VOLUME 16, N° 2 2022
Journal of Automation, Mobile Robotics and Intelligent Systems

Machine Learning and Artificial Intelligence Techniques for Detecting Driver Drowsiness

Submitted: 9th February 2022; accepted 3rd May 2022

Boppuru Rudra Prathap, Kukatlapalli Pradeep Kumar, Javid Hussain, Cherukuri Ravindranath Chowdary

DOI: 10.14313/JAMRIS/2-2022/17

Abstract
The number of automobiles on the road grows in lockstep with the advancement of vehicle manufacturing. Road accidents appear to be on the rise, owing to this growing proliferation of vehicles. Accidents frequently occur in our daily lives, and are the top ten causes of mortality from injuries globally. It is now an important component of the worldwide public health burden. Every year, an estimated 1.2 million people are killed in car accidents. Driver drowsiness and weariness are major contributors to traffic accidents this study relies on computer software and photographs, as well as a Convolutional Neural Network (CNN), to assess whether a motorist is tired. The Driver Drowsiness System is built on the Multi-Layer Feed-Forward Network concept CNN was created using around 7,000 photos of eyes in both sleepiness and non-drowsiness phases with various face layouts. These photos were divided into two datasets: training (80% of the images) and testing (20% of the images). For training purposes, the pictures in the training dataset are fed into the network. To decrease information loss as much as feasible, backpropagation techniques and optimizers are applied. We developed an algorithm to calculate ROI as well as track and evaluate motor and visual impacts.

Keywords: *Artificial Intelligence, Machine Learning, Drowsiness Detection, Image Processing, Convolutional Neural Networks, AI Visuals*

1. INTRODUCTION

Road accidents are the top cause of death from injuries worldwide. They now form an essential part of the global burden of illness. Every year, 1.2 million people are killed in accidents, and an estimated 50 million are injured. Developing nations bear the burden, reporting 85% of all yearly deaths and 90% of life-threatening injuries because of road damage. Road injuries mainly affect men (73% of total) between the ages of 15 and 44; this burden creates significant economic imbalance.

The number of automobiles on the road grows with the advancement of car manufacturing, and the number of incidents looks to be gradually increasing, owing to the rapid proliferation of automobiles [1]. People get hurt in our everyday lives. The Driving Discovery System is built using concepts based on non-disruptive machine learning. The system uses a small security camera that monitors the driver's face and eyes. A driver's warning [2] is given when fatigue is detected. This report explains how to find whether the eyes are open or closed. The algorithm developed differs from other papers. This system works using the application details gained from the grayscale image to find the edges of the eyes. The magnitude of changes in the position of the eyes determines the opening or closing of the eyes. Blindfolds are used to cover a large distance. When the driver's eyes remain closed for five consecutive frames, the system concludes the driver has fallen asleep and sounds an alert to wake them up. Working in typical illuminated conditions, the device can also identify when the eyes are not focused.

2. LITERATURE REVIEW

Personal recognition procedures use a wide range of stimulants, obtained from many senses, if not all, including sound, touch, etc. For the most part, conditions and context information are used, as an example, to play a vital role in facial recognition in how to go and find. It is useless to try and try to improve the system with current technology, which will copy the unique face's ability to identify people. However, as a brain, it has its limits in the number of people it can "remember" accurately. The primary benefit of a computer program is its ability to process large numbers of facial images. For the most part, photo applications have only present in the form of one or more views of two-dimensional data, so that input to a face recognition algorithm is only visual. For this, books updated in this field are limited.

Drowsiness in humans [3] is defined as when they are almost asleep or in a dormant state. It is a sign that they cannot stay awake or that they need to sleep [4]. Fatigue is another cause of drowsiness that is frequently and is defined as a weary physical condition, resulting in a state of mental and physical tiredness. Both drowsiness and fatigue happen as a result of overall physical tiredness. The level of weariness may also be a metric for attentiveness or observance. The condition of alertness is defined as the absence of sleepiness, and the state of being observant is one of being keenly focused on something [5].

A detailed review of relevant literature in psychophysics and neuroscience follows. We describe the applicable requirements for the face recognition system

[6] Facial recognition has been previously regarded as a digital process that differs from other awareness activities. Proof of dedicated presence in the face-to-face program comes from several sources [7]:

1. Faces are easier for humans to remember than anything else when presented directly standing.

2. Prosopagnosia patients may not be able to remember a familiar face, but often they might not have deep agnosia. They use voice, hair color, dress code, etc., to remember people. It should be kept in mind that patients with prosopagnosia can see whether the given object has a face or not, but have difficulty remembering the face.

The differences in facial expressions and object detection can be summed up according to strong notions: (1) the effects of composition, (2) technology, (3) the differences discussed, (4) comparisons of polarity and lighting direction, (5) metric diversity, (6) Depth of field (related in the case of variability in machine recognition systems), and (7) rotation on the plane / shifted surface real-time eye, gaze, and face pose tracking system for driver attentiveness monitoring proposed by Qiang and Yang [8] and Chauhan et al. [9] introduced a face recognition system which would use IoT, the cloud, and deep learning. The authors used a cloud environment, which is a tree-based profound foundation for programmed facial recognition. Pillai et al. [10] proposed a model on frame diminution techniques for speed detection of vehicles. With a stationary camera, the author used the VDFT background subtraction approach to estimate vehicle speed. This is a recursive technique for calculating the rate of movement that involves obtaining the moving object's area moving property and comparing it to the calibrated speed. The Support Vector Machine (SVM) [11] is used to improve eye recognition. In Principal Component Analysis (PCA) and Logical Analysis of Data (LAD), statistical characteristics like sparseness and kurtosis are employed to identify eye state. Crime analysis and prediction using data from road accidents has been done by Prathap and Ramesha et al. [12].

