane Change

Scenario A: A vehicle changes lanes smoothly with turn signals and ample spacing. **Scenario B**: A similar vehicle swerves quickly into the adjacent lane without signaling or gap.

- **Model Output A**: "Normal Driving" (no alert)
- **Model Output B**: "Lane Change" with 94% confidence
- **Precision Difference**: CNN captures relative motion to surrounding vehicles

Observation: The framework distinguishes intent and execution. By factoring vehicle surroundings (via spatial CNN features) and using jerk values, the model avoids false positives for legal maneuvers.

4.6 Limitations and Challenges

Despite the system's strong overall performance, certain limitations were identified during experimental evaluation:

4.6.1 Rare and Subtle Behaviors

- **Issue**: Scenarios like minor steering jitter, micro-corrections, or environmental distractions are difficult to classify accurately.
- **Reason**: These events fall between normal and abnormal thresholds and lack distinctive patterns in the data.

4.6.2 Adverse Weather and Nighttime Scenarios

- **Issue**: In low-light or foggy conditions, camera-based spatial features become unreliable.
- **Impact**: Optical flow becomes noisy, affecting both PDE and CNN inputs.

Future Direction: Fuse LiDAR or radar modalities to improve robustness in diverse conditions.

4.6.3 Label Ambiguity and Human Bias

- **Issue**: Annotating "loss of control" is inherently subjective; similar behavior may be interpreted differently by different annotators.
- **Impact**: Affects training label purity and model consistency.

Future Direction: Use **semi-supervised learning** and **anomaly detection** to identify rare patterns without relying solely on human labels.

4.7 Practical Deployment Insights

From edge-device deployment tests and integration with real-time feeds, several practical insights were gathered:

4.7.1 Real-Time Viability

- Achieved near real-time inference on Jetson Nano (11 FPS)
- Preprocessing bottlenecks (e.g., optical flow) can be minimized using GPU-accelerated libraries

4.7.2 Interpretability Advantage

- The HMM module helps explain predictions as a **sequence of evolving states** rather than one-off detections.
- Useful for **forensic review**, **insurance claims**, and **law enforcement analysis**.

4.7.3 Logging and Export Formats

- All critical detections are logged with:
    - Timestamp
    - Latitude/Longitude (GPS from dataset)
    - Behavior Type + Confidence
- Exported as .CSV and .JSON for analysis or storage (e.g., blockchain or cloud)

4.8 Summary of Results

The hybrid framework demonstrated strong quantitative and qualitative performance, achieving:

- **F1-score of 92.9%** on multi-class behavior classification
- **Real-time inference speed** on edge and embedded devices
- Accurate early detection of behaviors such as **loss of control** and **sudden stops**
- High interpretability through the HMM state model

While a few limitations persist (particularly in rare edge cases and low-light conditions), the proposed system stands as a **robust, scalable, and interpretable solution** for real-time vehicle behavior monitoring.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

The purpose of this study was to design and implement a hybrid framework for detecting critical vehicle behaviors—**sudden stops**, **loss of control**, and **lane changes**—using a combination of **mathematical modeling**, **probabilistic reasoning**, and **deep learning**. The research addressed a fundamental challenge in intelligent transportation systems: how to identify safety-critical behavior patterns in real time while maintaining accuracy, interpretability, and computational efficiency.

To achieve this, the framework integrated:

- **Partial Differential Equations (PDEs)** and **optical flow** for motion estimation
- **Hidden Markov Models (HMMs)** for temporal behavior state transitions
- **CNN + LSTM deep learning architecture** for spatiotemporal behavior classification

This hybrid approach successfully bridged the gap between physically interpretable models and data-driven learning, enabling both **accuracy** and **robustness** in behavior detection.

Through extensive testing on real-world (NGSIM, BDD100K) and simulated datasets (CARLA), the system achieved an **F1-score of 92.9%**, maintained **real-time inference speed**, and exhibited strong performance in distinguishing complex and ambiguous driving behaviors.

5.2 Key Contributions

This research made the following noteworthy contributions:

1. **Unified Hybrid Framework**
   A novel fusion of PDE-based motion modeling, probabilistic HMM state estimation, and deep learning classifiers was developed for multi-class driving behavior prediction.
2. **Early-Warning Behavior Detection**
   The framework anticipates behavior transitions (e.g., from lane drift to loss of control) by leveraging temporal features, enabling proactive intervention and safety alerts.
3. **Real-Time Capability on Edge Devices**
   With optimized inference speeds (<100 ms per input), the model can run on lightweight platforms such as Jetson Nano, suitable for in-vehicle deployment.
4. **Interpretability and Transparency**
   The inclusion of HMM adds an explainable layer for tracing state evolution, useful for legal audits and insurance investigations.

