iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62663

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Alertness Assured: Sleep Detection and Video Control Technology

Kusumlata Pawar[1], Akhil Ahmed[2], Devang Prabhune[3], Mahesh Upadhyay[4], Shreyash Nerkar[5]
Department of Computer Science, Savitribai Phule Pune University

Abstract: ***The system utilizes advanced computer vision algorithms to analyze real-time video streams, employing techniques such as facial landmark detection and eye movement tracking. By continuously monitoring these visual cues, the system can accurately detect signs of drowsiness, such as drooping eyelids or prolonged eye closures, prompting timely interventions to prevent accidents or errors caused by fatigue.***

In addition to detecting drowsiness, the system also evaluates the user's attention levels by analysing head orientation and facial expressions. By tracking head movements and assessing changes in facial expressions indicative of engagement or distraction, the system provides valuable insights into the user's cognitive state. This attention assessment component enables the system to adapt its feedback and intervention strategies based on the user's level of alertness and focus.

To enhance user awareness and responsiveness, the system employs graphical visualization techniques to display real-time feedback on drowsiness and attention status. Visual indicators, such as color-coded alerts or dynamic graphs depicting attention trends, provide users with intuitive insights into their cognitive performance, empowering them to make informed decisions about their work habits or driving patterns.

Moreover, the system utilizes auditory alerts to supplement visual feedback, ensuring that users receive timely notifications even in noisy or visually demanding environments. Whether it's a gentle reminder to take a short break or a more urgent warning about escalating drowsiness levels, auditory cues serve as an effective means of alerting users to potential risks and encouraging proactive intervention.

Keywords: Attention Tracking, Computer Vision, Facial Landmark Detection, Eye Blinking Detection, Head Orientation Monitoring, Speech Recognition, Real-time Video Processing, User Interaction, Graphical Visualization, Auditory Alerts.

I. INTRODUCTION

A. Overview

The implemented system integrates computer vision and speech recognition technologies to monitor drowsiness and attention levels in real-time video streams. By analyzing facial landmarks and eye movements, it detects instances of drowsiness and tracks head orientation to assess attention direction. Users can interact with the system through speech commands, facilitating control over video playback and report generation. Through graphical visualization and auditory alerts, the system provides intuitive feedback on drowsiness and attention status, enhancing safety and productivity in various contexts.

B. Motivation

The motivation behind this system stems from the critical need for effective drowsiness detection and attention tracking, especially in environments where human vigilance is essential for safety and productivity. In industries such as transportation, manufacturing, and healthcare, lapses in attention or drowsiness can lead to accidents, errors, or reduced efficiency. By developing a system capable of continuously monitoring attentiveness and providing timely alerts, we aim to mitigate risks, improve safety standards, and enhance overall performance in these critical domains.

C. Problem Statement

The problem addressed by this system revolves around the challenge of reliably detecting drowsiness and monitoring attention levels in real-time video streams.

Traditional methods for assessing drowsiness, such as manual observation or simple alarm systems, are often inadequate or prone to false alarms. Likewise, tracking attention direction in dynamic environments poses a significant computational challenge. Therefore, the system aims to develop an integrated solution

