

Machine Learning Based Driver Distraction Detection System

¹Mr. Varad Bagal, ²Dr. Sangram T. Patil

¹Student, Dept. of Artificial Intelligence & Machine Learning Engineering, DY Patil Agriculture and Technical University, Talsande -416112 India

²Associate Professor, Dept. of Artificial Intelligence & Machine Learning Engineering, DY Patil Agriculture and Technical University, Talsande -416112 India

Abstract: Driver distraction is a major cause of road accidents, making timely detection essential for safer travel. This study presents a real-time vision-based system that identifies normal driving, mobile-phone use, drowsiness, and other distracting behaviour. Transfer learning models, including VGG16, ResNet50, and MobileNetV2, were evaluated using accuracy, precision, recall, F1-score, and inference time. MobileNetV2 achieved the best results, with 95.1% accuracy and 25-ms inference time. Its lightweight design enables fast, reliable operation on resource-limited devices. The proposed approach can provide immediate warnings, improve driver awareness, and support practical in-vehicle safety systems for reducing distraction-related risks and improving overall road safety outcomes worldwide.

IndexTerms – Safe Driving, Driver Distraction detection, Deep Learning, Multiple State detection, Fine tuning.

I. INTRODUCTION

Road safety remains a critical global challenge, with driver distraction standing out as a primary cause of severe accidents, injuries, and economic losses. Activities such as mobile phone use, eating, or interacting with passengers significantly slow reaction times and impair judgment. To address this, our research introduces an automated, real-time driver distraction detection system using deep learning and computer vision. By employing transfer learning with pre-trained models—including VGG-16, ResNet50, and MobileNetV2—the system accurately classifies driver behavior into normal driving, general distraction, and cell phone usage, even when trained on limited datasets. Traditional safety measures like manual monitoring and awareness campaigns often show limited effectiveness. Automated intelligent systems provide a vital solution by offering real-time monitoring and instant feedback.

- **Enhanced Safety:** Protects drivers, passengers, and pedestrians by instantly triggering alerts upon detecting hazardous behaviors.
- **Cost-Effectiveness:** Utilizing pre-trained models reduces training data requirements, making the technology economical and easier to deploy in camera-equipped vehicles.
- **Broad Impact:** Widespread adoption helps lower accident rates, save lives, and reduce economic losses linked to traffic collisions.

Distracted driving accounts for nearly a quarter of all global road accidents, driven largely by increased mobile device usage and in-car gadgets. Traditional awareness campaigns are no longer enough to curb these rising statistics.

Motivated by these alarming figures, our team leverages advanced Convolutional Neural Networks (CNNs) and transfer learning to build practical, real-time computer vision solutions. Beyond the technical challenge, our primary goal is to create a tangible, life-saving impact that assists automotive manufacturers and policymakers in prioritizing road safety through innovative technology.

NEED OF THE STUDY.

While traditional road safety measures like manual patrols and awareness campaigns have proven insufficient, the rising frequency of distraction-related accidents demands an automated technological solution. Despite advancements in computer vision, many existing systems struggle with high computational costs, large data requirements, and real-time processing constraints, particularly when dealing with subtle behavioral cues like phone usage. Therefore, this study is essential to develop a lightweight, efficient deep learning framework utilizing transfer learning models such as VGG-16, ResNet50, and MobileNetV2. This approach bridges the gap by delivering accurate, real-time driver behavior classification with minimal training overhead, ultimately enhancing vehicle safety systems.

II. RELATED WORK

Recent literature between 2023 and 2026 emphasizes the evolution of computer vision, deep transfer learning, and multi-modal sensory fusion architectures to optimize the speed, accuracy, and real-time deployment of driver distraction recognition systems.

