DRIVER DROWSINESS DETECTION

A main Project Report submitted in the partial fulfillment of the

requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

P.Narendra (19471A0542)

Sk. Fardhin Basha (19471A0557)

M.Gopi Chandu (19471A0536)

Under the esteemed guidance of

Dr. S.V.N.Sreenivasu,M.Tech,Ph.D

Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE

Accredited by NAAC with A+ Grade and NBA (Tier -1) NIRF

rank in the band of 251-320 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDAROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the main project entitled **"DRIVER DROWSINESS DETECTION"** is a bonafide work done by P.Narendra (19471A0542), Sk.Fardhin Basha (19471A0557),M.Gopi Chandu (19471A0536)partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the department of **COMPUTER SCIENCE AND ENGINEERING during 2022-2023.**

PROJECT GUIDE
Dr S.V.N.Sreenivasu ,M.Tech., Ph,D
Professor

PROJECT CO-ORDINATOR
Dr.M.Sireesha ,M.Tech., Ph.D
Assoc.professor

HEAD OF THE DEPARTMENT
Dr.S.N.TirumalaRao, M.Tech., Ph.D.
Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M.V.Koteswara Rao,B.Sc.,** who took keen interest in us in every effort throughout this course.

We owe out sincere gratitude to our beloved principal **Dr. M. Sreenivasa Kumar, M.Tech., Ph.D., MISTE., FIE(I).,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao, M. Tech., Ph.D.,** H.O.D. Of CSE department our guide **Dr. M. Sireesha,M.Tech.,Ph.D** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish ourproject successfully in time.

We extend our sincere thanks towards **Dr.M.Sireesha,M.Tech.,Ph.D.,** Project Coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

P.Narendra(19471A0542)

Sk.FardhinBasha(19471A0557)

M.Gopi Chandu(19471A0536)

ABSTRACT

Many of the accidents occur due to drowsiness of drivers. It is one of the critical causes of roadways accidents now-a-days. Latest statistics say that many of the accidents were caused because of drowsiness of drivers. Vehicle accidents due to drowsiness in drivers are causing death to thousands of lives. More than 30% accidents occur due to drowsiness. For the prevention of this, a system is required which detects the drowsiness and alerts the driver which saves the life. In this project, we present a scheme for driver drowsiness detection. In this, the driver is continuously monitored through webcam. This model uses image processing techniques which mainly focuses on face and eyes of the driver. The model extract the drivers face and predicts the blinking of eye from eye region. We use an algorithm to track and analyze drivers face and eyes to measure Perclos. If the blinking rate is high then the system alerts the driver with a sound.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching ,imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems relatedto academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex

engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts ,and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature

CO425.4: Design and Modularize the project

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		√											√		
C425.2	√		√		√								√		
C425.3				√		√	√	√					√		
C425.4			√			√	√	√					√	√	
C425.5					√	√	√	√	√	√	√	√	√	√	√
C425.6									√	√	√i		√	√	

Course Outcomes – Program Outcome correlation

		PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**
2. **Medium level**
3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a **Driver Drowsiness Detection System.**	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into **six modules**	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our four members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the **driver drowsiness detection system is using to prevent the road accidents by using system.**	PO4, PO7
CC4.2.8 CC4.2.	**The physical design includes hardware components like eye tracking sensors, camera**	PO5, PO6

INDEX

LIST OF FIGURES

1.INTRODUCTION

1.1INTRODUCTION

Drowsines of one of the key issues for majority of road accidents. Drowsiness threatens the road safety and causes severe injuries sometimes, resulting in fatality of the victim and economical losses. Drowsiness implies feeling lethargic, lack of concentration, tired eyes of the drivers while driving vehicles. Most of the accidents happen in India due to the lack of concentration of the driver. Performance of the driver gradually deteriorates owing to drowsiness. To avoid this anomaly, we developed a system that is able to detect the drowsiness nature of the driver and alert him immediately. This system captures images as a video stream through a camera, detects the face and localizes the eyes. The eyes are then analysed for drowsiness detection using Perclos algorithm. Based on the result, the driver is alerted for drowsiness through an alarm system.

To prevent such accidents, our team has come up with a solution for this. In this system, a camera is used to record user's visual characteristics. We use face detection and CNN techniques and try to detect the drowsiness of driver, if he/she is drowsy then alarm will be generated. So that the driver will get cautious and take preventive measures. Driver drowsiness detection contributes to the decrease in the number of deaths occurring in traffic accident

1.1 EXISTING SYSTEM:

Face detection and eye region extraction Whole face region may not be required to detect the drowsiness but only eyes region is enough for detecting drowsiness. At first step by using the Viola-jones face detection algorithm face is detected from the images. Once 2the face is detected, Viola-jones eye detection algorithm is used to extract the eyeregion from the facial images. it is the first algorithm used for face detection**.**

For the face detection the Viola-Jones algorithm having three techniques those are Haar-like features, Ada boost and Cascade classifier.In this work, Viola-Jones object detection algorithm with Haar cascade classifier was used and implemented using OPEN CV with python. Haar cascade classifier uses Haar features for detecting the face from images.

1.2 PROPOSED SYSTEM:

Face and Eye Detection by OpenCV Algorithm:

In this system, instead of Support Vector Machine (SVM) we use a Classification Model based on Convolutional Neural Networks (CNN). Deep Learning is concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Convolutional Neural Networks are a type of Artificial Neural Networks which are widely used for image classification and even Multi-Class classification of images

Convolution layers in a CNN consist of a set of learnable filters. During forward propagation, we slide each filter across the whole input step by step where each step is called stride. We propose CNN since the accuracy of the system is improved by using a CNN. The driver's face is continuously captured using a camera.

1.3 SYSTEM REQIREMENTS

1.3.1 Software Reqirements:

- System type : intel®core™2 i5-5500UCPU@2.50gh
- Cache memory : 4 MB
- RAM : 8 GB
- Hard Disc : 8 GB

1.3.2 Hardware Reqirements:

- Operating system : windows 10
- Coding language:Python
- Python Distribution:Anaconda ,Flask

2.LITERATURE SURVEY

2.1 MACHINELEARNING:

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience withoutbeing explicitly programmed. Machine Learning is defined as the study of computer programs that leverage algorithms and statistical models to learn through inference and patterns without being explicitly programed. Machine Learning field has undergone significant developments in the last decade. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. There are also some types of machine learning algorithms that are used in very specific use-cases, but three main methods used today are: Supervised Machine Learning Algorithm Unsupervised Machine Learning Algorithm Reinforcement Machine Learning Algorithm Among these, the type of machine learning algorithm we used in our system is supervised machine learning algorithm.

1. **Supervised machine learning:**

This can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values.

2. **Unsupervised machine learning:**

It holds the advantage of being able to work with unlabeled data. This means that human labor is not required to make the dataset machine-readable, allowing much larger datasets to be worked on by the program. In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings. The creation of these hidden structures is what makes unsupervised learning algorithms versatile. Instead of a defined and set problem statement, unsupervised learning algorithms can adapt to the data by dynamically changing hidden structures. This offers more post deployment development than supervised learning algorithms.

3. **Reinforcement Machine Learning :**

It directly takes inspiration from how human beings learn from data in their lives. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favorable outputs are encouraged or ‘reinforced’, and nonfavorable outputs are discouraged or ‘punished’. Based onthe psychological concept of conditioning, reinforcement learning works by

putting the algorithm in a work environment with an interpreter and a reward system. In every iteration of the algorithm, the output result is given to the interpreter, which decides whether the outcome is favorable or not. In case of the program finding the correct solution, the interpreter reinforces the solution by providing a reward to the algorithm. If the outcome is not favorable, the algorithmis forced to reiterate until it finds a better result. In most cases, the reward systemis directly tied to the effectiveness of the result. In typical reinforcement learning use-cases, such as finding the shortest route between two points on a map, the solution is not an absolute value. Instead, it takes on a score of effectiveness, expressed in a percentage value. The higher this percentage value is, the more reward is given to the algorithm. Thus, the program is trained to give the best possible solution for the best possible reward.

2.2. DROWSINESS DETECTION THROUGH REGION OF INTEREST:

Region of interest (ROI) can detect a driver's face. As can be seen in the blue rectangle is the region of interest. The way to create an ROI area is to first obtain the green rectangle area from the Haar Cascade Classifier in the first frame, which includes height, width. Then, the rectangle is scaled up to create region of interest. Thereare several steps to calculate the ROI area and we have to calculate ROI for each and every region of interest

Fig 2.2 Drowsiness Detection through ROI

DISADVANTAGES OF REGION OF INTEREST:

1. It is uses extra frames or squares to detect face detection.

2. It can’t find in low light.

3. Why to use again region of interest while Haar cascade classifier can do the same process?

4. It can’t detect while using glasses in driving

2.3 EXISTING SYSTEM:

The driver behaviour is noticed in many conditions such as wearing spectacles and also in the dark condition inside the vehicle. The proposed system will be continuously monitoring the retina of the driver and all the monitored signals are sent to the microcontroller.

The system is capable of detecting the drowsiness condition within the duration of more than two seconds. After the detection of abnormal behaviour it is alerted to the driver through alarms and the parking lights will be on that will stop the vehicle which reduces the accidents due to drowsiness of the driver.

The existing system of driver drowsiness detection system has following disadvantages. Mainly, using of two cameras in the system one for monitoring the head movement and the other one for facial expressions[]. The other disadvantage is aging of sensors and all these sensors are attached to the driver's body which may affect the driver. So to overcome all these disadvantages we designed a system in which a live camera is used for monitoring the driver drowsiness condition and alert the driver which reduces the road accidents

2.4 PROPOSED SYSTEM:

This is the architecture for detecting the drowsiness of the driver. First of all the system captures images through the webcam and after capturing it detects the face through haar cascade algorithm. It uses haar features which can detect the face. If the system founds it as face the it will proceed for next phase i.e eye detection. The eye is also detected using haar cascade features and it is used for blink frequency

The state of eye will be detected using perclos algorithm. Through this algorithm we can find the percentage of time the eye lids remains closed. If it found eyes in closed state then it detects driver in drowsy state and alerts him by an alarm. In some cases distraction can be measured by 23 | P a g e continuous gazing. The drivers face is analysed continuously to detect any distraction. If found then alarm is activated by the system

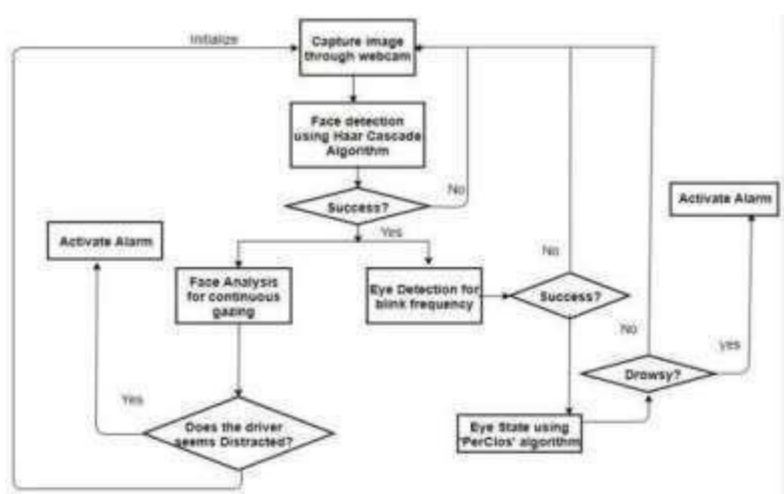


Fig 2.4 Proposed System

2.4.1 EYE TRACKING AND DYNAMIC TEMPLATE MATCHING:

To avoid road accidents, real time driver fatigue detection system based on vision is proposed. Firstly, system detects the face of driver from the input images using HSI color model. Secondly, Sobel edge operator is used to locate the eyes positions and gets the images of eye as the dynamic template for the tracking of eye. Then the obtained images are converted to HSI colour model to decide that whether the eyes are close or open to judge the drowsiness of driver. The experiments use four test videos for the tracking of eyes and face detection. The proposed system is compared with the labelled data which is annotated by the experts. The average correct rate of proposed system reaches up to 99.01 % and the precision to 88.90 %.

