IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

Implementation of Real Time Distracted Driver Detection using Artificial Intelligence

Lipakshi Banpurkar[1], Prof. Diksha Fulzele[2], Prof. Aditya Lavhale[3]
U.G. Students, Department of Computer Science and Engineering[1]
Professor, Department of Computer Science and Engineering[2,3]
Tulsiramji Gaikwad-Patil Institute of Engineering & Technology, Mohgaon, Nagpur, Maharashtra, India
lnbkar6@gmail.com, dikshafulzele73@gmail.com

Abstract: *Around the world, there has been a lot of worry about traffic fatalities. There is seldom a day when there isn't some news about traffic fatalities. nThe major reasons for the sharp increase in the number of traffic accidents are thought to be changing environmental circumstances and the failure of traffic sign recognition systems to materialise. As the world moves closer to a day when driverless or autonomous vehicles will be the norm, this research suggests a cutting-edge method to assist motorists in driving safely by focusing primarily on a technique for detecting traffic signs on the road The suggested system is a component of the Intelligent Transport System (ITS), which may one day become a Smart City dimension. The work done here mostly focuses on the difficulties that could be overcome to prevent traffic accidents by implementing a device that can help drivers with low vision see traffic signs while driving precedence to business signals. We fail to see business signs for a variety of reasons, including difficulty fastening, fatigue, and sleep deprivation.*

Keywords: The work-done describes the usage of Binarization and Region Of Interest to preprocess pictures for traffic sign detection and recognition using Convolution Neural Network(CNN) classification model using built-in OpenCV functionalities (ROI).

I. INTRODUCTION

In India, there are 400 business accidents per day, according to sanctioned statistics. Road signs insure the safety of both motorcars and climbers by precluding accidents from being. also, business signals reduce the prevalence of business offences by icing that motorists follow certain laws. The operation of business signals facilitates route navigation as well. All druggies of the road, including climbers and motorcars, should give precedence to business signals. We fail to see business signs for a variety of reasons, including difficulty fastening, fatigue, and sleep deprivation.

Other reasons for ignoring the indicators include impaired vision, the outside world's influence, and crucial. car's front-facing camera is used by image-based traffic- sign recognition algorithms to identify signals. They assist the motorist by issuing warnings. The main elements of a vision-based traffic sign recognition system are the identification and recognition modules. While the recognition module identifies the sign, the detecting module locates the sign area in the picture or video During the detection procedure, the sign areas with the highest probability are chosen and sent into the recognition system to classify the sign. Different machine learning techniques, including SVM, KNN, and Random Forest, can be utilised to recognise traffic signs [6]. The main drawback of these algorithms, however, is that feature extraction must be carried out independently; CNN, on the other hand, will carry out feature extraction on its own [1]. The suggested approach therefore uses a convolutional neural network. The input preprocessing module will get the vehicle camera's image ready for the recognition stage first. After recognition, the motorist will hear a text warning message.

II. LITERATURE SURVEY

Driver distraction is one of the leading causes of road accidents and traffic-related fatalities worldwide. It significantly reduces a driver's ability to perceive and react to road events, posing a serious threat to public safety. Distractions are typically classified into three categories: visual distraction (taking eyes off the road), manual distraction (taking hands off the steering wheel), and cognitive distraction (taking the mind off driving). Understanding and detecting these types of distractions has been a central focus in intelligent transportation systems and human-machine interaction research. As vehicles become smarter, the demand for real-time and accurate driver monitoring systems is increasing. Initial efforts in distraction detection focused on traditional methods using physical and physiological sensors. These included data from steering wheel angle sensors, braking patterns, lane-keeping behavior, heart rate monitors, electroencephalograms (EEG), and electrocardiograms (ECG). Although effective, these sensor-based systems were invasive and required complex setups, making them less practical for widespread consumer use. The integration of physiological data into vehicles also raised concerns regarding driver privacy and system cost.

With the rise of computer vision and image processing, researchers shifted towards camera-based approaches. These methods typically use dash-mounted or infrared cameras to monitor the driver's face, eyes, and hands. Visual features like gaze direction, blink rate, head pose, yawning, and eye closure duration became key indicators for detecting distraction and drowsiness. Early vision-based techniques employed classical methods like Haar cascades for face detection and Kalman filters for tracking facial landmarks.

These methods provided a foundation for real-time driver monitoring but often struggled in varying lighting conditions and occlusions caused by sunglasses, hats, or hand movements.

