



Drowsiness Detection System

Dr. P. Shanmuga Priya¹, Chandramouli Saravan Sowri², N.S.Vigneshwara Reddy³

¹Associate professor, Department of Computer Science and Engineering, SCSVMV, Kanchipuram

²B.TECH graduate (IV year), Department of Information technology, SCSVMV, Kanchipuram

³ B.E graduate (IV year), Department of Computer science and engineering, SCSVMV, Kanchipuram

ABSTRACT:

Now -a- days road accidents, medical errors, sleeping during the online classes etc., are happening for every individual at any point of time which can lead to a major loss by small mistakes. For these problems, a solution is derived to detect the drowsiness in a driver while he is driving, in a surgeon while he is operating with drowsiness and a student can study entirely according to his schedule etc. This project is used to detect, alert & put awake an individual whenever he is feeling the drowsiness.

It is decided to use deep learning as domain, with keras, tensorflow & convolutional neural network (CNN) to train the dataset, classify the region of interest (ROI) is according to dataset respectively using the python as platform. It can alert the user of this application, with wakeup alarm using pygame. We could use the interactive GUI to make the good interaction with user. It can work even in low and high lights without causing vision blurring effect.

Keywords: Drowsiness detection, Deep Learning, Python, Open CV project, Tensorflow project.

I. INTRODUCTION:

There is a need of the drowsiness detection system in the real world, where the major problems are happening due to small mistakes of a person. Accidents are happening in many sectors due to drowsiness of the individual such as a surgeon who is in the state of drowsiness has failed the surgery by the cuts and shakes in hands due to the drowsiness also there are many accidents in different sectors like transportation, education, vigilance, researches etc.

For our project, the used domain is deep learning in order to analyze the dataset using the CNN algorithm in the platform of python to develop and implement the logic of this application. As python is an easy language for interpretation, vast library support, dynamic implementation, robust, etc. are the main qualities to consider it as a platform for our project.

Also there are many factors to be considered to develop our application like how to use the data for the project, configuring CNN with its layers and Relu activation etc. which is used to decide the objectives, scope of this project, existing systems and its drawbacks.

Objectives:

To develop a drowsiness detection system in such a way that it can detect the drowsy user using CNN algorithm with the open CV, Tkinter, pygame libraries in python to detect user, to develop GUI and history of drowsiness, alert sounds respectively.

Scope of the project:

The scope of the project is in such a way that, there is no person with drowsiness in his/her eyes while using this application. By using this application, it can prevent the major accidents in the real world in many sectors as like hospitals, transportation, education, vigilance, researches etc. it uses the face and eye detection to identify the user is in drowsiness or not. Also there is more scope for the project when it is used in various other sectors.

Existing systems:

There were many systems like the drowsiness detection by using various techniques such as detect my sleep, awake, etc. where a person can use the application in order to alert himself from the drowsiness so that his work can be done.

Drawbacks of existing systems: But there are limitations for these drowsiness detection systems. They are as below:

Complex execution,

No GUI,

Extra physical sensors,

Not working in low and high lights,

Vision blurring effect etc.,

By these limitations they are not so compatible in the real world situations. Hence, a new system has been proposed.

II. LITERATURE REVIEW

It's necessary to have a real time system that supervises an individual drowsiness continuously. To develop that operation we're using the deep- learning as our domain [5]. Deep learning is a machine learning fashion that teaches computers to do what comes naturally to humans learn by illustration. It is used for detecting the objects, faces and so numerous other purposes. As the Hamilton & Steve said in their work former findings were cumulated showing that visual analytics is vital in perfecting client engagement and immersive virtual gestic in digital commerce. [7]

The python as the platform is used to develop the operation. As the python is an interpreting language also python is protean. Generally, "Python programming language is extensively used for web development, operation development, system administration, developing games etc." [6][8][9][10][11]

For this design, the convolutional neural network (CNN) is used. It is an algorithm where it uses the conception of the artificial intelligence. It has three layers videlicet input sub caste, hidden sub caste, output sub caste. The input sub caste does the input functions and operations in the algorithms. [1, 3, 4] When it comes to open CV, it's another important library which is place in the input sub caste of the CNN. It's the reason to pierce the camera in the device. The open source computer vision library (opencv) open source computer vision and machine learning software library.[2] Open CV was used to give a common structure for computer vision operations and to accelerate the use of machine perception in the marketable products. Being an Apache 2 certified product, Open CV makes it easy for businesses to use and modify the law. A simple and effective stir discovery system that supports the user to interact with the machine and made it possible to pierce and prize information from the internet on a PC. [12][13][14][15][16]

III. Problem statement:

There are many drowsiness systems where they use the user's face, yawn, eyes etc. But some systems do not have the GUI, history of drowsiness, proper sensing methods, vision blur effects, high and low lights effects which may reach limitations of the drowsiness system. So, a new drowsiness detection system is created in such a way that, it can cross the limitations of the existing systems as above as such, it can work almost everywhere where a drowsy user is in front of this drowsiness system.