2.4.2 MOUTH AND YAWNING ANALYSIS :

Fatigue is the major reason for road accidents. To avoid the issue, Sarada Devi and Bajaj proposed the driver fatigue detection system based on mouth and yawning analysis. Firstly, the system locates and tracks the mouth of a driver using cascade of classifier training and mouth detection from the input images. Then, the images of mouth and yawning are trained using SVM. In the end, SVM is used to classify the regions of mouth to detects the yawning and alerts the fatigue. For experiment, authors collect some videos and select 20 yawning images and more than 100 normal videos as dataset. The results show that the proposed system gives better results as compared to the system using geometric features. The proposed system detects yawning, alerts the fatigue earlier and facilitates to make the driver safe

2.4.3 FACIAL EXPRESSIONS METHOD :

Laboratory condition using Finite Element Analysis is used by the researchers which is a complex system that contains the database of facial expression as a template and detect the drowsiness on the basis of results from database.

Rahmati present the hardware-based Driver Drowsiness Detection system based on facial expressions. The hardware system uses infrared light as it has giving many benefits like ease of use, independent of lightning conditions of environment. The system firstly uses the technique of background subtraction to determines the face region from the input images. Then using horizontal projection and template matching, facial expressions are obtained. After that in the tracking phase, elements found earlier are followed up using template matching and then investigates the incidence of sleepiness using the determination of facial states from the changes of the facial components. Changing in the three main elements such as eye brow rising, yawning and eye closure for the certain period are taken as the initial indications for drowsiness and the system generates the alert. The experiment is performed in the real driving scenario. For testing, images are acquired by the webcam under different conditions of lighting and from different people.The results investigate that the system produces appropriate response in the presence of beard or glasses and mustache on the face of driver

.

2.4.4 YAWNING EXTRACTION METHOD:

Fatigue or drowsiness is the major reason for road accidents. To prevent the issue, Alioua proposed the efficient system for monitoring the driver fatigue using Yawning extraction. Firstly, face region is obtained from the images using Support Vector Machine (SVM) technique to reduce the edge required cost. The proposed method

is used to localize the mouth, detection technique is used to detects facial edges, then

compute vertical projection on the lower half face to detect the right and left region boundaries and then compute the horizontal projection on the resulting region to detect the upper and lower limit of mouth 9 | P a g e and mouth localized region is obtained. Finally, to detect the yawning, Circular Hough Transform (CHT) is executed on the images of mouth region to identify the wide-open mouth. If the system finds notable number of continuous frames where the mouth is widely open, system generates the al ert. The results are compared with the other edge detectors like Sobel, Prewitt, Roberts, Canny. The experiment uses 6 videos representing real driving conditions and results are presented in the form of confusion matrix. The proposed method achieves 98% accuracy and outperforms all other edge detection techniques.

2.4.5 EYE CLOSURE AND HEAD POSTURES METHOD:

Teyeb proposed the Drowsy Driver Detection using Eye Closure and Head postures. Firstly, video is captured using webcam and for each frame of video, following operations are performed. To detect the ROI (face and eyes), viola-jones method is used. The face is partitioned in to three areas and the top one presenting the aye area is browsed by the Haar classifier. Then to detect the eye state, Wavelet Network based on neural network is used to train the images then the coefficients learning images is compared with the coefficients of the testing images and tells which class it belongs. When the closed eye is identified in the frames then the eye closure duration is calculated, if the value exceeds the predefined time then the drowsiness state is detected. Then the

developed system estimates the head movements which are: left, right, forward, backward inclination and left or right rotation. The captured video is segmented into frames and extract the images of head and determines the coordinates of image. Then theimages are compared to determine the inclined state of head and same case with other head postures. Finally, the system combines the eye closure duration and head posture estimation to measure the drowsiness. To evaluate the system, experiment is performed on 10 volunteers in various situations. And results show that the systems achieve the accuracy of 80%. 10. This module takes input from the camera and tries to detect a face in the video input. The detection of the face is achieved through the Haar classifiers mainly, the Frontal face cascade classifier. The face is detected in a rectangle format and converted to grayscale image and stored in the memory which can be used for training the model.

2.4.6 REALTIME ANALYSIS USING EYE AND YAWNING :

Kumar proposed the real time analysis of Driver Fatigue Detection using behavioral measures and gestures like eye blink, head movement and yawning to identify the drivers' state. The basic purpose of the proposed method is to detect the close eye and open mouth simultaneously and generates an alarm on positive detection. The system firstly captures the real time video using the camera mounted in front of the driver. Then the frames of captured video are used to detect the face and eyes by applying the viola-jones method, with the training set of face and eyes provided in OpenCV. Small rectangle is drawn around the center of eye and matrix is created that shows that the Region of Interest (ROI) that is eyes used in the next step. Since the both eyes blink at the same time that's why only the right eye is examined to detect the close eye state.

If the eye is closed for certain amount of time it will be considered as closed eye. To determine the eye state, firstly the eye ball color is acquired by sampling the RGB components on the center of eye pixel. Then the absolute thresholding is done on the eye ROI based on eye ball color and intensity map is obtained on Y-axis that show the distribution of pixels on y-axis which gives the height of eye ball and compared that value with threshold value which is 4 to distinguish the open and close eye

Then to detect the yawning motion of the mouth, contour finding algorithm is used to measure the size of mouth. If the height is greater than the certain threshold. It means person is taking yawning. To evaluate the performance of the proposed system, system has been measured under different conditions like persons with glasses, without glasses, with moustache and without moustache for 20 days in different timings. The system performs best when the drivers are without glasses.

2.4.7. EYE BLINK DETECTION METHOD:

Ahmad and Borolie proposed the Driver Drowsiness System based on non-intrusive machine-based concepts. The system consists of a web camera which is placed in front of the driver. Online videos as well as saved videos for simulation purposed are considered. Firstly, camera records the facial expressions and head movements of the driver. Then the video is converted into frames and each frame is processed one by one.

Face is detected from frames using Viola-jones algorithm. Region of interest on face is indicated by rectangles. Here the main attribute of detecting drowsiness is eyes blinking, varies from 12 to19 per minute normally and indicates the drowsiness if the frequency is less than the normal range. Instead of calculating eye blinking, average drowsiness is calculated. The detected eye is equivalent to zero (closed eye) and non-zero values are indicated as partially or fully open eyes. The equation (2) is used to calculate the average. %d = No. of closed eyes found no. of frames

------ Eq: (2) 12 | P a g e If the value is more than the set threshold value, then system generates the alarm to alert the driver. Moreover, yawning is also considered to generate the alert. Online and offline are videos are used for experiment which are performed on two different systems. The results show that the system achieves the efficiency up to 90%.

2.4.8. EYE CLOSENESS DETECTION METHOD :

Khunpisuth creates an experiment the calculates the drowsiness level of driver using Raspberry Pi camera and Raspberry Pi 3 model B. Firstly Pi camera captures video and to detect face regions in the images, Haar cascade classifier from Viola-Jones method is used. Several images are trained in different lights. The percentage of 83.09 % is achieved based on the case study with 10 volunteers. Blue rectangle shows the Region of Interest (ROI) that is face. Again, Haar cascade classifier is applied on the last obtained frame which reduces the size of ROI. After the face detection, drowsiness level is calculated using eye blink rate.

Eye region is detected using template matching on theface and authors uses three templates to check the eye blink and aye area. CV_TM_CCOEFF_NORMED from OpenCV is considered as it gives improved results than other methods of template matching. The integration of eyes and face detection permits the checking of an eye blinking and closeness rate face and authors uses three templates to check the eye blink and aye area. CV_TM_CCOEFF_NORMED from OpenCV is considered as it gives improved results than other methods of template matching. The integration of eyes and face detection permits the checking of an eye blinking and closeness rate. If the eyes are closed, then the value of closed eye is higher than the open the eyes and opposite case if eyes are open. Authors assumed that Haar cascade classifier will work if the face is front facing position. That why authors proposed the method to rotate the tilted face back in to the front-facing position. Firstly, determines whether the head is tilt or not then calculates the degrees of rotation (angle). After the accurate detection of face and eyes, drowsiness level of driver is determined. If the drivers blink eyes too frequently, he system indicates he drowsiness. When the level reaches to one hundred, a loud sound will be generated to alert the driver.when driver wake up to prevent the accidents

3.SYSTEM ANALYSIS

3.1 System Architecture

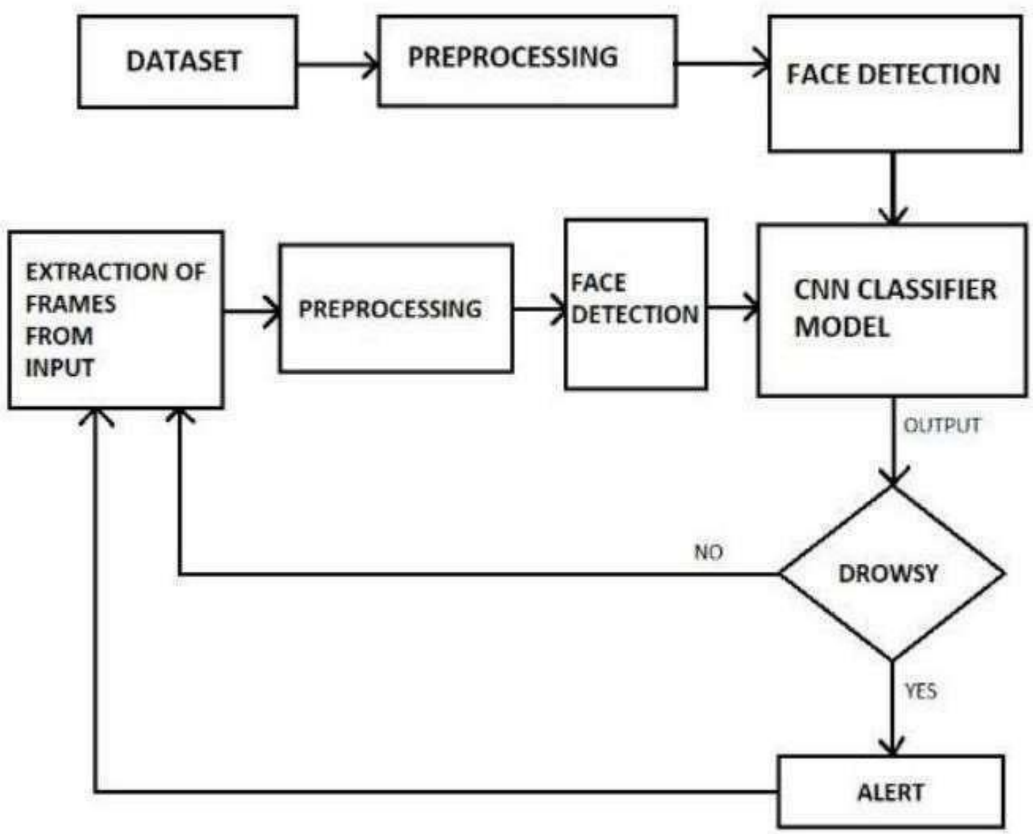


Fig 3.1 system architecture

3.2 MODULAR DIVISION:

The entire architecture is divided into 6modules.

1. Face Detection
2. Eye Detection
3. Face Tracking
4. Eye Tracking
5. Drowsiness Detection
6. Distraction Detection

1. Face Detection:

This module takes input from the camera and tries to detect a face in the video input. The detection of the face is achieved through the Haar classifiers mainly, the Frontal face cascade classifier. The face is detected in a rectangle format and converted to grayscale image and stored in the memory which can be used for training the model.

