International Journal of Advanced Technology in Engineering and Science
Vol. No. 11, Issue No. 04, April 2023
www.ijates.com
ijates
ISSN 2348 - 7550

Fatigue Detection of Drivers Using Machine Learning

(Under the guidance of Dr. R. Lalu Naik, Professor, Head of the Department, Department of Computer Science and Engineering, Tirumala Engineering College)

B.Yamini , A.Ramya , A.Mohana Lakshmi, B.Sushma

ABSTRACT

One of the main factors in accidents around the world is driver weariness. One of the most accurate ways to gauge driver weariness is by observing the driver's level of drowsiness. In this project, we want to create a drowsiness detection system prototype.

The driver's eye blink is recognised in this project. The motorist is deemed to be drowsy and an alarm is sounded if their eyes are closed for longer than a predetermined amount of time.

The initiative also searches for yawning, and if it does, the driver is deemed to be sleepy and a warning is raised. The Haar cascade library for face detection and other Machine Learning libraries for the programming are both used in OpenCV.

1. INTRODUCTION

In a substantial number of auto accidents, driver weariness plays a crucial role. According to recent figures, collisions caused by driver drowsiness result in an estimated 1,200 fatalities and 76,000 injuries per year. A significant obstacle in the field of accident avoidance systems is the creation of technology that can detect or prevent tiredness at the wheel. Methods for reducing the effects of drowsiness are needed because of the danger it poses while driving. This project's goal is to create a sleepiness detecting system prototype. Designing a system that can precisely track the driver's eyes' open or closed status in real time will be the main focus. It is thought that the signs of driver fatigue can be caught early enough to prevent an automobile collision by keeping a watch on the eyes.

2. LITERATURE SURVEY

Road accidents are thought to cost the Indian economy between 3 and 5 percent of GDP annually, according to a 2016 World Bank report. The relatives of the victims of road accidents have a heavy financial burden, driving the poor into poverty and debt. The human resources of the nation are significantly impacted by traffic accidents.

Road accidents are distinguished by age group .

Figure shows the data of road accidents distinguished by age group. Figure depicts that around 76.2 percent of individuals killed in road accidents in India are between the ages of 18 and 45. This implies that the country loses a significant portion of its workers due to traffic accidents each other.

3. IMPLEMENTATION

• The main algorithm is a really simple one. First, we employed a camera that is placed up in a car in a desirable location and searches for faces.

• If a face is found, the region around the eyes is removed and the facial landmark detection task is used.

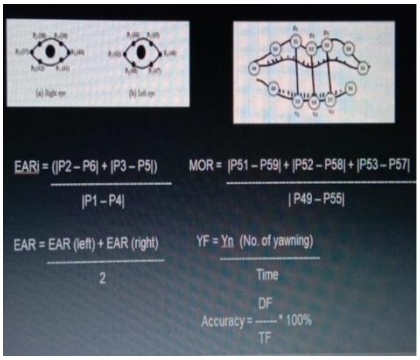
• After obtaining the eye region, we compute the eye aspect ratio to determine whether the eyelids are down for a prolonged period of time.

• The alert will sound loud to rouse the driver awake if the Eye Aspect Ratio indicates that the eyes have been closed for a disproportionately lengthy period of time.

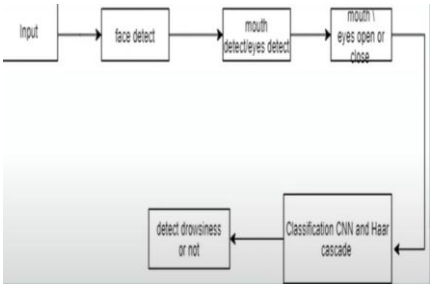
We have utilised OpenCv, dlib, and Python to implement the system's features and to make it operate effectively.

• Putting the drowsiness into effect the system consists of machine learning algorithms, which are then a part of OpenCv ML algorithms. Although there are several ML methods, we simply needed the face detecting algorithm for our needs.