Proposed system:

A new drowsiness detecting system is proposed to work more interactively by tkinter & opencv GUI's, pygame for alerting alarm. The advantages of the proposed system are:

- Simple execution,
- Interactive GUI,
- No Extra physical sensors,
- Working in low and high amount of light,
- No Vision blurring effect
- History of drowsiness etc.,

Module's description:

In this project, we have 3 main modules. They are:

- 1) **Camera capturing & image processing:** In this module we have the libraries and files of the input data collectors such as opencv file for accessing the camera, cascade files to compare the face data with the references in cascade file's images.
- 2) **CNN MODEL:** This is the second module in the project where we process the input data using the our algorithm ie., CNN. In this module we have the libraries and files of the configured CNN model.
- 3) **GUI Output display:** In this module we have the libraries and files such as Audio file to wake up the user, Tkinter GUI for displaying the disclaimer of the project with detect button and opencv capturing window for showing the user's face in the screen.

Algorithm:

In this project we are using the CNN algorithm for the processing the dataset according to the input given via open CV window then the algorithm shall return the output layer's bits for further processing. The sequence of execution has given below:

- 1) Import all the required libraries and files.

- 2) Place a start button in Tkinter window.
- 3) Get the face data from the user using openCV window.
- 4) Using the CNN model file, classify the input data from the dataset of cascade files and process the data.
- 5) If user closes the eyes, sound the alarm after 15 seconds and reduce the seconds count to zero as the eyes are open.
- 6) If user opens his eyes, standby the alarm.
- 7) End the application when “q” is pressed.

IV. Architecture:

The system architecture consists of the entire project’s operational modules as dataset, cascade file, model.py, drowsiness detection.py etc. The initial sequence of the project will begin when user clicks in the application in his working directory. Then the GUI in the system architecture will be initialized. Then it will proceed with the second stage in the architecture i.e., process initiation at front end.

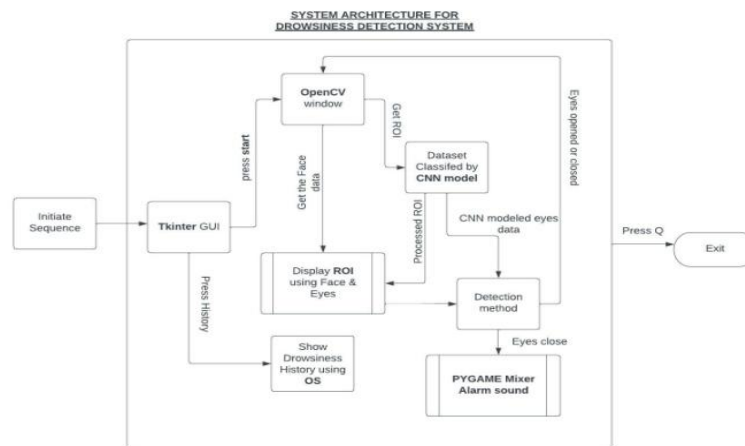


Figure 01: System architecture of the drowsiness detection system

V. Process:

The initial sequence of the project will begin when user clicks in the application in his working directory. Then the Tkinter frontend sequence is initialized. The tkinter GUI has designed in such a way that, the geometry of the UI is the standard metric of 700x500 which can be resizable. It also shows the standard guidelines to use the application in a proper way. The guidelines are simple. It will remove ambiguity of the new user. After clicking the start button in the GUI, the user can view the open CV window and also initializes the backend sequence of the application. The open CV window will be initialized by the loop, in order to get the continuous input from the user using the ROI (region of interest).

The ROI is decided by the methods of the detecting the faces and eyes as the continuous for loops. The configured GUI will be responsible for the processing the input data by referring the dataset in the form of the cascade files. After processing the input, the CNN output layer is responsible for locating the ROI. The ROI is responsible for the detecting the user’s face and his drowsiness. The open eyes in ROI shall remain constant until or unless the eyes were closed for 15 seconds.

Once the eyes are closed in the ROI, the alarm will alert the drowsy user in such a way that, it will keep beeping the sound of alarm until the eyes are opened. Meanwhile, the application will take the photos or snapshot of the drowsy user once the user closed his eyes and not opened till 15 seconds. It will store all the history of drowsiness in the file of ./history folder. The history folder is used to show the record of user’s drowsiness. In this way, the application will continuously work to detect the drowsy user in front of the camera.