2. Eye Detection:

Since the model works on building a detection system for drowsiness we need to focus on the eyes to detect drowsiness. The eyes are detected through the video input by implementing a haar classifier namely Haar Cascade Eye Classifier. The eyes are detected in rectangular formats

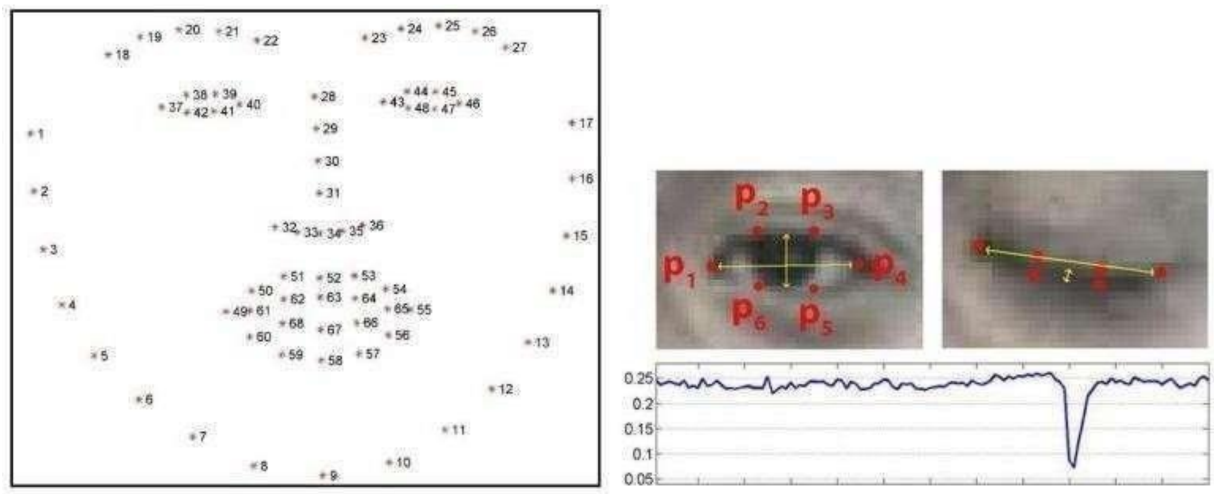


Fig 3.2 face and eye detection

3.Face Tracking:

Due to the real-time nature of the project, we need to track the faces continuously for any form of distraction. Hence the faces are continuously detected during the entire time

4.Eye Tracking:

The input to this module is taken from the previous module. The eyes state is determined through Perclos algorithm.

5.Drowsiness detection:

In the previous module the frequency is calculated and if it remains 0 for a longer period then the driver is alerted for the drowsiness through an alert from the system

6.Distraction detection:

In the face tracking module the face of the driver is continuously monitored for any frequent movements or the long gaze of the eyes without any blinks which can be treated as lack of concentration of the driver and is alerted by the system for distraction.

3.3 Implementation Of Machine Learning

3.3.1 Convolutional Neural Network:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with

enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area. A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights.

In other words, the network can be trained to understand the sophistication of the image better. The term "Convolution" in CNN denotes the mathematical function of convolution which is a special kind of linear operation whereintwo functions are multiplied to produce a third function which expresses how the shape of one function is modified by the other. In simple terms, two images which can be represented as matrices are multiplied to give an output that is used to extract features from the image. There are two main parts to CNN architecture: • A convolution tool thatseparates and identifies the various features of the image for analysis in a process calledas Feature Extraction • A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages. There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers

are stacked, CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

Convolutional Layer :

Convolutional layers are the major building blocks used in convolutional neural networks. This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

If the filter is designed to detect a specific type of feature in the input, then the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image. The output is termed as the Feature map which gives us information aboutthe image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

The Convolution Operation:

We analyze the influence of nearby pixels by using something called a filter. We move this across the image from top left to bottom right. When building the network, we

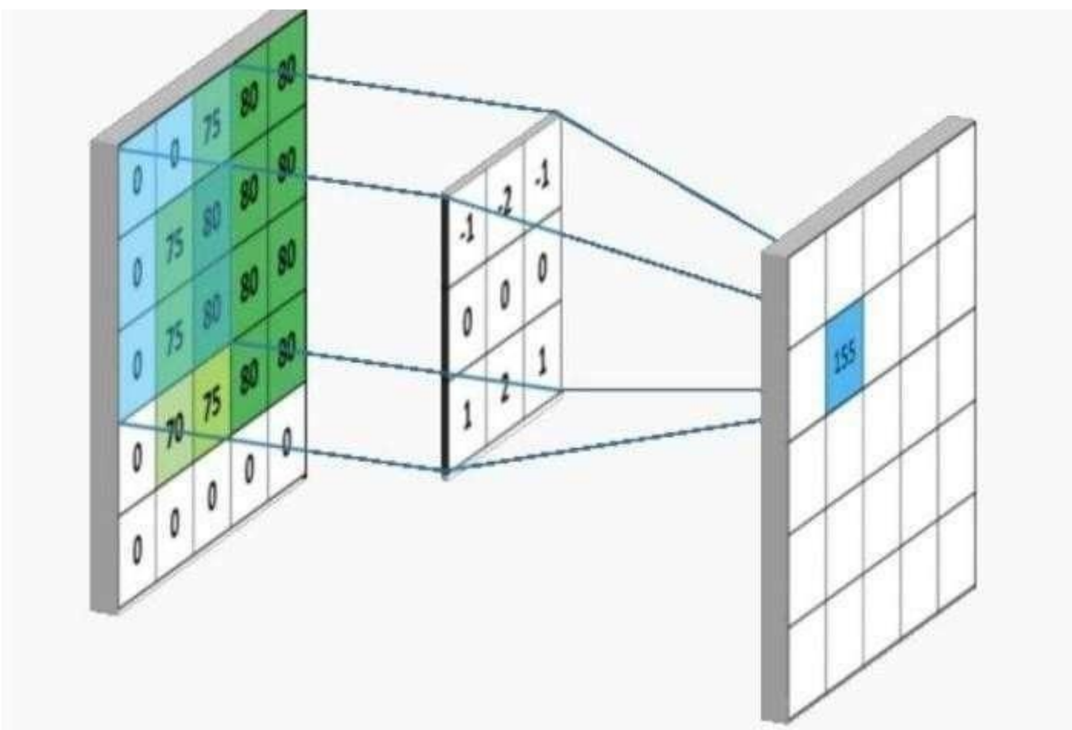
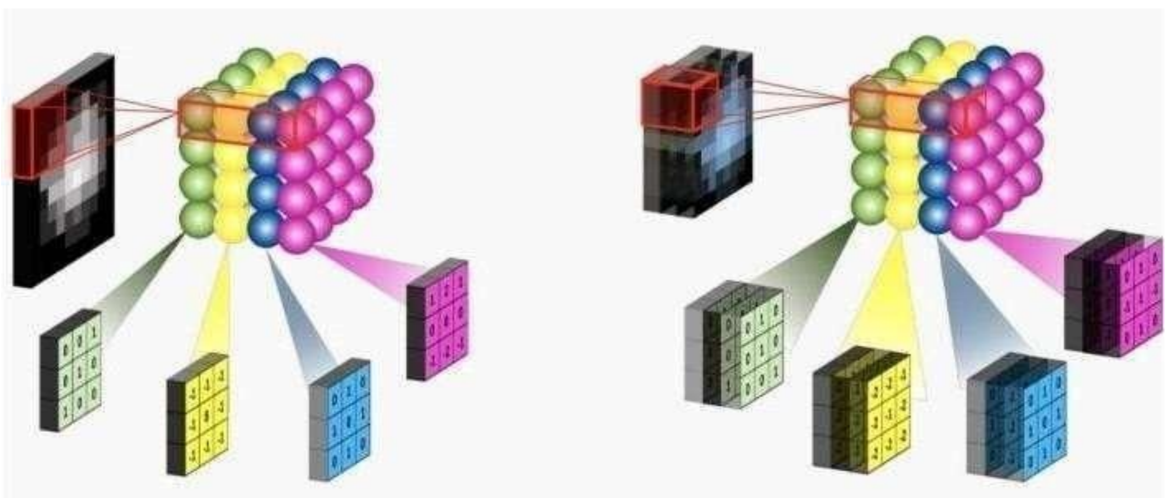


Fig 3.3.1.CNN

randomly specify values for the filters, which then continuously update themselves as the network is trained. After the filters have passed over the image, a feature map is generated for each filter. These are then taken through an activation function, which decides present at a given location in the image.

We have colour images which have 3 channels i.e., Red, Green and Blue. Each filter is convolved with the entirety of the 3D input cube but generates a 2D feature map. Be

I.Stride



Convolutional layer with four 3x3 filters on a black and white image (just one channel)

Convolutional layer with four 3x3 filters on an RGB image. As you can see, the filters are now cubes, and they are applied on the full depth of the image..

Stride is the number of pixels shifts over the input matrix. The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

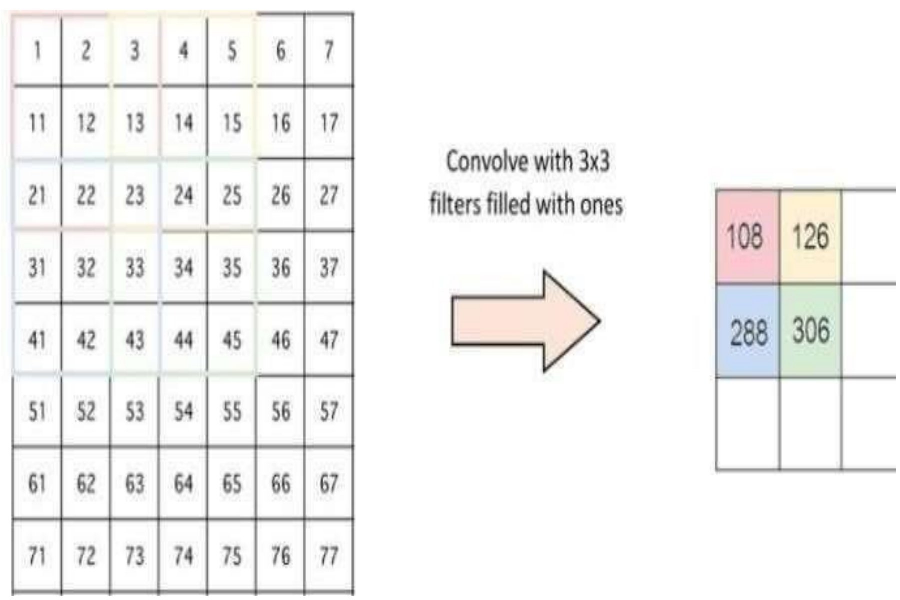


Fig 3.3.3 stride

II. Pooling Layer:

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used,there are several types of Pooling operations. In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

- **Max Pooling :**

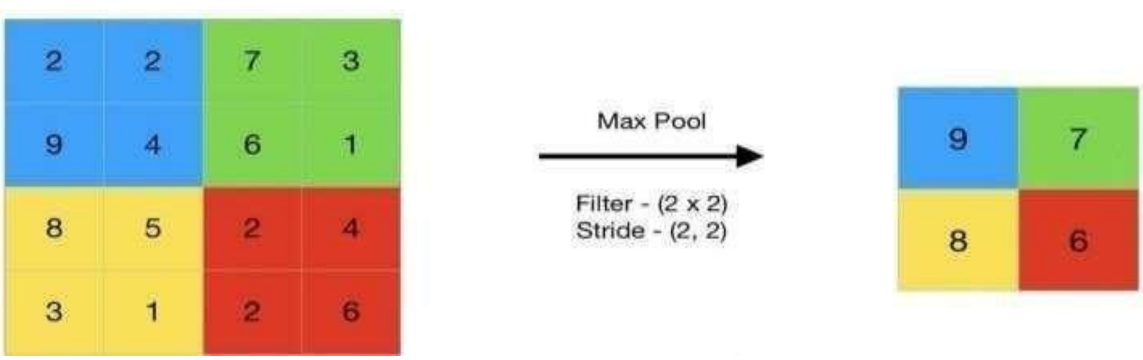


Fig 3.3.4 max polling

Max pooling is a pooling operation that selects the maximum elementfrom the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map A

- **Average Pooling:**

Average pooling computes the average of the elements present in the region of feature map covered by the filter. Thus, while max pooling gives the most prominent feature in a particular patch of the feature map, average pooling gives the average of features present in a patch.

