International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

FATIGUE DETECTION OF DRIVERS USING MACHINE LEARNING

Dr. R. Lalu Naik
PROFESSOR
Department of Computer Science and Engineering
Tirumala Engineering College

KURAPATI RADHA PRASANTHI
Student
Computer Science and Engineering
Tirumala Engineering College

JETTI PRASANNA
Student
Computer Science and Engineering
Tirumala Engineering College

KOMMAVARAPU LIKHITHA ANKADEVI
Student
Computer Science and Engineering
Tirumala Engineering College

KADIYALA GAYATHRI
Student
Computer Science and Engineering
Tirumala Engineering College

***Abstract* - In this project, the aim is to develop a prototype drowsiness detection system to address driver fatigue, a major cause of accidents worldwide. The system detects driver drowsiness by monitoring eye blinks and yawning. If the driver's eyes remain closed for an extended period or if yawning is detected, an alarm is sounded. Programming for this system is conducted in OpenCV, utilizing the Haar cascade library for face detection and other machine learning libraries to detect facial features. The system operates in real-time and is nonintrusive, prioritizing driver safety without being obtrusive. By effectively monitoring driver alertness and issuing timely warnings, this system contributes significantly to enhancing road safety.**

Keywords - *Driver fatigue, drowsiness detection, eye blink, yawn detection, OpenCV, Haar cascade, machine learning, facial features.*

I. INTRODUCTION

Driver fatigue contributes significantly to a considerable number of vehicle accidents. Recent statistics suggest that annually, fatigue-related crashes result in approximately 1,200 deaths and 76,000 injuries. Developing technologies aimed at detecting or preventing drowsiness while driving poses a significant challenge in the realm of accident avoidance systems. Given the danger posed by drowsiness on the road, it is imperative to devise methods for mitigating its impact. This project seeks to create a prototype drowsiness detection system with a focus on accurately monitoring the driver's eye state in real-time. By observing eye behavior, it is hypothesized that signs of driver fatigue can be identified early enough to avert potential accidents. Detecting fatigue entails analyzing eye movements and blink patterns within a series of facial images. Throughout history, humans have innovated transportation technologies, enhancing convenience and safety in daily life. Today, transportation is integral to our routines, whether by personal or public means. However, operating vehicles comes with a responsibility to remain alert, as negligence can lead to tragic consequences. Fatigue-induced errors contribute significantly to accidents, especially in countries like India where infrastructure challenges exacerbate risks.

Detecting and addressing driver drowsiness is a critical focus for safety. While alcohol and drug impairment have clear indicators, fatigue is subtler and harder to detect. This challenge has spurred research into drowsiness detection systems, aiming to enhance accuracy and effectiveness. In India, car accident fatalities in 2015 highlighted the significant role of fatigue-induced errors, yet awareness and prevention efforts remain insufficient.

Efforts to address fatigue-related accidents require a multi-pronged approach, including awareness campaigns and economic incentives for responsible driving. Despite some countries implementing driving hour regulations, enforcement remains inconsistent. Thus, comprehensive solutions are needed, integrating both awareness and structural reforms in transportation.

In 2015, India witnessed a staggering 148,707 fatalities from car accidents, with at least 21 percent attributed to fatigue-induced errors, though this figure likely underestimates fatigue's true impact amid numerous accident causes. In developing nations like India, where infrastructure deficiencies compound risks, fatigue poses a grave threat. Unlike alcohol and drug impairment, fatigue is elusive to measure or observe, complicating prevention efforts. Promoting awareness of fatigue-related risks and encouraging drivers to acknowledge fatigue are crucial solutions, albeit challenging and costly to implement. Economic factors also play a significant role, as driving long hours can be financially lucrative, driving drivers to make risky decisions for financial gain. Despite some countries imposing driving hour restrictions, enforcement remains inadequate, highlighting the ongoing need for comprehensive measures to address fatigue-related accidents effectively.

To contribute to this endeavor, our project utilizes OpenCV's Haar Cascade classifier for fast face detection and Dlib's Shape predictor for facial feature extraction. By monitoring eye and mouth aspect ratios, we aim to detect signs of drowsiness and mitigate potential accidents.

II. LITERATURE SURVEY

A 2016 World Bank survey indicated that road crashes exact a heavy toll on the Indian economy, accounting for an estimated 3 to 5 percent of GDP annually. The financial repercussions extend beyond economic figures, impacting victims' families by thrusting them into poverty and debt due to injuries and fatalities. Furthermore, road accidents have far-reaching consequences on the nation's human capital, underscoring the profound impact on India's societal and economic landscape.

