Human Activity Recognition

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Tanish Mahajan (201206), Hridhima Sen (201211),

Vaibhav Sharma (201259)

Under the guidance & supervision of

Dr. Diksha Hooda and Prof. Dr. Jata Shankar

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

Certificate

This is to certify that the work which is being presented in the project report titled "**Human Activity Recognition**" in partial fulfillment of the requirements for the award of the degree of B.Tech in Information Technology and submitted to the Department of Computer Science & Engineering And Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "**Tanish Mahajan (201206), Hridhima Sen (201211), and Vaibhav Sharma (201259)**" during the period from August 2023 to December 2023 under the supervision of **Dr. Diksha Hooda** (Assistant Professor, Department of Computer Science & Engineering and Information Technology) and **Prof. Dr. Jata Shankar** (Professor, Department of Biotechnology and Bioinformatics).

(Student Signature with Date)
Student Name: Tanish Mahajan
Roll No.: 201206

(Student Signature with Date)
Student Name: Hridhima Sen
Roll No.: 201211

(Student Signature with Date)
Student Name: Vaibhav Sharma
Roll No.: 201259

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr. Diksha Hooda

Designation: Assistant Professor

Department: Computer Science and Engineering

Dated: 16-05-2024

(Co-Supervisor Signature with Date)

Supervisor Name: Prof. Dr. Jata Shankar

Designation: Professor

Department: Biotechnology and Bioinformatics

Dated: 20-05-2024

Candidate's Declaration

We hereby declare that the work presented in this report entitled **'Human Activity Recognition'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Diksha Hooda** (Designation, Department of Computer Science & Engineering and Information Technology) and **Prof. Dr. Jata Shankar** (Professor, Department of Biotechnology and Bioinformatics).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Tanish Mahajan
Roll No.: 201206

(Student Signature with Date)
Student Name: Hridhima Sen
Roll No.: 201211

(Student Signature with Date)
Student Name: Vaibhav Sharma
Roll No.: 201259

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr. Diksha Hooda

Designation: Assistant Professor

Department: Computer Science and Engineering

Dated: 16-05-2024

(Co-Supervisor Signature with Date)

Supervisor Name: Prof. Dr. Jata Shankar

Designation: Professor

Department: Biotechnology and Bioinformatics

Dated: 20-05-2024

Acknowledgement

We want to express our genuine gratitude to our respected project mentors, Dr. Diksha Hooda and Prof. Dr. Jata Shankar, for their invaluable guidance, unwavering support, and continuous supervision throughout this project. Their expertise and encouragement were indispensable, and without them, we would have faced considerable challenges.

The mentorship provided by our supervisors has been crucial in steering us in the right direction, offering not just essential guidance but also insightful information crucial to the success of the project. We appreciate their consistent motivation, which played a pivotal role in guiding us toward the successful completion of this endeavor.

Additionally, we extend heartfelt thanks to our parents and the JUIT community for their generous cooperation and encouragement. Their support has been a driving force behind our accomplishments.

Recognition is also due to the invaluable contributions of our colleagues, whose skills and collaboration were integral to the development of this project. Their combined efforts elevated the overall quality and success of our work.

In conclusion, the completion of this project has been a collective effort, and we sincerely appreciate everyone who played a role, whether directly or indirectly, in bringing it to fruition.

Tanish 16-05-24

Tanish Mahajan (201206)

Hridhima 16-05-2024

Hridhima Sen (201211)

Vaibhav Sharma (201259)

Table of Content

List of Tables

List of Figures

List of Abbreviations, Symbols or Nomenclature

S. No.	**Abbreviation**	**Expansion**
1	Open CV	Open Source Computer Vision
2	AI	Artificial Intelligence
3	FPS	Frames Per Second
4	RGB	Red, Green, Blue
5	API’s	Application Programming Interfaces
6	EAR	Eye Aspect Ratio
7	MAR	Mouth Aspect Ratio
8	CPU	Central Processing Unit
9	UI	User Interface
10	OS	Operating System

Abstract

The Human Activity Recognition project is an innovative initiative aimed at improving road transportation safety through real-time analysis of actual images using the OpenCV library. The primary goal is to identify risky driver behaviors like eye closure, yawning, and abnormal head movements, which can indicate distracted or drowsy driving.

This project is mainly focused on cab drivers associated with ride-sharing services like Ola and Uber. By detecting risky behaviors, the system is designed to trigger automatic alarms beforehand, paving the way for immediate control mechanisms. Additionally, in case of any abnormality detected, an automated text message will be sent to the registered mobile number of the vehicle's owner or the driver's manager.

This proactive approach addresses the overall problem of distracted driving, recognized as a major cause of crashes. The project presents a novel solution that combines computer vision technologies to enable swift responses to crucial events and enhance the accuracy of risk prediction.

A key distinguishing feature of this project is its unwavering commitment to security. Whenever a warning is triggered, an alarm is sent without fail, ensuring that no potential risk goes unaddressed. This strategy directly tackles the reasons behind driver-associated crashes, improving road safety by examining and managing the root causes.

Ultimately, the "Human Activity Recognition" project aims to modernize road safety practices by adopting contemporary IT solutions that promote secure driving conditions. It showcases the power of technology in saving lives on the move, demonstrating an innovative approach to a critical issue through cutting-edge computer vision techniques.

Chapter 1: Introduction

1.1 Introduction

The "Driver Activity Recognition" project endeavors to develop a sophisticated system capable of real-time monitoring and analysis of driver behavior. Its primary objective is to augment road safety by swiftly identifying potential distractions or hazardous activities that could impair a driver's ability to effectively operate a vehicle. Through the utilization of advanced computer vision and Python-based image processing techniques, this system will deliver prompt alerts and warnings, empowering drivers to promptly address their actions and diminish the likelihood of accidents.

Operationally, the system continuously captures video frames from an in-vehicle camera, predominantly focused on the driver's perspective. These frames undergo initial preprocessing by conversion to grayscale, a step designed to accentuate relevant features while concurrently reducing computational complexity. This conversion process involves the extraction of the 59 luminance component from the color information, thereby eliminating hue and saturation components and retaining solely brightness values. Consequently, this technique enhances object-background contrast, facilitating the identification and extraction of pertinent features for subsequent analysis.

Furthermore, the grayscale frames undergo supplementary preprocessing stages to bolster the quality and dependability of subsequent analyses. These stages encompass noise reduction, contrast enhancement, and normalization techniques. Noise reduction serves to eliminate unwanted distortions or artifacts within the image, whereas contrast enhancement algorithms enhance the visibility and distinctiveness of relevant features. Concurrently, normalization techniques ensure uniform illumination and brightness levels across frames, thus mitigating the impact of fluctuating lighting conditions on the detection process.

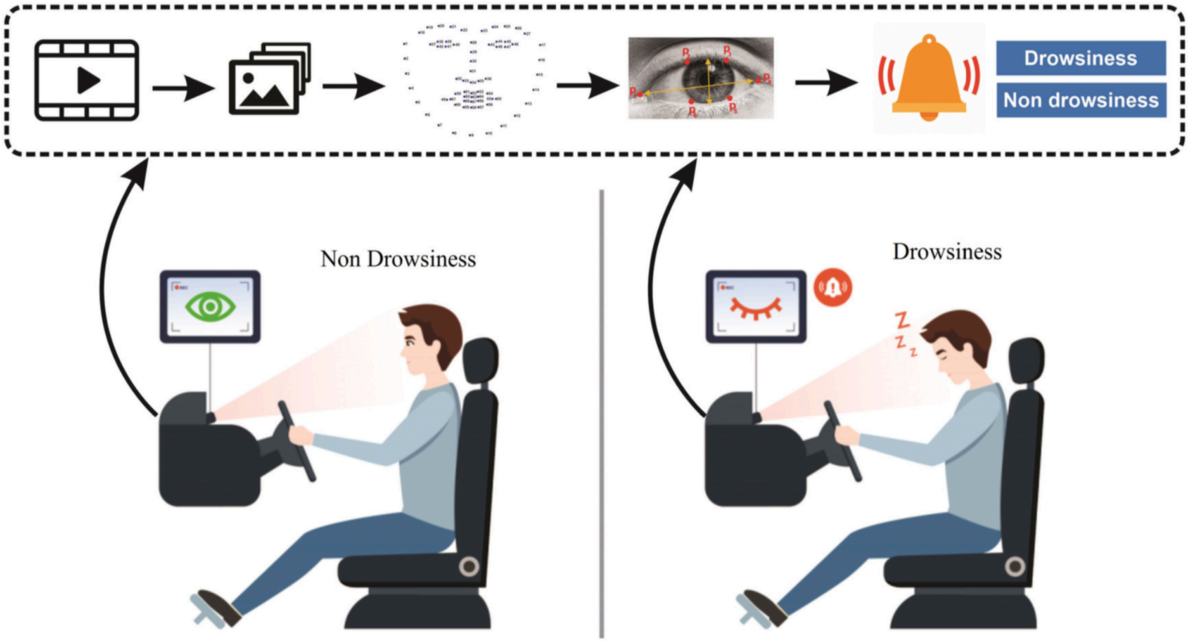

Fig. 1.1 Working of the Detection System [30]

The grayscale frames undergo a sophisticated phase of image processing algorithms and computer vision techniques to discern and categorize diverse driver activities, encompassing tasks such as texting, eating, drinking, or engaging in other potentially distracting behaviors. These algorithms commonly employ methodologies including edge detection, contour analysis, shape recognition, and pattern matching. By scrutinizing both the spatial and temporal attributes of the driver's movements and gestures, the system adeptly distinguishes and classifies various activities.

Once a specific activity is identified, the system promptly generates an appropriate alert or warning, disseminating it via visual or auditory cues. These cues serve to promptly notify the driver, prompting them to redirect their focus towards the road. This proactive approach to alerting not only heightens driver awareness but also facilitates timely intervention, thereby mitigating potential safety hazards and promoting responsible driving practices.

1.1.1 Key Components and Aspects

1. **Classification of Drivers:** Employing computer vision, machine learning algorithms, and signal processing to interpret and categorize diverse driving behaviors, thereby enhancing safety measures and vehicle maintenance.

2. **Exhaustion and Drowsiness Detection:** Utilizing facial expressions, yawning, eye movements, blinking patterns, and head tilts to detect signs of exhaustion or drowsiness, providing crucial insights into driver discomfort or reduced alertness.