- **Global Pooling :**

Global pooling reduces each channel in the feature map to a single value. Thus,an nh x nw x nc feature map is reduced to 1 x 1 x nc feature map. This is equivalent to using a filter of dimensions nh x nw i.e. the dimensions of the feature map.

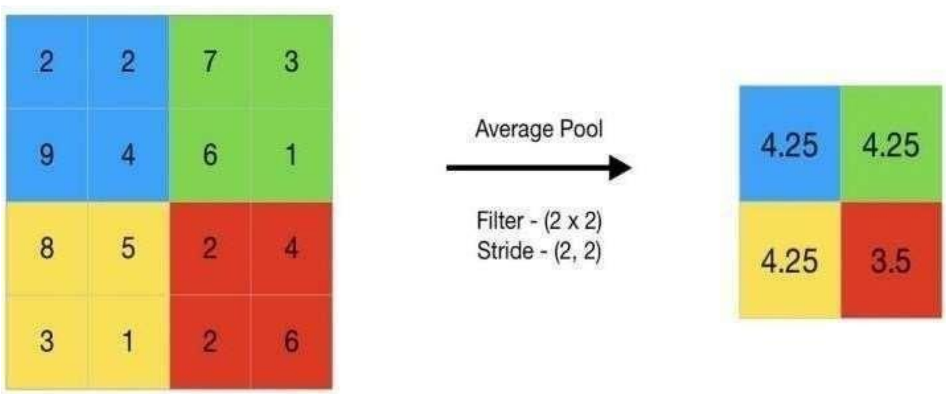


Fig. 3.3.7.5. Average Pooling

III Fully Connected layer: :

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture. In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place

IV Dropout:

When all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data. To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

3.4 LIBRARIES USED

3.4.1 OpenCV (Open Source Computer Vision Library) :

OpenCV is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

3.4.2 Numpy :

NumPy stands for Numerical Python. NumPy was created in 2005 by Travis Oliphant. NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the nd array object. It provides a lot of supporting functions that make working with nd array very easy. Arrays are very frequently used in data science, where speed and resources are very important. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

3.4.3 Tensor flow:

TensorFlow is an open-source software library. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

with TensorFlow accessible to most programmers. Google's Tensorflow engine has a unique way of solving problems This unique way allows for solving machine learning problems very efficiently. as the name indicates, is a framework to define and run computations involving tensors. A tensor is a generalization of vectors and matrices to potentially higher dimensions. Internally, TensorFlow represents tensors as n-dimensional arrays of base datatypes. Each element in the Tensor has the same data type, and the data type is always known. The shape (that is, the number of dimensions it has and the size of each dimension) might be only partially known. Most operations produce tensors of fullyknown shapes if the shapes of their inputs are also fully known, but in some cases it's only possible to find the shape of a tensor at graph execution time.

3.4.4 Keras: :

Keras is a python based open-source library used in deep learning (for neural networks). Keras was released in March 2015. It can run on top of TensorFlow, Microsoft CNTK or Theano. It is very simple to understand and use, and suitable for fast experimentation. It is designed to be fast and easy for the user to use. Keras models can be run both on CPU as well as GPU. Keras is the best platform out there to work on neural network models. The API that Keras has a user-friendly where a beginner can easily understand. Keras has the advantage that it can choose any libraries which support it for its backend support. Keras provides various pre-trained models which help the user in further improving the models the user is designing..

3.4.5 Classification :

Metrics The sk learn. metrics module implements several loss, score, and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values, or binary decisions values. Most implementations allow each sample to provide a weighted contribution to the overall score, through the sample_weight parameter. 35 The confusion_matrix function evaluates classification accuracy by computing the confusion matrix with each row corresponding to the true class (Wikipedia and other references may use different convention for axes). By definition, entry i,j in a confusion matrix is the number of observations actually in group i, but predicted to be in group j. Here is an example: Intuitively, precision is the ability of the classifier not to label as positive a sample that is negative, and recall is the ability of the classifier to find all the positive samples.

3.4 .6Play sound module in Python :

The play sound module is a cross platform module that can play audio files. This doesn’t have any dependencies, simply install with pip in your virtual environment and run. Implementation is different on platforms. It uses wind ll winm on Windows, App Kit. NS Sound on Apple OS X and G Streamer on Linux. The play sound module contains a function named play sound(). It works with both WAV and MP3 file

4. MODULES

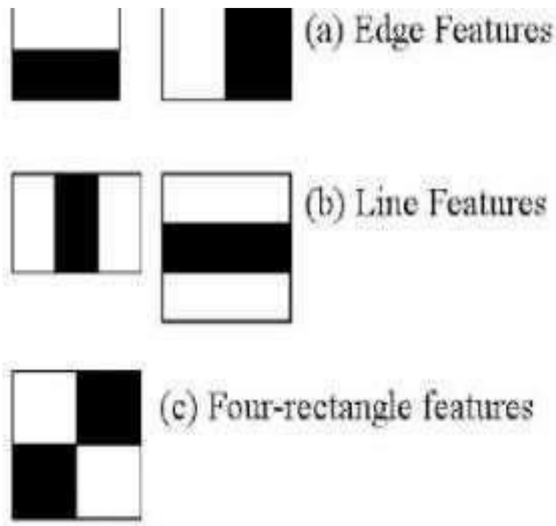
4.1 Collecting the Dataset:

The first step is collecting the dataset. To work with machine learning projects, we need a huge amount of data, because, without the data, one cannot train ML/AI models. Collecting and preparing the dataset is one of the most crucial parts while creating an ML/AI project. A dataset called Drowsiness_dataset from Kaggle which contains images with not yawning faces and yawning faces. This dataset contains images of many people captured while driving when they are yawning and not-yawning. This dataset also includes images with the people wearing spectacles. Another dataset from Github is considered in which the images of 3 people are recorded while driving. This dataset contains 3 sets of images which are alert images, images with closed eyes and images with yawning faces. The images of closed eyes and yawning faces are combined as drowsy images. We combined these images into 2 sets of images i.e., alert and drowsy.

4.2 Face Detection Module:

In this module, all the images in the dataset are preprocessed. We have the images for 2 classes (alert and drowsy) in two directories. We use the “os” module to list the directory names and list all the image files in each directory. Then, we can access all the

. image files in each directory We use OpenCV to read and resize the images. Instead of giving the entire image to classification model, only the face region is extracted and given to the model since the background and other portion of the image is unnecessary. For this, we use Face detection which isa computer vision technology that locates human faces in digital images. Haar Cascades Algorithm, also known Viola-Jones Algorithm is used for Face detection which was proposed by Paul Viola and Michael Jones in their 2001 paper. Working of the algorithm: This algorithm uses Haar features to extract objects.



The algorithm applies the features on windows of the image, gradually changing the window size after each turn. If a particular window is not classified as a face, that window is not processed in further steps. Instead of applying all the features, it makes use of cascade of classifiers concept. The features are grouped into different stages of classifiers and applied one-by-one. Only if a window passes a stage, the next stage is applied on that or else the window will be discarded and no longer considered as face region. A face region is the one which passes all the stages. The open CV detect multiscale() returns the starting coordinates, width and height of the detected face, using which the face region is extracted from the corresponding image and resized to a fixed size 100x100. We append all the resized images to a list and the corresponding image labels to another list. We convert both the lists to numpy arrays and we give these arrays to the Classification model. The parameters used in detect multiscale() method are image, scale Factor and min Neighbors. The scale Factor parameter specifies how much the image size is reduced at each image scale. The min Neighbor parameter specifies how many neighbors each candidate rectangle should have to retain in. This parameter affects the quality of the detected faces.

4.3 The Classification Module:

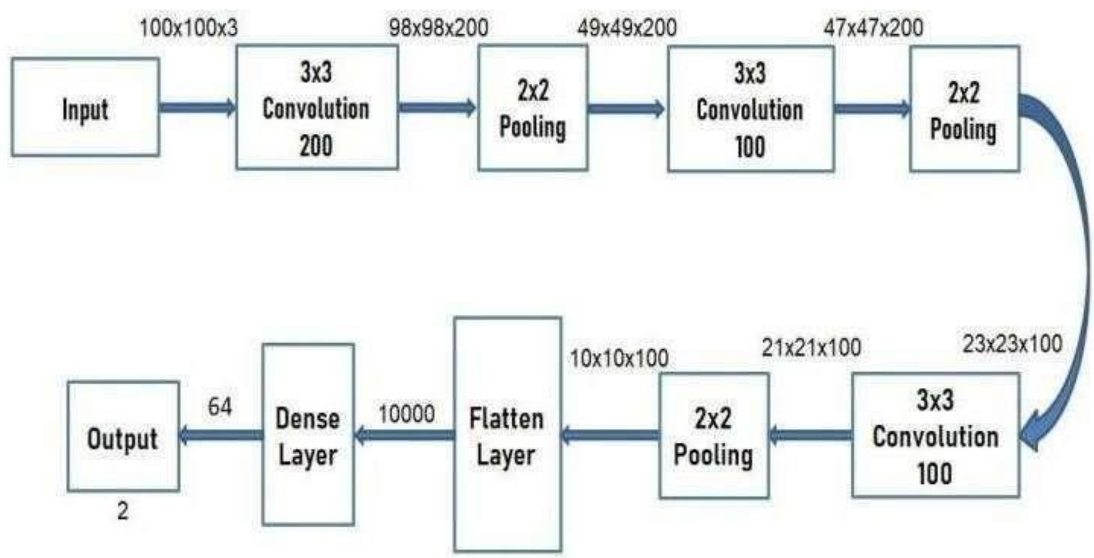
In this module, we build a Deep Learning Binary Classification model to classify the images as alert / drowsy. We used sequential model. In machine learning, classification refers to a predictive problem where a class label is predicted for a given example of input data. A model will use the training dataset and will calculate how to best map examples of input data to specific class labels.Binary classification refers to those classification tasks that have two class labels.

Convolutional Neural Networks, or CNNs, were designed to map image data to an output variable. The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image. A convolution is essentially sliding a filter over the input. Each convolutional layer contains a series of filters known as convolutional kernels. The filter is a matrix of integers that are used on a subset of the input pixel values. Each pixel is multiplied by the corresponding value in the kernel, then the result is summed up for a single value for simplicity representing a grid cell, like a pixel, in the output channel/feature map. 39 Fig. 6.3.1. CNN Architecture We define the size of the input for the input layer as 100x10

We define the size of the input for the input layer as 100x100. Weadd the following layers to the model: • A 2D convolutional layer with ReLU activation function. The hyper parameters for this layer are 200 (number of kernels) and (3x3) (each of size 3x3). The shape of each input image to this layer is (100x100) and the output shape is (98x98x200). Here, 200 are the number of kernels. • A Max Pooling layer of pool size (2x2). The output shape of this layer is(49x49x200). • A 2D convolutional layer with ReLU activation function. The hyper parameters for this layer are 100 (number of kernels) and (3x3) (each of size 3x3).

The output shape of this layer is (47x47x100). Here, 100 are the number of kernels. • A Max Pooling layer of pool size (2x2). The output shape of this layer is (23x23x100). • A 2D convolutional layer with ReLU activation function. The hyper parameters for this layer are 100 (number of kernels) and (3x3) (each of size 3x3). The output shape of this layer is (21x21x100). Here, 100 are the number of kernels. • A Max Pooling layer of pool size (2x2). The output shape of this layer is (10x10x100). • A flatten layer which consists of 10000 neurons. The shape of input to this layer is (10x10) with 100 kernels. So, the output from this layer has 10000 neurons. • A dense layer with 64 neurons with ReLU activation function. • The output layer with 2 neurons as this is a binary classification model. It has sigmoid activation function

fig classification model



4.4 Training:

We shuffle the data and split all the available images into train-data(80%) and test data(20%). We run 20 epochs during training. While training the model, some part of the training data is used as validation data. We use 10% of the training data for validation. The validation data is used to evaluate the performance of the model at the end of each epoch but not used to train the model. We use checkpoints in our model to save the state of the model each time an improvement is observed during training. The checkpoint stores the weights of the model. The weights will be updated after each improvement and the best model will be saved when the training is completed.