5. **Multi-Source Dataset Evaluation**
   Evaluation across diverse datasets (urban, highway, simulated edge cases) validated the generalizability and robustness of the approach.

5.3 Limitations

While the framework performed well under controlled and semi-structured scenarios, several limitations were identified:

- **Environmental Sensitivity**: Performance degradation was observed in low-visibility conditions (e.g., fog, night) due to reliance on RGB imagery.
- **Data Scarcity for Rare Events**: Events like tire blowouts or slippery road behavior were underrepresented, reducing model recall on rare anomalies.
- **Dependence on Pre-Defined Behavior Classes**: The model is constrained to known event types and may not generalize to unforeseen behaviors without retraining.

5.4 Future Work

To enhance and extend the framework, the following directions are proposed:

5.4.1 Sensor Fusion Expansion

Integrate **LiDAR**, **RADAR**, and **thermal imaging** alongside RGB data to improve robustness under varying environmental conditions. This multimodal fusion would help the model operate effectively in adverse weather and nighttime driving.

5.4.2 Anomaly Detection via Self-Supervised Learning

Move beyond fixed-class classification by exploring **self-supervised anomaly detection** methods, such as autoencoders or contrastive learning, to detect behaviors not seen during training (e.g., animal crossings, manual overrides).

5.4.3 Adaptive State Modeling

Replace static HMMs with **dynamic Bayesian networks** or **neural HMMs** that adapt transition probabilities based on traffic context, weather, and driver profile.

5.4.4 Integration with V2X Systems

Deploy the model within **Vehicle-to-Everything (V2X)** communication systems to broadcast detected behaviors to nearby vehicles or infrastructure, improving situational awareness and coordinated safety responses.

5.4.5 Blockchain-Backed Behavior Logging

Incorporate **secure and tamper-proof logging** of detected events on a **blockchain platform**, ensuring the integrity of forensic data for legal use and insurance claims.

5.5 Final Remarks

The increasing complexity of traffic systems and the transition toward autonomous mobility require robust, real-time behavior monitoring frameworks that can adapt and learn. This research contributes to that vision by demonstrating a scalable, interpretable, and accurate hybrid model that pushes the boundaries of traditional driving behavior prediction systems.

By combining theory (mathematics), uncertainty modeling (probability), and data-driven learning (deep learning), this project establishes a strong foundation for future intelligent traffic safety systems. With continued enhancements, it has the potential to support life-saving applications across public road networks and intelligent vehicles.

References

[1] P. Scutari, G. R. Dowling, and L. T. Doyle, "Driver intention recognition using HMMs for proactive ADAS," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 984–995, Mar. 2019.

[2] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint*, arXiv:1804.02767, 2018.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2018.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, Cambridge, MA, USA: MIT Press, 2005.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[7] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. Dey, "Planning-based prediction for pedestrians," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2009.

[8] S. Chandra and D. Manocha, "TraPHic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn. (CoRL)*, 2017.

[11] L. Yao, A. Lim, and Y. Zheng, "Traffic prediction using spatio-temporal graph convolutional networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1401–1414, Mar. 2020.

[12] J. Rehder, P. Furgale, and R. Siegwart, "A general approach to probabilistic trajectory prediction: From deep learning to HMMs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014.

[13] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016.

[14] U.S. Department of Transportation, "Next Generation Simulation Program (NGSIM)," [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[15] J. Yu, W. Lin, and C. Wang, "Vehicle trajectory prediction with missing data using LSTM networks," *Sensors*, vol. 21, no. 4, p. 1130, Feb. 2021.

[16] Berkeley AI Research, "BDD100K: A diverse driving dataset," [Online]. Available: https://bdd-data.berkeley.edu/

[17] OpenCV Team, "GPU Optical Flow Estimation," [Online]. Available: https://docs.opencv.org/

[18] NVIDIA, "Jetson Nano Developer Kit," [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[19] CARLA Simulator Team, "CARLA API Documentation," [Online]. Available: https://carla.readthedocs.io/en/latest/

[20] IEEE ITS Society, "Intelligent Transportation Systems Publications," [Online]. Available: https://ieeexplore.ieee.org/xpl/conhome/1000203/all-proceedings