3. **Feature Extraction:** Extracting pertinent features such as eye movements and facial landmarks from camera feeds or external webcams to furnish essential inputs for activity recognition systems.

4. **Real-time Monitoring and Alerts:** Minimizing potential risks by delivering real-time feedback or alerts to the driver or monitoring systems, such as warning signals in case of fatigue or distraction indicators.

1.1.2 Applications

1. **Driver Safety:** This system functions as a diligent companion, swiftly identifying and notifying drivers of hazardous behaviors or signs of fatigue. Its pivotal role lies in promoting safer driving practices, significantly contributing to the overall safety of road users.

2. **Vehicle Management:** Within the domain of commercial vehicle operations, this system assumes a managerial function, offering real-time insights into driver behavior.

Its capabilities empower vehicle managers to enhance productivity, mitigate accident rates, and optimize route planning, thereby maximizing operational efficiency.

3. **Insurance and Risk Assessment:** Through an analysis of driving risk factors, this system facilitates the determination of insurance premiums by assessing driver behavior. Its precise risk assessment supports the provision of tailored insurance policies, fostering responsible driving practices and reducing potential liabilities.

4. **Research and Development:** This system serves as a valuable asset for conducting comprehensive studies aimed at gaining deeper insights into driving habits, refining safety features, and optimizing vehicle architecture. Its provision of rich data and analysis contributes to ongoing advancements in automotive technology, ultimately enhancing road safety and driving experiences.

1.2 Problem Statement

Road accidents represent a critical menace to public safety, with a distressing number of fatalities occurring annually, particularly prevalent in nations like India. Driver fatigue and errors stand out as significant contributors, accounting for a staggering 78% of road accidents [1]. Moreover, distractions such as smoking or drinking exacerbate the risk of accidents by inducing a lapse in consciousness or compromising attention [2]. The severity of this issue is compounded by inadequate access to prompt medical assistance, amplifying the repercussions of these avoidable mishaps. As per research outlined in [1], India records approximately 1.5 lakh fatalities on its roads each year, translating to an average of 1130 accidents and 422 deaths daily, or a chilling 47 accidents and 18 deaths hourly.

Existing driver drowsiness detection systems grapple with various constraints, including intricate computations, expensive apparatus, and the imposition of uncomfortable devices on drivers. These limitations impede widespread adoption and introduce potential distractions and discomfort, rendering them unsuitable for real-world driving conditions where adaptability and user-friendliness are paramount.

To tackle these hurdles, an innovative solution is urgently required. Such a solution must effectively identify and mitigate incidents stemming from driver fatigue, as well as distractions induced by activities like smoking or drinking. It should be cost-effective, practical, and customized to the distinctive driving scenarios prevalent in India and analogous environments globally. By proactively addressing driver fatigue and distraction, this solution aspires to enhance road safety, diminish the likelihood of accidents, and ultimately safeguard lives in motion. The overarching objective is to develop a pragmatic, streamlined, and accessible solution conducive to widespread adoption, particularly in resource-constrained settings, in combating the disconcerting statistics of road accidents attributed to driver errors, fatigue, and distraction cited in [1] and [2].

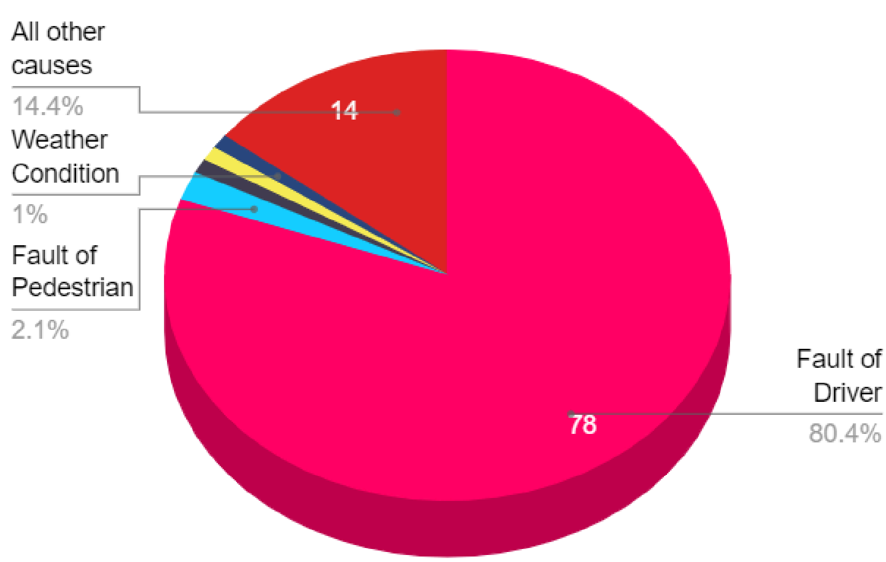

Fig. 1.2 Causes of Road Accidents in India [31]

1.3 Objectives

Driver activity recognition is mainly focused on improving road safety by using the technology to observe and analyze a driver's behavior which, in turn, reduces the dangers that come with distracted, tired or unsafe driving. The key targets include:

1. **Collecting and Preprocessing of Data:**

- Capture and preprocess video data from in-vehicle cameras or other video sources.
- Apply techniques such as grayscale conversion, noise reduction, and contrast enhancement to enhance the quality and reliability of the input data.
- Identify and extract relevant facial landmarks, including eyes, mouth, nose, and head position, for subsequent analysis.

2. **Real-time Monitoring:**

- Continuously monitor and observe the driver's behavior and activities in real-time.
- Analyze facial expressions, head movements, and other cues to detect signs of drowsiness, distraction, or unsafe driving practices.
- Provide immediate alerts or warnings to the driver and relevant stakeholders (e.g., fleet managers, emergency contacts) when potential risks are identified.

3. **Testing and Validation of the Proposed System:**

- Develop a comprehensive testing strategy to validate the system's performance, accuracy, and reliability.

- Conduct various testing scenarios, including manual testing with simulated inputs, threshold validation, module and function testing, integration testing, boundary testing, and alarm testing.
- Assess the system's robustness under different environmental conditions, such as varying lighting conditions, facial orientations, and camera distances.
- Evaluate the system's usability and user-friendliness through user testing with representative drivers and administrators.
- Continuously refine and improve the system based on testing results and feedback.

4. **Data Analysis and Insights:**

- Make the data on driving behaviors available, and then, process and analyze the data to determine the trends, patterns, and the areas which need to be improved.

1.4 Significance and Motivation of the Project Work

1.4. 1 Significance

1. **Improving Road Safety:** This project addresses critical safety issues by monitoring driver sleepiness, yawning, and distraction. The primary aim is to reduce the risk of accidents caused by distracted or fatigued drivers by developing intelligent systems that can recognize these behaviors in real-time [3].

2. **Technological Innovation:** Utilizing OpenCV for real-time video analysis to monitor driver activities signifies an innovative use of computer vision technology. The project explores advanced algorithms and AI techniques to create a sophisticated system that contributes to road safety measures [4].

3. **Human-Centric Design:** The project's emphasis on human-centric design aligns with the increasing need for technology to prioritize user safety. By integrating warning systems into the driver interface in a non-intrusive manner, it aims to maintain focus without disrupting driving attention [5].

1.4.1 Motivation

1. **High Incident Rates:** The motivation behind the project arises from concerning statistics indicating a substantial number of road accidents are attributed to driver fatigue or distraction [6]. The aim is to address this issue by utilizing advanced technology to identify and mitigate potential risks associated with these factors.

2. **Potential for Impact:** The project is driven by its ability to greatly affect public safety and health. Through the prevention of the accidents that are caused by the inattentiveness of the drivers, it can save lives, cut down the injuries and the damage of the property, thus getting the roads safer for everybody [7].

3. **Advancements in Computer Vision:** One of the main reasons for the project is the study of computer vision technologies, especially OpenCV. Its goal is to create strong systems that are able to identify complicated human behaviors and expressions in real-time [4].

4. **Academic and Industrial Relevance:** The project holds relevance not only within academic circles but also in industry. Academically, it contributes to the advancement of knowledge and development of AI-based driver monitoring systems. Industrially, there is potential for applications in automotive safety technologies, bridging the gap between research findings and practical implementation [8].

1.5 Organization of Project Report

This detailed report is structured into six key chapters, each offering valuable insights into the Human Activity Recognition project.

Chapter 1: Introduction

This first chapter is the project's starting point, it introduces the problem, sets the goals, and gives the reasons for the project. It is the basis of what is to come after.

Chapter 2: Literature Survey

In this chapter, we delve into existing knowledge, exploring reputable sources such as books and technical papers from the last five years. The aim is to grasp the current landscape and identify gaps for our project to address.

Chapter 3: System Development

The heart of the project unfolds in this chapter, covering the journey from understanding requirements to designing and implementing the system. We discuss challenges faced during development and the strategic solutions applied.

Chapter 4: Testing

This section highlights the thorough testing phase, detailing the approach and instruments employed. Here, we outline our test cases and results, giving a clear overview of the system's reliability.

Chapter 5: Results and Evaluation

Focusing on tangible outcomes, this chapter interprets findings and compares them with existing solutions if applicable. It provides an in-depth evaluation of our results.

Chapter 6: Conclusions and Future Scope

In summary, this chapter presents the main results, acknowledging the limitations, and suggests possible future directions for research and development.

Chapter 2: Literature Survey

2.1 Overview of Relevant Literature

The literature on driver cognition offers many methods and tools to understand and predict driver behavior. Researchers are exploring a variety of techniques, including computer vision, machine learning, and sensor systems, to identify and analyze activities such as impaired driving, fatigue, and agitated behavior. Much of the research focuses on analyzing data from onboard sensors, cameras, and remote vehicle data to create algorithms that can identify patterns that indicate driving is different. Additionally, adaptive learning and deep learning are attractive because they can extract complex features from raw data and images, thereby increasing the accuracy of the research and the power of its data. Overall, the data shows interest in developing intelligent systems to improve safety and driver assistance.