Fatigue is an important issue to consider while monitoring traffic accidents, as studied by Hamada et al. [13], Barr et al. [14], Eriksson and Papanikolopoulos [15], Eskandarian and Mortazavi [16], and Grace et al. [17]. Around 20% of all traffic accidents are caused by driver weariness while on the road. When driving while tired, the driver's performance deteriorates, resulting in road accidents.

Several vehicle accidents happen due to driver inattention [18]. Detecting driver drowsiness and warning the motorist before an accident happens is unquestionably advantageous and could save lives. The most significant cause of road accidents is driver weariness. Detecting and preventing driver weariness is crucial to avoiding accidents and can be accomplished with new technology. Shahverdy et al. [19] presented a vision-based Intelligent Transportation Systems technique for detecting tired drivers and distractions by monitoring the system. A unique eye-detecting algorithm is utilized in this system, which integrates adaptive convolutional networks, dynamic boosting, and blob recognition with eye verification. It cuts down on processing time considerably. According to Valsan, Mathai, and Babu [20], the central feature of Driver Sleepiness Detection is obtaining an image of the driver's face, analyzing the image obtained, and estimating the drowsiness level, SVM (Support Vector Machine) is a sort of machine learning technique that employs. To achieve this, we need proper hardware. The research looks at the Raspberry Pi 3 Model B as a primary computer. The Raspberry Pi camera was chosen as the visual device. It is simple hardware to work with when used in tandem with well-defined software. As a result, software plays a vital role in the machine. The researchers have used OpenCV as an image and video library in their study.

Baek et al. [21] proposed an algorithm for detecting faces using a classifier known as Adaboost MCT, and showed how facial landmark detection works using an LBF regressor. Also, the running is swift and accurate on the embedded camera, as shown in the flow chart diagram of the drowsiness detection algorithm. The image is received from the camera after the video query. Next, it is preprocessed to remove the noise using conducted surveys to comprehend the needs and prerequisites of the general population. To do so, they went through different sites and applications and looked for the fundamental data. They made an audit that helped us get new thoughts and make different arrangements for our task based on these data. We determined demand for such an application and thought we had made some headway in this area.

Based on the above literature, the driver drowsiness detection system is based on eye closedness detection. The system uses the mathematical formula for finding the closedness of the eye. The cascades are used to detect the face and eyes in the represented video. The eye closedness is noticed by drawing the landmarks over the edges and peaks of the eye. These landmarks or points represent the condition of the eye and the driver's state. The total combination of these points is defined as a Region of Interest (ROI). Once these edges and the peak points align horizontally, the algorithm predicts the result as drowsiness by giving an alert notification. Many of these are entirely embedded systems using Raspberry Pi and some other modules.

3. METHODOLOGY

The driver drowsiness system works over the Multi-Layer Feed Forward Network concept, especially on Convolutional Neural Networks (CNN). CNN is developed with the help of around 7000 images of eyes with different facial architecture in both drowsy and non-drowsy states. These images are split into training (80% of images) and testing (20% of images) datasets. The images under the training dataset are given as the network's input for training purposes.

A backpropagation algorithm and optimizer are used, which reduce data loss as much as possible. Figure 1 shows the system architecture of the driver drowsiness system. This facial preprocessing identifies the images from the camera and gives the input to network model for classifying the state of the driver, which is using a CNN classifier.

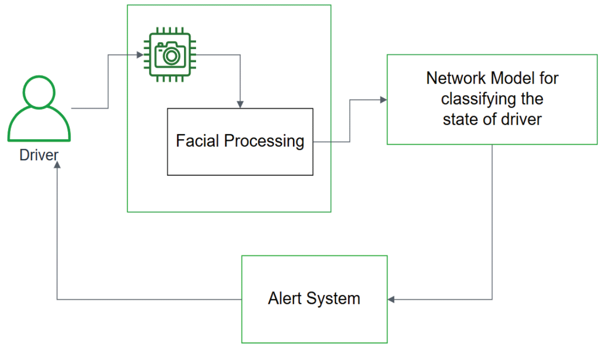


Fig. 1. System architecture diagram

A system architecture shows the structural and behavioral view of the whole system. An architecture design is a formal design representing a system in such a way that it clearly shows the structures and behaviors of the system. Figure 1 shows that the system architecture contains system components and the sub-systems developed, working together to implement the overall system functioning. The architecture of this system consists of 5 parts. They are listed as (1) Driver, (2) Camera, (3) Facial Processing, (4) Network Model, and (5) Alert System. The driver interacts with the camera to capture the video. This video captured by the camera is given as input for the facial processing. It detects the *Region of Interest (ROI)* and transfers it to the network model. The network model calculates the driver's drowsiness level with the help of output given by the facial processing step. Based on the result produced by the network model, an alert system notifies the driver if they are drowsy.

Figure 2 depicts the flow of events in a driver drowsiness system. In addition to representing a workflow, a flowchart may be a diagrammatic depiction of an algorithm, such as a step-by-step approach for accomplishing a job. The flowchart illustrates several kinds of boxes and the order in which they are connected by arrows. In Figure 2, new images are given as an input to adjust brightness and contrast, where the face will be detected first. Based on the research, the main focus in detecting drowsiness is identifying the eye movement in the face detection. Based on the eye movement, it will give an alarm.