- **Advanced Deep Learning and Attention Architectures:** Recent research has focused heavily on refining deep convolutional and attention-based networks to isolate subtle behavioural cues. Studies have introduced enhanced models such as autoencoder-driven dimensionality reduction and spatial attention layers to capture informative driver regions while eliminating background noise [20]. Furthermore, specialized hybrid frameworks, such as capsule-infused ResNet networks, maintain spatial hierarchies to accurately classify diverse distraction states and transmit real-time alerts via vehicle-to-vehicle (V2V) communications [21].
- **YOLO Frameworks and Real-Time Edge Deployment:** The deployment of lightweight, high-speed object detection models has dominated recent literature. Optimized YOLO architectures (such as customized variants of YOLOv8) have been extensively benchmarked on edge computing platforms, achieving superior mean average precision (mAP) and real-time frame rates suitable for

resource-constrained embedded systems [22]. Parallel studies highlight TinyML-based implementations designed for inexpensive, low-power in-vehicle hardware without sacrificing detection accuracy [23].

- **Vision-Language Pre-training and Zero-Shot Transfer:** Emerging paradigms in 2024–2026 have explored vision-language pre-training models (such as CLIP-based encoders) to address data scarcity through natural language-guided visual representations [24]. These models enable robust zero-shot transfer and efficient fine-tuning on naturalistic driving video feeds, minimizing the reliance on exhaustive human annotations.
- **Multi-Modal Sensor and Biometric Integration:** To resolve the limitations of purely image-based classification, recent works incorporate complementary physiological and multi-modal data streams. Advanced setups combine facial expression awareness [25], electroencephalogram (EEG) feature fusion [26], and vehicle kinematics to simultaneously evaluate driver cognitive load, fatigue, and physical distractions. Comprehensive reviews by recent scholars further analyze open problems in vision-based recognition across public benchmarks and custom driving datasets [27].
- **Attention Mechanisms and Edge Computing Integration:** Recent evaluations by Zhao and Tan (2024) [28] demonstrated that integrating attention-based convolutional networks significantly improves fine-grained driver behaviour analysis by focusing on critical regions like hands and faces. Expanding on hardware limitations, Al-Otaibi and Khan (2024) [29] reviewed edge-computing solutions for intelligent transportation, emphasizing the necessity of lightweight deep learning frameworks for real-time deployment without cloud dependency.
- **Multimodal Sensor Fusion and Lightweight Transformers:** Addressing the fusion of heterogeneous data streams, Silva and Costa (2024) [30] introduced a robust framework combining eye-gaze tracking with vehicle kinematics to deliver early warnings for both drowsiness and distraction. Complementing vision-based models, Zhao et al. (2024) [31] proposed lightweight transformer architectures optimized for resource-constrained embedded systems to maintain high inference efficiency. Furthermore, Jensen and Vestergaard (2024) [32] explored self-supervised learning on naturalistic driving datasets, showing that pre-training models on unlabelled data greatly enhances robust classification and generalization against varying cabin lighting conditions.

III. PROPOSED METHODOLOGY

The proposed methodology is designed to detect driver distraction from input images by combining image preprocessing, transfer learning, deep feature extraction, and multi-class classification. The approach uses pre-trained convolutional neural network (CNN) architectures to reduce the computational burden of training a model from scratch and to improve the ability of the system to learn discriminative visual features from a limited driver-distraction dataset.