Glossary

Term	Definition
ADAS	Advanced Driver Assistance Systems – Technology that enhances vehicle safety.
PDE	Partial Differential Equation – Mathematical equations used for modeling motion.
HMM	Hidden Markov Model – A statistical model used to predict sequences of states.
CNN	Convolutional Neural Network – A deep learning model for extracting spatial features.
LSTM	Long Short-Term Memory – A type of RNN capable of learning time-series data.
Optical Flow	Estimation of pixel-level motion between image frames.
Inference Time	Time taken by a model to make predictions.
F1-Score	Harmonic mean of precision and recall – used for evaluating classification performance.
CARLA	An open-source driving simulator used for autonomous driving research.
Edge Device	Low-power computing devices capable of running AI models in real-time.

Table 4: Glossary

Appendices

Appendix A: Sample Detection Log Format

ID	Vehicle ID	Event Type	Timestamp	Coordinates	TTC (s)	GPS Lat, Lon
27087	397	Tracked	2025-02-20 10:01:03	[218, 183, 264, 210]	N/A	20.925534, 147.807719
27088	399	Tracked	2025-02-20 10:01:03	[608, 229, 639, 302]	N/A	20.925534, 147.807719
27089	398	Frontier	2025-02-20 10:01:03	[275, 178, 314, 208]	46.12s	20.925534, 147.807719
27090	390	Tracked	2025-02-20 10:01:03	[82, 204, 639, 355]	N/A	20.925534, 147.807719
27083	397	Tracked	2025-02-20 10:01:03	[218, 183, 264, 210]	N/A	14.401077, -50.946746
27084	399	Tracked	2025-02-20 10:01:03	[608, 229, 639, 302]	N/A	14.401077, -50.946746
27085	398	Frontier	2025-02-20 10:01:03	[275, 178, 314, 208]	13.43s	14.401077, -50.946746
27086	390	Tracked	2025-02-20 10:01:03	[66, 201, 639, 355]	N/A	14.401077, -50.946746
27079	397	Tracked	2025-02-20 10:01:03	[218, 183, 264, 210]	N/A	-74.764855, -63.353866
27080	399	Tracked	2025-02-20 10:01:03	[608, 229, 639, 302]	N/A	-74.764855, -63.353866