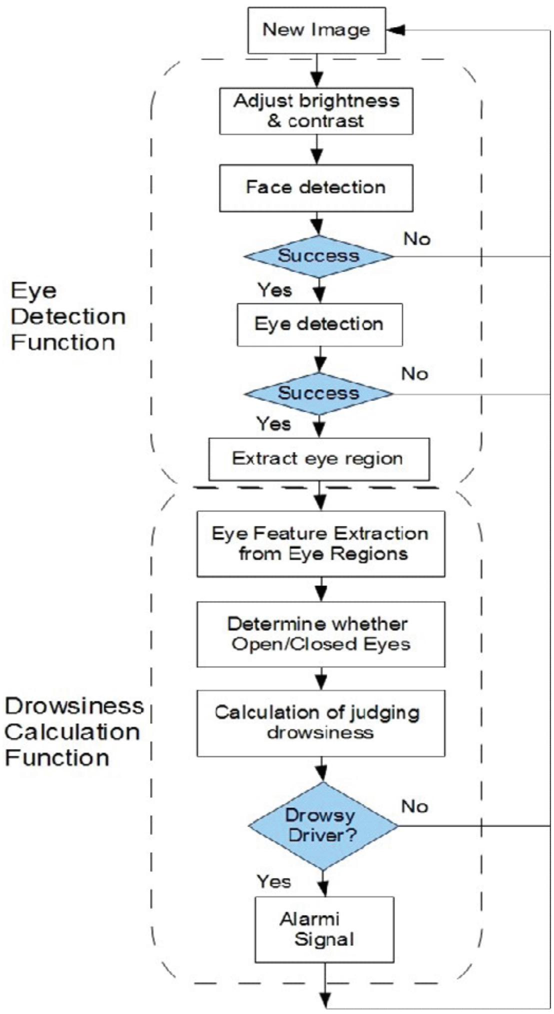


Fig. 2. Detailed framework of the driver drowsiness system

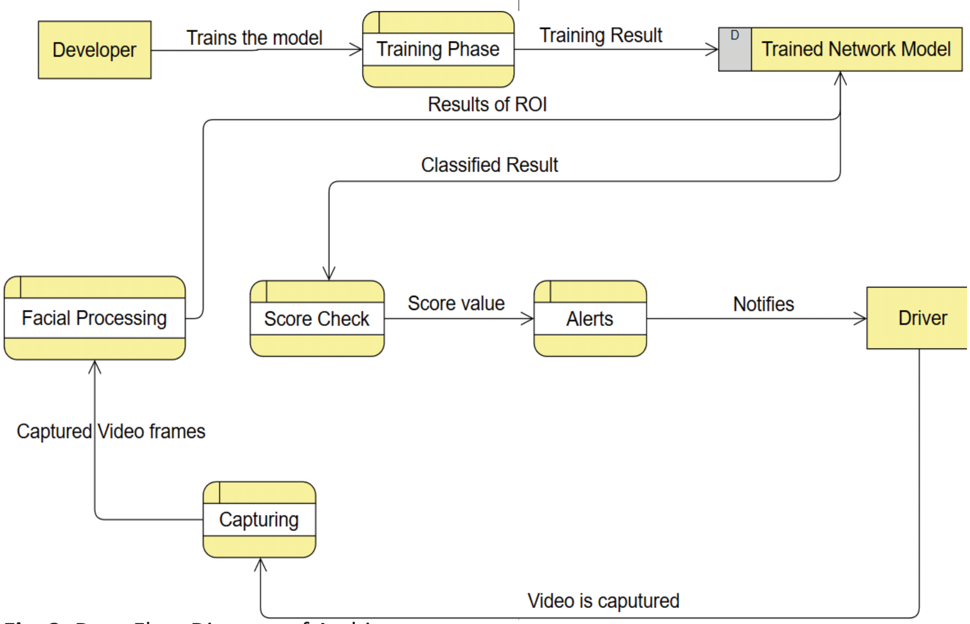


Fig. 3. Data-Flow Diagram of Architecture

Figure 3 shows the architecture's data flow diagram (DFD) (usually an information system). Here we also get the info about various aspects of an entity, like outputs and inputs of the process. A data-flow diagram does not have any control flow; there are no decision rules or loops in it.

3.1 Implementation Process and Modules

The whole system is comprised of various smaller modules or components or working groups which are described as follows: (1) Video Capturing and Image Processing, (2) Video Capturing, and (3) Processing video. A detailed explanation of each module is as follows.

Video Capturing and Image Processing

The video captured by the camera is to be refined, and faces are to be detected from them. Once seen, these faces are refined to provide information about the eyes. These eyes are to be transferred to network model to get classified. Several steps are to be followed in video capturing and image processing.

Video Capturing

The video is captured with the help of the camera of a laptop. These videos will be sent to the preprocessing stage to find the region of interest or ROI.

Processing Video

Once the camera begins to capture video, the video is translated into frames, which are used as input for the processing software. This processing occurs in the background so that users cannot see the processing steps. From Figure 4, RGB frames are converted to grayscale (which comprises the range between black to white on the scale of 255 to 0) for easier detection of the facial landmarks.

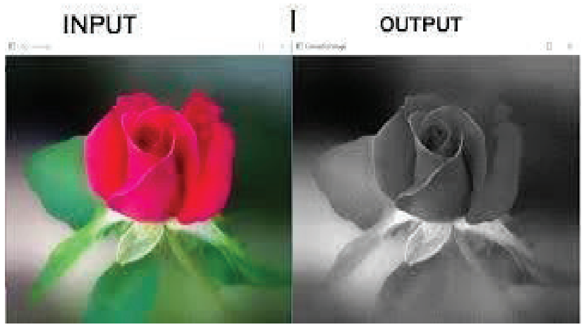


Fig. 4. Color to Grayscale conversion

Figure 5 shows the highlights to detect the region of interest (ROI). Once the ROI is detected, it is sent to the model for classification. Based on the results gathered from the model and the score value (effect of drowsiness), the system alerts the user. Once the conversion is completed, face and eye cascades are considered to detect face and eyes over frames. The edges are calculated using the *Edge Gradient* function given above. Using this function, the edges of the face are determined. Once the boundaries are determined, the face is detected, and using the cascades, the ROI (eyes region) is detected and transferred to the neural network for classification.