•Fundamentally, it is a revolutionary application for item discovery. With the OpenCv circulation, prepared frontal face identification is also available. Overall, it operates effectively and smoothly. With the necessary software, it may also be used to detect a variety of different item kinds.



Data flow diagram:



4. WORKING

Drivers face is monitored throughout using a video or web camera. In order to detect the drowsiness the first step is to detect the face using the set of frames taken by the camera. Then the location of the eyes is detected and retina of the eye is continuously monitored. The captured image is sent to the processor for image processing. It converts the received image to digital signal using Open CV. The digital signal is transmitted from transmitter to the receiver. Both the transmitter and the receiver are paired up. The signal is then passed to the LPC2148, the microcontroller. If the signal crosses the threshold value of EAR for a given number of frames, then the alarm beeps and the speed of the vehicle is automatically reduced

Real time work

5. TYPES OF TESTINGS

Unit Testing

The creation of test cases for unit testing verifies that the core program logic is working correctly and that program inputs lead to legitimate outputs. It is important to verify the internal code flow and all decision branches. It is the testing of the application's separate software components. Before integration, it is done following the completion of each individual unit. This type of intrusive structural testing requires an understanding of how it was built. Unit tests run basic tests at the component level and evaluate a particular setup for a system, application, or company.

Integration Testing

Integration tests are created to check whether integrated software parts function as a single application. Event-driven testing is more focused on the fundamental consequence of screens or fields. Integration tests show that even though the individual components were satisfactory, the combination of the components is accurate and consistent, as successfully demonstrated by unit testing.

Integration testing is especially designed to highlight issues that result from combining components.

Functional Testing

Functional tests offer methodical proof that functional tests are available in accordance with the technical and business requirements, system documentation, and user manuals. Focus of functional testing is on the following areas: Determine classes of legitimate input that must be received. Determine which categories of invalid inputs need to be rejected.

Functions: Exercised identities require exercised functions.

output: categorize the application outputs that need to be tested, Procedures: It is necessary to activate interface systems or processes.

Functions tests are organised and prepared with a focus on requirements, important functions, or unique test cases. Moreover, testing should take into account systematic coverage of data fields, established processes, and subsequent processes, as well as the flow of identified business processes. More tests are found before functional testing is finished, and the effective value of test is determined.

System Test

System testing makes ensuring that the integrated software system as a whole complies with specifications. In order to provide known and predictable outcomes, it tests a setup. Configuration-oriented system integration testing is an illustration of on-site testing.

White Box Testing

It is a type of testing when the software tester is familiar with the inner workings, language, and structure of the software, or at the very least its intended use. It is utilised to examine regions that are inaccessible from a black box level.

Black Box Testing

White Box Without having any prior knowledge of the inner workings, structure, or language of the module being tested, testing the software. Black box tests, like the majority of other types of tests, must be written from

a clear source text, such as the requirements section of a specification. It is The programme being tested is viewed as a "black box" that you cannot see inside. Without taking into account how the software functions, the test generates inputs and responds to outputs.

6. TEST PLAN

Test plan is a report outlining the objectives, strategy, available tools, and timetable for the planned testing activities. It lists test items, features to be tested, testing tasks, who will complete each task, degree of tester independence, test environment, test design techniques, entry and exit criteria to be used, and the justification for their selection, among other things. It also lists any risks that call for contingency planning. It serves as a log of the test preparation procedure.

Examine the setup:

Only when the tester is familiar with a system or product, such as how it operates, who its end customers are, what software or hardware it employs, what it is used for, etc., can it be thoroughly assessed.

Design the Test Strategy:

figuring out the expenses and efforts necessary to accomplish the system's goals in order to design a test strategy for every form of functional hardware.

The test strategy for each project may be created by: • defining the testing's scope.

• Determining the necessary testing type
• Risks and problems
• establishing testing logistics.