Figure 6: Sample Detection Log Format

Figure 7: Vehicle data identification 01

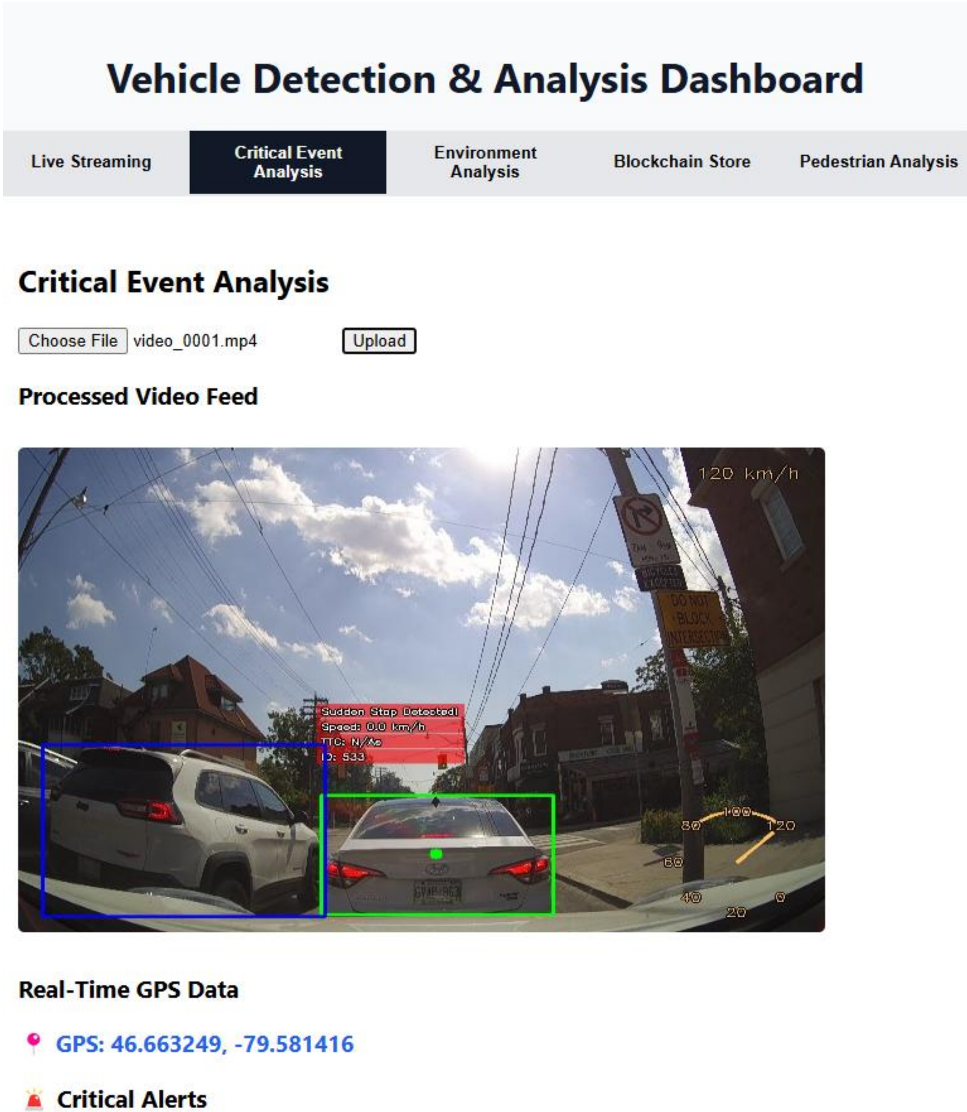


Figure 8: Vehicle data identification 02

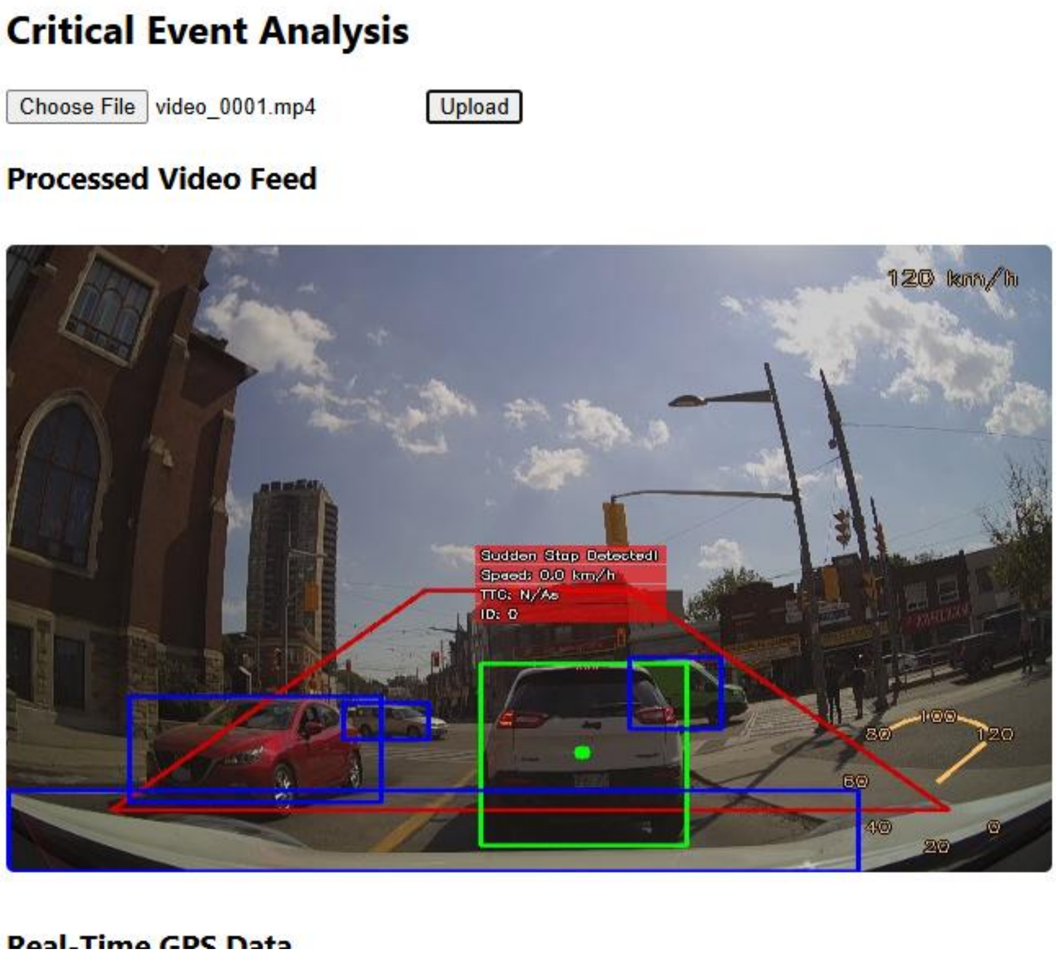