4.4.1 Evaluation:

After training the model, it's performance should be measured. There are different metrics to evaluate a Classification model like Loss, Accuracy, Precision and Recall. We use Matplotlib to plot the graphs of different metrics of the model

4.4.2 Testing with test data:

Testing is used to evaluate the generalizing ability of the model by giving unseen data to it. The data is previously split into training data and test data. The 20%of the data which is not known to the model is given to the model and the model will predict the classes for the test data. We evaluate the performance of the model by finding loss and accuracy with the test data. We also find the Precision and Recall values for the model

Predicting the images captured from the camera .After the model is trained with the given dataset, we can use the model to predict the class of the images which are captured from the camera. We use OpenCV to capture the images from the camera. We continuously capture image frames from the camera. The same preprocessing steps which are applied on the dataset are applied on each frame captured i.e., detecting the face from the image frame, extract the Region of Interest and then resizing the Region of Interest to a fixed size (100x100). Then we convert the images into array format to give as input to the model. Then, we can give a set of images to the trained CNN Classification model to predict the labels for the images. If the driver is feeling drowsy, a voice alert is generated. An audio file is played using the play sound module in python

5. IMPLEMENTATION CODE

```
import os
import shutil
import glob
from tqdm import tqdm
Raw_DIR= r'D:\Python37\Projects\iNeuron Intership Projects\CV_Driver_Drowsiness_Detection\MRL Eye Data\mrlEyes_2018_01'
for dirpath, dirname, filenames in os.walk(Raw_DIR):
   for i in tqdm([f for f in filenames if f.endswith('.png')]):if
      i.split('_')[4]=='0':
         shutil.copy(src=dirpath+'/'+i, dst=r'D:\Python37\Projects\iNeuron Intership Projects\CV_Driver_Drowsiness_Detection\MRL Eye Data\Prepared_Data\Open Eyes 1')

      elif i.split('_')[4]=='1':
!pip install tensorflow
import tensorflow as tf
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dropout,Input,Flatten,Dense,MaxPooling2D
from tensorflow.keras.optimizers import Adam
fromtensorflow.keras.preprocessing.imageimportImageDataGenerator
         shutil.copy(src=dirpath+'/'+i, dst=r'D:\Python37\Projects\iNeuron Intership
Projects\CV_Driver_Drowsiness_Detection\MRL Eye Data\Prepared_Data\Closed
Eyes 1')
from tensorflow.keras.preprocessing.image import ImageDataGeneratortrain_datagen=
ImageDataGenerator(rescale=1./255, rotation_range=0.2,shear_range=0.2,
   zoom_range=0.2,width_shift_range=0.2,
   height_shift_range=0.2, validation_split=0.2)
```

train_data= train_datagen.flow_from_directory(r'D:\Python37\Projects\iNeuron Intership Projects\CV_Driver_Drowsiness_Detection\MRL Eye Data\Prepared_Data\train',

target_size=(80,80),batch_size=batchsize,class_mode='categorical',subset='training')

validation_data= train_datagen.flow_from_directory(r'D:\Python37\Projects\iNeuron Intership Projects\CV_Driver_Drowsiness_Detection\MRL Eye Data\Prepared_Data\train',
target_size=(80,80),batch_size=batchsize,class_mode='categorical', subset='validation')

```
bmodel=InceptionV3(include_top=False,weights='imagenet',
input_tensor=Input(shape=(80,80,3)))
hmodel = bmodel.output
hmodel = Flatten()(hmodel)
hmodel = Dense(64, activation='relu')(hmodel)
hmodel = Dropout(0.5)(hmodel)
hmodel = Dense(2,activation= 'softmax')(hmodel)
model = Model(inputs=bmodel.input, outputs= hmodel)
for layer in bmodel.layers:
   layer.trainable = False
from tensorflow.keras.models import Model
checkpoint = ModelCheckpoint(r'D:\Python37\Projects\iNeuron Intership Projects\CV_Driver_Drowsiness_Detection\models\model.h5',
                      monitor='val_loss',save_best_only=True,verbose=3)

earlystop = EarlyStopping(monitor = 'val_loss', patience=7, verbose= 3, restore_best_weights=True)

learning_rate = ReduceLROnPlateau(monitor= 'val_loss', patience=3, verbose= 3, )

callbacks=[checkpoint,earlystop,learning_rate]
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 80, 80, 3)]  0           []

 conv2d (Conv2D)                (None, 39, 39, 32)   864         ['input_1[0][0]']

 batch_normalization (BatchNorm  (None, 39, 39, 32)  96          ['conv2d[0][0]']
 alization)

 activation (Activation)        (None, 39, 39, 32)   0           ['batch_normalization[0][0]']

 conv2d_1 (Conv2D)              (None, 37, 37, 32)   9216        ['activation[0][0]']

 batch_normalization_1 (BatchNo  (None, 37, 37, 32)  96          ['conv2d_1[0][0]']
 rmalization)

 activation_1 (Activation)      (None, 37, 37, 32)   0           ['batch_normalization_1[0][0]']
```

```
!pip install opencv-python
!pip install pygame

import cv2
import tensorflow as tf
from tensorflow.keras.models import load_model
import numpy as np
from pygame import mixer
face_cascade=cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
```

```
acc_vr, loss_vr = model.evaluate_generator(validation_data)
print(acc_vr)
print(loss_vr)
```

```
C:\Users\papab\AppData\Local\Temp\ipykernel_24976\408175674
ll be removed in a future version. Please use `Model.evalua
  acc_vr, loss_vr = model.evaluate_generator(validation_dat
```

```
0.30093714594841003
0.8691250681877136
```

```
acc_test, loss_test = model.evaluate_generator(test_data)
print(acc_tr)
print(loss_tr)
```

```
C:\Users\papab\AppData\Local\Temp\ipykernel_24976\365547188
ll be removed in a future version. Please use `Model.evalua
  acc_test, loss_test = model.evaluate_generator(test_data)
```

```
0.07290811091661453
0.9712789058685303
```

```
while True:
   ret,      frame      =      cap.read()
   height,width = frame.shape[0:2]
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   faces= face_cascade.detectMultiScale(gray, scaleFactor= 1.2, minNeighbors=3)
   eyes= eye_cascade.detectMultiScale(gray, scaleFactor= 1.1, minNeighbors=1)
   cv2.rectangle(frame, (0,height-50),(200,height),(0,0,0),thickness=cv2.FILLED)
        for (x,y,w,h) in faces:
      cv2.rectangle(frame,pt1=(x,y),pt2=(x+w,y+h),  color=  (255,0,0),  thickness=3  )for
      (ex,ey,ew,eh) in eyes:
      #cv2.rectangle(frame,pt1=(ex,ey),pt2=(ex+ew,ey+eh),          color=          (255,0,0),
thickness=3 )
      # preprocessing steps
      eye= frame[ey:ey+eh,ex:ex+ew]
      eye=       cv2.resize(eye,(80,80))
      eye= eye/255
      eye= eye.reshape(80,80,3)
      eye= np.expand_dims(eye,axis=0)
      # preprocessing is done now model prediction
      prediction = model.predict(eye)


      # if eyes are closed
      ifprediction[0][0]>0.30:
         cv2.putText(frame,'closed',(10,height-
```

```
20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,fontScale=1,color=(255,2
55,255),

                  thickness=1,lineType=cv2.LINE_AA)
         cv2.putText(frame,'Score'+str(Score),(100,height-
20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,fontScale=1,color=(255,2
55,255),
                  thickness=1,lineType=cv2.LINE_AA)
         Score=Score+1
         if(Score>5):
            try:
               sound.play()
            except:
               pass


      # if eyes are open
      elif prediction[0][1]>0.90:
         cv2.putText(frame,'open',(10,height-
20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,fontScale=1,color=(255,2
55,255),
                  thickness=1,lineType=cv2.LINE_AA)
         cv2.putText(frame,'Score'+str(Score),(100,height-
20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,fontScale=1,color=(255,2
```

```
55,255),
                    thickness=1,lineType=cv2.LINE_AA)
          Score = Score-1
          if (Score<0):
             Score=0
  cv2.imshow('frame',frame)
    if  cv2.waitKey(33)  &  0xFF==ord('q'):
       break


cap.release()
cv2.destroyAllWindows()
```

