IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14494

Driver Assistance System: Utilising Machine Learning for Reducing Accidents, Vehicle and Road Safety

Brunda S[1], Namratha M V[2], Shreyas A S[3], Pranitha R[4], Gopika M[5]

Assistant Professor, CSE Department, JSS STU, Mysuru, India[1]

Students, CSE Department, JSS STU, Mysuru, India[2,3,4,5]

Abstract: The road safety is an important aspect in the present scenario. The project aims to improve the road safety by using machine and deep learning to monitor and classify driver behaviour in real time. It identifies ten types of activities of driver including- safe driving, texting, phone usage, drinking and more. The system uses advances CNNs, transfer learning models like VGG16 and ResNet50, and YOLOv8 for object detection. It also includes a drowsiness detection module to alert drivers showing signs of fatigue. The project uses the state farm distracted driver detection dataset for training and evaluation, and flask-based web app for real-time monitoring and alerts. Performance is measured using Accuracy, Precision, Recall, and F1-score, showing high effectiveness in enhancing driver awareness and reducing accidents. This system is suitable for modern vehicle safety and fleet management solutions. The drowsiness module is also integrated to alert the driver feeling drowsy and improve the safety. It utilizes the standard dataset of open and closed eyes for training and detects the drowsy behaviour in real time.

Keywords: ML, Road safety, VGG16, ResNet50, YOLOv8, CNN, Real-time monitoring.

I. INTRODUCTION

Road safety has become a major concern due to the increasing number of traffic-related incidents worldwide. The World Health Organization reports that around 1.3 million people die each year from road traffic crashes, with many more suffering non-fatal injuries. Driver distraction, such as texting, talking on the phone, and other activities, is a leading cause of these accidents. Traditional methods like traffic enforcement and awareness campaigns have not been sufficient to address this issue. To tackle this problem, the "Driver Assistance System" project leverages artificial intelligence and deep learning technologies. The system uses computer vision and models like CNNs, VGG16, ResNet50, and YOLOv8 to detect and classify driver behaviours in real-time. It identifies activities such as texting, talking on the phone, drinking, and more. Additionally, a drowsiness detection module monitors the driver's eyes and triggers alerts if signs of fatigue are detected. The project uses the "State Farm Distracted Driver Detection" dataset for training, which includes images of ten specific driver behaviours. The system processes real- time webcam input and issues alerts for unsafe behaviours. It is implemented as a Flask based web application, making it suitable for integration into vehicle dashboards, mobile apps, and surveillance systems. This proactive approach aims to enhance road safety by reducing accidents caused by distracted driving.

II. LITERATURE REVIEW

I.-R. Adichie et al. (2020) proposed a non-intrusive driver monitoring system using ECG and EOG sensors to track physiological signals indicating driver fatigue. The fusion of bio-signal monitoring with real-time processing techniques for timely warning generation was highlighted, though challenges in signal stability and sensor intrusiveness were noted [1].

Md. Ebrahim Shaik (2023) conducted a meta-analysis of various detection models, classifying methods into visual cues, biological signals, and vehicular behavior. The review identified CNN, LSTM, and hybrid deep learning models as promising techniques, discussing limitations like sensor intrusiveness and lighting sensitivity [2].

Das et al. (2024) presented an intelligent system combining deep learning with IoT technologies. The model used a U-Net-based convolutional network to segment facial features and track micro-expressions, integrated with IoT devices for real-time response and high precision in facial feature segmentation [3].

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025
DOI: 10.17148/IJARCCE.2025.14494

Yashar Jibraeel (2024) presented a CNN-based system optimized using Genetic Algorithms (GA) to improve detection accuracy and computational efficiency. The integration of GA allowed dynamic selection of optimal hyperparameters, leading to better classification of drowsiness levels from facial features [4].

Muhammad Ramzan (2024) proposed a custom deep learning model merging CNNs with attention mechanisms to focus on critical facial features like eyelids and yawning patterns. The study demonstrated superior performance metrics and runtime efficiency compared to baseline architectures [5].