Figure 9: Vehicle data identification 03

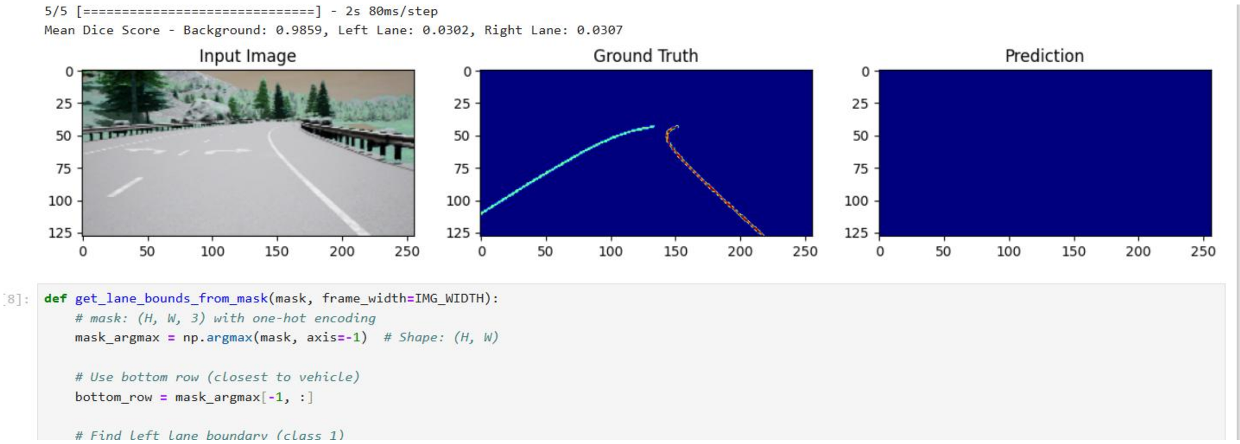


Figure 10: Training models to identify lanes

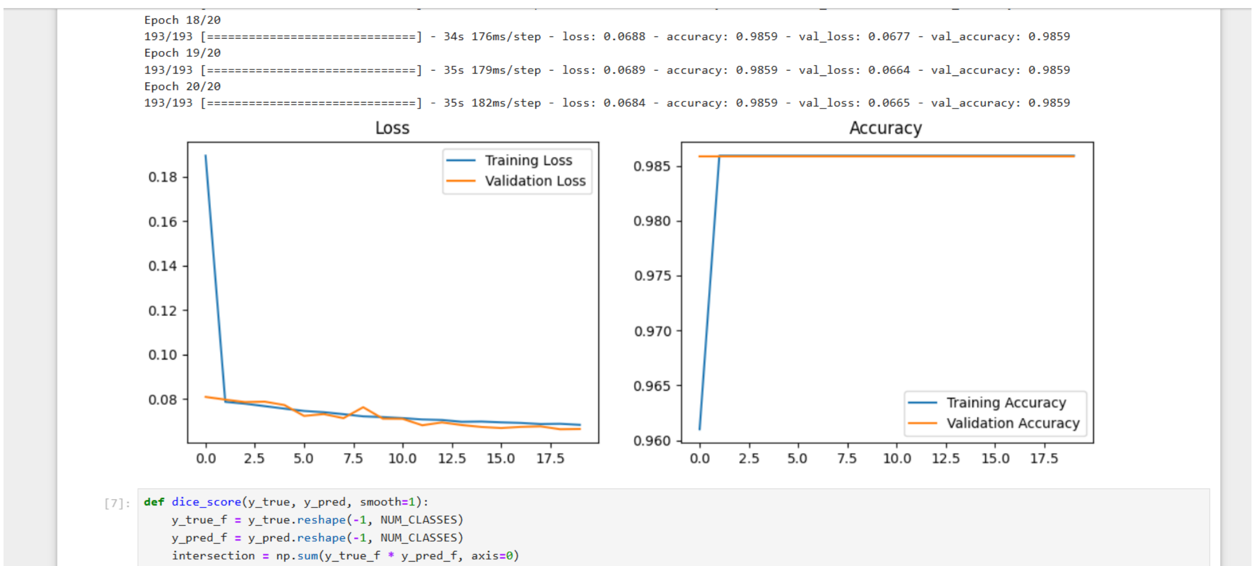


Figure 11: Training Evaluations