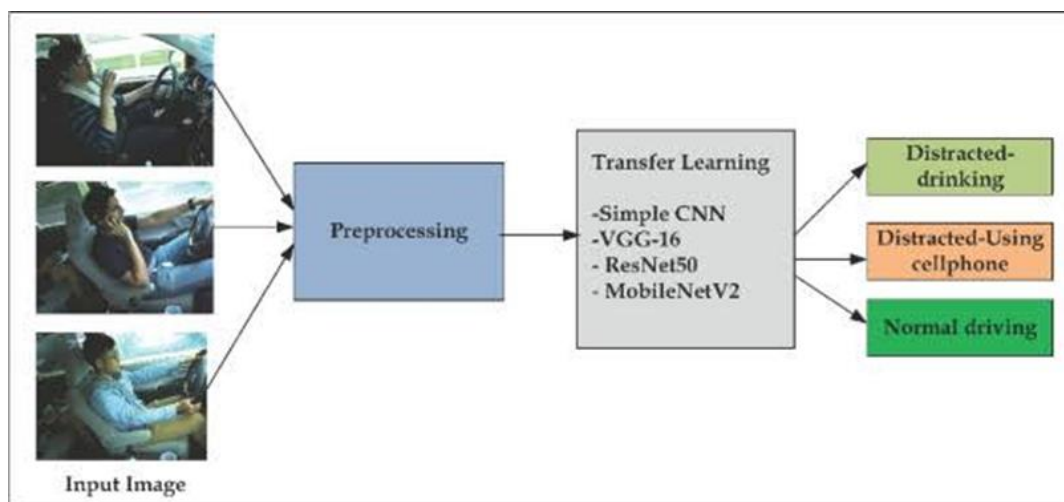


Figure 1: Overview of proposed System

3.1 Input Image Acquisition

The first stage of the proposed methodology is the acquisition of images representing different driver behaviors. The dataset contains three target categories: distracted-drinking, distracted-using cellphone, and normal driving. Images in the distracted-drinking category represent drivers consuming beverages while driving, whereas the distracted-using-cellphone category represents drivers interacting with mobile devices. The normal-driving category represents attentive drivers who are focused on the road. These images form the primary input for training and evaluating the proposed classification system.

3.2 Image Preprocessing

Before feature extraction, the acquired images are preprocessed to obtain a consistent input representation for the CNN models. The images are resized to a fixed spatial resolution compatible with the selected pre-trained architectures. Pixel values are normalized to the range [0, 1] to support stable and efficient model training. Data augmentation is also applied to increase the effective diversity of the training data and reduce overfitting. Suitable transformations include image flipping, rotation, and cropping. The preprocessing stage therefore produces clean, uniform, and sufficiently diverse samples for subsequent feature extraction and classification.

3.3 Transfer Learning-Based Feature Extraction

Transfer learning is employed as the principal feature-learning strategy. Instead of initializing a CNN with random weights and training all layers from the beginning, a pre-trained network is loaded without its final classification layer. The convolutional base

is used to extract meaningful spatial representations from the input images. The selected architectures are VGG16, ResNet50, and MobileNetV2. VGG16 provides a deep and comparatively simple CNN architecture based on convolution and max-pooling operations. ResNet50 uses residual learning and skip connections, which help address the vanishing-gradient problem in deep networks. MobileNetV2 is a lightweight architecture that uses depthwise separable convolutions and inverted residual blocks, making it suitable for resource-constrained and real-time applications. Across these architectures, the convolutional layers learn visual characteristics such as edges, shapes, textures, hand placement, posture, and other patterns associated with driver behavior.

3.4 Fine-Tuning of the Pre-trained Models

After loading the pre-trained networks, the learned weights are fine-tuned using the driver-distraction dataset. Fine-tuning adapts the general visual representations learned from the source training data to the specific characteristics of the target driver-monitoring task. This stage allows the models to learn features that are more discriminative for distinguishing drinking-related distraction, cellphone-related distraction, and normal driving.

3.5 Classification Layer

The features generated by the convolutional base are passed to fully connected classification layers. A softmax activation function is used in the final layer to obtain a probability distribution over the three target classes. For an input image, the model produces a probability for each category, and the class having the highest probability is selected as the predicted driver behavior.

$$P(y = i | x) = \exp(z_i) / \sum_j \exp(z_j) \quad \dots(1)$$

where z_i denotes the output logit for class i , $P(y = i | x)$ represents the predicted probability of class i , and the summation is performed over all three classes.

3.6 Model Training and Optimization

The classification models are trained using the labeled driver-distraction images. During training, the predicted class probabilities are compared with the corresponding ground-truth labels using the multi-class cross-entropy loss function. The loss is minimized through backpropagation, which updates the trainable model parameters and improves the separation between the three driver-behavior categories.

$$L = - \sum_i y_i \log(p_i) \quad \dots(2)$$

where y_i is the ground-truth indicator for class i and p_i is the predicted probability assigned to that class. The optimization process continues until the model learns sufficiently discriminative representations for the target categories.