III. SYSTEM ARCHITECHTURE

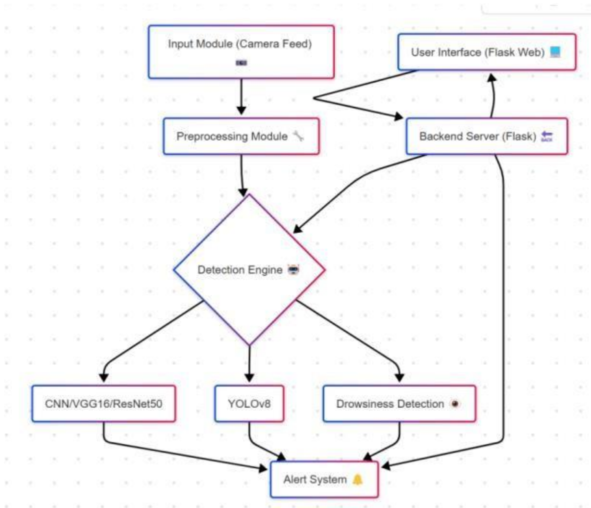

Fig. 1 System Architecture

A. User Interface Layer (Frontend Web Application):

Built with HTML, CSS, JavaScript, and Bootstrap within a Flask framework. It includes functionalities like start/stop detection buttons, live camera feed, detection results display, and alert status. The UI is responsive and user- friendly.

B. Camera Input Module:

Captures live video feed using a webcam or USB camera, streaming video at a consistent frame rate (20–30 fps) for smooth real-time inference.

C. Image Preprocessing Module:

Prepares frames for deep learning models by resizing, normalizing, and extracting regions of interest (ROI) like faces or eyes.

D. Detection Engine (Deep Learning Inference Layer):

Core processing unit with submodules:

a) *Driver Behavior Classification Model*: Uses CNN-based models (e.g., VGG16, ResNet50) to classify driver activities.
b) *YOLOv8 Model:* Detects objects and gestures indicating distraction.
c) *Drowsiness Detection Model:* Monitors eye openness and flags drowsiness if eyes remain closed for 2–3 seconds.

E. Alert System Module:

Activates upon detecting risky behavior or drowsiness, generating audio and visual alerts. It can also send SMS/email notifications to fleet managers or emergency contacts.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025
DOI: 10.17148/IJARCCE.2025.14494

F. Backend Layer (Flask Server):

Connects the frontend, camera module, detection engine, and alert system. Manages HTTP routes, real-time video feed via OpenCV, and serves detection results to the UI.

IV. METHEDOLOGY

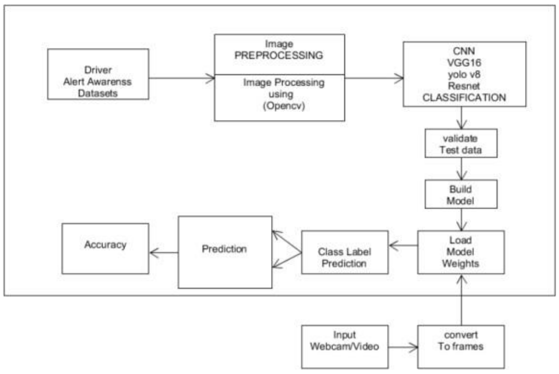

Fig. 2 Flow Diagram

A. Data Collection and Preprocessing:

a) Uses the State Farm Distracted Driver Detection dataset from Kaggle, categorized into ten classes (e.g., safe driving, texting, talking on the phone, etc.).
b) Images are resized (e.g., 224x224 pixels) and normalized.
c) Data augmentation techniques (rotation, zoom, flipping, brightness adjustments) are applied to increase diversity and reduce overfitting.

B. Model Selection and Training:

a) Multiple models trained: Custom CNN, VGG16, ResNet50, YOLOv8.
b) Training involves categorical cross- entropy loss, Adam optimizer, and tuning batch size and learning rate.
c) Training-validation split (e.g., 80:20) used to monitor performance and prevent overfitting.

C. Drowsiness Detection Module:

a) Separate CNN model monitors eye aspect ratio (EAR) using OpenCV and facial landmarks.
b) Detects drowsiness if eyes remain closed for more than 15 consecutive frames, triggering an alert.

D. Real-Time Detection and Integration:

a) Models deployed using Flask backend.
b) Webcam input captured in real time using OpenCV, processed frame-by- frame.
c) Predictions displayed on web interface, unsafe actions or drowsiness trigger alerts.