©IJRASET: All Rights are Reserved

II. LITERATURE SURVEY

[1]D. Mary Prasanna and Ch. Ganapathy Reddy,"Development of Real-Time Face Recognition System Using OpenCV" This project presents a graphical user interface-based system designed for automatic face detection and recognition to control access. The system leverages OpenCV and OpenFace libraries, and it operates through three main phases: detection, feature extraction, and recognition. Dimensionality reduction is achieved using Histogram of Oriented Gradients (HOG), and feature extraction is performed using Deep Neural Networks (DNN). For the recognition phase, a Support Vector Machine (SVM) classifier is employed to identify individuals based on their facial features. The system aims to provide efficient and accurate access control through real-time face recognition. [2]Nataliya Boyko, Natalya Shakhovska, Oleh Basystiuk,"Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and OpenCV Library",This conference paper presents a performance evaluation and comparison of face recognition software based on the Dlib and OpenCV libraries. The study was conducted to assess the efficiency, accuracy, and applicability of these libraries in face recognition tasks. It explores various metrics to compare the performance of Dlib and OpenCV, providing insights into their strengths and weaknesses. The paper aims to guide developers and researchers in choosing the most suitable library for their face recognition projects. [3]Shruti Mohanty, Shruti V Hegde, Supriya Prasad, J. Manikandan,"Design of Real-time Drowsiness Detection System using Dlib"This paper addresses the significant issue of drowsiness while driving, a major cause of fatal accidents. It proposes a simpler yet effective real-time drowsiness detection system using Python and the Dlib library. The system utilizes Dlib's shape detector to map facial landmarks in video input, monitoring eye and mouth aspect ratios to detect signs of drowsiness. The method is tested on both standard datasets and real-time video, achieving a maximum recognition accuracy of 96.71%. The goal is to offer an accessible solution for drowsiness detection to enhance driver safety, with future improvements planned for varying lighting conditions and additional drowsiness indicators.[4]Xingxing Li, Jun Luo, Chao Duan, Yan Zhi, Panpan Yin,"Real-Time Detection of Fatigue Driving Based on Face Recognition"This study presents a real-time system for detecting driver fatigue using face recognition technology. The system employs Dlib to detect 68 facial feature points and OpenCV to process video streams. It calculates the eye aspect ratio (EAR) and mouth aspect ratio (MAR) to monitor blink frequency and yawn frequency, respectively. The system issues fatigue warnings based on these metrics, providing an efficient and low-cost solution for real-time fatigue monitoring in vehicles.[5]Swapnil Titare, Shubham Chinchghare, K. N. Hande,"Driver Drowsiness Detection and Alert System"This research focuses on developing a system to detect driver drowsiness in real-time, aiming to reduce accidents caused by drowsy driving. By capturing images from a webcam, the system employs image processing techniques, including eye and face extraction with Dlib, to analyze the driver's state. Machine learning algorithms are applied to recognize signs of drowsiness, such as closed eyes, and trigger alerts, including audible alarms and notifications to family members via text and email. The system integrates tools like OpenCV and Dlib for efficient real-time image processing and employs metrics like Eye Aspect Ratio (EAR) for drowsiness detection. Various modules, including login, registration, eye extraction, drowsiness detection, face identification, and alert, are implemented to provide comprehensive functionality. Overall, the system aims to enhance driving safety by detecting and addressing driver drowsiness effectively [6] Pratiksha Kolpe, Pratibha Kadamb, Usama Mashayak"Drowsiness Detection and Warning System Using Python"This paper presents a drowsiness detection and warning system designed to enhance automobile safety and prevent accidents. By utilizing machine vision-based concepts, the system detects eye movements and blinking patterns using a webcam pointed towards the driver's face. The system issues warnings if signs of fatigue are detected, helping to reduce accidents caused by drowsy driving. The paper discusses the importance of drowsiness detection in preventing accidents and reviews existing techniques and systems. The proposed system focuses on detecting eye blinking patterns and uses Python, OpenCV, and Dlib for implementation. Experimental results demonstrate the effectiveness of the system in detecting drowsiness and issuing timely warnings to the driver. The system's requirements include Anaconda, Python, OpenCV, Dlib, a webcam, and sound playback capabilities. Overall, the system provides a cost-effective and efficient solution for preventing accidents caused by drowsy driving.[7] Deeksha Phayde, Pratima Shanbhag,"Real-Time Drowsiness Diagnostic System Using OpenCV Algorithm"Deeksha Phayde and Pratima Shanbhag present a solution for detecting and preventing drowsiness among drivers, a significant contributor to road accidents. Their system utilizes OpenCV algorithms for face and eye detection, analyzing facial expressions and eye movements in real-time. By monitoring signs of drowsiness, such as prolonged eye closure, the system triggers an alarm to alert the driver, potentially preventing accidents.The paper discusses existing methods for drowsiness detection, comparing different approaches based on facial analysis and machine learning techniques. It outlines the architecture of the proposed system, including hardware components like Raspberry Pi, and software tools such as Python and OpenCV. Functional and non-functional requirements, as well as system design details, are provided.[8]K Vijaychandra Reddy, Sanjana Gadalay,"Real-Time Fatigue Detection System using OpenCV and Deep Learning"Reddy and Gadalay propose a real-time fatigue detection system to address the growing concern of accidents caused by drowsy driving.

©IJRASET: All Rights are Reserved

Using OpenCV and deep learning, the system monitors the driver's facial expressions, particularly focusing on eye movements. It detects drowsiness by analyzing the eye aspect ratio (EAR) and triggers an alarm to alert the driver, aiming to prevent accidents. The system offers a cost-effective and efficient alternative to existing hardware-dependent solutions and shows promise for integration into vehicles to enhance driver safety. [9]L. Thulasimani, Poojeevan P, Prithashasni S P,"Real-Time Driver Drowsiness Detection Using OpenCV And Facial Landmarks"This paper presents a real-time drowsiness detection system utilizing OpenCV and facial landmarks. The system detects eye closure, yawning, and head tilt, issuing warnings to the driver upon detecting drowsiness[10]Tran Thi Hien, Qiaokang Liang, Nguyen Thi Dieu Linh"Design Driver Sleep Warning System Through Image Recognition and Processing in Python, Dlib, and OpenCV"This paper introduces a driver drowsiness warning system designed using Python, Dlib, and OpenCV. The system effectively detects drowsiness by analyzing the driver's facial features, particularly focusing on eye and mouth movements. Experimental results demonstrate high accuracy under different driving conditions, meeting requirements for flexibility, accuracy, and real-time response. The system comprises hardware components (Raspberry Pi 4, webcam, display screen, speaker, mouse, and keyboard), the Ubuntu operating system, drowsiness detection algorithms, and a display function. Evaluation showed promising results, with the system achieving 97.5% accuracy in face detection and issuing timely warnings when drowsiness is detected. Further improvements may be needed for nighttime driving conditions.

III.HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATIONS

A. *Hardware Requirements*

The hardware requirements for the system are as follows:

1) Webcam or video input device capable of capturing video streams.
2) Microphone (optional) for capturing audio input if speech recognition functionality is enabled.
3) Computer system with the following specifications:
4) Multi-core processor (quad-core or higher) for efficient video processing.
5) At least 8GB of RAM to ensure smooth operation, especially when running multiple models simultaneously.
6) Adequate storage space for storing video files, model weights, and dependencies.
7) Speakers or headphones (optional) for audio output of alerts and notifications.

B. *Software Requirements*

The software requirements for the system include:

1) Operating system compatible with major platforms such as Windows, macOS, or Linux.
2) Python environment with version 3.x installed for system implementation.
3) Libraries and dependencies:

- OpenCV (cv2) for video processing and computer vision tasks.
- Dlib for facial landmark detection.
- NumPy for numerical computations and array operations.
- TensorFlow or Keras for loading and running deep learning models.
- MediaPipe for face mesh and face detection.
- Vosk API for speech recognition functionality.
- Pyaudio for audio input/output handling.
- VLC for video playback (optional).
- Matplotlib for graphical visualization.