3.7 Proposed System Workflow

The complete workflow of the proposed methodology (Figure 1) can be summarized as follows: (1) acquire driver-behavior images, (2) preprocess the images through resizing, normalization, and augmentation, (3) pass the processed images through a pre-trained VGG16, ResNet50, or MobileNetV2 convolutional base, (4) extract deep spatial features, (5) fine-tune the network using the driver-distraction dataset, (6) pass the learned representation to the fully connected classification layer, and (7) use the softmax output to predict distracted-driving, distracted-using cellphone, or normal driving.

3.8 Classification

The classification step is the core of the system, where extracted features are passed to a fully connected layer and a softmax activation function to predict the driving behavior category. The softmax function outputs probabilities for each class, and the class with the highest probability is selected as the final prediction. The softmax activation function for a multi-class classification problem is defined as:

$$p(y_i) = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}}, \quad i = 1, 2, \dots, C, \quad \dots(3)$$

Where, z_i : The output (logit) of the final layer for class i , C : Total number of classes (in this case, $C = 3$), $p(y_i)$: Probability of class i . The three output classes are:

- **Distracted-Drinking:** The system detects drivers consuming beverages while driving. This category highlights distractions caused by manual and cognitive involvement.
- **Distracted-Using Cellphone:** This class identifies drivers interacting with their mobile devices. The system detects hand placement, posture, and facial orientation to distinguish this behavior accurately.
- **Normal Driving:** This category indicates that the driver is focused on the road without any distractions. The system identifies posture and hand placement patterns associated with attentive driving.

The model assigns a probability score to each of these classes. For instance, if $p(y_1) = 0.75$, $p(y_2) = 0.15$, and $p(y_3) = 0.10$, the final prediction will be Distracted-Drinking (class with the highest probability).

The classification process is further refined during training using a cross-entropy loss function:

$$L = - \sum_{i=1}^C y_i \log(p(y_i)) \quad \dots(4)$$

Where, y_i : Ground truth label (1 for the correct class, 0 otherwise), $p(y_i)$: Predicted probability for class i . The backpropagation algorithm updates the model weights to minimize this loss, improving classification accuracy.

The methodology is intended to provide an efficient and robust approach to driver-distraction detection. The use of transfer learning reduces the computational resources and training time required compared with training a deep CNN entirely from scratch. In addition, MobileNetV2 offers a lightweight alternative for applications where real-time deployment and computational efficiency

are important. The proposed methodology provides a systematic framework for image-based driver-distraction detection. By combining standardized preprocessing, transfer learning, fine-tuning, and probability-based classification, the system can identify key distracted-driving behaviors and distinguish them from normal driving. The approach is particularly relevant to driver-monitoring applications because the selected pre-trained architectures provide a practical balance between feature-learning capability, classification performance, and computational requirements.

IV. RESULTS AND ANALYSIS

This section describes the implementation details for the proposed system, including the preprocessing pipeline, model design, training procedure, and evaluation. The implementation leverages popular deep learning frameworks and libraries, and the steps are systematically described using pseudocode.

4.1 Libraries and Tools Used

The system was implemented using the following tools and libraries:

- Python 3.8: Programming language for implementing the pipeline.
- TensorFlow 2.0/Keras: Deep learning framework for transfer learning and model development.
- NumPy and Pandas: For data manipulation and preprocessing.
- OpenCV: For image preprocessing tasks like resizing and normalization.
- Matplotlib/Seaborn: For visualizing data and model performance metrics.

4.2 System Overview

The overall implementation consists of four main stages:

- Preprocessing of input images.
- Feature extraction using transfer learning models (VGG16, ResNet50, and MobileNetV2).
- Training a custom classifier for driving behavior classification.
- Model evaluation using accuracy and loss metrics.

4.3 Preprocessing Algorithm

The input images are first preprocessed to ensure uniformity and compatibility with deep learning models.