Specify the test objectives. The test objective is the general objective that must be accomplished for the exam to pass. The system must be free of bugs and ready for usage by end users before objectives can be created. The goal of the test and the software elements that must be tested in order for it to be successful can be used to establish the test objective.

Define Test Criteria:

A test procedure or test judgement can be based on test criteria, which are standards or rules. Two such test criteria are as follows: Exit criteria, which outlines the requirements that signify a test phase's successful conclusion, and suspension criteria, which state that if a predetermined number of test cases fail, the tester should halt all ongoing test cycles until the criteria is met.

Planning resources:

The project task's resource requirements are listed in depth in the resource plan. Resources include people, tools, and supplies needed to finish a project.

Plan the test environment:

A testing environment is a configuration of hardware and software where the testing team will run test cases.

Schedule & Estimation:

To reduce the chance of failing to finish the project by the deadline, it is essential to create a timetable for the various testing phases and estimate the amount of time and labour required to test the system. The creation of the test specification, test execution, test reporting, and test delivery are all included.

Choose the test outputs:

Deliverables are the materials that must be created, updated, and maintained to support the testing process. Deliverables for testing are offered prior to, throughout, and following the testing phase.

The following test plan and various situations should be taken into account for this project:

1. The purpose of this study is to identify driver attention and sleepiness while operating a non-two-wheeled vehicle
2. Several project modules, including those for face, eye, and face-and-eye tracking, drowsiness, and distraction detection, are all put through unit and functional testing. Via functional testing, determine whether the system is able to extract the necessary aspects of the face after the face has been detected. API testing and database testing are not covered by the test plan because this project lacks any kind of api. Currently, there will be no testing of nonfunctional aspects like stress, performance, or logical databases. Finally, system testing for the alarm while the driver is sleepy or distracted is used to check the functionality of the complete system.
3. Every team member in the project group is introduced to the testing procedure in order to reduce the risks of any team member failing tocomprehend the testing. To make the testing faster, two members of this project group have been assigned the responsibility to test this system. The functional and system testing of this system were carried out by G. Vamsi Krishna and P.R. Krishna Priya, respectively.
4. The testing cycle for this project is suspended if 50% of the test cases are deemed to have failed, and the development team, which consists of D. Saiesh and J. Likith, has performed the necessary code improvements. 95% of the test cases need to pass in order to pass the exit criteria. We have achieved this success under low light conditions and good light conditions.
5. Two members of the project team made up the resource planning for testing: G. Vamsi Krishna, who identified various testing scenarios and can be regarded as the test developer or administrator, and P.R. Krishna Priya, who carried out the tests, recorded the results, and alerted the project team to any flaws.
6. The test environment and system resources consist of a Windows 10 PC with an i5 6th Gen processor, 8GB RAM, and a 1TB HDD. The complete project is executed and tested using the WebCam-attached PyCharm IDE.

The project's timetable is as follows:

• G. Vamsi Krishna developed the test specification, which called for testing the project's many components in a variety of scenarios while ignoring others.

• P.R. Krishna Priya carried out the test execution, which involved carrying out all of the different test cases supplied by G. Vamsi Krishna. She meticulously records the findings and communicates them, along with any flaws.

• Both members of the testing team provide the test reports in the appropriate manner. The development team, which also comprised D. Saiesh and J. Likith, handles flaws appropriately. These test cases have since been evaluated once more and have been shown to be effective in producing the desired results.

TEST REPORT

7. CONCLUSION

An automated technique for identifying driver tiredness was created in the current study. Drowsiness is determined by reading the continuous visual stream from the system. Using the Haar cascade method, it is discovered. Haar characteristics are used by the haar cascade algorithm to find faces and eyes. Predefined Haar characteristics are employed to detect a variety of items. The haar characteristics are added to the image, and the perclos method is used to determine blink frequency. If the reading stays at 0, it recognises that the user is drowsy and informs the driver by sounding an alarm. The motorist is considered to be preoccupied and an alarm is sent off if the value stays the same for extended periods of time