2.2 Key Gaps in the Literature

The case study highlights some of the limitations of existing DAR data, identifying areas for further research and development:

1. **Adapting to real-world variables:** Current DAR systems rely on past learning, which limits their ability to adapt to real-world variables such as changing driving and environment.[1]

2. **Lighting issues:** DAR systems often have problems managing performance in different lighting conditions, affecting the accuracy and confidence of operating information.[2] ,[3]

3. **Special Equipment Requirements:** Many DAR systems require special equipment; this creates challenges for large-scale deployment in the industry and may prevent large-scale adoption.[4]

4. **Competitive Training of 3D ResNet:** Training of 3D ResNet from scratch can be affected by stochastically driven data and configuration data, resulting in overfitting and It may cause problems such as decreased model performance.[5]

5. **Reliance on Eye Tracking for Sleep Detection:** Some DAR systems rely on tracking the driver's eyes to detect sleep; This may not be enough to prevent the situation in important cases.[6]

6. **Dataset sparsity:** Dataset sparseness poses a problem for DAR research and needs to be improved, especially when training 3D ResNet from scratch. Standardized data collection to reduce the risk of overwork. [7]

7. **Sensor Accuracy and Placement:** Sensor sensitivity (especially in alcohol testing) will vary depending on location and distance from the driver. More research is needed to improve accuracy and reliability, especially in different driving conditions.[8]

8. **Limited Training Data:** Limited training data may affect the overall performance of the DAR model, especially in real situations with different lighting conditions and camera angles. Limitations require collaborative research involving advances in sensor technology, algorithm development, and data collection techniques. Future research should focus on improving the flexibility, stability, and efficiency of DAR systems to improve safety and vehicle performance.[9]

S. NO.	PAPER TITLE	JOURNAL/ CONFERENCE	TOOLS/ TE CHNIQUES	RESULTS	LIMITATIONS
1	Driver Drowsiness Detection using OpenCV and YOLOV5 [9]	2023	YOLOv5, a real-time object detection algorithm, deployed for fatigue detection via eye closure and yawn detection, trained on Roboflow dataset across four groups.	• Average Object Detection Accuracy: 50.9% • Common misclassifications due to higher IoU threshold.	• Initial 200 images may not capture real-life scenes. • Variable lighting and camera angles may affect model's effectiveness.
2	Human Activity Recognition using OpenCv & Python [1]	2022	OpenCV2 (CvHMM version) for activity recognition and Hidden Markov Model (HMM) for optical flow estimation were implemented.	• Recognizes walking, running, sitting, standing.	• Adapting to Real-World Variables.
3	Real Time Driver Behaviour Monitoring system in vehicle using Image Processing [3]	2022	Python and OpenCV are used to make graphics and computer projects.	• Effective in detecting driver fatigue and distraction.	• System accuracy varies with lighting conditions.
4	Federated learning based Driver Activity Recognition	2021	The federal government proposes a learning method for analyzing driving	• FedGKT outperforms FedAvg, offering enhanced	• Research drivers rely on specialized hardware.

	for Edge Devices[4]		effects, overcoming computational and communication constraints without compromising model accuracy.	privacy protection and communication efficiency.	• Hinders large-scale industry deployment.
5	Real Time Human Activity Recognition using 3D ResNet. [5]	2021	Prior learning from data dynamics Transfer learning using 3D ResNet.	• 3D ResNet shows high accuracy in influencing driver behavior recognition. .	• Training 3D ResNet from scratch is affected by random driver data
6	Driver Drowsiness Detection using opencv and keras[6]	2021	Python,Opencv,Keras,webcam, alarm system.	• Alarm is generated upon detection. • Constant eye monitoring.	• Webcam quality impacts performance.
7	Driver Activity Recognition through Deep Learning [7]	2021	3D kernel ResNet used for detecting impaired driving, leveraging spatiotemporal video features, impact detection, and model design with limited data on impact drivers.	• Highlights importance of high-quality datasets. • Potential for further enhancements with combined datasets.	• Dataset sparseness: Few samples. • Refine process to reduce overfitting risk.
8	A Real Time Drowsiness Detection [2]	2020	OpenCV, DLIB	This method was able to identify the driver yawning and eye blinking.	Was not showing good results under different lighting conditions.
9	Driver Alertness Detection system using OpenCV [8]	2019	The image processing process includes blurring, RGB to HSV conversion, HSV thresholding, and blob detection to	• Detects drowsy drivers with 94.6% accuracy. • Maintains 87.69% accuracy with	• Accuracy relies on sensor distance. • Proper sensor placement is

			detect face and head movement, ensuring safety and visual impairment.	different data. • Proves the system's effectiveness in varied scenarios.	crucial. • More research needed for better accuracy.

Table 2.1 Literature Review

Chapter 3: System Development

3.1 Requirements and Analysis

3.1.1 Functional Requirements

1. **Live Video Processing:**

- The system must be able to process live video in the driving environment in real-time.
- Videos should be optimized to minimize latency and provide fast feedback.

2. **Feature Extraction:**

- Develop algorithms to extract facial features including eyes, mouth, and head movements.
- The system must identify symptoms of fatigue, yawning, and head tilt by specifically analyzing the images.

3. **Warning Mechanism:**

- Design and integrate a warning mechanism that will alert the driver immediately when fatigue, yawning, or excessive head tilt is detected.
- Warnings should be diverse, clear, and unobtrusive, utilizing sounds or visuals to attract the driver's attention.

4. **Notification System:**

- Whenever any unusual driver activity or behavior is detected, the system should send a text message alert to a registered phone number, potentially informing a designated emergency contact or authority.

5. **Integration Possibility:**

- Ensure seamless integration with existing systems such as traffic and alarm notifications.
- Compatibility with various car brands and models, offering solutions for different vehicles.

3.1.2 Non-Functional Requirements

1. **Accuracy and Reliability:**

- Achieve a high level of accuracy in identifying driving activities, reducing false positives and false negatives.
- Ensure a reliable system with minimal errors or malfunctions in real-world driving conditions.

2. **Speed:**

- The system must process video data and analyze facial expressions with minimal latency to provide timely warnings.
- Aim for response times suitable for real-time applications, considering the critical nature of driver safety.

3. **Scalability:**

- Ensure that the system can scale to accommodate different vehicle models and hardware configurations.
- The solution should be adaptable to future advancements in camera technology and computing capabilities.

4. **Usability:**

- Develop an intuitive user interface for both drivers and system administrators.
- Provide clear and easily understandable alerts and instructions for drivers to respond appropriately to warnings.

5. **Robustness:**

- The system should work flawlessly in different lighting situations, including daylight, nighttime, and adverse weather conditions.
- Robustness is the key to keep the machine working properly even in the changing driving environments and the occlusions of the facial features.

3.2 Project Design and Architecture

3.2.1 Project Design

1. **Objective:**

- Create a real-time system for the detection of driver activity and drowsiness using the facial landmarks captured through a webcam or in-vehicle camera, which will improve road safety by providing timely alerts and notifications.

2. Components:

2. 1 Data Processing and Collection:

- The program employs OpenCV to record video frames from the webcam or the in-vehicle camera, thus providing efficient and low-latency video capture.
- Uses Dlib's cutting-edge shape predictor (shape_predictor_68_face_landmarks.dat) to precisely detect 68 facial landmarks, paying attention to lips, eyes, nose, and head position.
- The sensor determines the lip distance, eye aspect ratio, and head tilt angle with the precision of the detection of yawning, sleepiness, and unusual head movements, which are the common indicators of driver fatigue or distraction.

2.2 Alert System Based on Thresholds:

- The thresholds are defined empirically (EYE_AR_THRESH, YAWN_THRESH, HEAD_TILT_THRESH) for the distance between the lips, the eye aspect ratio, and the head tilt angle to alarm accurately and with a low probability of false positives.
- Frames tracking (EYE_AR_CONSEC_FRAMES, YAWN_CONSEC_FRAMES, HEAD_TILT_CONSEC_FRAMES) to decrease the number of false positives and to confirm the drowsiness, yawning, and unusual head movements over a number of consecutive frames, hence, making the detection reliable.
- Creates an audible or visual alarm and sends a text message to a registered mobile number when the unusual activity is detected continuously for a certain number of frames, thus giving a notification to the driver and the designated contacts immediately.

2.3 Real-time Monitoring:

- Displays the video stream with intuitive overlays, highlighting the contours of lips, eyes, and facial landmarks in real-time, allowing the driver to understand the system's detection capabilities.
- Provides clear and concise textual feedback about detected states (drowsiness, yawning, unusual head movements) using OpenCV, ensuring the driver is aware of potential hazards.

3. Execution Flow:

3.1 Initialization:

- Loads pre-trained models (shape predictor, cascade classifier) and necessary libraries (OpenCV, Dlib) for efficient and accurate facial landmark detection and video processing.
- Sets up variables, thresholds, and flags (alarm_status, alarm_status2, alarm_status3, saying) for alert control and tracking, ensuring proper state management and consistent behavior.
- Initializes the video capture object and the text messaging service (e.g., Twilio, SMS gateway), enabling seamless integration with external notification systems.

3.2 Loop for Video Processing:

- Captures real-time video frames using OpenCV's optimized VideoStream from the webcam or in-vehicle camera, ensuring minimal latency and efficient resource utilization.
- Utilizes cv2.CascadeClassifier (Haar cascade classifier) to accurately detect faces in grayscale frames, enabling precise facial landmark detection.

- Applies Dlib's shape predictor to identify 68 facial landmarks with high precision, extracting lip, eye, nose, and head position coordinates for subsequent analysis.

3.3 Compute and Analyze Features:

- Evaluates drowsiness, yawning, and unusual head movements by computing lip distance, eye aspect ratio, and head tilt angle based on the detected facial landmarks, leveraging robust algorithms for accurate detection.
- Creates lips, eye, and head contours, adding feature-related text to the video frames, providing visual feedback to the driver and supporting debugging and development processes.

3.4 Real-time Visualization and User Interaction:

- Displays annotated video frames with textual feedback and visual indicators using OpenCV's efficient cv2.imshow and cv2.putText functions, ensuring smooth and responsive visualization.
- Allows the user to end the application by pressing 'q', providing a user-friendly exit mechanism.

3.5 Alert Generation and Notification:

- If unusual activity (drowsiness, yawning, or unusual head movements) is detected continuously for a certain number of frames, it generates an audible or visual alarm to alert the driver, using attention-grabbing and intuitive cues.
- Sends a text message to a registered mobile number, potentially informing a designated emergency contact or authority about the detected activity, enabling prompt intervention and assistance.