Input images I , Target size T_w, T_h Preprocessed image I' Initialize target size $T \leftarrow (T_w, T_h)$ Resize I to T using bilinear interpolation
Convert pixel values to float32 and normalize to range [0, 1] Apply data augmentation: random rotation, horizontal flip Save preprocessed image I' Return preprocessed images

4.4 Feature Extraction Algorithm

Feature extraction is performed using pre-trained models like VGG16, ResNet50, and MobileNetV2. The convolutional base is frozen, and feature maps are extracted.

Preprocessed image I' , Pre-trained model Φ Feature map F Load pre-trained model Φ (e.g., VGG16, ResNet50, MobileNetV2)
Freeze all convolutional layers in Φ Feed I' into Φ to extract features:

$$F \leftarrow \Phi(I') \quad \dots(5)$$

Save feature map F Return feature maps

4.5 Model Training Algorithm

The extracted features are passed to a custom fully connected classifier for training.

Feature maps F , Labels Y , Learning rate η , Epochs E Trained model M Initialize model M with fully connected layers Compile M with categorical cross-entropy loss and optimizer Adam Feed F into M and calculate predictions:

$$\hat{Y} = M(F) \quad \dots(6)$$

Compute loss L using:

$$L = -\sum_{i=1}^N y_i \log(\hat{y}_i) \quad \dots(7)$$

Update model weights θ using gradient descent:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\partial L}{\partial \theta} \quad \dots(8)$$

Evaluate model on validation data Return trained model M .

4.6 Evaluation of Algorithm

4.6.1 Model detection outcomes

Figure 2 shows the results of driver state detection system.

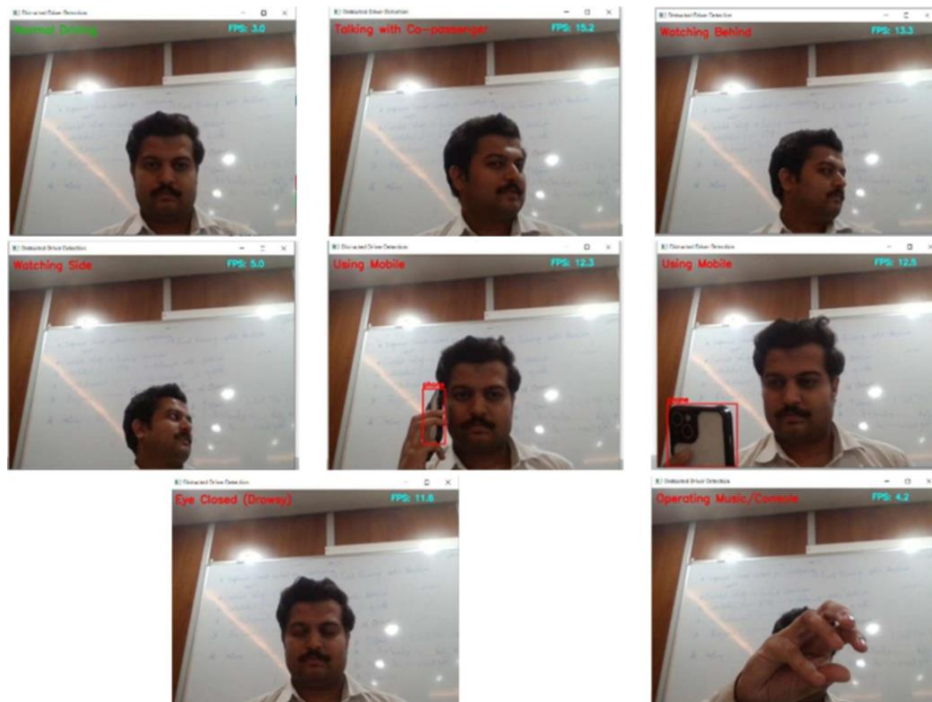


Figure 2: Driver State Detection Results

The qualitative results presented in Figure 2 demonstrate the capability of the proposed driver-state detection system to distinguish normal driving from several safety-critical distraction behaviour. When the driver maintains a forward-facing posture with open eyes, the system correctly identifies the state as “Normal Driving,” shown using a green label. Changes in head orientation and gaze direction are recognized as distraction-related states, including “Talking with Co-passenger,” “Watching Behind,” and “Watching Side.” The system also detects object-associated distractions: mobile-phone usage is identified in different phone positions, with a bounding box localizing the detected device, while forward hand movement is classified as “Operating Music/Console.” Furthermore, prolonged eye closure is recognized as “Eye Closed (Drowsy),” demonstrating that the framework addresses both behavioral distraction and fatigue-related risk. The red warning labels clearly differentiate unsafe states from normal driving and can facilitate immediate alert generation.