Figure 1: Road accidents are distinguished by age group

Figure 1 illustrates the distribution of road accidents across different age groups. The data indicates that approximately 76.2 percent of fatalities in Indian road accidents occur among individuals aged 18 to 45. This highlights a substantial loss of the country's workforce to traffic accidents annually.

A. Driver Drowsiness Detection System and Techniques

Experts note that drivers who don't take breaks face increased drowsiness risks, with studies indicating that accidents often stem from fatigue rather than drink-driving. Attention assist systems can detect inattentiveness and drowsiness across various speeds, notifying drivers of their fatigue level and time since their last break. These systems offer adjustable sensitivity and, upon issuing a warning, suggest nearby service areas via the COMAND navigation system.

B. Implementation of the Driver Drowsiness Detection System

This paper focuses on enhancing car intelligence and interactivity to alert or intervene in unsafe conditions, providing real-time information to rescue services, police, or the owner. Addressing driver fatigue, a significant factor in rising accidents, we present a real-time safety prototype that regulates vehicle speed based on driver fatigue detection. Our aim is to develop an advanced fatigue detection system.

C. Detecting Driver Drowsiness Based on Sensors

Researchers have attempted to determine driver drowsiness using the following measures: vehicle-based measures, behavioural measures, physiological measures.

D. Eye Tracking Based Driver Drowsiness Monitoring and Warning System

This project aims to create an interface for detecting driver drowsiness by continuously monitoring their eyes using DIP algorithms. Micro sleeps, brief periods of sleep lasting 2 to 3 seconds, serve as reliable indicators of fatigue. By monitoring the driver's eyes with a camera, the system detects signs of drowsiness and issues timely warnings. The project focuses on developing advanced hardware for driver safety, integrating controllers and image processing. The product alerts the driver with alarms and reduces vehicle speed upon detecting drowsiness. Additionally, it employs continuous distance monitoring using an ultrasonic sensor to warn and slow down the vehicle in the presence of obstacles.

III. Technology Used

A. PYTHON

Python is a high-level, interpreted programming language created by Guido Van Rossum in 1991. Known for its emphasis on code readability and significant use of whitespace, Python supports various programming paradigms and is dynamically typed and garbage collected. It offers language constructs and an object-oriented approach to facilitate clear, logical code for projects of all scales.

B. PYCHARM

PyCharm, developed by JetBrains, is an Integrated Development Environment (IDE) tailored for Python programming. It features code analysis, a graphical debugger, integrated unit testing, and version control system (VCS) integration. PyCharm supports web development with Django and data science with Anaconda, and it is compatible with Windows, macOS, and Linux. The community edition is open-source under the Apache License, while the Professional Edition offers additional proprietary features.

C. OPENCV

OpenCV (Open-source Computer Vision) is a library of programming functions primarily designed for real-time computer vision tasks. Initially developed by Intel and later supported by Willow Garage and Itseez (acquired by Intel), OpenCV is cross-platform and free to use under the BSD license. It supports models from deep learning frameworks like TensorFlow, Torch, PyTorch (after conversion to an ONNX model), and Caffe, promoting a portable format known as Open Vision Capsules.

D. SCIPY

The SciPy package offers various toolboxes addressing common challenges in scientific computing. Its submodules cater to different applications such as interpolation, integration, optimization, image processing, statistics, and special functions. SciPy relies on NumPy, which facilitates convenient and swift N-dimensional array manipulation. Together, they provide efficient numerical practices like numerical integration and optimization, compatible with all major operating systems and quick to install, free of charge. NumPy and SciPy utilize multidimensional arrays as the fundamental data structure, with NumPy also offering functions for Linear Algebra, Fourier Transforms, and Random Number Generation.

E. NUMPY

NumPy, a Python library, augments the language with support for large, multi-dimensional arrays and matrices, alongside a vast collection of mathematical functions for array operations. Initially developed by Jim Hugunin, with contributions from various developers, NumPy was revamped in 2005 by Travis Oliphant, merging features of the competing Num array into Numeric with substantial modifications. As open-source software, NumPy boasts numerous contributors.

F. DLIB

Dlib, a modern C++ toolkit, encompasses machine learning algorithms and tools tailored for solving real-world problems in various domains, including robotics, embedded devices, mobile phones, and high-performance computing environments. Widely used in both industry and academia, Dlib's open-source licensing enables free usage in any application without installation or configuration steps required.