To overcome these limitations, machine learning algorithms were introduced. Models such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, and Random Forests were trained on features extracted from images or videos to classify driver behavior. The availability of benchmark datasets, such as the StateFarm Driver Distraction dataset, allowed researchers to train and evaluate these models on various classes of distraction, including texting, talking on the phone, drinking, adjusting controls, and more. These supervised models improved detection accuracy but required extensive manual feature engineering. The advent of deep learning, particularly Convolutional Neural Networks (CNNs), revolutionized driver distraction detection. CNNs eliminated the need for manual feature extraction by learning hierarchical features directly from raw image data. Pretrained networks like VGG16, ResNet50, MobileNet, and Inception were successfully used for transfer learning, enabling high-accuracy classification even on relatively small datasets. Deep learning models demonstrated superior performance in identifying distraction behaviors across different lighting conditions and driver profiles.

Lightweight CNN architectures, such as DRIVENET, were developed to ensure that deep learning could be deployed on edge devices in real-time.

Some researchers introduced attention mechanisms and facial region weighting in CNN models to focus on critical areas like eyes, mouth, and hands. This further improved model robustness and interpretability. Beyond single-modal image input, multimodal distraction detection systems emerged. These systems integrate video data with audio, GPS coordinates, steering patterns, acceleration data, and even in-cabin speech to provide a more holistic understanding of the driver's condition. Multimodal fusion improves the reliability of detection, especially in ambiguous scenarios.

Despite technical advancements, the field faces several persistent challenges. One major challenge is the variability in lighting conditions, such as night-time driving, sun glare, or changes in ambient light. Another issue is facial occlusion due to accessories like sunglasses, masks, or caps. These occlusions hinder the system's ability to detect gaze and facial expressions accurately. Moreover, datasets used in training often suffer from class imbalance, where some distraction types (e.g., talking on the phone) are overrepresented, while others (e.g., eating or reaching for objects) have limited samples.

Driver diversity is another concern. Many models perform well in controlled environments but struggle with generalization across different ages, skin tones, facial structures, and cultural behaviors. Real-world deployment requires systems that are robust to such variations.

Additionally, achieving low latency and high throughput on embedded systems (e.g., Raspberry Pi or Jetson Nano) without sacrificing accuracy is still a major technical hurdle. Privacy and ethical considerations also play a critical role.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

Using cameras inside the cabin raises concerns about surveillance, data misuse, and consent. Designers of these systems must ensure data is anonymized, securely stored, and used only for safety enhancement. Regulatory guidelines and driver acceptance will significantly influence the adoption of such technologies in commercial vehicles. Future research in driver distraction detection is moving toward optimizing models for edge computing. Techniques like quantization, pruning, and knowledge distillation are being applied to make CNNs more efficient for real-time use in cars. Researchers are also exploring unsupervised and semi-supervised learning to reduce dependency on large labeled datasets. Furthermore, 3D vision systems and stereo cameras are being investigated to capture depth information and improve spatial accuracy.

Integration with Advanced Driver Assistance Systems (ADAS) is becoming a key objective. Real-time distraction alerts can be used to trigger lane-keeping assist, adaptive cruise control adjustments, or emergency braking. Some systems are designed to notify the driver with audio or haptic feedback, while others may take partial control to prevent accidents. Companies like Tesla, Volvo, and Toyota are already incorporating distraction detection modules into their latest vehicle models.

Continued efforts are being made to build large, diverse, and annotated datasets that represent real-world driving scenarios. These include variations in race, age, environment, and driving habits. Augmentation techniques, synthetic data generation, and simulation environments (like CARLA or Unity-based simulators) are also being used to train and validate models under various edge cases.

In conclusion, driver distraction detection is a critical area in intelligent transportation and automotive safety. The evolution from traditional sensors to AI-powered computer vision has greatly enhanced the ability to monitor and mitigate distracted driving. While challenges remain in achieving real-time, generalizable, and privacy-aware solutions, ongoing advancements in deep learning, multimodal fusion, and embedded AI offer promising avenues for making roads safer through proactive driver monitoring systems.

Together, these studies build a strong foundation for the development of Agro Vision. Unlike existing agri-portals that rely on one-way information delivery, Agro Vision offers a bi- directional, open discussion model. It is designed to enable collaboration between students and farmers, provide regionally relevant content, and support scalable deployment through its backend architecture built in Spring Boot (Java). By leveraging RESTful APIs, structured categories, and moderated forums, the platform meets the evolving demands of Indian agriculture in a digital age.

III. METHODOLOGY

Real-time image-based driver distraction detection is a prominent topic in the disciplines of machine learning and computer vision, and several models and algorithms are proposed and researched by researchers.

Two of the main topics covered are the selection of a classification model and an image preprocessing method.

It was hypothesised that utilising feature extraction rather than image flattening directly could, but wasn't guaranteed to, boost prediction accuracy when it came to picture preprocessing.

One of the main methods for the classification model is convolutional neural network (CNN)-based models.

When compared to CNN-based models, SVM models don't obtain the maximum accuracy, but their learning curves are shorter and their processing costs are lower .