6. OUTPUT:

6.1 Open Eye

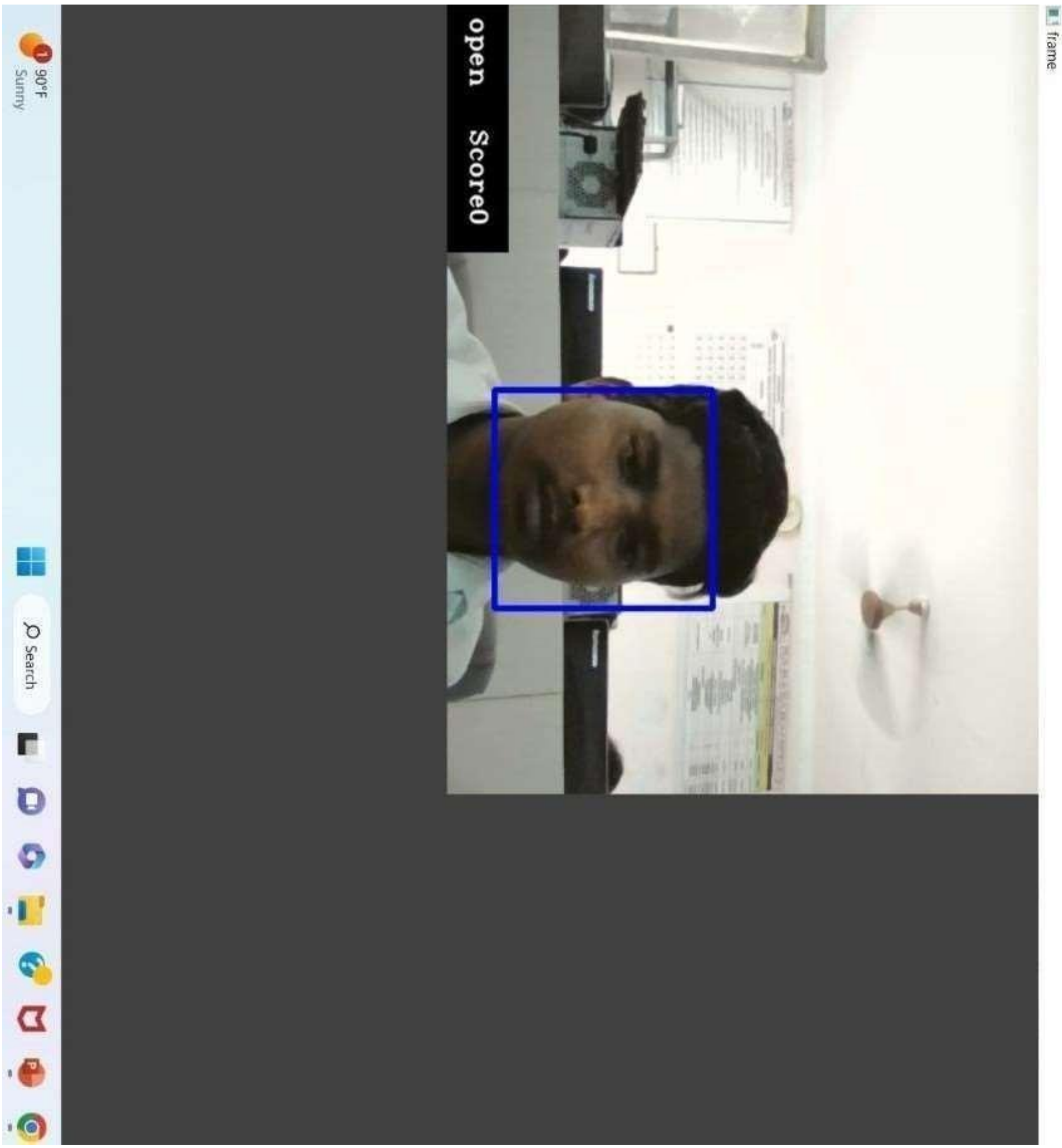


Fig 6.1 OPEN EYE

6.2 Closed Eye

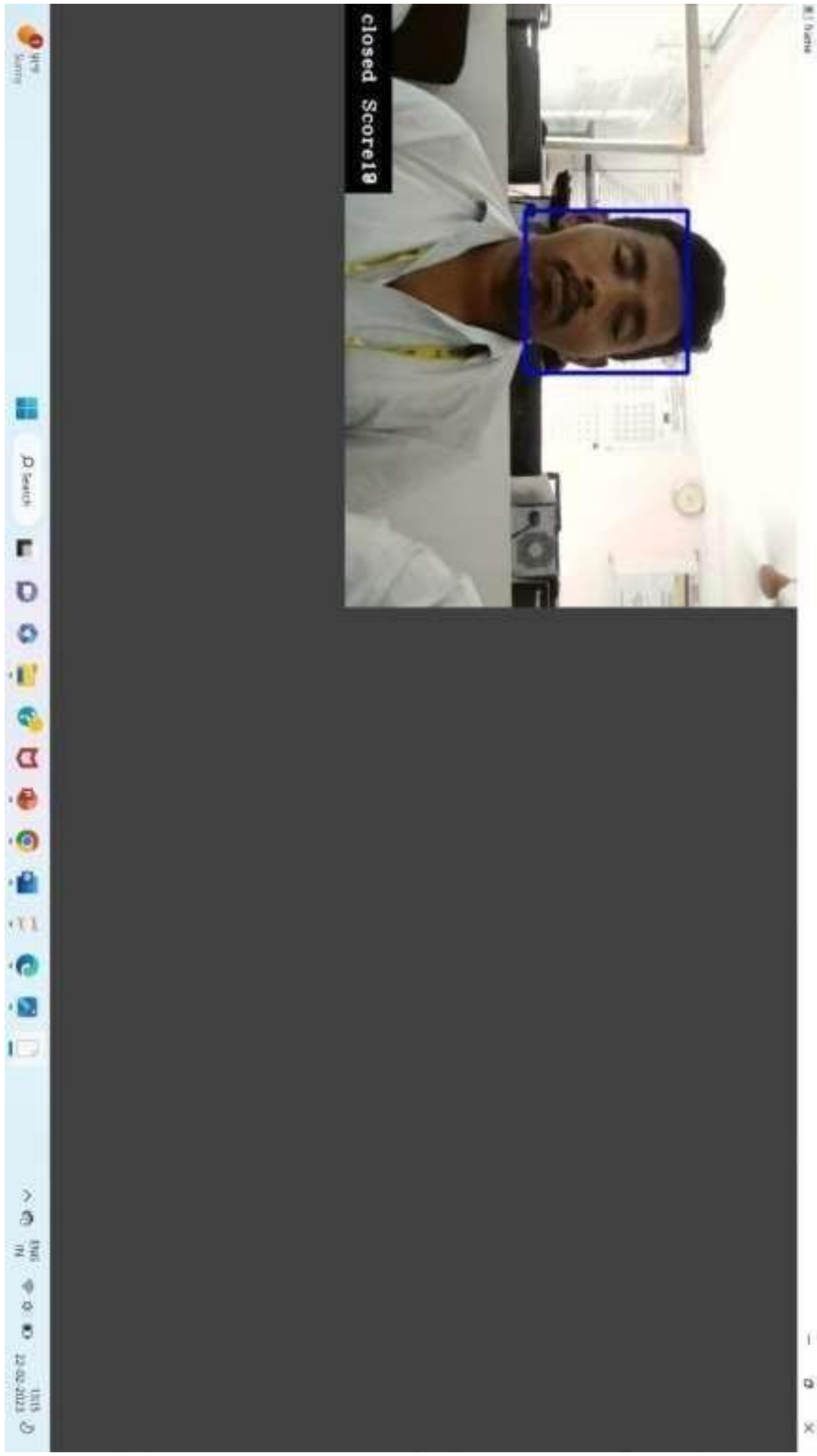


Fig 6.2:Closed eye

7.CONCLUSION AND FUTUREWORK

CONCLUSION:

The current study developed an automated system for detecting drowsiness of the driver. The continuous video stream is read from the system and is used for detecting the drowsiness. It is detected by using haar cascade algorithm. The haar cascade algorithm uses haar features to detect face and eyes. Haar features are predefined are used for detecting different things. The haar features are applied on the image and blink frequency is calculated using perclos algorithm. If the value remains 0 for some amount of time then it detects as sleepy and alerts driver by activating an alarm. If the value remains constant for longer periods then the driver is said to be distracted then also an alarm is activated.

- **FUTURE WORK:**

The work can be extended by extracting the features of mouth where the driver can be detected as drowsy through yawning. If the driver yawns repeatedly for more number of times then we can say that he is in sleepy mode. If the number exceeds a limit then we can alert the driver. This work can also be extended by implementing in full night light using IR web cam.

8. BIOLIOGRAPHY

1. Rosebrock, A. (2017, May 8). Retrieved from Pyimagesearch: https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/

[2] Awais M, Badruddin N, Drieberg M. A Hybrid Approach to Detect Driver Drowsiness

Utilizing Physiological Signals to Improve System Performance and Wearability.

Sensors (Basel). 2017 Aug 31;17(9):1991. doi: 10.3390/s17091991. PMID: 28858220;

PMCID: PMC5620623. https://pubmed.ncbi.nlm.nih.gov/28858220/

[3] Arefnezhad, S., Samiee, S., Eichberger, A., & Nahvi, A. (2019). Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection. Sensors (Basel, Switzerland), 19(4), 943. https://doi.org/10.3390/s19040943

[4] B. K. Savaş and Y. Becerikli, "Real Time Driver Fatigue Detection Based on SVM Algorithm," 2018 6th International Conference on Control Engineering & Information Technology (CEIT), 2018, pp. 1-4, doi: 10.1109/CEIT.2018.8751886. https://ieeexplore.ieee.org/abstract/document/8751886/

[5] Vesselenyi, T., Moca, S., Rus, A., Mitran, T., & Tătaru, B. (2017, October). Driver drowsiness detection using ANN image processing. In IOP Conference Series:

Materials Science and Engineering (Vol. 252, No. 1, p. 012097). IOP Publishing. https://iopscience.iop.org/article/10.1088/1757-899X/252/1/012097/meta

[6] Paul Viola and Michael J. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, 2004.

[7] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[8] Chirra, V. R. R., ReddyUyyala, S., & Kolli, V. K. K. (2019). Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State. Revue d'Intelligence Artificielle, 33(6), 461-466

Driver Drowsiness Detection using MachineLearning

Narenda P [1], Sk.Basha[2], M.Gopichand[3],Dr S.V.N.Sreenivasu[4]

[1,2,3]Student, Dept. of CSE, Narasaraopeta Engineering College(Autonomous),Narasaraopet

[4]Professor, Dept. of CSE, Narasaraopeta Engineering College(Autonomous),Narasaraopet

jessyraj999@gmail.com[1], fardhindasha786@gmail.com[2] ,marlapatigopichand4065@gmail.com[3], drsvnsrinivasu@gmail.com[4]

Abstract *- Driver drowsiness detection is a critical area of research in the field of transportation safety. Drowsy driving can cause accidents and fatalities, particularly among drivers of commercial vehicles who often have to drive long distances without a break. To address this problem, researchers have developed various technologies and techniques for detecting signs of drowsiness in drivers.*

These technologies typically rely on a combination of sensors and algorithms to monitor the driver's behavior and determine if they are becoming drowsy. Some of the most common indicators of drowsiness that these systems monitor include eye movement, head movement, and steering behavior.

If the system detects signs of drowsiness, it can alert the driver through visual, auditory, or physical cues. For example, the system may emit a warning sound or vibration, prompting the driver to take a break, stretch their legs, or get some rest before continuing their journey.

.

I. INTRODUCTION

.

Driver tiredness detection is the process of using technology to detect signs of fatigue or drowsiness in drivers. Drowsy driving is a major problem on the roads, as it can lead to accidents and injuries. This is particularly true for drivers of commercial vehicles, such as long-haul truck drivers, who often have to drive for long periods of time without a break.

Drowsiness detection systems use various sensors and algorithms to monitor the driver's behavior and determine if they are becoming drowsy. Some of the most common indicators of drowsiness that these systems monitor include eye movement, head movement ,and steering behavior

If the system detects signs of drowsiness, it can alert the driver through visual, auditory, or physical cues, such as a warning sound or vibration. This can give the driver a chance to take a break, stretch their legs, or get some rest before continuing their journey.

Overall, driver drowsiness detection technology has the potential to improve road safety by reducing the number of accidents caused by drowsy driving

2. Related Work

Driver drowsiness detection has been an active area of research for many years, and there have been numerous studies and developments in this field. Here are some examples of related work:

0.Sensor-based systems: Various sensor-based systems have been developed for detecting drowsiness in drivers. These include systems that use eye-tracking technology, steering wheel sensors, and facial recognition software. One example is the eye-tracking system developed by Bosch, which uses infrared cameras to monitor the driver's eye movements and detect signs of drowsiness.

0. Machine learning algorithms: Machine learning algorithms have been used to analyze driver behavior and detect signs of drowsiness. These algorithms can analyze data from sensors and other sources, such as GPS and accelerometer data, to identify patterns and predict when a driver is likely to become drowsy.

1. Wearable devices: Wearable devices, such as smartwatches and fitness trackers, can also be used to monitor the driver's vital signs and detect signs of drowsiness. For example, the Garmin dēzl™ OTR500 trucking navigator includes a driver fatigue monitoring system that uses a wearable device to track the driver's heart rate and alertness.

2. Real-time alert systems: Real-time alert systems can provide warnings to drivers when signs of drowsiness are detected. These systems can include visual, auditory, or physical alerts, such as a vibration in the seat or a warning sound.

3. Driving simulation studies: Driving simulation studies have been conducted to test the effectiveness of driver sleepiness discovery organizations. These studies can help researchers and developers refine their algorithms and improve the precision of their systems.

Overall, the related work in driver drowsiness detection has resulted in many promising developments and technologies, which have the potential to significantly improve road safety and prevent accidents caused by drowsy driving.

3. The Proposed System:

A proposed driver drowsiness detection system might consist of several components, including sensors, data analysis algorithms, and real-time alert systems. Here is an indication of how such a system might work:

1. Sensors: The system would use a combination of sensors to screen the driver's performance and detect signs of drowsiness. These sensors could include a camera to monitor the driver's facial expressions and eye movements, as well as a steering wheel sensor to detect any erratic driving behavior.

2. Data analysis: The data collected by the sensors would be analyzed by an algorithm to identify patterns and signs of drowsiness. Machine learning algorithms could be used to analyze the data and predict when the driver is becoming drowsy.

3. Real-time alerts: If the system detects signs of drowsiness, it could alert the driver through real-time alerts. These alerts could include an auditory alert, such as a warning sound or voice prompt, or a visual alert, such as a flashing light on the dashboard. Additionally, the system could provide a physical alert, such as a vibration in the driver's seat

4. Driver feedback: The system could also include a feedback mechanism to inform the driver of their driving behavior and alertness levels. This feedback could be displayed on a dashboard screen or sent to the driver's smartphone.

5. Data logging: The classification could also log data on the driver's behavior and alertness levels. This data could be used to analyze the effectiveness of the system and identify areas for improvement.

Overall, a driver drowsiness detection system could help prevent accidents caused by drowsy driving by alerting the driver when they are becoming drowsy and encouraging them to take a break or rest before continuing their journey.