3.6 Cleanup and Termination:

- Destroys OpenCV windows and interrupts the video stream when the user closes the application, ensuring proper resource management and avoiding memory leaks.
- Releases resources and terminates the text messaging service, ensuring a clean and efficient shutdown process.

3.2.2 Architecture

1. **Initialization:**

- The system commences by initializing essential modules, including Dlib and OpenCV, ensuring seamless integration and efficient utilization of these powerful libraries for computer vision and image processing tasks.
- It sets up the video stream from the webcam or in-vehicle camera and configures the video capture settings for the best performance and compatibility with different hardware configurations.
- It loads the pre-trained face cascade classifier (detector) and facial landmark detector (predictor) from Dlib, thus making the facial feature analysis and landmark detection more accurate and robust.

2. **Loop for Frame Processing:**

- Each video frame is captured from the stream (frame = vs.read()) and converted to grayscale using cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY), reducing computational complexity and enhancing contrast for feature detection.

- The face cascade classifier is employed (detector.detectMultiScale()) to identify and locate faces within the grayscale frame, leveraging efficient algorithms for accurate face detection.
- For each detected face, the facial landmark detector (predictor) is applied to analyze and extract precise coordinates of 68 facial landmarks, including the lips, eyes, nose, and head position.
- Advanced algorithms are utilized to compute the lip distance, eye aspect ratio, and head tilt angle based on the detected facial landmarks, enabling the identification of yawning, drowsiness, and unusual head movements.

3. Display and Visualization:

- OpenCV functions like cv2.drawContours() and cv2.putText() are employed to overlay detected facial features and annotations on the video frame, providing intuitive visual feedback to the driver.
- Contours are drawn around the eyes, lips, and other relevant facial features, clearly highlighting the identified elements for ease of interpretation.
- Annotated video frames, including real-time feedback on lip distance, eye aspect ratio, head tilt angle, and corresponding alerts, are displayed using cv2.imshow(), ensuring a seamless and responsive user experience.

4. Alert Generation and Notification:

- Based on the computed lip distance, eye aspect ratio, and head tilt angle, the system evaluates the driver's state against predefined thresholds to detect potential drowsiness, yawning, or unusual head movements.

- If these activities are detected continuously for a certain number of frames, the system generates audible or visual alerts to notify the driver, utilizing attention-grabbing cues to prompt immediate action.
- Additionally, the system integrates with a text messaging service (e.g., Twilio, SMS gateway) to send real-time notifications to a registered mobile number, informing designated emergency contacts or authorities about the detected activities, enabling prompt intervention and assistance.

5. **User Interaction and Termination:**

- The system allows user interaction for termination by monitoring keyboard input, specifically the 'q' key (if key == ord("q")), providing a user-friendly exit mechanism for both the driver and the system administrator.
- Upon detecting the termination key, OpenCV windows are closed gracefully, and the video stream is terminated, ensuring proper resource management and preventing memory leaks or system instability.
- This termination process is designed to be seamless and efficient, ensuring a smooth user experience and minimizing disruptions to the driver's focus on the road.

3.3 Data Preparation

1. **Real-time Face Landmark Recognition:**

- The process initiates by leveraging OpenCV's functionality, specifically cv2.VideoCapture(), to commence and capture real-time video frames either from the webcam or a designated video file.
- This real-time video stream serves as the input source for subsequent facial feature analysis.

2. **Finding Facial Landmarks:**

- Facial features, such as eyes and lips, are identified through the utilization of either Dlib's pre-trained facial landmark detection model or OpenCV's Haar cascade classifiers (cv2.CascadeClassifier).
- Dlib provides a sophisticated approach to facial landmark recognition, offering detailed points for subsequent analysis.
- Additionally, OpenCV's Haar cascade classifier provides a viable option for face detection, specifically designed for real-time applications.

3. **Object Detection:**

- Besides facial landmark recognition, the system also carries out object detection to find possible distractions or dangerous items such as mobile phones, cigarettes or any other objects that may jeopardize the driver's safety.
- The detection of objects is accomplished through the application of pre-trained deep learning models, which are able to correctly detect and classify several objects in the video frames.
- These models are trained on big datasets of images which allows them to identify many objects with the highest accuracy.

4. **Data Preprocessing:**

- After the video frames, facial landmarks, and objects of interest have been detected, the system performs the data preprocessing steps to improve the quality and reliability of the input data.

- This can be achieved by techniques like image normalization, noise reduction, and contrast enhancement, which will guarantee the uniformity of the illumination and the clarity in different driving environments and lighting conditions.
- Besides, the data augmentation techniques, like the random rotations, flips, or scaling, can be used to make the training data more diverse and thus, the system will be more robust.

At this stage of data preparation, the groundwork for the future real-time face recognition is laid, thus, the system can capture video images, locate face regions, extract the necessary facial features and detect the potential distractions or hazardous objects. The amalgamation of Dlib, OpenCV, and deep learning object detection models is a complete way to data acquisition and preprocessing which allows the analysis of driver behavior and activities to be accurate and reliable.

3.4 Implementation

3.4.1 Code Snippets

- This code snippet imports the necessary libraries and modules required for the project.

```
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os
```

Fig 3.1 Importing Libraries

- This code defines the functions responsible for computing facial landmarks, such as eye aspect ratio and lip distance, which are crucial for detecting drowsiness and yawning.

```
def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear

def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
```

Fig 3.2 Defining Facial Landmark Functions

- This code sets up an argument parser, allowing the user to provide command-line arguments to configure the program's behavior.

```
ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
                help="index of webcam on system")
args = vars(ap.parse_args())
```

Fig 3.3 Setting up Argument Parser

- This code defines the constants and variables used throughout the program, such as thresholds for eye aspect ratio and lip distance.

```
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0
```

Fig 3.4 Defining Constants and Variables

- This code loads the pre-trained face detector and facial landmark predictor models from Dlib and OpenCV.

```
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
```

Fig 3.5 Load Face Detector and Facial Landmark Predictor

- This code initializes and starts the video stream from the webcam or a video file.

```
vs = VideoStream(src=args["webcam"]).start()
time.sleep(1.0)
```

Fig 3.6 Starting Video Stream

- This code represents the main loop of the program, where it detects faces in each frame, computes facial landmarks, and analyzes them for signs of drowsiness or yawning.

```
while True:

    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
        minNeighbors=5, minSize=(30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE)


    for (x, y, w, h) in rects:
        rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))
```

Fig 3.7 Main Loop - Detect and Analyze Faces

- This code handles the cleanup and shutdown process, including closing OpenCV windows and stopping the video stream when the program is terminated.

```
cv2.destroyAllWindows()
vs.stop()
```

Fig 3.8 Clean Up and Shutdown

- This code snippet adds bounding boxes around detected objects, such as persons, mobile phones, cigarettes, or wine glasses, for object detection purposes.

```python
# Extract co-ordinates of bounding box (with NMS)
indicies = cv2.dnn.NMSBoxes(bbox,confs,thres,nms_threshold)

# add boxes for each detection on each frame
for i in indicies:
    i = i[0] #Get the bounding box info
    box = bbox[i]
    x,y,w,h = box[0],box[1],box[2],box[3]
    cv2.rectangle(image,(x,y),(x+w,h+y),color = (0,255,0), thickness =2)
    cv2.putText(image,classNames[classIds[i][0]-1],(box[0]+10,box[1]+30),
                cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)

# Show output until CTRL+C
cv2.imshow("Output", image)
cv2.waitKey(1)
```

Fig 3.9 Adding Boxes For object Detection

- This code reads and processes images for object detection or other image-based analysis tasks.

```python
#Read in the image file
thres = 0.45 #Threshold to detect object
nms_threshold = 0.5 #NMS
cap = cv2.VideoCapture(0)

cap.set(3,1280)
cap.set(4,720)
cap.set(10,150)


#Import the class names
classNames = []
classFile = os.path.realpath(os.path.join(os.path.dirname(__file__), '..', 'config_files', 'coco.names'))


# Read object classes
with open(classFile, 'rt') as f:
    classNames = f.read().rstrip('\n').split('\n')

#Import the config and weights file
os.path.realpath(os.path.join(os.path.dirname(__file__), '..', 'config_files', 'coco.names'))
configPath =  os.path.realpath(os.path.join(os.path.dirname(__file__), '..', 'config_files', 'ssd_mobilenet_v3_large_coco_2020_01_14
weightsPath = os.path.realpath(os.path.join(os.path.dirname(__file__), '..', 'config_files', 'frozen_inference_graph.pb')) #Weights

#Set relevant parameters
net = cv2.dnn_DetectionModel(weightsPath,configPath)
```

Fig 3.10 Reading The Images

- This code is the main file responsible for executing the entire driver activity recognition system, including detecting and detecting drowsiness, written in Python.

```python
#Import relevant modules
import cv2

#Read in the image file
image = cv2.imread("config_files/lena.png")


#Import the class names
classNames = []
classFile = 'config_files/coco.names'

#### Extracts the classes into a list
#rt = open file for read
# rstrip('\n') removing white space
# split seperates each word at each line into a string
with open(classFile, 'rt') as f:
    classNames = f.read().rstrip('\n').split('\n')

#Import the config and weights file
configPath = 'config_files/ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weightsPath = 'config_files/frozen_inference_graph.pb'

#Set relevant parameters and initiate model
net = cv2.dnn_DetectionModel(weightsPath,configPath)
net.setInputSize(320,320)
net.setInputScale(1.0/127.5)
net.setInputMean((127.5,127.5,127.5))
net.setInputSwapRB(True)

#Extract, ClassIds, confidence and bounding box info
classIds, confs, bbox = net.detect(image, confThreshold = 0.5) #if 50% confident, predict
print(classIds, confs,bbox)
```

Fig 3.11 Image Main File

3.4.2 Algorithm

A. Detection of Facial Landmarks:

- Identify 68 facial expressions using facial expressions from the Dlib library. These important characters contain the main landmarks of the face and provide an overall map for later analysis. The image below shows the exact locations of 68 facial features that provide detailed information about facial features.