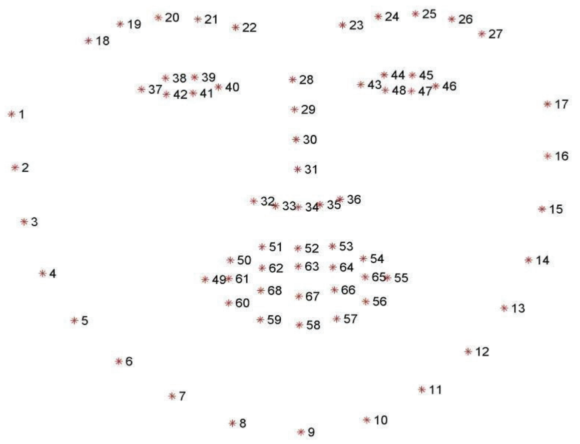


Fig. 5. Highlights to detect the region of interest

Network Model Building

The ROI from the frames captured is sent as input to the network for classification purposes. The network that mainly works over extensive dimensional data such as images is known as a convolutional neural network or CNN, and is built over around 7000 images of different facial architectures. There are several steps involved in the construction of this CNN.

Preprocessing the Data

The ROI transferred from module-1 (Video Capturing and Image Processing) is of high dimensionality. To reduce dimensionality, this data is transformed to grayscale, normalized in the range of 0 to 1. This preprocessed data is sent into the CNN model. The input is scaled to 32x32 dimensions.

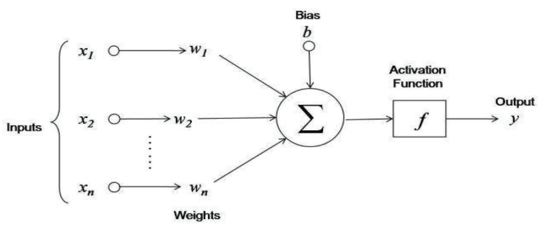


Fig. 6. Convolutional neural network model

Building a Network Model

Figure 6 shows the CNN Model. The CNN contains 3 layers. These are (1) input layer, (2) hidden layer and (3) output layer. The input layer includes 32 input nodes in total. Input is passed to these nodes. Every node of this layer is connected to each node of hidden layers. The data is transferred from the input node to the hidden node. The processed data from the hidden node are transferred to the output nodes. The output of the input nodes is the same as the input of the input nodes, but in the hidden nodes, the input is the summation of the product of the previous node taking into consideration the weight of the connection. Along with that, a bias is added to the value to compensate for the loss of transmission. The output of the hidden layers is charged using the activation function. This output is the input for the next or succeeding hidden layers or for the output layer. There are different activation functions in use. We have used specific tasks for this system.

Optimizing the CNN

The primary process is optimizing the neural network that is to be done for the accuracy maintenance of the network. The optimization algorithm improves the network's accuracy and decreases the network's error loss. This optimization step is responsible for the backpropagation process. Different optimizing algorithms like Gradient Descent, SGD, Adam, etc. We have used the Adam optimizer, which provides the most accurate results for its users. Neural networks for image processing are the most complex and large-build algorithm. To recover this situation dropout function is used. The dropout function is used to remove the not-used or unwanted nodes from the neural network. This function reduces the execution time in the training process by temporarily disabling the nodes. To minimize the processing step-over, the images' max pool step is performed. Max pool is the process of compressing the image without losing the contents. This process is done by reducing the size of the matrix by replacing the considerable matrix with a smaller one. This reduces the size of the image through which the processing of the image is reduced along with the reduction of the execution time.

Training and Testing the Network

Once the model is built, and optimization is done, the main step that needs to be done is training and testing. The training dataset is given as input for the network for training purposes, and the network is trained here. In the process of training, the network model dropout step is performed with a constraint. The trained network obtained is tested with the testing dataset to determine the accuracy of the network. The accuracy of the model is increased at every epoch with a reduction in data loss.

Alert System

Based on the output received by the model and the core value of the state, the model alerts the user if the state is drowsiness.

3.2 Working Methodology

The proposed research experiment is initially carried out by taking an input of a video stream with the camera. As the video stream is given as input, each frame is separated by initiating two threads running parallel. Figure 7 shows the working model of the proposed work. The input video streams will be converted into frames. The frames will be divided into thread1 and thread2, as explained in 3.2.1 and 3.2.2. Figure 8 explains the inception layer with filter concentration combined with the CNN algorithm. The proposed model works with a combination of the CNN model and Euclidian distance model to get an accurate result for the drowsiness identification of the driver.

Thread 1

In thread 1, the separated frame is given to the Haar cascade detector for detection and segmentation. Once the detection is done, the cropped image is sent to the Inception V3 model. The Inception V3 model is trained with 7000 images of eyes closed and open and has a training accuracy of 95%. T Inception V3 model classifies the cropped image with real-time prediction for eyes closed or open.

Thread 2

In thread 2, the separated frame is given to a pertained eye key point detector. The key point detector maps the face with a key point and allocates variables to each of the eye key points. With the assigned key points to the upper and lower edges of the eyes, we calculate the Euclidian distance formula to evaluate the distance between every two points. The eye key point is given by using Euclidian distance.