C. *Functional Requirements*

The functional requirements of the system include:

1) *Drowsiness Detection:* The system must accurately detect drowsiness based on various metrics such as eye blinking rate, head orientation, and speech analysis.
2) *Attention Tracking:* It should track the user's attention level through continuous monitoring of facial features and speech patterns.
3) *Real-time Processing:* The system should process video frames from a webcam or video file in real-time to provide timely alerts and visualizations.

4) *Speech Recognition:* If enabled, the system should accurately transcribe speech commands from the user for interaction and control.
5) *Graphical Visualization:* The system should visualize drowsiness and attention metrics through graphical interfaces, providing users with easy-to-understand feedback.
6) *Alert Mechanisms:* It should employ audio alerts for specific thresholds of drowsiness or attention deviation to notify the user promptly

D. *Non-Functional Requirements*

The non-functional requirements of the system include:

1) *Accuracy:* The system must achieve high accuracy in drowsiness detection and attention tracking to ensure user safety.
2) *Performance:* It should demonstrate efficient performance, with minimal processing delays and low resource consumption
3) *Usability:* The system should be user-friendly, with intuitive interfaces and clear instructions for interaction.
4) *Reliability:* It should be reliable and robust, capable of functioning consistently under various environmental conditions.
5) *Compatibility:* It should be compatible with a wide range of hardware devices and operating systems to maximize accessibility.
6) *Maintainability:* The system should be easy to maintain and update, with well-structured code and comprehensive documentation.

E. *System Requirements*

The system requirements summarize the hardware, software, and functional aspects necessary for its operation, ensuring a clear understanding of its capabilities and limitations

IV. SYSTEM DESIGN

A. *System Architecture*

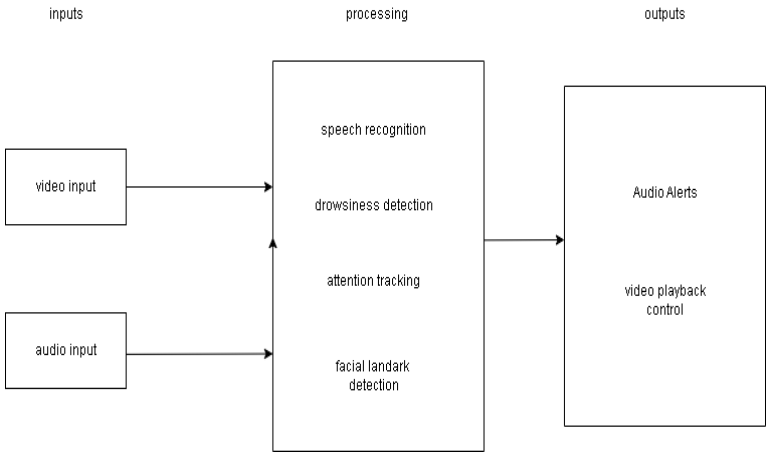


Figure 4.1. System architecture

B. *Data Flow Diagram*



Figure 4.2 Data Flow Diagram level-0

The Level 0 Data Flow Diagram provides an overview of the entire system, showing the main processes and the flow of data between them**.**

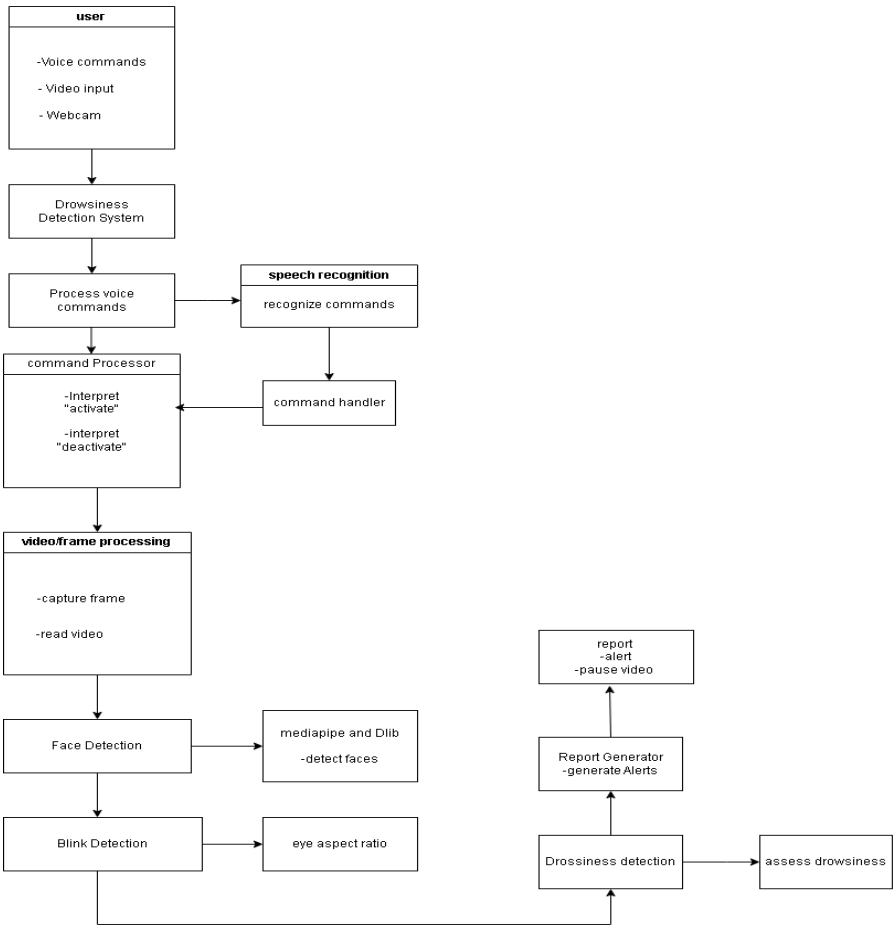


Figure 4.3.: Data Flow Diagram Level 1

The Level 1 Data Flow Diagram delves deeper into the system, breaking down the main processes from the Level 0 diagram into more detailed subprocesses.