IV. METHODOLOGY

Machine learning is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and

inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly told.

V. SYSTEM ANALYSIS

A. PROPOSED SYSTEM

Driver fatigue detection systems are widely used in vehicles, offering improved results over drowsiness detection systems. These systems, present in high-end vehicles, alert users when drowsiness reaches a critical level, though they are often unaffordable for the middle class. While qualitative, AI and neural networks convert drowsiness measures into quantitative data, enhancing reliability. Root Mean Squared Error (RMSE) calculations improve prediction accuracy, though not all users can easily implement these methods due to software limitations.

Literature surveys on drowsiness detection highlight key processes like face image extraction, yawning tendency, blink detection, and eye area analysis. MATLAB-based systems are effective but face limitations due to cost and accessibility. Python, with its open-source nature and powerful libraries, offers a viable alternative. Face detection technologies like MTCNN and HaarCascade classifiers are employed, with HaarCascade outperforming MTCNN in real-time performance.

Accurate identification of facial features is crucial for tasks like facial recognition, expression analysis, and gender classification. Facial landmarks play a key role in extracting features like eyes and mouth, enhancing the effectiveness of drowsiness detection systems.

B. PROPOSED SYSTEM

Driver's Drowsiness is detected using different machine learning libraries like OpenCV and Dlib. Here the face recognition is being done along with detecting weather the eyes are open or closed. If the eyes are seemed to be closed for a significant amount of time. An alarm is triggered in order to make the driver Alert.

A camera is setup that looks for faces in the input video stream and monitors frames of faces. In the event that a face is identified, facial milestone identification is connected and the eye district is removed from the edges of the video stream.

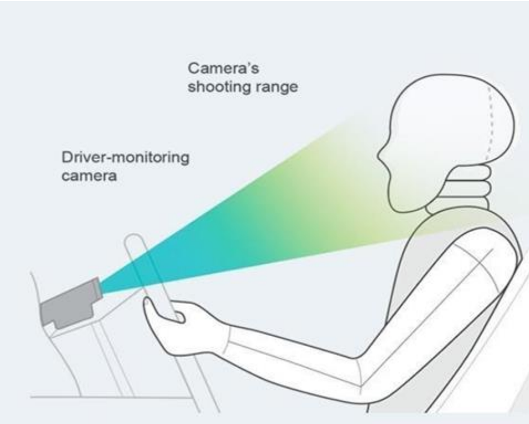

Figure 2: Practical design of the system

VI. SYSTEM DESIGN

A. DATA FLOW DIAGRAM

- The Data Flow Diagram (DFD), also known as a bubble chart, is a simple graphical tool that illustrates a system's input data, processing stages, and output data generation.
- DFDs are crucial modeling tools used to represent system components, including system processes, data utilized by these processes, external entities interacting with the system, and the flow of information within the system.
- They depict how information moves through a system and undergoes transformations, showcasing the flow of information and the alterations applied as data progresses from input to output.
- DFDs, or bubble charts, can represent systems at various levels of abstraction and can be partitioned into levels that illustrate increasing levels of information flow and functional detail.