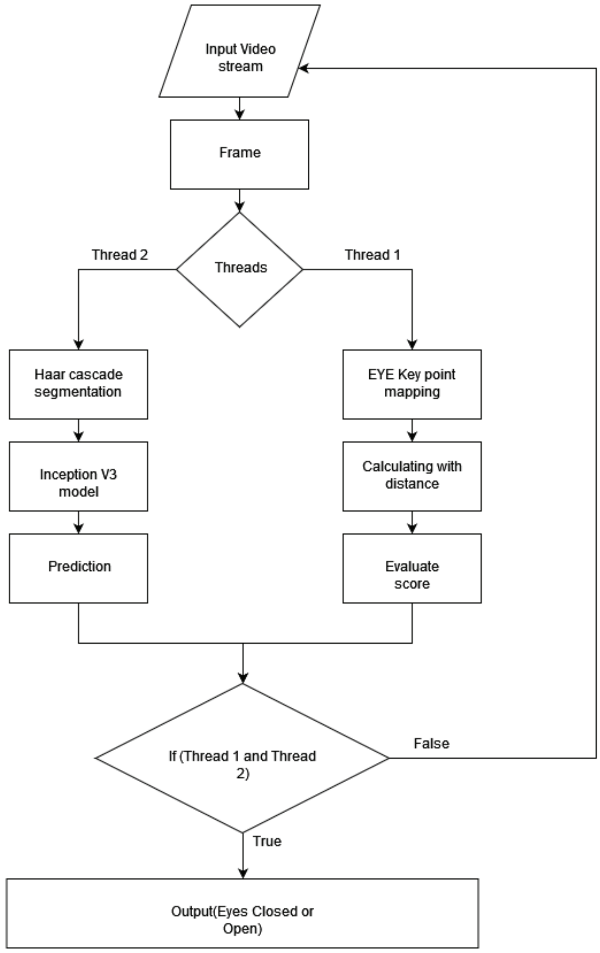


Fig. 7. Working model of the proposed methodology

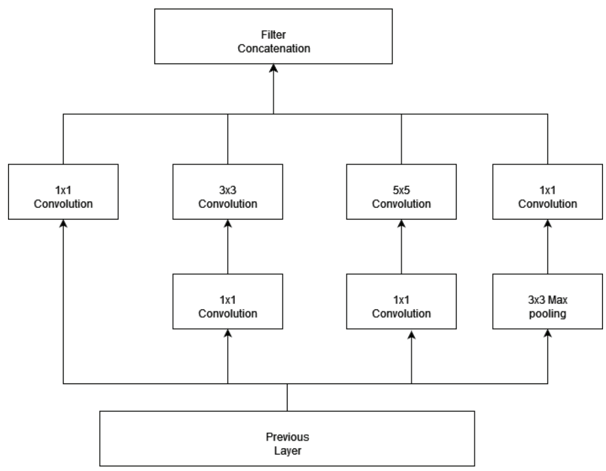


Fig. 8. Inception layer

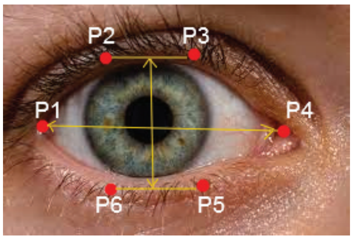


Fig. 9. Euclidian distance of eye points to detect drowsiness.

3.3 Novelty of Proposed Work

In this research, a module has been introduced to minimize the number of accidents due to driver fatigue, which is why this method significantly focuses on increased safety. This research also deals with automatically detecting driver drowsiness using AI and visuals associated with CNN. The CNN method, combined with Euclidian distance, helps in identifying driver drowsiness with more accuracy related to the proposed system. This works over the concept of Multi-Layer Feed-Forward Network, especially on Convolutional Neural Networks (CNN) and Euclidian distance, giving more accurate results.

Euclidean Distance and Its Significance in the Proposed Study

In mathematics, Euclidian distance is the length of a line segment between two particular points. Cartesian coordinates of the points help in calculating the distance using a simple Pythagorean theorem. Higher dimension calculations in computing and optimization use the square of the Euclidian distance.

The basic formula is as follows:

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]} \quad (1)$$

where (x_1,y_1) is the coordinate associated with the first point, (x_2,y_2) is the coordinate associated with the second point, and 'd' is the Euclidian distance.

Figure 9 gives a simple understanding of distance calculation between two points using the Pythagorean theorem, leading to the formation of the equation (1).

This simple concept of the distance-making formula is used in our work in combination with the CNN. This provides better accuracy in forming the results in the context of driver drowsiness. The same is applied in the software code of the AI-based application that has been developed.

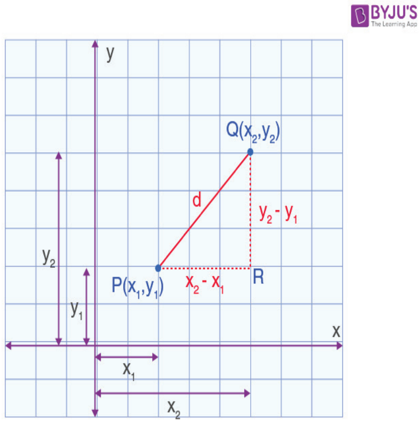


Fig. 9. A simple graphical illustration of Euclidian distance in view of the Pythagorean theorem