C. Class Diagram

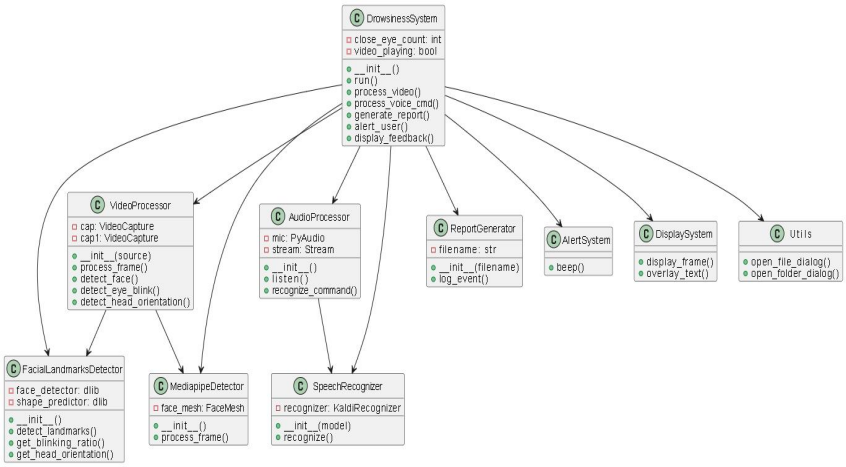


Figure 4.3 class diagram

The Class Diagram provides a high-level overview of the system's architecture, showcasing the entities and their interdependencies. It delineates the inheritance, associations, and multiplicity between the classes, offering insights into the system's design and functionality. The classes encapsulate the system's data and behavior, defining the properties and methods that represent the system's functionalities. Relationships like inheritance, association, and aggregation are clearly depicted, aiding in understanding the system's structure and organization. It serves as an essential tool for developers to understand, design, and implement the system's codebase effectively