E. Web Interface and Visualization:

a) Flask-based GUI allows viewing live video feed, real-time activity predictions, and instant alerts.
b) Log system can record time-stamped alerts for future analysis or fleet management.

F. Evaluation Metrics:

a) Models evaluated using accuracy, precision, recall, F1-score, and confusion matrix.
b) YOLOv8 assessed based on mean average precision (MAP) and frames per second (FPS) for real-time performance.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14494

V. SYSTEM ENVIRONMENT

A. Flask:

A lightweight Python web framework for building scalable applications with minimalistic design and essential components.

B. Python:

A versatile programming language with extensive libraries for machine learning and web development, known for its simplicity and readability.

C. Jupyter Notebook:

An interactive computing environment for prototyping, experimenting, and testing machine learning models with real-time data visualization.

D. TensorFlow:

A powerful open-source framework for building and training neural network models, developed by Google.

E. Keras:

A high-level neural networks API that simplifies deep learning model construction and training, operating atop TensorFlow.

F. NumPy:

A fundamental library for numerical computing in Python, supporting multi- dimensional arrays and mathematical functions.

G. Pandas:

A popular library for data manipulation and analysis, providing efficient handling of structured data through DataFrames.

H. Scikit-learn (sklearn):
A versatile library offering a wide array of machine learning algorithms and tools for building and deploying models.

VI. CHALLENGES AND LIMITATIONS

A. Dataset had the smaller training set and larger testing set.
B. Preprocessing tasks were lengthy.
C. Does not detect the objective if low light or poor lighting condition.
D. Drowsiness cannot be detected with the person wearing shades or goggles.
E. Made few trails and errors for the selection of optimum algorithm for implementation.
F. Required the usage of GPUs for training purpose.
G. Prolonged training period.

VII. EXPERIMENTAL RESULTS

A. Observations
a) *CNN Model (Behavior classification):*

Metric	Value
Accuracy	89.45%
Precision	88.60%
Recall	87.90%
F1-Score	88.24%

Fig. 3 CNN Analysis

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14494

b) *VGG16 Transfer Learning Results:*

Metric	Value
Accuracy	94.32%
Precision	93.90%
Recall	94.50%
F1-Score	94.20%

Fig. 4 VGG16 Analysis

c) *YOLOv8 Real*-Time Detection:

Test Case	Detection Accuracy	Average FPS	Inference Time
Phone Usage	97%	22 FPS	0.045s
Texting	95%	20 FPS	0.050s
Drinking	93%	21 FPS	0.047s

Fig. 5 YOLOv8 Analysis

d) *Drowsiness Detection (eye ratio, CNN):*

Scenario	Detection Accuracy	False Positives	Alert Delay
Eyes closed for >3 seconds	98.2%	Low	1.2s
Rapid blinking (not drowsy)	95.5%	Moderate	0.9s
Eyeglass obstruction	92.7%	Slightly High	1.4s

Fig. 6 Drowsiness Module Analysis

B. Results

a) Home Page:

Fig. 7 Home Page

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14494

b) About Page:

Fig. 8 About Page

c) Video-feed detection:

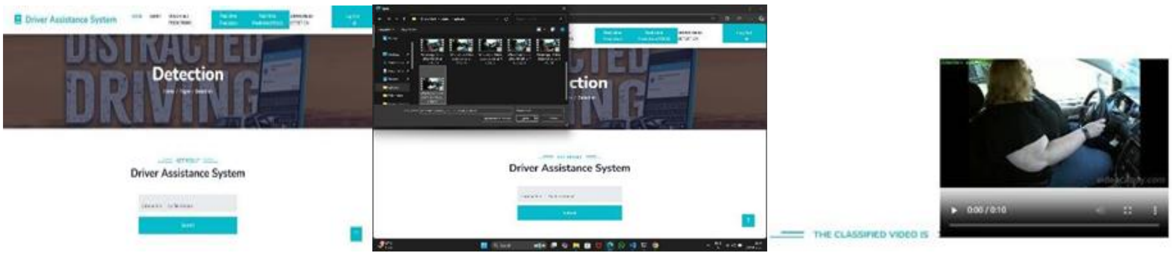

Fig. 9 Video-feed detection

d) Realtime detection:

Fig. 10.1 Talking on Phone

Fig. 10.2 Safe Driving

e) Realtime detection (YOLOv8):