4. Results and Analysis

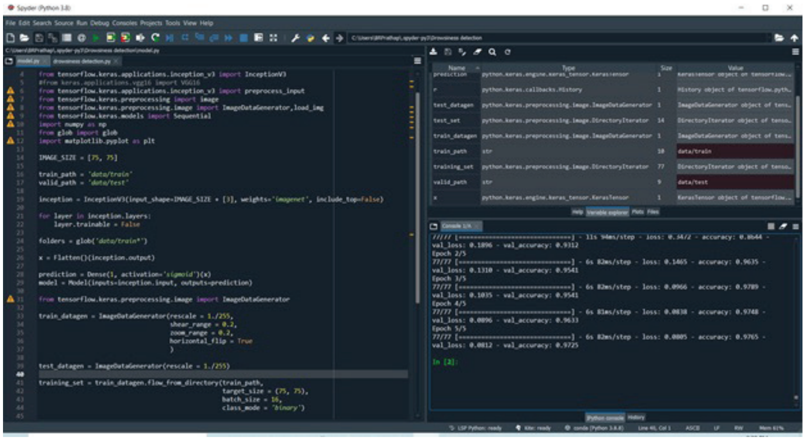
The sole objective of this system is to classify and notify the user of their drowsiness state. The working of our algorithm step-by-step process is as follows:

Step 1. Photo is taken from the camera as input. For accessing the webcam, we create an endless loop that holds each frame. We use CV2.capturevideo() to access the camera to capture images. Then we have CAP.READ(), which will set the captions, read the images frame by frame, and save them.

Step 2. Creating the region of interest (ROI): Our detection algorithm is based on OpenCV, so we need to convert the obtained color image to grayscale. We do not require color details to get the items. We use a Haar classifier to find the face. This sets our face partition = CV2. Cascade-Classifier ('here we add the path to our XML file'). After that, we made the discovery with the help of faces = face. Multiscaledetect(gray), which returns the coordinates (x,y), along with the height and width of the object. Now we can trim the face and create the border-box for each one.

Step 3. Capture the eyes within the ROI and feed it to the divider: The same facial recognition process is made to capture the eyes. Initially, we place the cascade separator with the eye on one and then the other, and then find the eye using left eye = eye:multiscaledetect(gray). Now we are left with the extraction of optical data from the whole image. This can be done by removing the border of the eye and then removing the image from the frame.

(a)



(b)

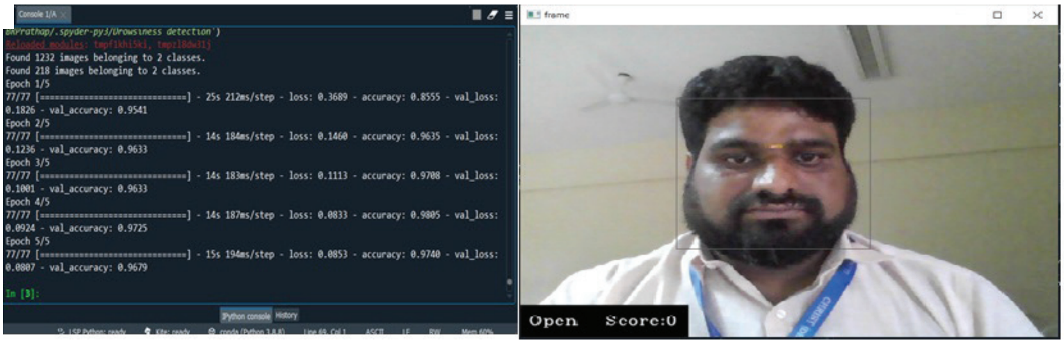
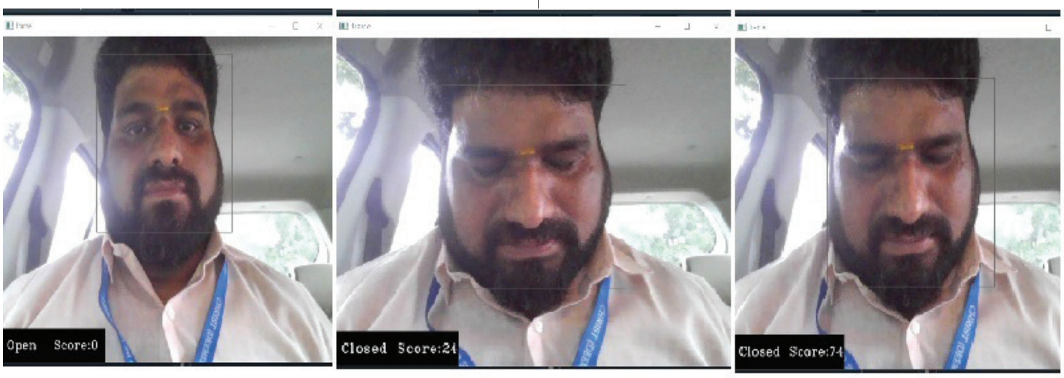


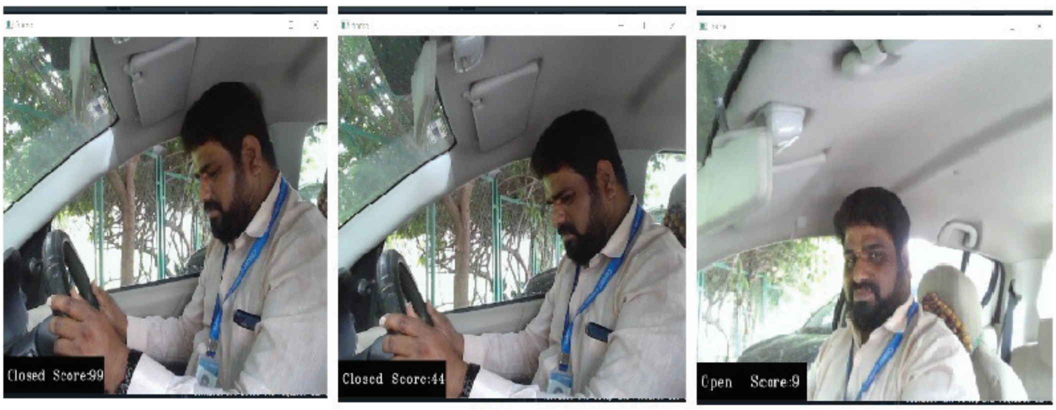
Fig. 7. Running the code and Accessing the GUI