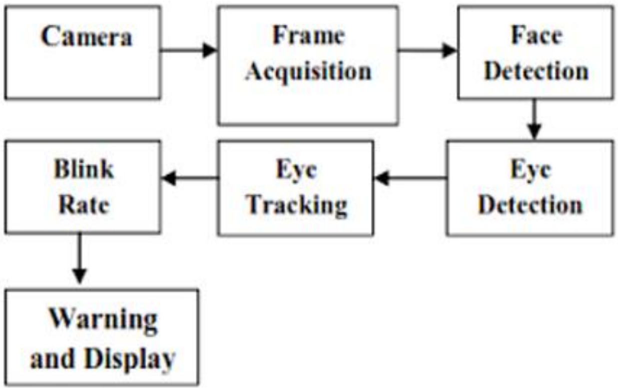

Figure 3: Data Flow Diagram

B. SYSTEM ARCHITECTURE

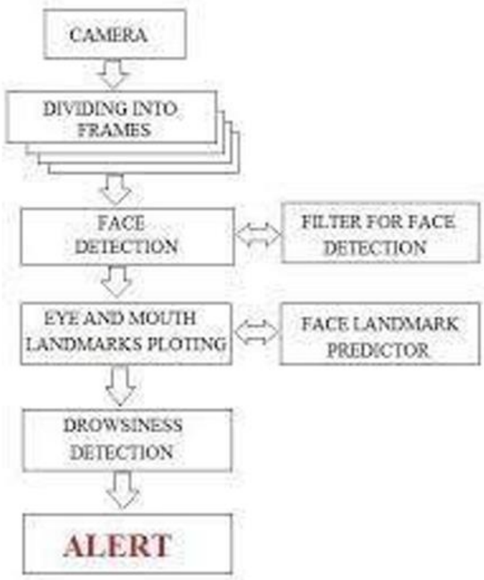

Figure 4: System Architecture

C. UML DIAGRAMS

UML, short for Unified Modeling Language, is a standardized modeling language widely used in object-oriented software engineering. Managed by the Object Management Group, its aim is to establish a common language for creating models of object-oriented software. UML consists of two main components: a Meta-model and a notation, with the possibility of adding a method or process in the future. It serves as a standard language for specifying, visualizing, constructing, and documenting software system artifacts, as well as for business and non-software systems modeling.

UML embodies a set of best engineering practices proven effective in modeling large and complex systems, making it a crucial aspect of object-oriented software development. Its predominantly graphical notations facilitate the expression of software project designs.

i. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

ii. USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a behavioral diagram derived from Use-case analysis. It provides a graphical overview of a system's functionality, showcasing actors, their goals (use cases), and any dependencies among these use cases. The primary goal of a use case diagram is to illustrate the system functions performed for each actor, including the depiction of actors' roles within the system.

iii. SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

iv. ACTIVITY DIAGRAM

Activity diagrams in the Unified Modeling Language (UML) are graphical representations of stepwise activities and actions within workflows, supporting choice, iteration, and concurrency. They describe business and operational workflows in a system, depicting step-by-step processes and the overall flow of control.

v. STATE CHART DIAGRAM

A state diagram, alternatively called a state machine diagram or statechart diagram, visually represents the various states an object can occupy and the transitions between these states within the Unified Modeling Language (UML).

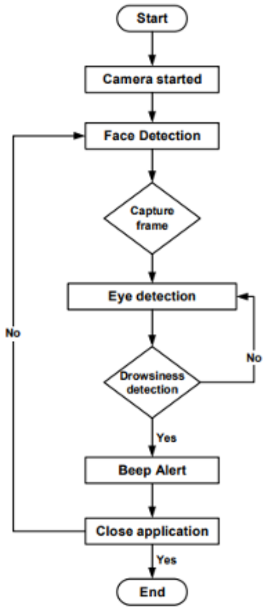

Figure 5: State chart diagram

VII. CONCLUSION

In this study, an automated system was developed to detect driver drowsiness using a continuous video stream. The system employs the Haar cascade algorithm to detect drowsiness, which in turn utilizes predefined Haar features to detect facial features like eyes. The algorithm calculates blink frequency using the PERCLOS algorithm. When the blink frequency remains at 0 for a certain duration, indicating drowsiness, the system activates an alarm to alert the driver. Similarly, if the blink frequency remains constant for extended periods, indicating driver distraction, the alarm is also activated.

VIII. REFERENCES

Assari, M. A., & Rahmati, M. (2011). Driver drowsiness detection using face expression recognition. 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA).

Tianyi Hong, Huabiao Qin, & Qianshu Sun. (2007). An Improved Real Time Eye State Identification System in Driver Drowsiness Detection. 2007 IEEE International Conference on Control and Automation. .

Warwick, B., Symons, N., Chen, X., & Xiong, K. (2015). Detecting Driver Drowsiness Using Wireless Wearables. 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems.

Dwivedi, K., Biswaranjan, K., & Sethi, A. (2014). Drowsy driver detection using representation learning. 2014 IEEE International Advance Computing Conference (IACC).

Yan, J.-J., Kuo, H.-H., Lin, Y.-F., & Liao, T.-L. (2016). Real-Time Driver Drowsiness Detection System Based on PERCLOS and Grayscale Image Processing. 2016 International Symposium on Computer, Consumer and Control (IS3C).

Alshaqaqi, B., Baquhaizel, A. S., Amine Ouis, M. E., Boumehed, M., Ouamri, A., & Keche, M. (2013). Driver drowsiness detection system. 2013 8th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA).

Tripathi, D.P., Rath, N.P.(2009). A novel approach to solve drowsy driver problem by using eye-localization technique using CHT. International Journal of Recent Trends in Engineering.

Subbarao, A., Sahithya, K. (2019) Driver Drowsiness Detection System for Vehicle Safety, International Journal of Innovative Technology and Exploring Engineering (IJITEE).