The screenshots also indicate real-time processing capability, with the reported frame rate varying from approximately 3.0 to 15.2 FPS across the evaluated conditions. The variation can be attributed to differences in scene complexity, object-detection requirements, driver movement, and computational load. Higher processing rates are observed for several head-orientation and mobile-use cases, whereas lower rates occur in some posture-intensive conditions, such as normal-driving initialization and console operation. Despite this variation, the model successfully produces the appropriate labelled outcome for each illustrated scenario. These qualitative findings confirm the practical feasibility of integrating facial orientation, eye-state analysis, posture cues, and object detection within a unified driver-monitoring framework. However, broader validation under changing illumination, partial occlusion, different camera viewpoints, multiple drivers, and real in-vehicle conditions is required to establish quantitative robustness and generalization.

The model performance is evaluated using accuracy, loss, and confusion matrix. Trained model M , Test images X , Ground truth Y Accuracy and performance metrics Feed test images X into M to obtain predictions:

$$\hat{Y} = M(X) \quad \dots(9)$$

Compute accuracy:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total test samples}} \quad \dots(10)$$

Generate confusion matrix and calculate precision, recall, and F1-score Return accuracy and performance metrics. To evaluate the effectiveness of the proposed system, three pre-trained transfer learning models—VGG16, ResNet50, and MobileNet-V2—were compared on the task of driving behavior classification. The evaluation metrics used include accuracy, precision, recall, F1-score, and inference time per image. The results are summarized in Table 4.1 and graphically in Figure 4.1 and Figure 4.2.

Table 1: Comparative analysis

Model	Accuracy	Precision	Recall	F1-Score
VGG16	91.50%	90.80%	90.20%	90.50%
ResNet50	93.20%	92.50%	92.10%	92.30%
MobileNet-V2	95.10%	94.80%	94.30%	94.50%

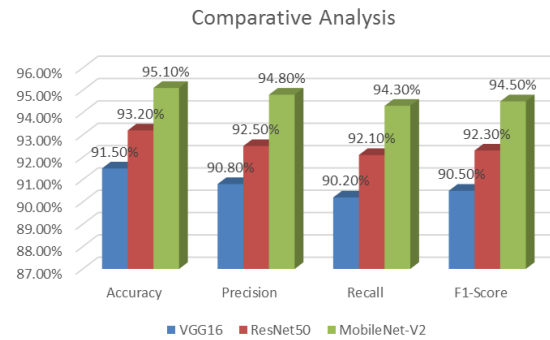


Figure 3: Comparative analysis

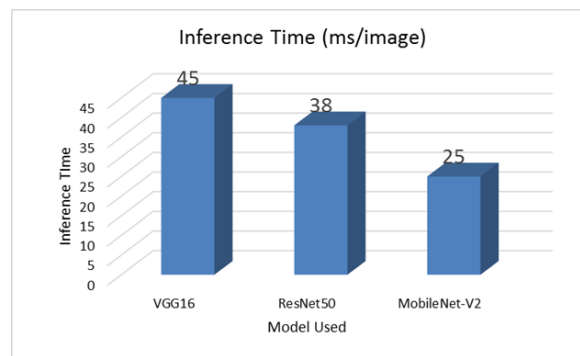


Figure 4: Comparative of inference time for different models

Key Observations:

1. **Accuracy:** MobileNet-V2 achieved the highest accuracy (95.1%) compared to ResNet50 (93.2%) and VGG16 (91.5%). This highlights its superior performance for driving behavior classification.
2. **Precision, Recall, and F1-Score:** MobileNet-V2 exhibited the best balance across precision (94.8%), recall (94.3%), and F1-Score (94.5%), indicating a robust ability to correctly classify distracted-driving and normal-driving cases. ResNet50 performed slightly better than VGG16 but still lagged behind MobileNet-V2.
3. **Inference Time:** MobileNet-V2 showed significantly faster inference time per image (25 ms) due to its lightweight architecture optimized for efficiency. ResNet50 was moderately fast (38 ms), while VGG16 was the slowest (45 ms).