Step 4. The separator will classify opened and closed eyes. We use CNN classification to predict eye conditions and give our image to the model because the model requires the right proportions to start with. First, we convert the color image to gray using reye = cv2:cvtColor(reye;cv2:COLORBGR2GRAY),

Step 5. Count Points for Checking Consciousness: Points are basically the amount we are going to use



(a)



(b)

Fig. 8. Face and eye recognition with respect to drowsiness

Fig. 9. Identification of driver drowsiness and alertness

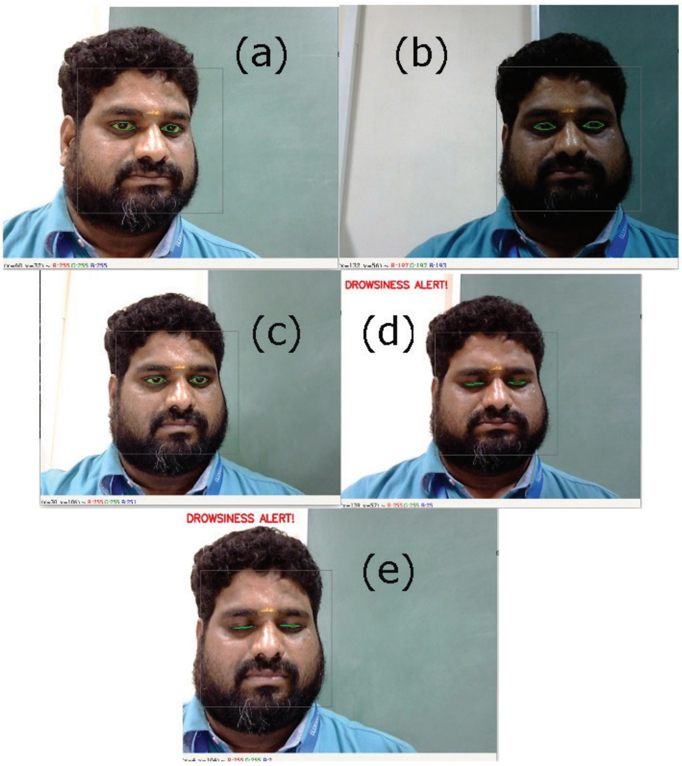


Fig. 10. Identification of driver drowsiness and alertness with Euclidian and CNN methods

to find out how long a person closes their eyes. When both the eyes are in closed state, we will continue to raise points and reduce points when the eyes open. We draw the effect on the screen using the cv2.putText () function to show the real-time statistics.

The following figures provide an explanation of the training the network and classifying the results.

Figure 7ab shows the GUI launching the camera. Clicking the "Okay" button launches the camera and the classification process. Once the camera is launched, the classification process is started, and it is shown at the bottom-left corner of the camera interface. "Open" and "closed" are the two classified outputs of the system. Based on the time of the "Closed" output, the alert system interacts with the driver. Here we considered 5 cycles of epoch as hyper parameters, as shown in Figure 7b.

Figure 8a shows the drivers' eyes as "closed" and the scores are given: 167, 154, and 154. Hence, the alarm has to be invoked. Figure 8b the driver's eyes and face is not being recognized at all, so the eyes are considered to be "closed" and the scores are given as 302, 207, and 249. Hence, the alarm has to be invoked. The eyes in Figure 8c are slightly closed so it is detected as "closed", and the scores are given as 40, 83, and 325 (since a complete eye is not recognized). In this case, the alarm system will turn on.

Figure 9a shows the driver's eyes completely shut, so it is detected as "closed." The score is increased to 53, and the alarm is triggered. Figure 9b shows the driver is not looking towards the road or camera at all, so it is detected as "closed" and the score increases to 116, and we have another alarm notification.

Figure 10 shows a simulation of driver drowsiness using Convolutional Neural Networks combined with Euclidian distances using the algorithm explained in Figures 7–9. Figure 10a shows an active driver simulation, and 10e shows a complete drowsiness alert.

5. CONCLUSION

Drowsy driving is becoming more of a concern throughout the world. It poses a risk and a serious dilemma for both the individual and the nation. According to a study [22], drowsy driving is responsible for 20% of all crashes. As a result, developing a system for detecting driver drowsiness is not only justified but also essential. The first proposed system is used to find the fatigue of the driver based on eye detection. The videos are captured through a camera or web camera for offline detection of the driver, and also through an Android mobile phone used for real-time detection of the fatigue of the driver. In every situation, a new invention or innovation has both positive and negative sides, but we as humans need to improve the world towards using the positive side and ignoring the negative side of the outcome. With our drowsiness models, we were able to reach a prediction accuracy of 97.65% by considering 1232 images belonging to 2 classes (Open & Closed). Combining the eye key point calculation and Inception V3 prediction, the evaluation score is calculated. With the help of the score, the drowsiness is measured and the driver is alerted with a buzzing sound. However, working with Euclidian distance in connection with the Convolutional Neural Network helped us understand how to obtain more accurate results for the proposed work in this specific context. This aids better safety and offers a newer dimension in mitigating the number of accidents.