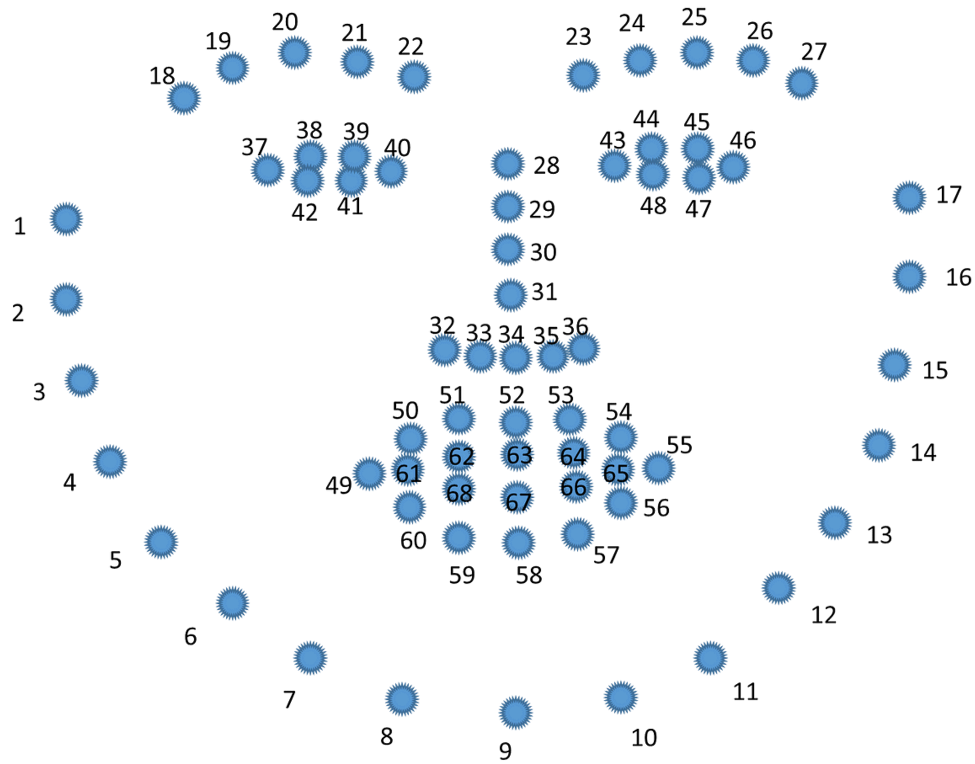

Fig 3.12 68 Facial Features Mapped on a Representative Face

Facial Elements	Landmarks	Number of Points
Facial Contour	0-14	15
Right Eyebrow	15-20	6
Left Eyebrow	21-26	6
Left Eye	27-30	4
Left Pupil	31	1
Right Eye	32-35	4
Right Pupil	36	1
Nose	37-45	9

Nose Center	67	1
Nostrils	46-47	2
Mouth Contour	48-59	12
Lower Lip	48, 60, 61, 62, 54	5
Upper Lip	54, 63, 64, 65, 48	5

Table 3.1 Facial Landmark Points

B. **Eye Aspect Ratio (EAR):**

- The system uses the position of the eye, specifically the distance between the field relative to the inside and outside of the eye, to calculate the Eye Response Ratio (EAR). < br>The ear monitoring system allows the system to detect signs of sleepiness by detecting changes in eye aperture, an important indicator of the driver's alertness.

C. **Mouth Aspect Ratio (MAR):**

- (MAR) is calculated as the distance between the upper and lower lips and the horizontal line between the ribs and the corners of the mouth. MAR identifies yawning and allows detection of those cases when a driver yawns.

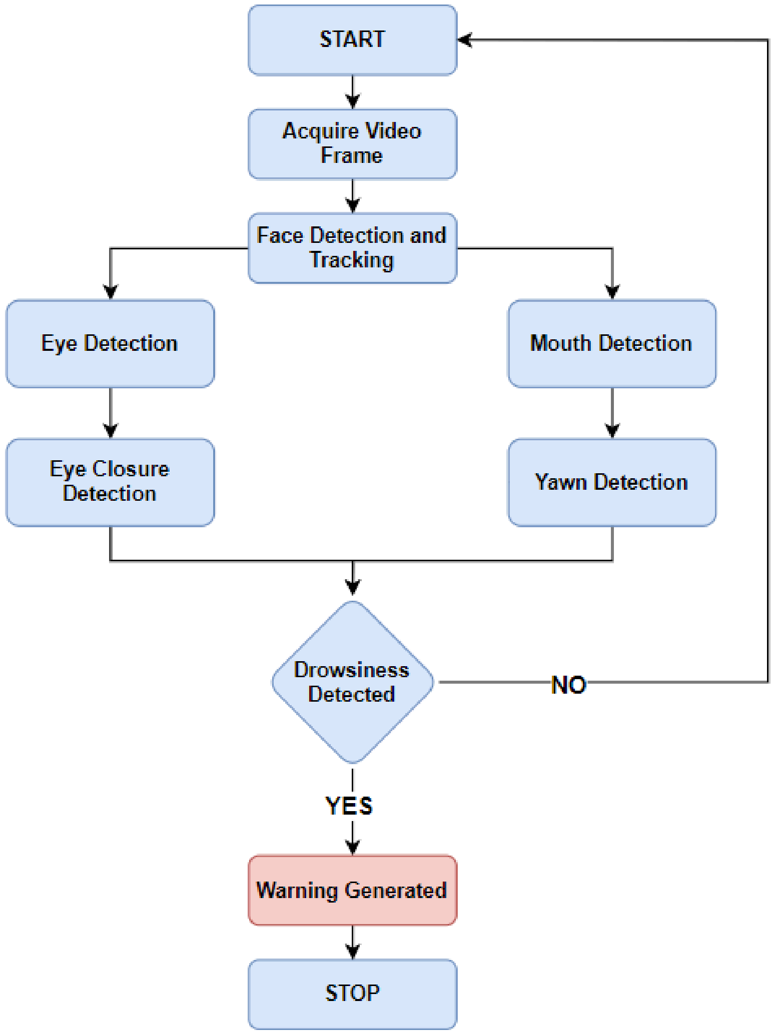

Fig 3.13 Flowchart of Eye and Yawn Detection for Drowsiness

D. Detection of Head Tilt:

- Facial features are analyzed to establish a connection in terms of positioning of eyes and nose so that the tilt of the head can be established. The angle is determined using the line which connects the two eyes and the nose, giving us an idea on how the driver's head is titled.

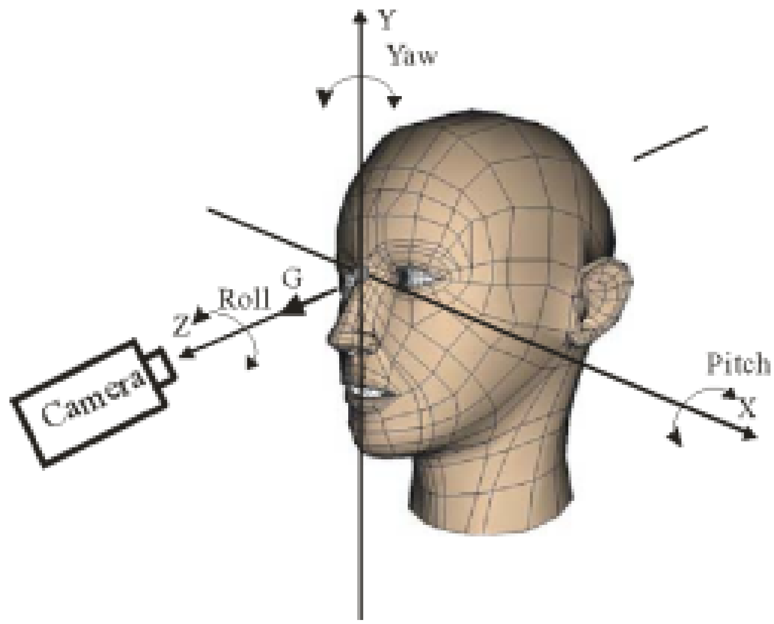

Fig 3.14 Head Tilt Detection based on Facial Recognition [32]

Eye recognition movement, head tilt detection as well as mouth dynamics make a crucial part of this identification system allowing to assess how effective the driver is at a time.

1. **Head Tilt Calculation:**

- This algorithm computes the gradient of the straight line drawn through the nose and into each eye.
- Head tilt is assumed with an occurrence of the tilt when it deviates from its initial state and hence signifying that a driver's head is too far away.

2. **Head Tilt Warning:**

- When the head tilt is detected a warning is displayed on the video frame addressed to the driver and system admin.
- This additional feature enhances the system's capability to identify potential signs of driver distraction or discomfort.

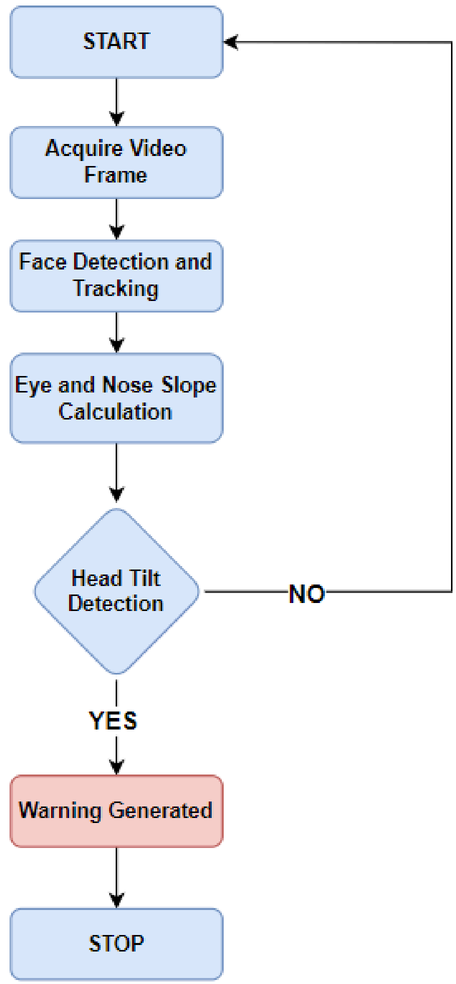

Fig 3.15 Flowchart Illustrating Head Tilt Detection Steps

The integration of head tilt detection into the overall analysis further refines the system's ability to ensure driver safety by capturing a broader range of behaviors that may impact driving performance.