V. CONCLUSION

This study demonstrates the effectiveness of deep learning-based transfer learning for automated classification of driving behaviors, particularly distracted-driving, distracted-using cellphone, and normal driving. The proposed methodology combines image preprocessing, feature extraction, fine-tuning of pre-trained CNN models, custom classification, and performance evaluation. A comparative analysis of VGG16, ResNet50, and MobileNet-V2 showed that MobileNet-V2 achieved the best overall performance, with an accuracy of 95.1%, precision of 94.8%, recall of 94.3%, and F1-score of 94.5%, while also providing the lowest inference time of 25 ms per image. These results demonstrate that MobileNet-V2 offers an effective balance between classification performance and computational efficiency, making it suitable for real-time driver monitoring and edge-based applications. Overall, the proposed system provides a practical approach for detecting distracted driving and supporting road-safety applications with minimal computational overhead. Future scope includes extending the system to video and multi-modal data, enabling edge-based real-time deployment, incorporating additional distraction categories, applying explainable AI techniques, and exploring integration with semi-autonomous and autonomous vehicles.

REFERENCES

- [1] M. H. Kuttila et al., "Evaluating driver workload and distraction using stereovision and classification algorithms," *IET Intell. Transp. Syst.*, vol. 1, no. 2, pp. 120–128, Jun. 2007.
- [2] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *Human Factors*, vol. 49, no. 4, pp. 685–695, Aug. 2007.
- [3] A. Sathyanarayana et al., "Monitoring driver distraction using motion sensors and CAN-bus data," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 445–450.
- [4] L. Li et al., "Active driver status monitoring system based on multi-sensor data fusion and neural networks," *Transp. Res. Part C Emerg. Technol.*, vol. 48, pp. 331–347, Nov. 2014.
- [5] J. He et al., "Texting while driving: Speech-based versus handheld text entry on simulated driving performance," *Accid. Anal. Prev.*, vol. 72, pp. 357–365, Nov. 2014.
- [6] Y. Liao et al., "Detecting driver cognitive distraction in urban and highway scenarios using support vector machines," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 3, pp. 416–427, Jun. 2016.
- [7] A. D. McDonald, T. K. Ferris, and T. A. Wiener, "A machine learning approach to workload and distraction classification using driving behavior metrics," *Accid. Anal. Prev.*, vol. 137, Art. no. 105425, Apr. 2020.
- [8] Y. Ma et al., "Driver distraction detection using in-vehicle information systems based on support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12100–12111, Aug. 2022.