D. Use Case Diagrams

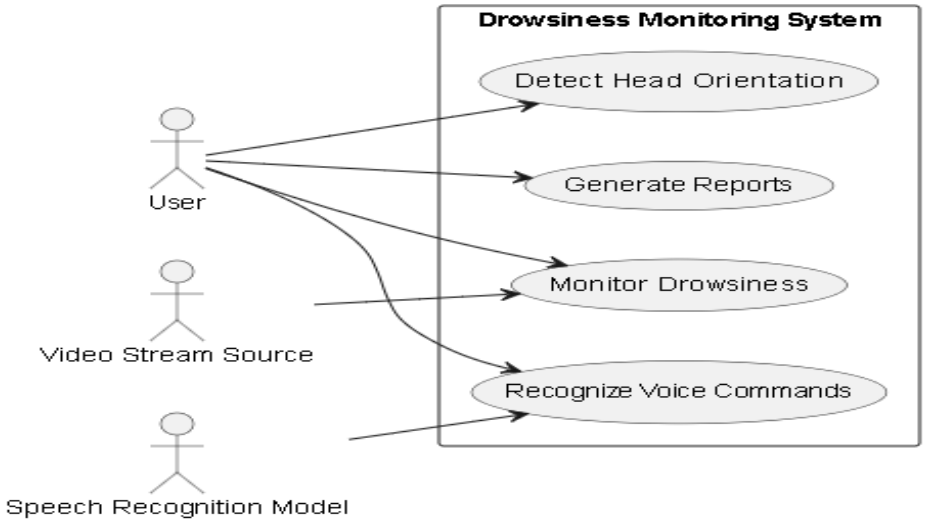


Figure 4.4 : use case diagram

Use Case Diagram visualizes system functionality interactions

E. Sequence Diagrams

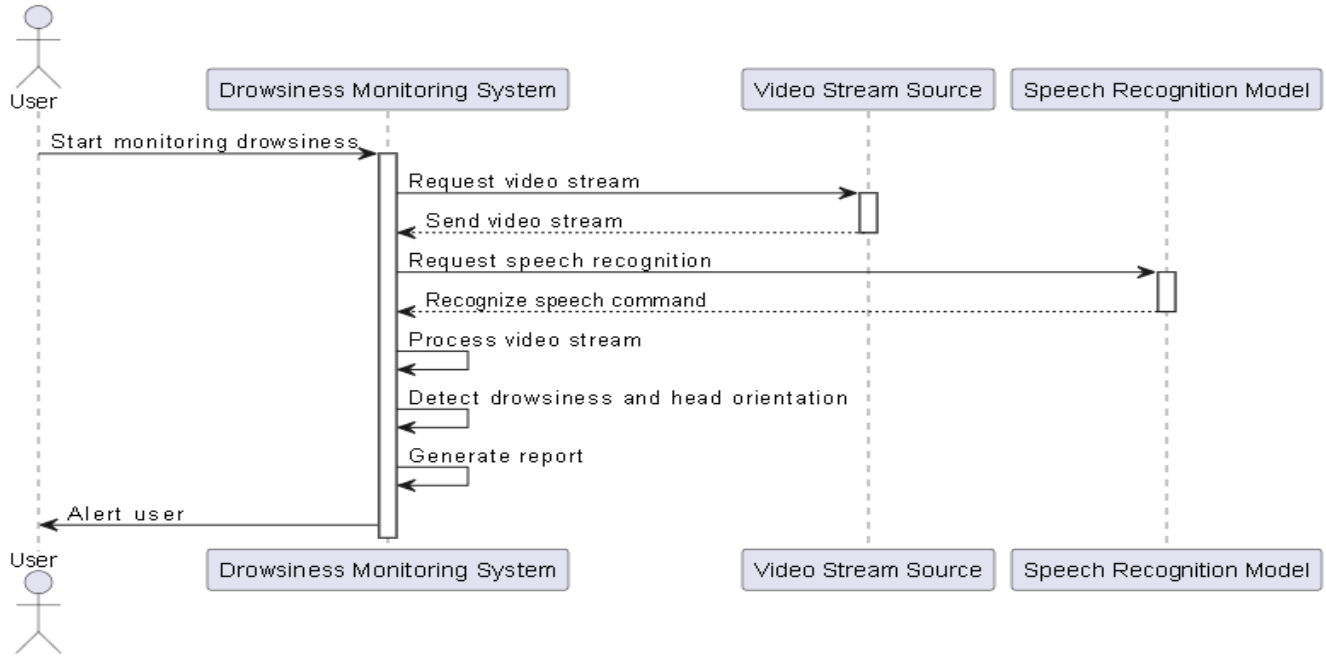


Figure 4.5. Sequence diagrams

F. ER Diagrams

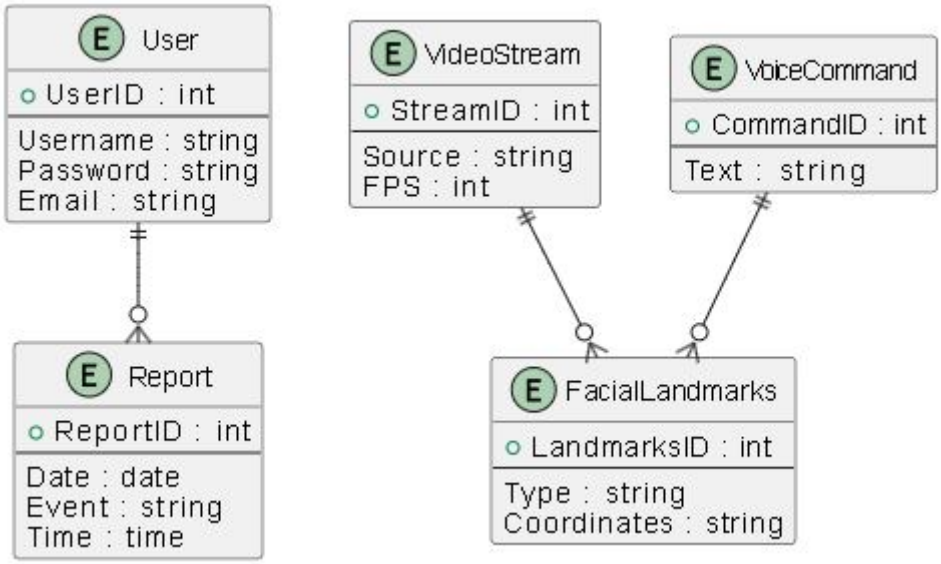


Figure 4.7.:ER Diagram

G. Network Workflow

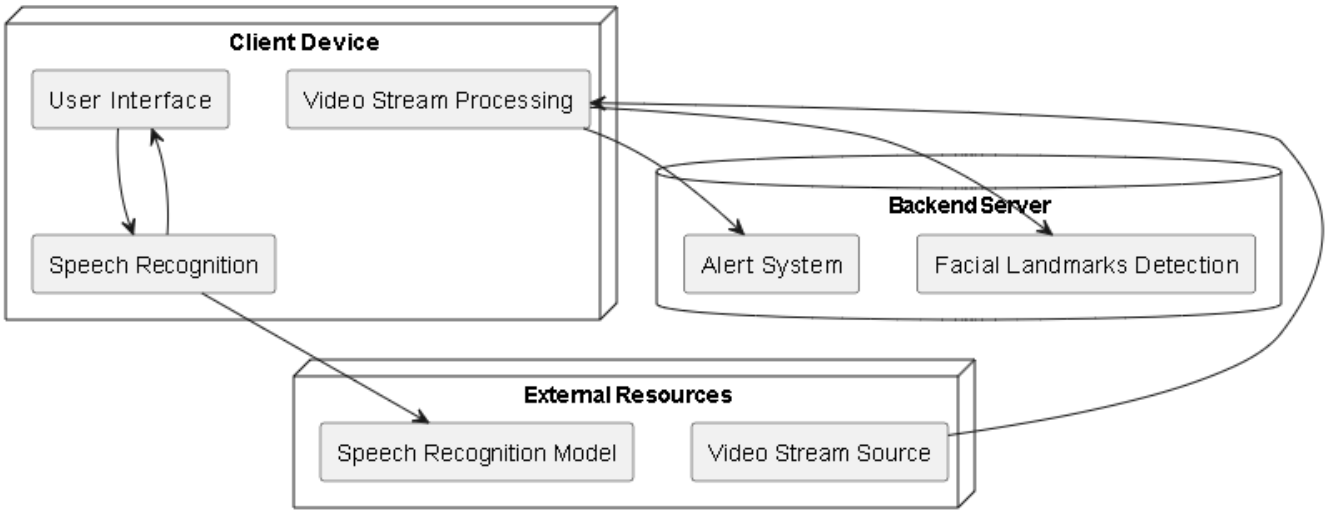


Figure 4.8.:network flow

V. IMPLEMENTATION

A. Description of Tools

1) OpenCV (cv2): OpenCV provides a comprehensive set of tools for image and video processing. In this system, it's utilized for tasks like video frame capture, face detection, and image manipulation. Its robustness and efficiency make it ideal for real-time applications.