3.4.3 Techniques Used

1. **OpenCV:**

- Utilizes OpenCV's VideoStream for efficient video stream handling, capturing frames from the webcam.
- Leverages image manipulation techniques, including contrast drawing, resizing, and grayscale conversion.
- Implements Haar Cascade Classifier for face detection, a fundamental component in identifying regions of interest.

2. **Dlib:**

- Incorporates Dlib for shape prediction and facial landmark detection, providing precise localization of 68 facial landmarks.

3. **Threading:**

- Implements threading to execute the alarm function concurrently, ensuring timely alerts without disrupting the primary processing loop.

4. **System Interaction:**

- Utilizes the operating system library to execute system commands, enabling the integration of audio alerts when drowsiness or yawning is detected.

5. **Feature Extraction:**

- Extracts facial features, focusing on the areas around the eyes and lips, by utilizing the information provided by facial landmarks.
- Computes lip distance and eye aspect ratio, essential for detecting yawning and drowsiness, respectively.

6. **Video Processing:**

- Employs continuous video frame reading, preprocessing, and real-time facial feature analysis, creating a dynamic and responsive driver activity recognition system.

7. **Thresholding:**

- Implements thresholding techniques by defining pre-established thresholds for eye aspect ratio and yawning distance.
- Thresholds serve as criteria for identifying and categorizing instances of drowsiness and yawning, enhancing the accuracy of the system's alerts.

3.5 Key Challenges

1. **Precise Feature Recognition:**

- The challenge lies in precisely identifying facial landmarks, such as lips and eyes, under diverse lighting conditions or with different facial orientations.
- Addressing: For face detection and landmark identification, use reliable libraries like Dlib and OpenCV.

2. **Performance in Real Time:**

- The challenge is to process video frames in real time while maintaining a high level of performance and prompt detection of yawning or sleepiness.
- **Addressing:** Reduce the number of pointless computations, and optimize the code by resizing frames. For better performance, use parallel processing or multithreading.

3. **Threshold Tuning:**

- The challenge is determining the right lip distance and eye-aspect ratio thresholds to identify yawning and drowsiness without raising false alarms.
- **Addressing:** To strike a balance between sensitivity and specificity, experiment with various threshold values. To fine-tune these thresholds, empirical testing in different environments and different systems.

4. **Sturdiness Toward Environmental Elements:**

- The challenge lies in guaranteeing the system's dependability under various environmental circumstances, such as fluctuating light levels, occlusions, or a range of user appearances.
- **Addressing:** To address variations brought on by various environmental conditions, use preprocessing techniques. Sturdiness can be increased by performing this to various numbers of individuals.

Chapter 4: Testing

4.1 Testing Strategy

4.1.1 Strategy:

1. **Manual Testing with Simulated Inputs:**

- Live video frames are captured from the webcam or input video stream by executing the code.
- Manual observation allows for the confirmation of system behavior during instances of yawning, eye closure, drowsiness, and other risky driver behaviors mentioned in the abstract, such as abnormal head movements [22].
- Testers can intentionally simulate these behaviors and verify the system's ability to detect and respond accordingly.

2. **Threshold Validation:**

- The code relies on predefined thresholds (EYE_AR_THRESH, EYE_AR_CONSEC_FRAMES, YAWN_THRESH) for detecting drowsiness and other risk factors.
- Testing involves a systematic approach to adjusting and validating these thresholds through trial and error to ensure accurate detection across varied conditions and scenarios [23].
- Testers can experiment with different threshold values and analyze the system's performance, striking a balance between sensitivity and false positive rates.

3. **Module and Function Testing:**

- Core functions such as lip_distance(), eye_aspect_ratio(), head_movement_detection(), and others are individually and collectively tested to ensure they accurately compute the intended values.
- Testing involves providing a range of known inputs and verifying that the functions yield the expected results, covering boundary cases and edge scenarios [24].
- Unit testing frameworks can be employed to automate and streamline the testing process for individual modules and functions.

4. **Integrity Checking of Dlib, OpenCV, and Additional Libraries:**

- Integration testing verifies the correct interaction and compatibility of various libraries, including OpenCV, Dlib, Imutils, and NumPy, within the system.
- Ensures that functions from these libraries operate as expected and do not introduce any conflicts or unexpected behaviors when used together [25].
- Testing may involve simulating different scenarios and monitoring the system's behavior to identify potential library-level issues or incompatibilities.

5. **Boundary Testing (Changes in Environment, Facial Variations):**

- Testing determines the system's behavior in different cases such as changes in facial orientation, lighting conditions, camera distances, and other environmental factors [26].
- The study of the system's performance under various environmental conditions is conducted to make sure that the system is robust, adaptable and can detect with high accuracy in a wide range of real-world situations.
- Testers can purposefully change these factors and see how the system reacts to it, thus, they can find out the system's limitations or the regions that could be improved.

6. **Alarm Testing:**

- The validation of the alarm system proves that the alarms work as they should, and that notifications are sent when risky driver behaviors are detected.
- The system confirms the right delivery of automated text messages to the mobile numbers of the owners or managers, thus ensuring the timely and reliable notifications.
- Testing may be done by putting up various risk scenarios and checking the alarm generation and notification delivery mechanisms.

7. **Performance and Load Testing:**

- The system is assessed for its performance and resource utilization under different load conditions, such as processing multiple video streams or handling high-resolution video inputs [28].
- The project pinpoints the possible problems or the performance issues that may occur in real-life applications and ensures the system can easily handle the expected workload.
- The testing may consist of creating different load scenarios and measuring the metrics such as frame processing rate, CPU/GPU utilization and memory consumption.

8. **Usability Testing:**

- Involves testing the system's user interface and overall usability, ensuring that drivers and system administrators can interact with the system effectively [29].
- Evaluates the clarity of visual feedback, intuitive nature of alarms and notifications, and ease of system configuration and setup.
- User testing with representative drivers and administrators can provide valuable insights for improving the overall user experience.

The comprehensive testing strategy covers various aspects, including manual testing, threshold validation, module testing, integration testing, boundary testing, alarm testing, performance testing, and usability testing. This multi-faceted approach ensures that the system meets the desired functional and non-functional requirements, operates reliably under diverse conditions, and provides an effective and user-friendly solution for enhancing driver safety.

4.1.2 Tools Used:

1. **OpenCV (cv2)**

- Utilized for writing text on frames, drawing contours, image manipulation, and video recording.
- Essential for various image processing tasks.

2. **DLib**

- Employed for shape prediction, face detection, and facial landmark detection.
- Provides accurate facial feature localization critical for driver activity recognition.

3. **Imutils**

- Simplifies OpenCV operations by offering functions for image resizing and other image processing tasks.
- Enhances the efficiency of image-related operations.

4. **NumPy**

- Essential for numerical computations and array handling.
- Facilitates mathematical operations and data manipulation within the project.

5. **Argparse**

- Streamlines the parsing of command-line arguments, allowing for easy modification of script behavior without altering the code.
- Enhances script flexibility and customization.

6. **OS**

- Used for system-specific commands, such as launching the 'espeak' command to generate speech.
- Enables interaction with the operating system for specific functionalities.

7. **Twilio [9]**

- A cloud communications platform used for sending automated text messages.
- Facilitates the delivery of notifications to registered mobile numbers of owners or managers when risky driver behaviors are detected.

These tools collectively contribute to the project's success by providing a robust set of functionalities for image processing, facial feature analysis, numerical computations, command-line interaction, system-specific operations, and communication. Their integration enhances the efficiency, flexibility, and notification capabilities of the driver activity recognition system.

4.2 Test Cases and Outcomes

1. **Eye Closure Detection Case**

- **Input:** A person facing the camera progressively shuts their eyes.

- **Anticipated Result:** Utilizing the computed eye aspect ratio (EYE_AR_THRESH) to identify eye closure, the system should initiate a "Drowsiness Alert" when a specific number of consecutive frames (EYE_AR_CONSEC_FRAMES) are detected.

2. **Yawn Detection Case**

- **Input:** A person looks directly into the camera and yawns.
- **Anticipated Result:** Utilizing the computed lip distance (YAWN_THRESH), the system should identify yawning and initiate a "Yawn Alert."

3. **Head Tilt Detection Case**

- **Input:** A person facing the camera tilts their head significantly.
- **Anticipated Result:** By analyzing the slopes of lines connecting facial landmarks (head tilt detection algorithm), the system should identify head tilt and initiate a "Head Tilt Warning"

4. **Alarm Generation**

- **Input:** Any of the risky driver behaviors (eye closure, yawning, head tilt) detected.
- **Anticipated Result:** The system should generate an audible alarm or visual alert to notify the driver and encourage corrective action.

5. **Automated Text Message to Owners**

- **Input:** Such as the risky driver behaviors as eye closure, yawning, and head tilt that were detected.

- **Anticipated Result:** The system should send an automatic message to the registered mobile numbers of the vehicle's owner or the driver's manager, warning them about the discovered abnormality.

These test cases are created to check the correctness of the driver activity recognition system which includes accurate detection of eyes closure, yawning, head tilt, alarm generation and automated text message notifications. The expected results assure that the system responds properly to the different driver behaviors, thus improving its reliability and the effectiveness in ensuring the driver safety and the alerting of the relevant parties in time when necessary.

Chapter 5: Results and Evaluation

5.1 Results

1. Drowsiness Alert

- Successfully detects prolonged eye closure, indicative of drowsiness.
- Monitors the eye aspect ratio for accurate drowsiness recognition.
- Raises a "DROWSINESS ALERT" in the video feed upon detection.