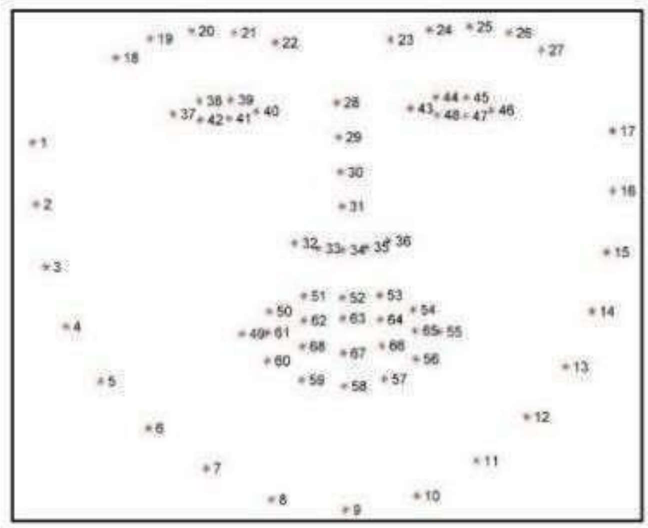
3.1 Face Detection and Facial Land Marking:

Face detection and facial landmarking are two techniques that are often used in driver drowsiness detection systems to monitor the driver's behavior and detect signs of drowsiness.

1. .Face detection is a computer vision technique used to identify the presence and location of human faces in an image or video. It works by analyzing an image or video frame for patterns that match a human face, such as the presence of two eyes, a nose, and a mouth. Once a face is detected, the system can track it over time and use other techniques, such as facial landmarking, to extract more detailed information about the face.

2. Facial landmarking is a technique used to identify specific features on a face, such as the position of the eyes, nose, mouth, and eyebrows. This information can be used to determine the orientation and expression of the face, and to track changes in the face over time. Facial landmarking typically involves the use of machine learning algorithms and deep neural networks to identify and track the features of the face.

In driver drowsiness detection systems, face detection and facial landmarking are often used together to monitor the driver's behavior and detect signs of drowsiness. For example, the system might use face detection to identify the driver's face, and then use facial landmarking to track the position and movement of their eyes, mouth, and other facial features. By analyzing changes in these features over time, the system can detect signs of drowsiness, such as drooping eyelids or a slack jaw, and provide alerts to the driver to take a break or rest before continuing their journey.



3.2 Yawning Detection:

Yawning detection is a technique used in driver drowsiness detection systems to monitor the driver's behavior and detect signs of drowsiness. Yawning is a common physiological response to fatigue and drowsiness, and can be a useful indicator of the driver's level of alertness.

Another approach to yawning detection is to use sensors to monitor the driver's physiological responses, such as changes in heart rate, breathing rate, and muscle activity. These sensors can be placed on the driver's body, such as on the chest or wrist, and can be used to detect subtle changes in their physiological responses that are indicative of drowsiness and yawning.

Once yawning is detected, the driver drowsiness detection system can provide an alert to the driver to take a break or rest before continuing their journey. This alert can be provided through auditory or visual cues, such as a warning sound or a flashing light on the dashboard.

Overall, yawning detection is a useful technique for driver drowsiness detection, and can be used in grouping with other techniques such as eye trailing and steering wheel monitoring to provide a comprehensive picture of the driver's behavior and level of alertness.

4. Result And Discussion:

The outcomes of a driver drowsiness detection system depend on the specific techniques and algorithms used ,as well as the quality and quantity of data available for training and testing the system. In general, however, driver drowsiness detection systems have shown promising results in improving road safety by detecting signs of drowsiness and alerting drivers to take a break or rest before continuing their journey.

One common evaluation metric used for driver drowsiness detection systems is the accuracy of the system in detecting drowsiness. This can be measured by comparing the system's output to a ground truth label, such as a manual observation of the driver's behavior or physiological state. Studies have reported high accuracy rates for driver drowsiness detection systems, with some systems achieving accuracy rates of over 90%

Another important factor to consider in the evaluation of driver drowsiness detection systems is the response time of the system. The response time refers to the time it takes for the system to detect signs of drowsiness and provide an alert to the driver. A fast response time is important to ensure that the driver can take action to prevent an accident before it occurs. Studies have reported response times of less than one second for some driver drowsiness detection systems.

One challenge in the evaluation of driver drowsiness detection systems is the variability in driver behavior and the effectiveness of the system in detecting drowsiness in different contexts. For example, the system may perform well in detecting drowsiness during daytime driving on highways, but may not be as effective in detecting drowsiness during nighttime driving or on winding roads. Further research is needed to evaluate the effectiveness of driver drowsiness detection systems in different driving contexts and under different conditions.

Overall, driver drowsiness detection systems have shown promising results in improving road safety by detecting signs of drowsiness and alerting drivers to takea break or rest before continuing their journey. However, further research is needed to optimize these systems and ensure their effectiveness in a range of driving contexts and conditions.

Fig :Closed Eye

Fig: open eye

5. Conclusion:

In conclusion, driver drowsiness detection systems are an important area of research and development in the field of road safety. Drowsy driving is a major cause of accidents on the road, and driver drowsiness detection systems have the potential to significantly reduce the incidence of these accidents.

There are a variety of techniques and algorithms used in driver drowsiness detection systems, including facial recognition, eye tracking, and yawning detection. These techniques have shown promising results in detecting signs of drowsiness and alerting drivers to take a break or rest before continuing their journey.

However, there are still challenges to be addressed in the development and implementation of driver drowsiness detection systems. These challenges include variability in driver behavior effectiveness of the system in different driving contexts, as well as issues related to privacy and data security.

6. Reference:

[1] Kyong Hee Lee, Whui Kim, Hyun Kyun Choi, Byung Tae Jang, "A Study on Feature Extraction Methods Used to Estimate a Driver's Level of Drowsiness", International Conference on Advanced Communications Technology (ICACT), 2019.

[2] Ashish Kumar, Rusha Patra, “Driver Drowsiness Monitoring System using Visual Behaviour and Machine Learning”, IEEE Conference, 2018.

[3] Cyun-Yi Lin, Paul Chang, Alan Wang, Chih- Peng Fan, “Machine Learning and Gradient Statistics Based Real-Time Driver

Drowsiness Detection”, 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE- TW).

[4] Ching-Hua Weng, Ying-Hsiu Lai, Shang-Hong Lai, “Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief

Network”, In Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video,

Taipei, Taiwan,Nov. 2016

[5] Anjith George, Aurobinda Routray, “Real-time Eye Gaze Direction Classification Using Convolutional Neural Network”, IEEE,2016.

[6] Media Research Lab, http://mrl.cs.vsb.cz/, 2018.

[7] Venkata Rami Reddy Chirra, Srinivasulu Reddy Uyyala, Venkata Krishna Kishore Kolli, “Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State”, Revue d'Intelligence Artificielle Vol. 33,No. 6, December,2019, pp. 461-466.

[8] Mkhuseli Ngxande, Jules-Raymond Tapamo, Michael Burke, “Driver drowsiness detection using Behavioral measures and machine learning techniques: A review of state-of- art techniques”, 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference(PRASA-RobMech).

[9] T. Vogelpohla, M. Kühn, T. Hummel, M.Vollrath, “Asleep at the automated wheel— Sleepiness and fatigue during highly automated driving”, https://doi.org/10.1016/j.aap.2018.03.013, Accident Analysis and Prevention, 2018.

[10] Xuan-Phung Huynh, “Detection of driver drowsiness using 3D deep neural network and semi-supervised

AB6

by Vamshi krishna Namani

Submission date: 09-Mar-2023 03:36PM (UTC+1000)

Submission ID: 2032768735

File name: HELLO.docx (171.53K)

Word count: 2133

Character count: 11809

Driver Drowsiness Detection using MachineLearning

Abstract *- Driver drowsiness detection is a critical area of research in the field of transportation safety. Drowsy driving can cause accidents and fatalities, particularly among drivers of commercial vehicles who often have to drive long distances without a break. To address this problem, researchers have developed various technologies and techniques for detecting signs of drowsiness in drivers.*

These technologies typically rely on a combination of sensors and algorithms to monitor the driver's behavior and determine if they are becoming drowsy. Some of the most common indicators of drowsiness that these systems monitor include eye movement, head movement, and steering behavior.

If the system detects signs of drowsiness, it can alert the driver through visual, auditory, or physical cues. For example, the system may emit a warning sound or vibration, prompting the driver to take a break, stretch their legs, or get some rest before continuing their journey.

I. INTRODUCTION

Driver tiredness detection is the process of using technology to detect signs of fatigue or drowsiness in drivers. Drowsy driving is a major problem on the roads, as it can lead to accidents and injuries. This is particularly true for drivers of commercial vehicles, such as long-haul truck drivers, who often have to drive for long periods of time without a break.

Drowsiness detection systems use various sensors and algorithms to monitor the driver's behavior and determine if they are becoming drowsy. Some of the most common indicators of drowsiness that these systems monitor include eye movement, head movement, and steering behavior

If the system detects signs of drowsiness, it can alert the driver through visual, auditory, or physical cues, such as a warning sound or vibration. This can give the driver a chance to take a break, stretch their legs, or get some rest before continuing their journey.

Overall, driver drowsiness detection technology has the potential to improve road safety by reducing the number of accidents caused by drowsy driving

2. Related Work:

Driver drowsiness detection has been an active area of research for many years, and there have been numerous studies and developments in this field. Here are some examples of related work:

1. Sensor-based systems: Various sensor-based systems have been developed for detecting drowsiness in drivers. These include systems that use eye-tracking technology, steering wheel sensors, and facial recognition software. One example is the eye-tracking system developed by Bosch, which uses infrared cameras to monitor the driver's eye movements and detect signs of drowsiness.

2. Machine learning algorithms: Machine learning algorithms have been used to analyze driver behavior and detect signs of drowsiness. These algorithms can analyze data from sensors and other sources, such as GPS and accelerometer data, to identify patterns and predict when a driver is likely to become drowsy.

3. Wearable devices: Wearable devices, such as smartwatches and fitness trackers, can also be used to monitor the driver's vital signs and detect signs of drowsiness. For example, the Garmin dēzl™ OTR500 trucking navigator includes a driver fatigue monitoring system that uses a wearable device to track the driver's heart rate and alertness.

4. Real-time alert systems: Real-time alert systems can provide warnings to drivers when signs of drowsiness are detected. These systems can include visual, auditory, or physical alerts, such as a vibration in the seat or a warning sound.

5. Driving simulation studies: Driving simulation studies have been conducted to test the effectiveness of driver sleepiness discovery organizations. These studies can help researchers and developers refine their algorithms and improve the precision oftheir systems.

Overall, the related work in driver drowsiness detection has resulted in many promising developments and technologies, which have the potential to significantly improve road safety and prevent accidents caused by drowsy driving.

3.The Proposed System:

A proposed driver drowsiness detection system might consist of several components, including sensors, data analysis algorithms, and real-time alert systems. Here is an indication of how such a system might work:

1. Sensors: The system would use a combination of sensors to screen the driver's performance and detect signs of drowsiness. These sensors could include a camera to monitor the driver's facial expressions and eye movements, as well as a steering wheel sensor to detect any erratic driving behavior.

2.Data analysis: The data collected by the sensors would be analyzed by an algorithm to identify patterns and signs of drowsiness. Machine learning algorithms could be used to analyze the data and predict when the driver is becoming drowsy.

3.Real-time alerts: If the system detects signs of drowsiness, it could alert the driver through real-time alerts. These alerts could include an auditory alert, such as a warning sound or voice prompt, or a visual alert, such as a flashing light on the dashboard. Additionally, the system could provide a physical alert, such as a vibration in the driver's seat

4.Driver feedback: The system could also include a feedback mechanism to inform the driver of their driving behavior and alertness levels. This feedback could be displayed on a dashboard screen or sent to the driver's smartphone.

5.Data logging: The classification could also log data on the driver's behavior and alertness levels. This data could be used to analyzethe effectiveness of the system and identify areas for improvement.

Overall, a driver drowsiness detection system could help prevent accidents caused by drowsy driving by alerting the driver when they are becoming drowsy and encouraging them to take a break or rest before continuing their journey.