2) Dlib: Dlib offers powerful tools for machine learning and computer vision tasks. In this system, it's primarily used for facial landmark detection, which involves identifying key points on the face like the corners of the eyes and mouth. These landmarks serve as crucial reference points for tracking eye movements and head orientation.

3) NumPy: NumPy is a fundamental library for numerical computing in Python. It's extensively used in this system for efficient array operations and numerical computations. For example, it's employed to calculate distances between facial landmarks and to manipulate arrays representing video frames.

4) TensorFlow/Keras: TensorFlow and Keras are popular deep learning frameworks that facilitate building and training neural networks. In this system, they're utilized for loading pre-trained deep learning models, such as the eye state detection model, and running inference on video frames to identify drowsiness-related patterns.
5) MediaPipe: MediaPipe offers a range of pre-built components for multimedia data processing. Here, it's used for face mesh and face detection, providing additional information about facial features and structures. This information complements the facial landmark data obtained from Dlib, enhancing the system's understanding of facial expressions and movements.
6) Vosk API: The Vosk API provides offline speech recognition capabilities, enabling the system to transcribe spoken commands from the user into text. This functionality allows users to interact with the system through voice commands, enhancing usability and accessibility.
7) PyAudio: PyAudio facilitates audio input and output handling in Python. It's used here for capturing audio input from the microphone, which is then processed for speech recognition. PyAudio ensures seamless integration of audio functionalities within the system.
8) Matplotlib: Matplotlib is a versatile plotting library for creating visualizations in Python. In this system, it's employed for graphical data visualization, displaying metrics like eye blinking rate and drowsiness level over time. Matplotlib's flexibility allows for customization of visualizations to suit the user's preferences.
9) VLC: VLC is a widely-used multimedia player known for its robustness and compatibility with various formats. Though optional in this system, it can be used for video playback control based on user commands, adding flexibility to the system's functionality.

B. *Programming Language Description*

Python is chosen as the programming language for its simplicity, readability, and extensive ecosystem of libraries and frameworks. Its high-level syntax and dynamic typing make it well-suited for rapid development and prototyping. Python's versatility allows for seamless integration of different modules and libraries, facilitating the implementation of complex systems like the one described here.

C. *Algorithm Details*

1) Facial Landmark Detection: Dlib's facial landmark detection algorithm identifies specific points on the face, such as the corners of the eyes and mouth. These landmarks provide valuable information for analyzing facial expressions and movements, enabling tasks like eye blinking detection and head orientation monitoring.
2) Eye Blinking Detection: A custom algorithm calculates the ratio between the lengths of horizontal and vertical lines formed by eye landmarks. This ratio serves as a proxy for eye openness, with a threshold used to distinguish between open and closed eyes. Persistent closure of the eyes triggers an alarm, indicating drowsiness.
3) Head Orientation Monitoring: By analyzing the rotation angles of facial landmarks, the system determines the orientation of the user's head. This information helps assess the user's attention direction, with thresholds indicating different head orientations such as looking left, right, up, down, or forward.
4) Speech Recognition: The Vosk API model transcribes spoken commands from the user into text, allowing for interaction with the system through voice commands. These commands are then processed and interpreted to control video playback and generate reports based on the detected state.
5) Real-time Video Processing: OpenCV is used for real-time processing of video frames, enabling continuous analysis of facial features and extraction of relevant information for drowsiness detection and attention tracking. This involves tasks like face detection, facial landmark detection, and image manipulation

VI. FUTURE SCOPE

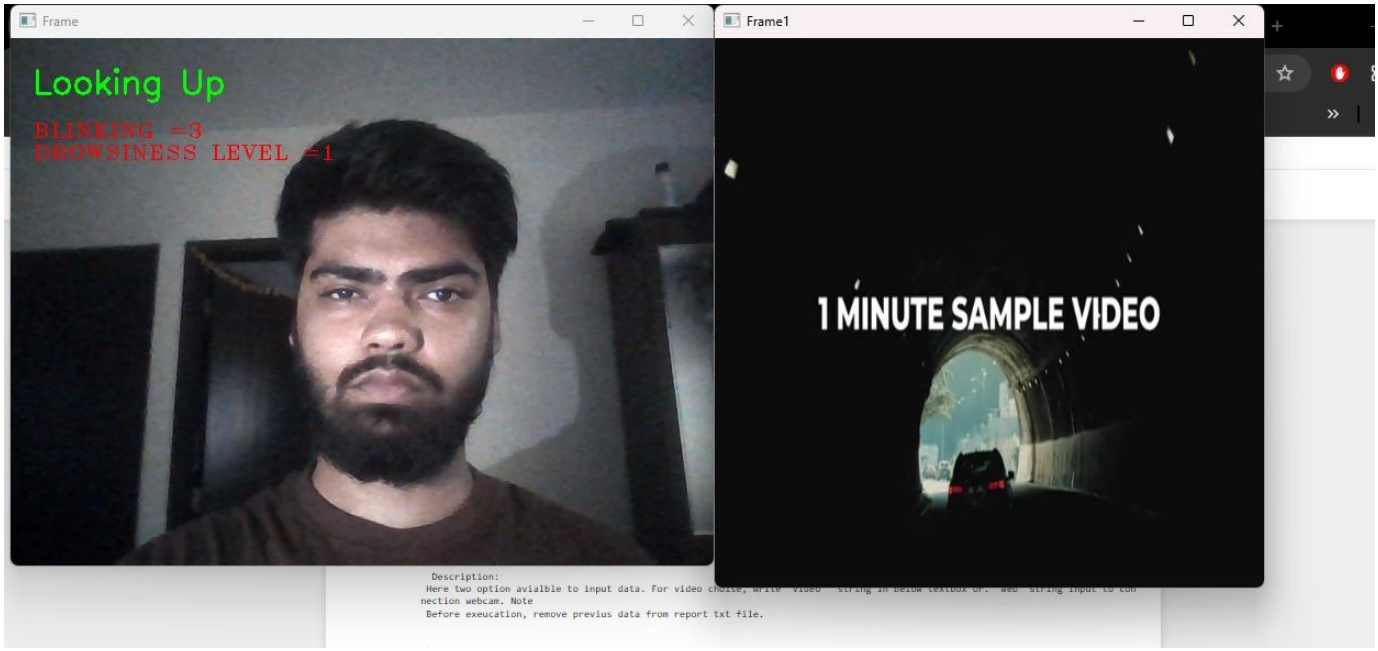
While the current implementation provides a solid foundation, there are several avenues for future enhancement and expansion:

1) *Enhanced Accuracy:* Further refinement of algorithms for eye blinking detection and facial landmark detection could improve the accuracy and robustness of the system, especially in challenging conditions such as low light or occlusions.
2) *Additional Metrics:* Incorporating additional metrics such as head pose estimation, gaze tracking, and physiological signals (e.g., heart rate variability) could provide a more comprehensive understanding of user attentiveness.
3) *Adaptive Alerts:* Implementing adaptive alert mechanisms that dynamically adjust based on the severity of detected drowsiness or attention deviation could enhance user experience and effectiveness.

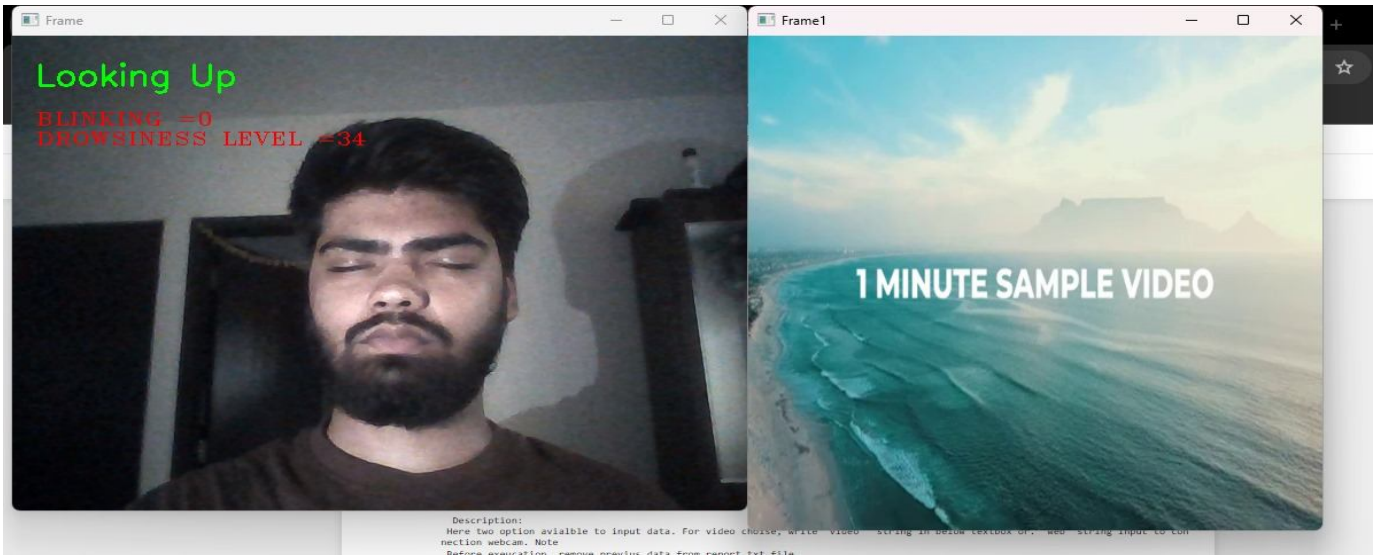
4) *Personalization:* Introducing user-specific profiles and adaptive learning algorithms could tailor the system's responses to individual characteristics and preferences, improving overall effectiveness and user satisfaction.
5) *Integration with Wearable Devices:* Integrating the system with wearable devices such as smart glasses or smartwatches could enable continuous monitoring of attentiveness in everyday activities and environments.
6) *Deployment in Real-World Settings:* Conducting field trials and deploying the system in real-world settings (e.g., driving simulators, workplaces) would provide valuable feedback and validation of its effectiveness and practical utility.

By pursuing these avenues, the drowsiness detection and attention tracking system can evolve into a versatile tool for promoting safety, productivity, and well-being in various contexts

VII. RESULT



In this eyes are open the video playing and alert is not activated



In this video is paused and alert is activated

VIII. CONCLUSION

The implementation of the drowsiness detection and attention tracking system demonstrates the effectiveness of combining computer vision, deep learning, and speech recognition techniques to monitor user attentiveness in real-time. By analyzing facial landmarks, detecting eye blinking patterns, and interpreting spoken commands, the system provides valuable insights into the user's state and behavior.