VI. Methodology:

The Tkinter’s graphical user interface is used for the application to initiate the project’s sequence. When the start button is pressed in the Tkinter window, the open CV window for detection will appear on the screen. That openCV window can capture the user data in the frontend while the CNN, tensorflow can process the input data to show the user is in drowsy state or not. If user opens eyes, standby the alarm. If user closes the eyes, Pygame audio mixer will sound the alarm after 15 seconds and reduce the seconds count to zero as the eyes will be opened due to the alarm.

The methods involved in the project is as below:

Table 01: methods involved in the project

<u>Method</u>	<u>Description</u>
Hist()	Stores the History of the Drowsiness
Main()	The main methodology of the project is included in the main function. It contains all the loops for detecting the ROI and drowsiness of the user.

VII. Implementation:

The implementation of the drowsiness detection system starts from the preparing the dataset for referring the facial data by the CNN input layer. The algorithm of CNN will get the input data from the opencv window which is initiated by the Tkinter start button label in the display GUI. The loop of the opencv window will be initiated by that start button label, then the input data is given by user itself in front of the camera.

The loop of the opencv window will be initiated by that start button label, then the input data is given by user itself in front of the camera. By that input data, the dataset is referenced using the CNN algorithm model.py. Then the processing of the hidden layer is done in main file and ready to display the output of the status whether the user closes or opens his eyes by the right and left eye functions. By that output given in the ROI (range of interest) the opencv window will use the user's face on the screen. The drowsiness history is noted down in the .history file and the pygame will use its audio Beep.wav to alert the user.

Requirement specifications:

- Latest version of python IDE to implement the code.
- Cascade files for the data.
- A device with good quality camera.
- A good speaker.
- A Storage Space for storing the drowsiness history.
- Last but not the least, any version of windows OS with minimum processing RAM.

VIII. Results:

The results of the project are quite fascinating when a user closes his eyes and got alarmed for the first time with the beep.wav file. This is the graphical user interface of the application where the guidelines and the button to start is shown on the screen using Tkinter window of size 700x500.

This GUI is the first one to appear to show that the application's initial sequence has been successfully initiated and the good to go. The results of the project are taken a snapshot as below:

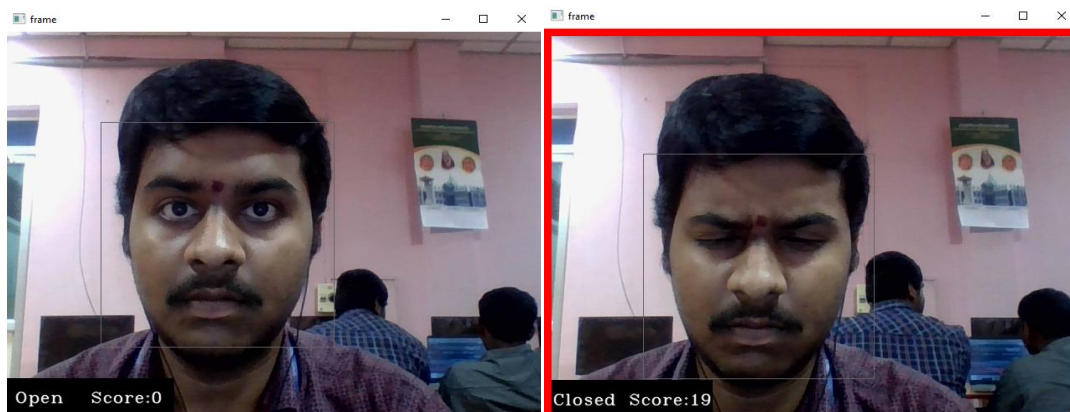


Figure 02a. 2b: Open CV window of the drowsiness detection system when eyes are opened and closed.

This is the GUI where many applications in the existing systems are lagging behind. By using this GUI, we are a step ahead while compared to the existing drowsiness detection systems.

As we can see in the figure, it represents the result of the open CV window when eyes are open. The loops of the right eye, left eye and the face will continuously detect the region of interest of face and eyes. The input data is processed by using the CNN model file import then, it will show the output of the processed data in the open CV window as in the figure. Here, the alarm will not ring in the first place, since the eyes were opened in the ROI, the alarm will stand by for the moment to close the eyes for 15 seconds.

When the user is in low light area, the user is detected even in the low light conditions given that the quality of the camera is good. The face in the low light is detected by the open CV window and the face data shall be classified using the low light images of the data set.