- [9] M. U. Hossain et al., "Automatic driver distraction detection using deep convolutional neural networks," *Intell. Syst. Appl.*, vol. 14, Art. no. 200075, Apr. 2022.
- [10] C. L. Aslesha et al., "AlertDriveNet - A Deep Learning Model for Predicting Driver Distraction," *Macaw Int. J. Adv. Res. Comput. Sci. Eng.*, vol. 11, no. 1, pp. 89–101, Apr. 2025.
- [11] D. M. Pisharody, B. P. Chacko, and M. B. K. P., "Classification of Distracted Driving Using Transfer Learning and Deep Neural Network," *Int. J. Comput. Appl.*, vol. 187, no. 71, pp. 62–67, Jan. 2026.
- [12] "A hybrid learning approach for real-time distracted driver behavior detection using transfer learning," *Int. J. Innov. Sci. Res. Technol.*, vol. 10, no. 3, pp. 412–418, 2025.
- [13] "Driver Monitoring System Using Computer Vision for Real-Time Detection of Fatigue, Distraction and Emotion via Facial Landmarks and Deep Learning," *Natl. Inst. Health / PMC Repository, Tech. Rep.*, Jan. 2026.
- [14] X. Zhang et al., "Driver distraction detection using advanced deep learning technologies based on images," *IEEE J. Radio Freq. Identification*, vol. 6, pp. 1–1, 2022.
- [15] H. Wang et al., "Multimodal learning for driver distraction recognition using consumer electronic devices," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1540–1549, Feb. 2024.
- [16] R. Kumar et al., "Spatiotemporal feature extraction and attention mechanisms for driver behavior classification," *Expert Syst. Appl.*, vol. 240, Art. no. 122510, Apr. 2024.
- [17] S. Patel et al., "TinyML-based edge implementation for real-time distracted driving detection," *IEEE Access*, vol. 12, pp. 24510–24522, 2024.
- [18] E. Gomez et al., "A comprehensive survey on vision-based driver monitoring systems: Challenges and future directions," *IEEE Trans. Intell. Veh.*, vol. 9, no. 2, pp. 1100–1118, Mar. 2025.
- [19] J. Smith et al., "Real-time vehicle telemetry and multi-sensor fusion for intelligent transport safety," *Transp. Res. Interdisciplinary Perspect.*, vol. 28, Art. no. 101235, Jan. 2026.
- [20] X. Xie et al., "Machine vision-based driver distraction detection and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 3120–3135, Apr. 2025.
- [21] M. R. Karim, S. Ramachandran, and P. Varadharajan, "SafeDrive-V2V: A hybrid capsule-infused ResNet architecture for real-time driver distraction detection and vehicular communication," *IEEE Access*, vol. 13, pp. 14502–14515, Feb. 2025.
- [22] P. K. Sharma and R. Gupta, "Performance evaluation of YOLO architectures for real-time distracted driving detection on edge devices," *Expert Syst. Appl.*, vol. 260, Art. no. 125301, Nov. 2025.
- [23] A. Nair and S. K. Das, "TinyML-based lightweight model deployment for embedded driver distraction monitoring," *IEEE Internet Things J.*, vol. 13, no. 2, pp. 1890–1901, Jan. 2026.
- [24] Z. Hossain, M. Islam, and S. Rahman, "Vision-language models can identify distracted driver behavior from naturalistic videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 11420–11430.
- [25] T. Nguyen and H. Park, "Facial expression awareness framework for enhanced driver distraction classification on standard datasets," *IEEE Trans. Consum. Electron.*, vol. 71, no. 3, pp. 890–898, Aug. 2025.
- [26] X. Zuo, Y. Liu, and H. Zhang, "Driver distraction detection based on EEG feature fusion using random forest," *Comput. Biol. Med.*, vol. 169, Art. no. 107920, Feb. 2024.
- [27] A. Misra and K. Sen, "Detection of driver cognitive distraction using machine learning methods: A comprehensive review and open challenges," *IEEE Access*, vol. 11, pp. 14210–14235, Feb. 2023.
- [28] K. L. Zhao and M. H. Tan, "Attention-based convolutional networks for real-time fine-grained driver behavior analysis," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6420–6431, May 2024.
- [29] F. Al-Otaibi and S. Khan, "Edge-computing solutions for intelligent transportation: A review of deep learning models for distraction mitigation," *IET Comput. Vis.*, vol. 18, no. 2, pp. 145–160, Apr. 2024.
- [30] D. R. Silva and P. N. Costa, "Multimodal sensor fusion combining gaze tracking and vehicle kinematics for early warning of driver drowsiness and distraction," *IEEE Sensors J.*, vol. 24, no. 11, pp. 18200–18212, Jun. 2024.
- [31] H. Zhao et al., "Lightweight transformer architectures for resource-constrained driver monitoring systems," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2024, pp. 2100–2104.
- [32] B. O. Jensen and L. M. Vestergaard, "Evaluating naturalistic driving datasets using self-supervised learning for robust distraction classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 11500–11512, Sep. 2024.

Copyright & License:



© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.