In the near future, there is a wide potential scope for AI, and the key for humans will be to maintain and balance its rise. Some people can still believe that developing AI can lead to the destruction of the human race, but it is far from our sight and time, and also near impossible. As part of the future extension of research, to reduce hardware expenses, we can implement this project in mobile app form. We could also add some additional features like speed control in case of drowsiness along with an alarm system.

AUTHORS

Boppuru Rudra Prathap*- Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, Karnataka, India, *E-Mail: boppuru.prathap@christuniversity.in.*

Kukatlapalli Pradeep Kumar- Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, Karnataka, India.

Javid Hussain- Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, Karnataka, India.

Cherukuri Ravindranath Chowdary- Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore-560074, Karnataka, India *E-Mail: boppuru.prathap@christuniversity.in.*

REFERENCES

[1] P. Inthanon and S. Mungsing, "Detection of Drowsiness from Facial Images in Real-Time Video Media using Nvidia Jetson Nano," *2020 17th International Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2020, pp. 246–249, doi: 10.1109/ECTI-CON49241.2020.9158235.

[2] A. Dasgupta, D. Rahman and A. Routray, "A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, 2019, pp. 4045–4054, doi: 10.1109/TITS.2018.2879609.

[3] M. Ramzan et al., "A Survey on State-of-the-Art Drowsiness Detection Techniques," *IEEE Access,* vol. 7, 2019 pp. 61904–61919, doi: 10.1109/ACCESS.2019.2914373.

[4] K.G. Seifert, T. Jan and T. Karnahl, "Don't Sleep and Drive – VW's Fatigue Detection Technology," *Proc. 19th International Technical Conf. Enhanced Safety of Vehicles (ESV)*, 2005.

[5] R.J. Sternberg, *Cognitive Psychology*, Cengage Learning Press, 2012.

[6] I. Biederman and P. Kalocsai, "Neural and Psychophysical Analysis of Object and Face Recognition." *Face Recognition*, Springer, 1998, pp. 3–25.

[7] A. Ellis and R.M. Grieger, *Handbook of Rational-Emotive Therapy*, *Vol. 2*, Springer, 1986.

[8] J. Qiang and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *International Journal of Real-Time Imaging*, vol. 8, no. 5, 2002, pp. 357–377, doi: 10.1006/rtim.2002.0279.

[9] D. Chauhan et al. "An effective face recognition system based on Cloud-based IoT with a deep learning model." *Microprocessors and Microsystems*, vol. 81, 2021, pp. 103726.

[10] V.J. Pillai et al., "Fixed Angle Video Frame Diminution Technique for Vehicle Speed Detection," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 2, 2021, pp. 3204–3210.

[11] S. Hu and G. Zheng, "Driver drowsiness detection with eyelid related parameters by Support Vector Machine," *Expert Systems with Applications*, vol. 36, no. 4, 2009, pp. 7651–7658, doi: 10.1016/j.eswa.2008.09.030.

[12] B.R. Prathap and K. Ramesha. "Spatio-Temporal Crime Analysis Using KDE and ARIMA Models in the Indian Context." *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 12, no. 4, 2020, pp. 1–19, doi: 10.4018/IJDCF.2020100101.

[13] T. Hamada et al., "Detecting method for Driver's drowsiness applicable to Individual Features," *IEEE Proc. Intelligent Transportation Systems*, vol. 2, 2003, pp. 1405–1410, doi: 10.1109/ITSC.2003.1252715.

[14] L. Barr et al., "A review and evaluation of emerging driver fatigue detection, measures and technologies," A Report of U.S. Department of Transportation, 2009.

[15] M. Eriksson and N.P. Papanikolopoulos, "Eyetracking for detection of driver fatigue," *IEEE Proc. Intelligent Transport Systems*, 1999, pp. 314–318, doi: 10.1109/ITSC.1997.660494.

[16] A. Eskandarian and A. Mortazavi, "Evaluation of a smart algorithm for commercial vehicle driver drowsiness detection," IEEE Intelligent Vehicles Symposium (IV'07), Istanbul, Turkey, 2007, pp. 553–559, doi: 10.1109/IVS.2007.4290173.

[17] R. Grace et al., "A Drowsy Driver Detection System for Heavy Vehicles," Digital Avionics Systems Conference, 1998. Proceedings, 17th DASC. The AIAA/IEEE/SAE, vol. 2, 1998, pp. 50–70, doi: 10.1109/DASC.1998.739878.

[18] M.T. De Mello et al., "Sleep disorders as a Cause of Motor Vehicle Collisions." *International Journal of Preventive Medicine*, vol. 4, no. 3, 2003, pp. 246–257.

[19] M. Shahverdy et al., "Driver Behavior Detection and Classification Using Deep Convolutional Neural Networks," *Expert Systems with Applications*, vol. 149, 2020, pp. 113240, doi.org/10.1016/j.eswa.2020.113240.

[20] V.A. Valsan, P.P. Mathai and I. Babu, "Monitoring Driver's Drowsiness Status at Night Based on Computer Vision," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 989-993 (2021). doi: 10.1109/ICCCIS51004.2021.9397180.

[21] J.W. Baek et al., "Real-time Drowsiness Detection Algorithm for Driver State Monitoring Systems," *2018 Tenth International Conf. Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 73–75, doi: 10.1109/ICUFN.2018.8436988.

[22] Rivelli, Elizabeth. "Drowsy Driving 2021 Facts and Statistics | Bankrate." Drowsy Driving 2021 Facts & Statistics | Bankrate, www.bankrate.com/insurance/car/drowsy-driving-statistics. Accessed 27 Jan. 2023.