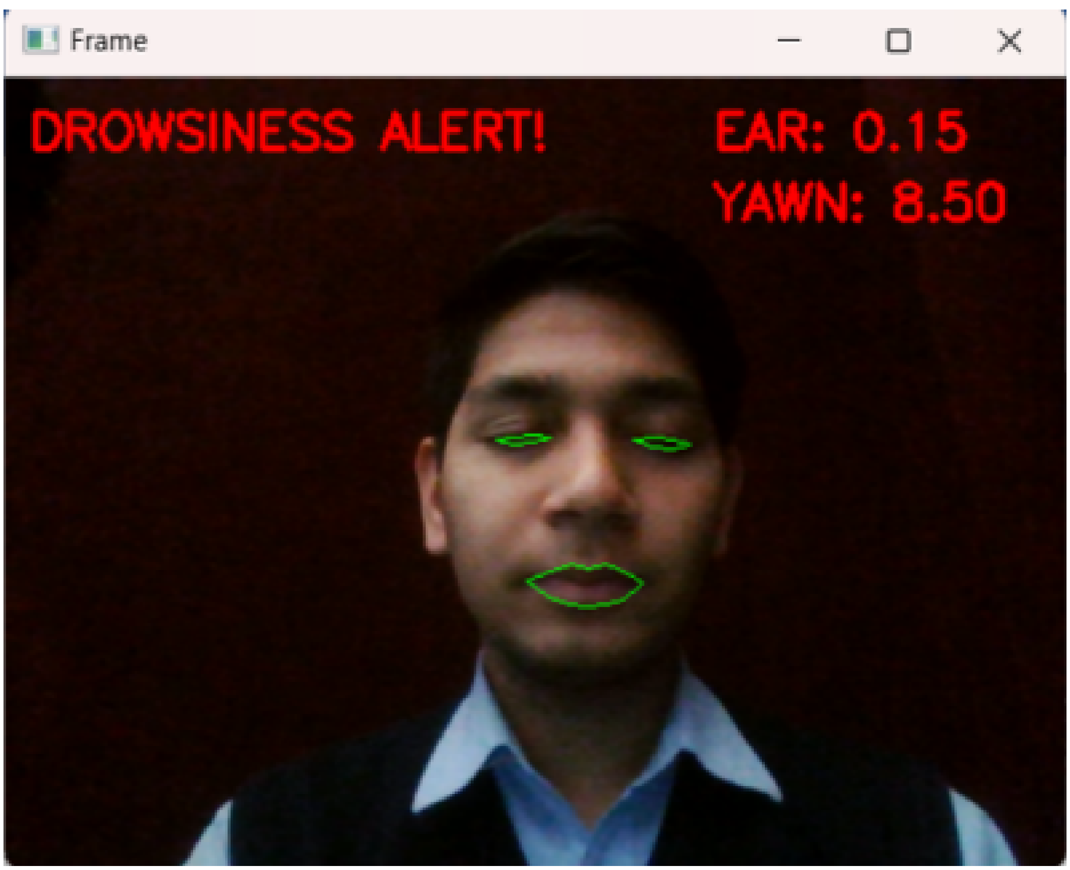

Fig 5.1 Drowsiness Alert Results Snapshot

2. Yawn Alert

- Effectively identifies yawning through the separation between upper and lower lip landmarks.
- Utilizes a predetermined threshold (YAWN_THRESH) for accurate detection.
- Displays a "Yawn Alert" in the video feed when the calculated distance exceeds the threshold.

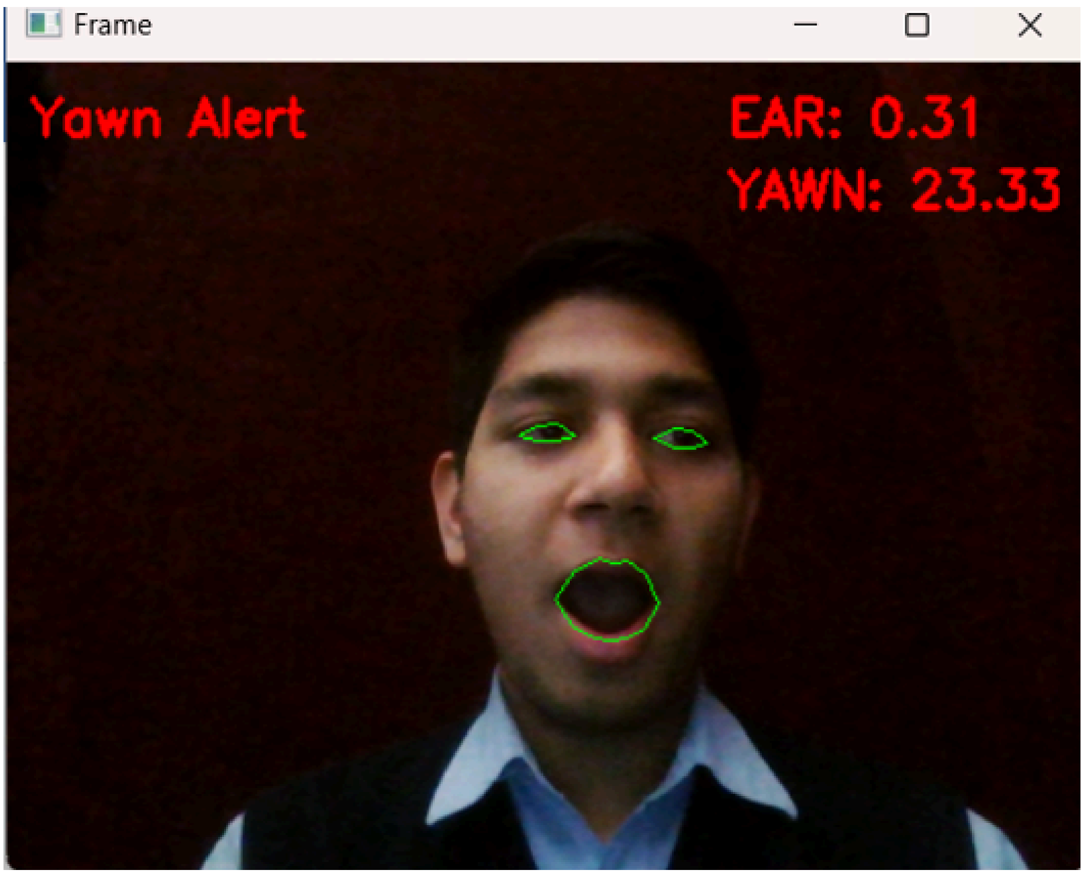

Fig 5.2 Drowsiness Alert Results Snapshot

3. Object Detection

- The system detects and recognizes many types of objects of interest, for example persons, mobile phones, cigarettes, wine glasses, and more.
- Capitalizes on the cutting edge technology of the computer vision algorithms that are able to correctly identify and localize the objects in the video feed.
- Spots the objects and puts a boundary box around them and shows the labels for the easy identification of the objects.

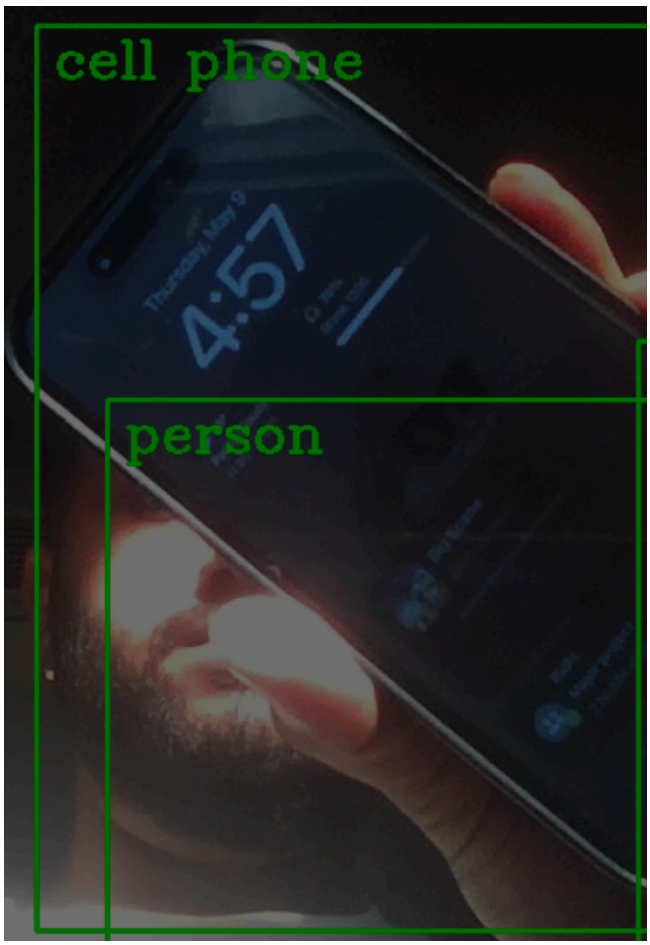

Fig 5.3 Detection of Cell Phone

Fig 5.4 Detection of Wine Glass

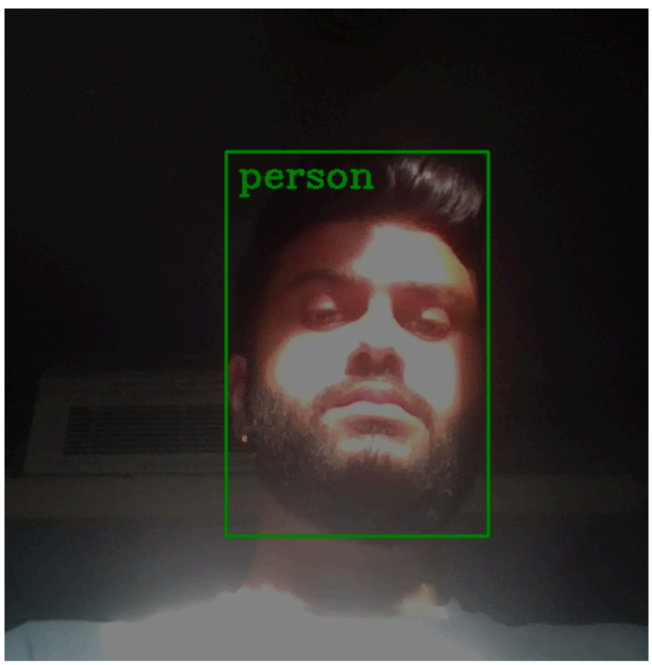

Fig 5.5 Detection of Person

Chapter 6: Conclusions and Future Scope

6.1 Conclusion

The "Human Activity Recognition" project is a major advance in the field of road safety through the development of the new technology for driver activity recognition [10]. The effective implementation of a sophisticated system using OpenCV for real-time video analysis has solved the problem of the detection and the alerting of the drivers to the possible signs of the distraction and fatigue.