IJARSCT

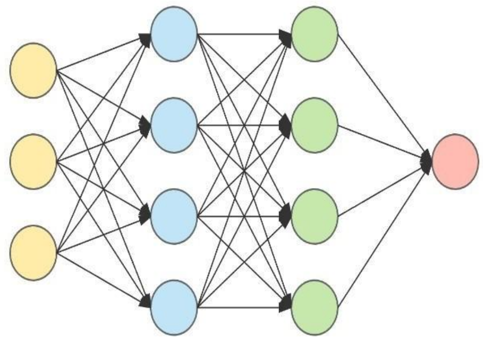
International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 13, April 2025

Impact Factor: 7.67



by Spring Boot's capability to rapidly produce scalable applications, its seamless integration with REST APIs, and built-in support for security, testing, and dependency management via Spring Initializer and Maven.

The methodology for driver distraction detection is structured into several key stages: data collection, preprocessing, model selection, training, evaluation, and deployment. Each stage is critical to ensure that the system can accurately detect different types of distractions in real- time environments Repository Layer: Uses Spring Data JPA to manage database operations. MySQL serves as the underlying relational database for user data, discussions, comments, tags, and interaction logs.

1. Data Collection

We used publicly available datasets such as the StateFarm Driver Distraction Dataset, which contains thousands of images of drivers performing various activities like texting, talking on the phone, drinking, reaching behind, grooming, and driving safely. Additional data was collected through in-vehicle camera recordings to capture real-world scenarios, ensuring diversity in lighting, backgrounds, driver demographics, and distraction behaviors.

2. Data Preprocessing

Captured images were standardized to a consistent size (e.g., 224x224 pixels) suitable for input to Convolutional Neural Networks (CNNs). Data augmentation techniques such as rotation, flipping, brightness adjustment, and scaling were applied to increase dataset variability and address class imbalance. Face detection algorithms (e.g., Haar cascades or Dlib facial landmark detection) were optionally used to crop and focus on the driver's face and hands, enhancing relevant feature extraction

3. Model Selection

Deep learning models, particularly CNNs, were selected for their high performance in image classification tasks. Transfer learning was employed by fine-tuning pretrained models

such as VGG16, ResNet50, and MobileNetV2 to leverage their ability to extract hierarchical features effectively.

Lightweight models were prioritized to enable potential deployment on resource-constrained edge devices.

4. Model Training

The selected models were trained using the processed dataset. The training process involved:

• Using categorical cross-entropy as the loss function. Applying the Adam optimizer with an appropriate learning rate (e.g., 0.0001).

• Splitting the dataset into training, validation, and test sets (e.g., 70%-15%-15%).

• Implementing early stopping and learning rate scheduling to avoid overfitting.

• Regularization techniques such as dropout and L2 regularization were employed to further enhance generalization.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

5. Evaluation Metrics

The performance of the models was evaluated using metrics such as:

• Accuracy: Correct predictions out of total samples.

• Precision, Recall, F1-Score: To assess class-wise performance.

• Confusion Matrix: To visualize misclassification among distraction categories.

• Inference Time: To ensure real-time applicability. Cross-validation was conducted to verify the robustness of the models across different data splits.

6. Real-Time Testing

The best-performing model was integrated with a simple camera setup to perform real-time distraction detection. Frame capture, preprocessing, model inference, and result visualization were implemented using OpenCV and TensorFlow/PyTorch frameworks. The system was tested under varying lighting conditions and different driver behaviors to simulate real-world driving.

7. Deployment (Optional)

For embedded applications, the model was optimized using techniques like model quantization and pruning. It was then deployed on lightweight platforms such as Raspberry Pi 4 or NVIDIA Jetson Nano to verify real-time performance outside the development environment.

This methodology ensures a systematic approach to building a reliable, accurate, and real-time driver distraction detection system using deep learning and computer vision techniques.

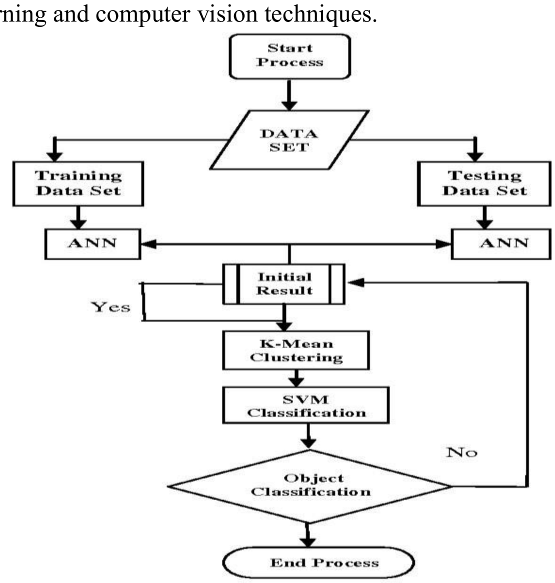


Fig.01 Architecture

1. Data Acquisition

• Collect images or video data of drivers performing various activities.