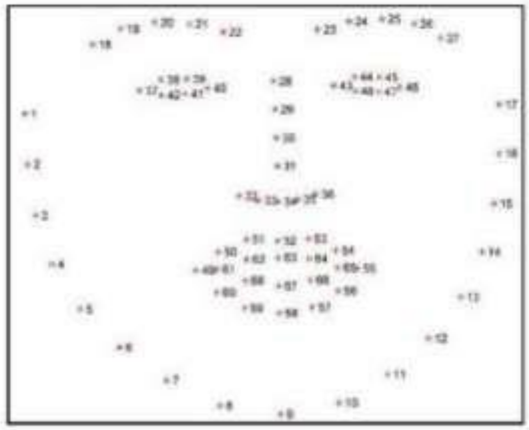
3.1 Face Detection and Facial Land Marking:

Face detection and facial landmarking are two techniques that are often used in driver drowsiness detection systems to monitor the driver's behavior and detect signs of drowsiness.

1..Face detection is a computer vision technique used to identify the presence and location of human faces in an image or video. It works by analyzing an image or video frame for patterns that match a human face, such as the presence of two eyes, a nose, and a mouth. Once a face is detected, the system can track it over time and use other techniques, such as facial landmarking, to extract more detailed information about the face.

2.Facial landmarking is a technique used to identify specific features on a face, such as the position of the eyes, nose, mouth, and eyebrows. This information can be used to determine the orientation and expression of the face, and to track changes in the face over time. Facial landmarking typically involves the use of machine learning algorithms and deep neural networks to identify and track the features of the face.

In driver drowsiness detection systems, face detection and facial landmarking are often used together to monitor the driver's behavior and detect signs of drowsiness. For example, the system might use face detection to identify the driver's face, and then use facial landmarking to track the position and movement of their eyes, mouth, and other facial features. By analyzing changes in these features over time, the system can detect signs of drowsiness, such as drooping eyelids or a slack jaw, and provide alerts to the driver to take a break or rest before continuing their journey.



3.3 Yawning Detection:

Yawning detection is a technique used in driver drowsiness detection systems to monitor the driver's behavior and detect signs of drowsiness. Yawning is a common physiological response to fatigue and drowsiness, and can be a useful indicator of the driver's level of alertness.

Another approach to yawning detection is to use sensors to monitor the driver's physiological responses, such as changes in heart rate, breathing rate, and muscle activity. These sensors can be placed on the driver's body, such as on the chest or wrist, and can be used to detect subtle changes in their physiological responses that are indicative of drowsiness and yawning.

Once yawning is detected, the driver drowsiness detection system can provide an alert to the driver to take a break or rest before continuing their journey. This alert can be provided through auditory or visual cues, such as a warning sound or a flashing light on the dashboard.

Overall, yawning detection is a useful technique for driver drowsiness detection, and can be used in grouping with other techniques such as eye trailing and steering wheel monitoring to provide a comprehensive picture of the driver's behavior and level of alertness.

4.Result And Discussion:

The outcomes of a driver drowsiness detection system depend on the specific techniques and algorithms used, as well as the quality and quantity of data available for training and testing the system. In general, however, driver drowsiness detection systems have shown promising results in improving road safety by detecting signs of drowsiness and alerting drivers to take a break or rest before continuing their journey.

One common evaluation metric used for driver drowsiness detection systems is the accuracy of the system in detecting drowsiness. This can be measured by comparing the system's output to a ground truth label, such as a manual observation of the driver's behavior or physiological state. Studies have reported high accuracy rates for driver drowsiness detection systems, with some systems achieving accuracy rates of over 90%.

Another important factor to consider in the evaluation of driver drowsiness detection systems is the response time of the system. The response time refers to the time it takes for the system to detect signs of drowsiness and provide an alert to the driver. A fast response time is important to ensure that the driver can take action to prevent an accident before it occurs. Studies have reported response times of less than one second for some driver drowsiness detection systems.

One challenge in the evaluation of driver drowsiness detection systems is the variability in driver behavior and the effectiveness of the system in detecting drowsiness in different contexts. For example, the system may perform well in detecting drowsiness during daytime driving on highways, but may not be as effective in detecting drowsiness during nighttime driving or on winding roads. Further research is needed to evaluate the effectiveness of driver drowsiness detection systems in different driving contexts and under different conditions.

Overall, driver drowsiness detection systems have shown promising results in improving road safety by detecting signs of drowsiness and alerting drivers to take a break or rest before continuing their journey. However, further research is needed to optimize these systems and ensure their effectiveness in a range of driving contexts and conditions.

Fig: open eye

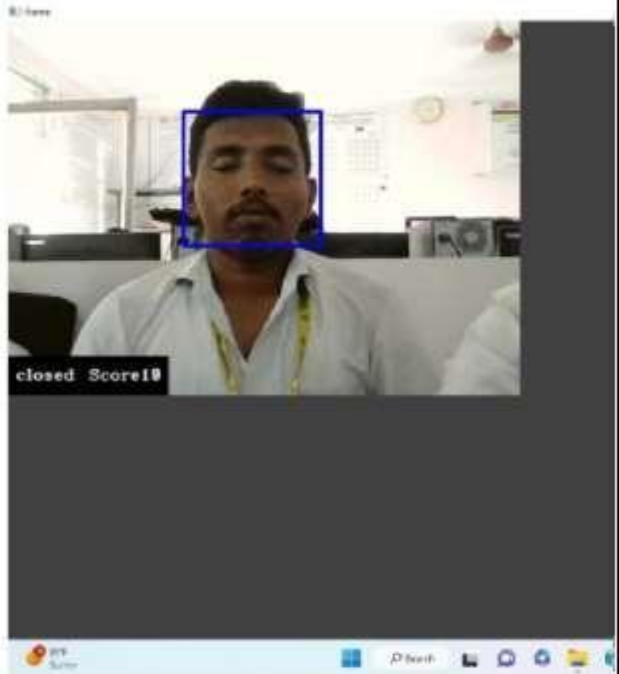


Fig :Closed Eye

5.Conclusion:

In conclusion, driver drowsiness detection systems are an important area of research and development in the field of road safety. Drowsy driving is a major cause of accidents on the road, and driver drowsiness detection systems have the potential to significantly reduce the incidence of these accidents.

There are a variety of techniques and algorithms used in driver drowsiness detection systems, including facial recognition, eye tracking, and yawning detection. These techniques have shown promising results in detecting signs of drowsiness and alerting drivers to take a break or rest before continuing their journey.

However, there are still challenges to be addressed in the development and implementation of driver drowsiness detection systems. These challenges include variability in driver behavior and the effectiveness of the system in different driving contexts, as well as issues related to privacy and data security.

6.Reference:

[1] Kyong Hee Lee, Whui Kim, Hyun Kyun Choi, Byung Tae Jang, "A Study on Feature Extraction Methods Used to Estimate a

Driver's Level of Drowsiness", International Conference on Advanced Communications Technology (ICACT), 2019.

[2] Ashish Kumar, Rusha Patra, "Driver Drowsiness Monitoring System using Visual Behaviour and Machine Learning", IEEE

Conference, 2018.

[3] Cyun-Yi Lin, Paul Chang, Alan Wang, Chih-Peng Fan, "Machine Learning and Gradient Statistics Based Real-Time Driver

Drowsiness Detection", 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW).

[4] Ching-Hua Weng, Ying-Hsiu Lai, Shang-Hong Lai, "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief

Network", In Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video, Taipei, Taiwan,

Nov. 2016

[5] Anjith George, Aurobinda Routray, "Real-time Eye Gaze Direction Classification Using Convolutional Neural Network", IEEE,

2016.

[6] Media Research Lab, http://mrl.cs.vsb.cz/, 2018.

[7] Venkata Rami Reddy Chirra, Srinivasulu Reddy Uyyala, Venkata Krishna Kishore Kolli, "Deep CNN: A Machine Learning

Approach for Driver Drowsiness Detection Based on Eye State", Revue d'Intelligence Artificielle Vol. 33, No. 6, December,

2019, pp. 461-466.

[8] Mkhuseli Ngxande, Jules-Raymond Tapamo, Michael Burke, "Driver drowsiness detection using Behavioral measures and

machine learning techniques: A review of state-of-art techniques", 2017 Pattern Recognition Association of South Africa and

Robotics and Mechatronics International Conference (PRASA-RobMech).

[9] T. Vogelpohla, M. Kühn, T. Hummel, M. Vollrath, "Asleep at the automated wheel—Sleepiness and fatigue during highly

automated driving", https://doi.org/10.1016/j.aap.2018.03.013, Accident Analysis and Prevention, 2018.

[10] Xuan-Phung Huynh, "Detection of driver drowsiness using 3D deep neural network and semi-supervised g

AB6

ORIGINALITY REPORT

13%	5%	8%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

%

1 Submitted to Liverpool John Moores University 4%
Student Paper

2 www.irjmets.com 1%
Internet Source

3 Mkhuseli Ngxande, Jules-Raymond Tapamo, Michael Burke. "Bias Remediation in Driver Drowsiness Detection systems using Generative Adversarial Networks", IEEE Access, 2020 1%
Publication

4. Changfu Liu, Bo Liu, Yang Zhou, Yan He, DexiaChi,Xingjun Gao, Qingkang Liu. "A real-time cutting temperature monitoring of tool in peripheral milling based on wireless transmission", International Journal of Thermal Sciences, 2023 1%
Publication

5. Submitted to Columbia College of Missouri 1%
Student Paper

6 Jennifer K. Johnson, Evelyn Vingilis, Amanda Terry. "Qualitative 1%
Study of Long-Haul Truck Drivers' Health and Healthcare Experiences",
Journal of Occupational & Environmental Medicine, 2020
Publication

7 Ghulam Masudh Mohamed, Sulaiman SaleemPatel, Nalindren 1%
Naicker. "Data Augmentationfor Deep Learning Algorithms that
Perform Driver Drowsiness Detection", International Journal of
Advanced Computer Science and Applications, 2023
Publication

8 J. Rajevenceltha, Vilas H. Gaidhane, V. Anjana."A novel 1%
for Drowsiness Detection using Local Binary Patterns and Histogram
of Gradients", 2019 International Conference on Electrical and
Computing Technologies and Applications (ICECTA), 2019
Publication

9 Naoto Nojiri, Xiangbo Kong, Lin Meng, Hiromitsu Discussion
on Machine Learning and Deep Learning basedM 1%
Considered Eye Status Recognition for Driver Drowsiness", Procedia
Computer Science, 2019
Publication

SpringerBriefs in Computer Science, 2014.

Publication

10 <1%

11 Vineetha Vijayan, Pushpalatha KP. "A Comparative Analysis of RootSIFT and SIFT Methods for Drowsy Features Extraction", Procedia Computer Science, 2020 <1%

Publication

12 Wenbin Xiang, Xuncheng Wu, Chuanchang Li, Weiwei Zhang, Feiyang Li. "Driving Fatigue Detection Based on the Combination of Multi- Branch 3D-CNN and Attention Mechanism", Applied Sciences, 2022 <1%

Publication

cse.anits.edu.in

Internet Source

13 ijircce.com <1%

Internet Source

14 orbi.uliege.be <1%

Internet Source

15 www.ijsrd.com <1%

Internet Source

16 Chien-Yu Chiou, Wei-Cheng Wang, Shueh- Chou Lu, Chun-Rong Huang, Pau-Choo Chung,Yun-Yang Lai. "Driver Monitoring Using SparseRepresentation With Part-Based Temporal <1%

<1%

17

Face Descriptors", IEEE Transactions on Intelligent Transportation Systems, 2019

Publication

Certificates

NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA, Website:www.nrtec.in

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

PAPER ID

NECICAIEA2K23027

Certificate of Presentation

This is to Certify that Papabathula Narendra , Narasaraopeta Engineering College has presented the paper title Driver Drowsiness Detection in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineering Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr.M.Sreenivasa Kumar


NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23027

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Sk Basha , Narasaraopeta Engineering College has presented the paper title Driver Drowsiness Detection in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr.M.Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

PAPER ID
NECICAIEA2K23027

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that M Gopichand , Narasaraopeta Engineering College has presented the paper title Driver Drowsiness Detection in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr.M.Sreenivasa Kumar

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23027

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Dr S V N SREENIVASU** , Narasaraopeta Engineering College has presented the paper title Driver Drowsiness Detection in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)