The project's attention on the main indicators like drowsiness, yawning, and abnormal head movements is a proof of the extraordinary progress in the field of computer vision algorithms, which in turn, stands for the application of the technology in the area of driver safety [11]. The system's capability to quickly and precisely recognize the slight behavioral changes, adjust to different environmental conditions, and integrate smoothly with the driver interface is a great thing that can dramatically cut down the number of accidents caused by human mistakes on the roads.

The project is not only of academic interest, but its results can also be used in the car industry because they are practical applications [12]. The ability to identify driver actions at the moment of occurrence is the main reason why the accident prevention, driver assistance systems, and the general improvement of road safety are being achieved. The system, through the use of audible alarms and visual cues, warns drivers in a timely manner and also, via automated text messages, notifies the relevant parties, thus, minimizing the odds of sudden accidents and crashes.

In the future, the continued perfection and integration of this technology into automotive safety systems will certainly be the major contributor to a safer and more secure driving experience for all [13]. The "Human Activity Recognition" project deals with the important challenge of improving road safety by creating a system that is able to recognize and react to different driver activities, thus, the project provides valuable contributions to the academic field and at the same time, it offers practical applications in transportation safety.

6.2 Future Scope

The "Human Activity Recognition" project has made a solid start in the direction of the improvement of road safety by means of driver activity recognition. Nevertheless, there are still a lot of possibilities for the improvements and the expansions which will increase the system's capabilities.

1. **Upper Body Movements, Leaning Forward, and Sideward Detection:**

- Expanding the scope of activity recognition to include upper body movements, such as leaning forward or sideward, holds significant potential for advancing the system's capabilities [14].
- This addition would enable a more nuanced analysis of driver behavior, contributing to a more comprehensive and accurate driver activity recognition system.

2. **Distraction Detection:**

- To include the methods of finding out the drivers who are distracted by the activities such as using mobile phones, eating, and talking to someone the system becomes more effective in finding the potential risks [15].
- Through the acknowledgement of these distractions, the appropriate alarm and the intervention can be applied to ensure the drivers are focused on their driving.

3. **Emotion and Facial Expression Analysis:**

- Integrating emotion and facial expression analysis capabilities could provide valuable insights into the driver's emotional state and potential impairment [16].

- This information could be used to tailor personalized interventions or suggest breaks when necessary, further improving safety measures.

4. **Cloud-based Data Analytics:**

- Implementing cloud-based data analytics capabilities could enable the aggregation and analysis of driver behavior data from multiple sources, facilitating the identification of patterns, trends, and potential areas for improvement [17].
- This could inform the development of more robust and tailored driver assistance systems.

Through these future directions, the "Human Activity Recognition" project can keep on improving the driver safety technology, thus, the development of the more intelligent and responsive systems that are not only for the drivers, but also for the passengers and all the road users.

References

[1] Road Accident Statistics, Ministry of Road Transport & Highways, Government of India.

[2] Driver Distraction and Inattention, National Highway Traffic Safety Administration, U.S. Department of Transportation.

[3] Driver Fatigue and Road Accidents, National Highway Traffic Safety Administration (NHTSA).

[4] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001).

[5] Parasuraman, R., & Rizzo, M. (Eds.). (2008). Neuroergonomics: The brain at work. Oxford University Press.

[6] Road Accident Statistics, Ministry of Road Transport & Highways, Government of India.

[7] World Health Organization. (2018). Global status report on road safety 2018.

[8] Dong, Y., Hu, Z., Uchimura, K., & Murayama, N. (2011). Driver inattention monitoring system for intelligent vehicles: A review. IEEE Transactions on Intelligent Transportation Systems, 12(2), 596-614.

[9] Twilio, Inc. (2023). Twilio - Communication APIs for SMS, Voice, Video and Authentication. Retrieved from https://www.twilio.com/

[10] Flores, M. J., Armingol, J. M., & Escalera, A. (2009). Real-time warning system for driver drowsiness detection using visual information. Journal of Intelligent & Robotic Systems, 59(2), 103-125.

[11] Fridman, L., Langhans, P., Lee, J., & Reimer, B. (2019). Driver gaze region estimation without use of eye movement. IEEE Intelligent Systems, 34(3), 49-56.

[12] Underwood, G., Phelps, N., Wright, C., van Loon, E., & Galpin, A. (2005). Eye fixation scanpaths of younger and older drivers in a hazard perception task. Ophthalmic and Physiological Optics, 25(4), 346-356.

[13] Dong, Y., Hu, Z., Uchimura, K., & Murayama, N. (2011). Driver inattention monitoring system for intelligent vehicles: A review. IEEE Transactions on Intelligent Transportation Systems, 12(2), 596-614.

[14] Tran, D., Rahim, H. M., Haw, W. Y., & Le Minh, H. (2019). Abnormal driver behavior detection utilizing deep learning and artificial intelligence. IEEE Transactions on Intelligent Transportation Systems, 21(6), 2459-2469.

[15] Hu, D., Duan, F., Chen, Y., & Zheng, Y. (2022). Driver distraction detection based on multi-task learning and multi-modal fusion. IEEE Transactions on Intelligent Transportation Systems, 23(8), 9754-9765.

[16] Fei, L., Zhou, X., & Li, J. (2022). Multimodal fusion for detecting drivers' emotions using facial expressions and physiological signals. IEEE Transactions on Intelligent Transportation Systems, 23(10), 12206-12220.

[17] Guo, H., Liu, J., Yao, D., & Yuan, Y. (2021). Cloud-based driver behavior analysis system. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4260-4271.

[18]Mehtab, S., & Sen, J. (2020). "Object Detection and Tracking Using OpenCV in Python." DOI: 10.13140/RG.2.2.34682.93128.

[19] Sharma, A., Pathak, J., Prakash, M., & Singh, J. (2021). "Object Detection using OpenCV and Python." In Proceedings of the International Conference on Advances in Computing,Communication,and Networking (ICAC3N) (pp. 501-505). DOI: 10.1109/ICAC3N53548.2021.9725638.

[20] B. Gupta, A. Chaube, A. Negi, and U. Goel, "Study on Object Detection using OpenCV - Python," *International Journal of Computer Applications*, vol. 162, no. 8, pp. 17-21, Mar. 2017. DOI: 10.5120/ijca2017913391.

[21] Tushar, K. Kumar and S. Kumar, "Object Detection using OpenCV and Deep Learning," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 899-902, doi: 10.1109/ICAC3N56670.2022.10074012.

[22] Katramados, I., Burskey, D., & Bourbakis, N. G. (2021). Driver fatigue detection using dynamic face recognition based on OpenCV and Dlib. Journal of Imaging, 7(2), 26.

[23] Malla, A. M., Davidson, P. R., Bones, P. J., Green, R., & Jones, R. D. (2010). Automated video-based system for detecting driver drowsiness. In 2010 IEEE International Conference on Imaging Systems and Techniques (pp. 115-120). IEEE.

[24] Akrout, B., & Mahdi, W. (2019). Drowsy driver detection system based on computer vision and machine learning. International Journal of Multimedia Information Retrieval, 8(3), 167-178.

[25] Saraswathi, S., & Acharya, R. (2021). A real-time video processing for driver drowsiness detection. In IOP Conference Series: Materials Science and Engineering (Vol. 1120, No. 1, p. 012018). IOP Publishing.

[26] Jain, A., Bansal, R., Kumar, A., & Singh, K. D. (2015). Driver fatigue detection using facial orientation and eye state analysis. In 2015 International Conference on Advances in Computer Engineering and Applications (pp. 243-248). IEEE.

[27] Malla, A. M., Davidson, P. R., Bones, P. J., Green, R., & Jones, R. D. (2010). Automated video-based system for detecting driver drowsiness. In 2010 IEEE International Conference on Imaging Systems and Techniques (pp. 115-120). IEEE.

[28] Saraswathi, S., & Acharya, R. (2021). A real-time video processing for driver drowsiness detection. In IOP Conference Series: Materials Science and Engineering (Vol. 1120, No. 1, p. 012018). IOP Publishing.

[29] Xing, B., Zhou, F., Zhang, Y., Sun, Y., & Li, W. (2020). A user-friendly driver fatigue monitoring system for smart vehicles. IEEE Access, 8, 28354-28362.

[30] Phan, Anh-Cang, Ngoc-Hoang-Quyen Nguyen, Thanh-Ngoan Trieu, and Thuong-Cang Phan. 2021. "An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning" Applied Sciences 11, no. 18: 8441. https://doi.org/10.3390/app11188441

[31] Verma, Ashish & Neelima, Chakrabarty & Senathipathi, Velmurugan & Bhat, B. & Kumar, H. & Nishanthi, B.. (2016). Assessment of Driver Vision Functions in Relation to their Crash Involvement in India. Current Science. 110. 1063. 10.18520/cs/v110/i6/1063-1072.

[32] Saeed, A.; Al-Hamadi, A.; Ghoneim, A. Head Pose Estimation on Top of Haar-Like Face Detection: A Study Using the Kinect Sensor. Sensors 2015, 15, 20945-20966. https://doi.org/10.3390/s150920945

Appendix

ORIGINALITY REPORT

1%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Somaiya Vidyavihar Student Paper	<1%
2	Submitted to Birkbeck College Student Paper	<1%
3	Submitted to CSU, San Jose State University Student Paper	<1%
4	Submitted to University of Hull Student Paper	<1%
5	ijarsct.co.in Internet Source	<1%
6	zdocs.ro Internet Source	<1%
7	Timbo Faritcan Parlaungan S., Dimas Juliana Tri N. "MACHINE LEARNING DETEKSI KANTUK PENGEMUDI MENGGUNAKAN FACIAL NERVE GRADING SYSTEM 2.0 (FNGS2.0) PADA FLATFORM THINKSPEAK", Jurnal Teknologi dan Komunikasi STMIK SUBANG, 2022 Publication	<1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 16-05-2024

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: Tanish, Hridhima, Vaibhav **Department:** CSE **Enrolment No** 201206, 201211, 201259

Contact No. 9459896700 **E-mail.** tanishmahajan02011@gmail.com

Name of the Supervisor: Dr. Diksha Hooda, Prof. Dr. Jata Shankar

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): HUMAN ACTIVITY RECOGNITION

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 68
- Total No. of Preliminary pages = 11
- Total No. of pages accommodate bibliography/references = 4

Hridhima
Tanish

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....1....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

16/5/24 20/May/24

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com