Fig. 11 Drinking

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14494

f) Drowsiness detection:

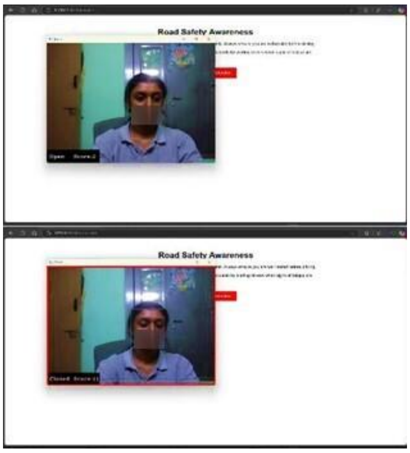

Fig. 12 Closed and Open Score

VIII. CONCLUSION

The "Driver Assistance System" project enhances road safety by using deep learning algorithms (CNN, ResNet50, YOLOv8) to detect distracted and drowsy driving behaviours. It identifies ten classes of distractions and includes a drowsiness detection module based on Eye Aspect Ratio (EAR) and CNN. Deployed as a Flask web application with real-time webcam integration, it provides audio alerts for unsafe behaviours. Extensive evaluation showed high accuracy and responsiveness, validating its potential to reduce accidents and save lives. This scalable framework can be adapted for future intelligent Transportation system.

A. Algorithm comparison and integration

Algorithm	Purpose	Speed (FPS)	Accuracy	Role in System
CNN	Baseline behaviour classification	Medium	Moderate	Classify driver behaviours
ResNet50	Enhanced classification	Medium	High	Fine-tuned for high-accuracy output
YOLOv8	Real-time object detection	High	High	Live detection of unsafe actions
Drowsiness CNN	Eye state monitoring	Very High	High	Alert on continuous eye closure

Fig. 13 Integrated Analysis

B. Comparative analysis

Model	Accuracy	F1 Score	Real-Time Performance
CNN	89.45%	88.24%	Medium (0.12s/frame)
ResNet50	94.32%	94.20%	Medium (0.18s/frame)
YOLOv8	96%+	95%+	High (22 FPS)
Drowsiness CNN	98.2%	97.5%	High (1.2s Alert Delay)

Fig. 14 Comparative Analysis

IX. FUTURE SCOPE

A. Mobile and Embedded Deployment: Convert models using TensorFlow Lite or PyTorch Mobile for efficient operation on Android or embedded platforms like Raspberry Pi. This would enable real-time in-vehicle deployment without external hardware dependency.

B. Edge Computing and IoT Integration: Integrate the system with edge devices and IoT sensors to aggregate multiple behavioral parameters like head tilt, heart rate, and steering patterns. This holistic approach would enhance driver profiling accuracy.

DOI: 10.17148/IJARCCE.2025.14494

C. Night Vision and Infrared Support: Extend the system to support infrared or night vision cameras for functionality in low-light or nighttime conditions. This would ensure reliable 24/7 operation.

D. Context-Aware Detection: Enhance the system to analyse road context using road-facing cameras to distinguish between safe and unsafe distractions. This context awareness would reduce false positives and increase system intelligence.

REFERENCES

[1]. I. -R. Adochiei et al., "Drivers' Drowsiness Detection and Warning Systems for Critical Infrastructures," 2020 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 2020, pp. 1-4, doi: 10.1109/EHB50910.2020.9280165.

[2]. Md. Ebrahim Shaik,A systematic review on detection and prediction of driver drowsiness,Transportation Research Interdisciplinary Perspectives,Volume 21,2023,100864,ISSN 2590-1982.

[3]. Das, S.; Pratihar, S.; Pradhan, B.; Jhaveri, R.H.; Benedetto, F. IoT-Assisted Automatic Driver Drowsiness Detection through Facial Movement Analysis Using Deep Learning and a U-Net-Based Architecture. Information 2024, 15, 30.

[4]. YASHAR JEBRAEILY Driver Drowsiness Detection Based on Convolutional Neural Network Architecture Optimization Using Genetic Algorithm Received 1 February 2024, accepted 22 March 2024, date of publication 27 March 2024, date of current version 2 April 2024. Digital Object Identifier 10.1109/ACCESS.2024.3381999.