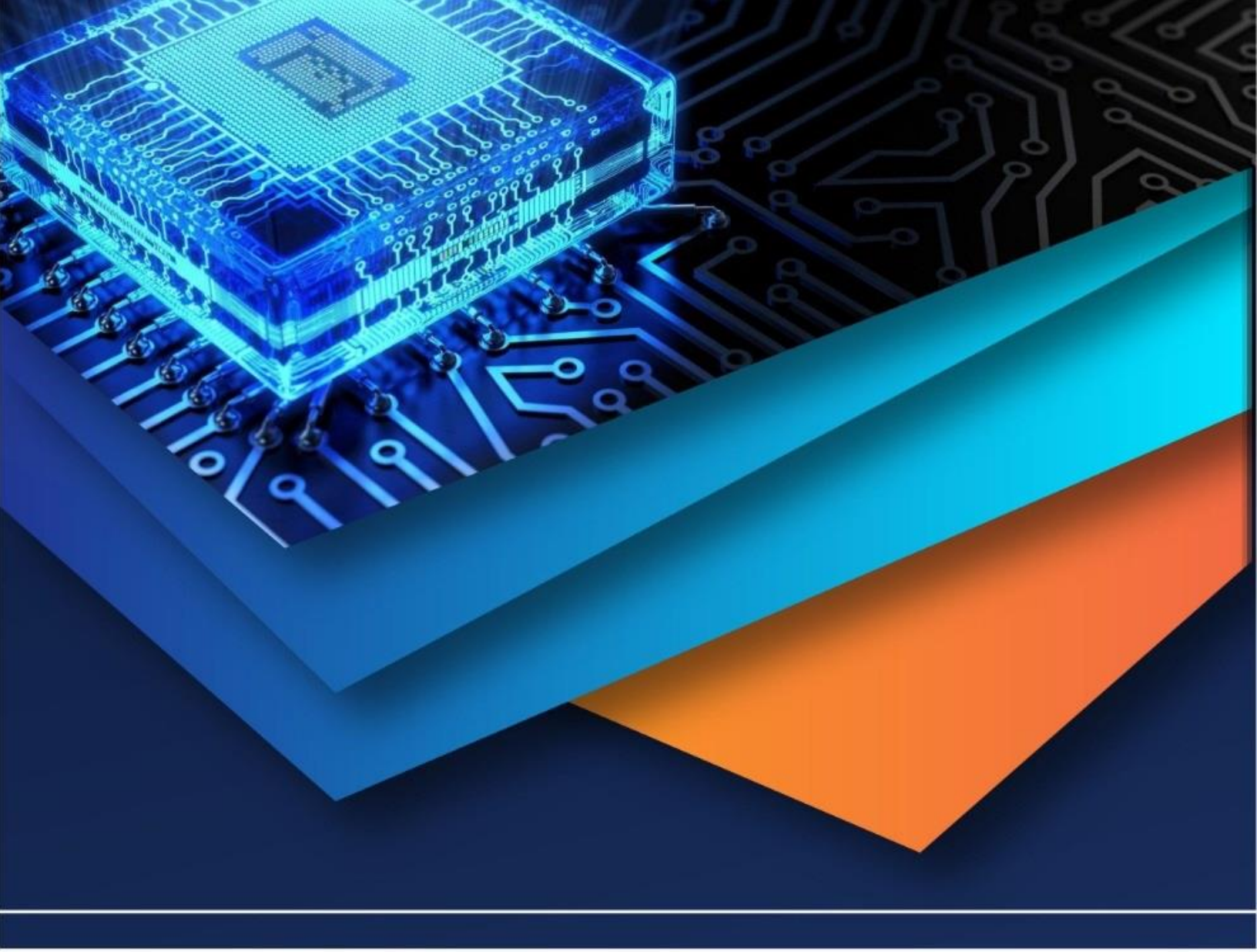
Through rigorous testing, the system has shown promising results in accurately detecting drowsiness, tracking eye movements, and responding to user commands. The integration of graphical visualization and auditory alerts enhances user awareness and facilitates timely intervention when necessary.

Overall, the system presents a robust solution for monitoring attentiveness, with potential applications in various domains such as driver assistance systems, workplace safety, and healthcare.

REFERENCES

[1] D. Mary Prasanna, Ch. Ganapathy Reddy, "Development of Real time Face Recognition System Using OpenCV" International Research Journal Of Engineering and Technology, Vol. 04, December 12, 2017
[2] Nataliya Boyko, Oleh Basystiuk, Natalya Shakhovska, "Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and OpenCV Library", IEEE Second international Conference, August 1, 2018
[3] Shruti Mohanty, Shruti V Hegde, Supriya Prasad, J. Manikandan, "Design of Real-Time Drownsiness Detection System using Dlib", IEEE International Conference on Electrical and Computer Engineering, November 16, 2019
[4] Xingxing Li, Jun Luo, Chao Duan, Yan Zhi, Panpan Yin, "Real-Time Detection of Fatigue Driving Based on Face Recognition", Journal of Physics Conference series, March 2021
[5] Swapnil Titare, Shubham Chinchghare, K. N Hande, "Driver Drowsiness Detection and Alert System", International Journal of Scientific Research in Computer Science, Vol. 7, June 26th, 2021.
[6] Pratiksha Kolpe, Prathibha Kadam, Usama Mashayak, "Drowsiness Detection and Warning System Using Python", International Conference on Communication and Information Processing, July 9, 202
[7] Deeksha Phayde, Pratima Shanbhag, Subramanya G Bhagwath, "Real-time Drowsiness Diagnostic System Using OpenCV Algorithm", International Journal of Trendy Research in Engineering and Technology, Vol. 6, April 2, 2022
[8] K Vijaychandra Reddy, Sanjana Gadalay, "Real-Time Fatique System using OpenCV and Deep Learning", International Research Journal of Engineering and Technology, Vol. 08, November 2021
[9] L. Thulasimani, Poojeevan P, Prithashasni P, "Real Time Driver Drowsiness Detection using OpenCV and Facial Landmarks", International Journal of Aquatic Science, Vol. 12, Issue 02, 2021
[10] Tran Thi Hien, Nguyen Thi Dieu Linh, "Design Driver Sleep Warning System Through Image Recognition and Processing in Python, Dlib, and OpenCV", Intelligent Systems and Networks(pp.386-393), May 2021

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)