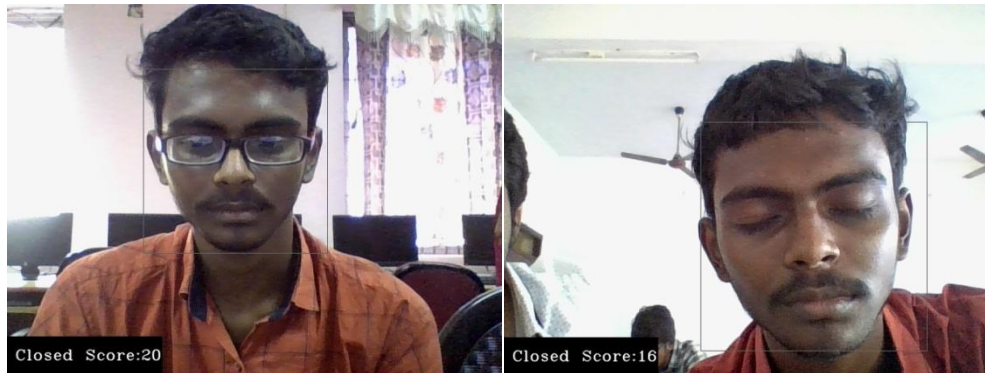


Figure 03a 3b: Open CV window of the drowsiness detection system when eyes are closed in the low and high light areas.

When the user is in high light area, the user is detected even in the high light conditions given that the quality of the camera is good. The face in the high light is detected by the open CV window and the face data shall be classified using the high light images of the data set.

IX. Conclusion:

Thus a drowsiness detection system can be implemented with the interactive GUI using a camera of Open CV while storing the drowsiness history in a file when the user is alerted by the alarm of beep with the help of pygame in a drowsiness detection system generated with the help of a platform of python using the CNN algorithm on the cascade files. This system is more efficient when compared to the current existing systems.

X. Future scope:

This drowsiness system has a very good scope when it is using in the various sectors like medical, education, science, sport etc., when it is customized accordingly. Customization of the application is also possible according to user interest. When it has more public reach, it can save a lot of damages when it is developed as advanced mobile app.

REFERENCES:

- [1] Aneesa al redhaei, Yamanalbadawi, saifamohamed, alialnoman : **Driver Drowsiness detection using machine learning – IEEE Xplore** 18-03-2022
- [2] Pranjitkumar das, parthoprothim barman et.al : **Application of python – opencv to detect contour of shapes & color of real image – ResearchGate** 15-05-2022
- [3] Serdar yegulalpet.al : **What is tensorflow? The machine learning library - Infoworld** 03-06-2022
- [4] Cordny et.al **Top 10 GUI Frame works – Active state** 05-06-2022
- [5] Md. Tanvir Dipu, syedasumbul et.al : **Real time driver drowsiness detection using deep learning -IJACSA** 14-07-2021
- [6] Vidya M Ayer et. al : **Why scientists should learn to program in python – ResearchGate** 05-07-2022
- [7] Steve Hamilton et.al : **Deep Learning Computer Vision Algorithms, Customer Engagement Tools, and Virtual Marketplace Dynamics Data in the Metaverse Economy. 2022 – Addle ton academic publishers**
- [8] Nirial Modi : **Python: the scope of the future – Tops technologies** - November 2022
- [9] Ekamjot Kaur : **Importance of Python programming language and how upskilling and studying Python can help you advance your career in the IT industry** : 4th April 2022 – **Zenodo**.
- [10] Sergi Castella I sape : **Research on python AI, ML - ICACITE** – March 2022
- [11] Elena magan et.al. **Driver drowsiness detection system by applying the sequence of images – MDPI** - December 2021
- [12] Anoop Jaiswal : **CNN Based on Transfer Learning Models Using Data Augmentation and Transformation for Detection of Concrete Crack – ResearchGate** - 15 August 2022
- [13] Gizemozgul, fatamapatlar et.al **wearable sensor device for posture monitoring and analysis during daily activities: Preliminary study. – International advanced researches and engineering Journals** – 15th April 2022
- [14] Elena Megan et.al **Driver drowsiness detection by applying Deep learning techniques to sequences of images - MDPI** – 22-01-2022
- [15] Sakshi bhor, sahib preetsingh et.al **Driver Drowsiness detection – Research Gate** – May 2022.
- [16] Ferdin Joe Joseph; sarayut nonsiri; **Keras & tensorflow:A hands on experience – ResearchGate** – July 2021.

AUTHOR'S PROFILE:

1. Dr. P. SHANMUGA PRIYA Associate Professor, Department of Computer Science and Engineering at SCSVMV, Enathur, Kanchipuram, India.
2. Mr. CHANDRAMOULI SARAVAN SOWRI, 11199M009, Student, B.TECH Information Technology at SCSVMV, Enathur, Kanchipuram, India.
3. Mr. N. S. VIGNESWARA REDDY 11199A259, Student, B.E Computer Science and Engineering at SCSVMV, Enathur, Kanchipuram, India.