• Use public datasets like StateFarm or record custom in-cabin video footage.

2. Dataset Selection / Creation

Use public datasets like: StateFarm Driver Distraction Dataset AUC Distracted Driver Dataset

OR Create a custom dataset: Install in-car cameras to record behavior. Get ethical clearance/consent from participants.

Record videos in varied lighting, vehicle types, and weather conditions.

This architecture is designed to ensure that each component operates independently, making it easier to scale and maintain. For instance, as the platform grows and more users access it, the backend can scale horizontally by adding additional service instances.

3. Data Annotation
• Manually label frames or use semi-automated tools (e.g., CVAT, LabelImg).
• Annotate actions per frame for supervised learning.
• Use multiple annotators and resolve conflicts (inter- rater agreement).

4. Preprocessing
• Resize/crop images.
• Normalize pixel values.
• Apply noise reduction (optional).
• Apply data augmentation: flip, rotate, shift, blur, zoom, brightness changes.
• Handle class imbalance via oversampling or class weighting.

5. Feature Extraction (if not using end-to-end DL)
• Extract keypoints (face, hands, eyes) using:
• OpenPose, MediaPipe, Dlib
• Calculate head pose, gaze direction, eye aspect ratio (for blinking/yawning).
• Feed features into traditional ML models (SVM, KNN, etc.) if applicable.

6. Model Selection and Design
• Choose between:
o Classical ML: SVM, Random Forest
o Deep Learning: CNN, RNN, hybrid models
o Transfer Learning: MobileNetV2, EfficientNet, ResNet50
• Design architecture (if custom CNN is used).

7. Model Training
• Split dataset: Train (70%), Validation (15%), Test (15%)
• Use data generators for batch training.
• Loss: Categorical Cross-Entropy
• Optimizer: Adam / SGD
• Use callbacks: EarlyStopping, ReduceLROnPlateau
• Train with GPU acceleration if possible.

8. Model Validation
• Plot loss and accuracy curves.
• Tune hyperparameters:
o Learning rate
o Batch size
o Number of epochs
• Use k-fold cross-validation to verify stability.
• Evaluate with:
o Accuracy
o Confusion Matrix
o Precision, Recall, F1 Score

9. Real-Time Integration Pipeline
• Capture video stream using OpenCV or webcam module.
• Process video into frames.
• Run frames through model.
• Apply softmax + prediction thresholding.
• Display result (label, probability, action alert).

10. Distraction Alert System
• Trigger:
o Visual alert (GUI pop-up)
o Audio alert (beep, voice message)
o Haptic alert (vibration, seat nudge)
• Log events with timestamp and image snippet.

11. Privacy & Ethical Compliance
• Mask or blur facial identities in saved frames.
• Encrypt video data.
• Obtain user consent.
• Comply with GDPR/data privacy laws.

V. FUTURE SCOPE

One more module that can recognize traffic signs and issue instructions via voice messages will be developed in the upcoming development.
In this session, we will use machine learning techniques to identify the images that we display the system in order for it to produce a voice message for us.
For image recognition in this system, we are utilizing CNN, Numpy, the OS Library, and Open- Cv.

VI. CONCLUSION

In this research, we reviewed the literature on traffic sign identification using machine learning approaches and conducted a comparative analysis of these methods.
With the help of hyper parameter adjustment, CNN works well for recognition, and accuracy or recognition rate can be increased.
As a result, we used CNN for traffic sign identification in the suggested scheme to create a warning traffic sign detection system for drivers.
During the image acquisition step, the pictures will be taken with a camera mounted on the automobile, and after preprocessing, the CNN algorithm will be used to identify the pictures.
When a traffic sign is detected, the machine sounds a warning. This concept can be applied in situations when accurate navigation is necessary.

REFERENCES

[1]. Shreyas Satardekar Int. Journal of Engineering Research and Application www.ijera.com
[2]. ISSN : 2248-9622, Vol. 8, Issue4 (Part - III) April 2018, pp51-55
[3]. T. H. N. Le, Y. Zheng, C. Zhu, K. Luu and M. Savvides, "Multiple Scale Faster-RCNN Approach to Driver‟s Cell-Phone
[4]. Usage and Hands on Steering Wheel Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),Las Vegas, NV, 2016, pp. 4653
[5]. Yehya Abouelnaga, Hesham M. Eraqi, and Mohamed N. Moustafa, “Realtime DistractedDriver Posture Classification”, arXiv preprint arXiv:1706.09498

[6]. Kaggle. (2016). State Farm Distracted Driver Detection.
[7]. Abulkhair, M. F., & Alsulaiman, M. (2021). IEEE Access, 9, 50354–50375. DOI: 10.1109/ACCESS.2021.3068717.