Dayananda Sagar College of Engineering

Department of Electronics and Communication Engineering

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru – 560 078.

(An Autonomous Institute affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment and Accreditation Council (NAAC) with 'A' grade

Assignment

Program: B.E. Branch: ECE

Course: Machine Learning Semester : 7

Course Code: 19EC7DEMAL Date:22/12/2023

A Report on

Drowsiness Detection using IR Image

Submitted by

USN	NAME
1DS20EC130	PRABHAT KUMAR
1DS20EC137	PRASHANT ARYAN
1DS20EC196	SITESH KUMAR

Faculty In-charge

Dr. K N Pushpalatha

Signature of Faculty In-charge

TABLE OF CONTENTS

INTRODUCTION

Tiredness could be a state* of near-sleep, such that a person includes a lusty craving or want for sleep. Sleepiness is hazardous or vicious while performing certain duty/chores that requires a perpetual centralization, like motoring an automobile. When an individual is adequetly fatigued while motoring, they're going to encounter fatigue ness and this ends up in escalating the norms of road accidents.

Current tiredness detection system* monitoring the person who is controlling the auto mobile's condition which demands multiplex calculation and high rated appliances or stuff, agonizing or having problems to wear during motoring an automobile and isn't decent enough for driving norms; as an example, Electroencephalography*(EEG) and Electrocardiographyo(ECG), the event of technologies for noticing or dispensing tiredness while controlling an automobile may be a magnificent dare within the area of mishap avoidance system. thanks to the menace that tiredness presents on the roadarea, different ways have to be made for counteracting its affects. A tiredness detection system which uses a piece of video equipment placed before the motive force is highly acceptable to be utilized. We devised a mechanism that can detect fatigue and inform instantly. This technology uses a webcammer to record images as a video stream, recognises the faces, and locates the eyes. The CNN algorithm is then used to identify tiredness in the eyes. As a result of this, the driver is alerted for drowsiness via a gadget.

LITERATURE SURVEY

2.1 Real time object detection and tracking using deep learning and opencv.

G Chandan, Ayush Jain, Harsh Jain, et al. Real time object detection and tracking using deep learning and opencv. In 2018 International Conference on inventive research in computing applications (ICIRCA), pages 1305–1308.
IEEE, 2018. *https://ieeexplore.ieee.org/abstract/document/8597266*

2.2 Study of grayscale image in image processing.

Pramod Kaler. Study of grayscale image in image processing. International Journal of Recent and Innovation Trends in Computing and Communication, 4(11):309–311, 2016

2.3 Digital eye strain detection system based on svm.

Ramandeep Kaur and Ankit Guleria. Digital eye strain detection system based on svm. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pages 1114–1121. IEEE, 2021. *(https://ieeexplore.ieee.org/abstract/document/9453085)*

2.4 One millisecond face alignment with an ensemble of regression trees

Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1867–1874, 2014.

2.5 Convolutional neural network based image processing system

Hankil Kim, Jinyoung Kim, and Hoekyung Jung. Convolutional neural network based image processing system. Journal of information and communication convergence engineering, 16(3):160–165, 2018.
https://www.koreascience.or.kr/article/JAKO201831854885756.page

2.6 Eye blink detection using facial landmarks.

Tereza Soukupova and Jan Cech. Eye blink detection using facial landmarks. In 21st computer vision winter workshop, Rimske Toplice, Slovenia, 2016.

2.7 Supervised-learning

https://www.ibm.com/cloud/learn/supervised-learning

2.8 deep-learning-with- matlab

https://in.mathworks.com/campaigns/offers/deep-learning-with- matlab.html.

2.9 Open CV

https://opencv.org/about/

2.10 Digital image processing

.

https://sisu.ut.ee/imageprocessing/book/

SYSTEM ANALYSIS

As mentioned before, we are trying to create a software for fast and accurate detection of drowsiness. It was observed the accuracy of prediction vary before pre-processing of data and after pre-processing of data. More accurate results were obtained on trained data sets.

SVM and CNN both can be used but SVM is less efficient as compared to CNN. Therefore, CNN Algorithm is being used for this study. For developing the drowsiness detection software, we need hundred percent accuracy. CNN provides both.

SYSTEM DESIGN

This project is intending to develop a drowsiness detection system using machine learning(ML).

(ML)Machine Learning is the advancement ,development of computers that can adapt and learn with the help of statistical models and algos without following explicit instructions for patterns in data are analysed and interference is drawn. Algorithms used in ml have old data as input and gives prediction of new values as output.

4.1 ALGORITHMS USED

MACHINE LEARNING ALGORITHMS:

Creation application of computerr structure that is reading and adjusting without any explicit or special instruction by doing analyses andd drawing out interference from patterns of data with the help of statistical algorithms models is machine learning. the algorithms of ml take past data as input and then anticipate new numbers for output. Machine learning with supervision is a subfield of machine learning. It is distinguished by the use of datasets which are lablled to help in training of algorithms in order to accurately identify data or forecast outcomes. Unsupervised Machine Learning models do not use training datasets to be supervised. Instead, the model searches the data for hidden patterns and insights on its own. The necessity for tagged training data is the key difference between un supervised and supervisedd machine learning. Un supervised machine learning uses unmarked or raw data, whereas in the case of supervised machine learning it uses tagged input and output data for training.

CNN :

The (CNN)Convolutional NeuralNetwork is a kind of' deep learning neurall network that are most typically used to analyse visual information and do some tasks as

image identification, object detection, segmentation. In a nutshell, consider CNN to be a machine learning framework that can accept an input image and attach important (learning weights and biases) to different aspects/objects in the picture, as well as differentiate in them. It is a strong image processing algorithm. These methods are currently the best we have for image processing that is automated. Many businesses utilise these algorithms to do tasks such as identifying the objects in a photograph

The 3 types of layer present in "CNN" are :

1) ConvolutionalLayer: Each input neuron are basically typical neuraal network which are connected to the very nexthidden layer. Only a little fraction of 1st layers neuron joins the neuron in hiddenn layer in CNNs

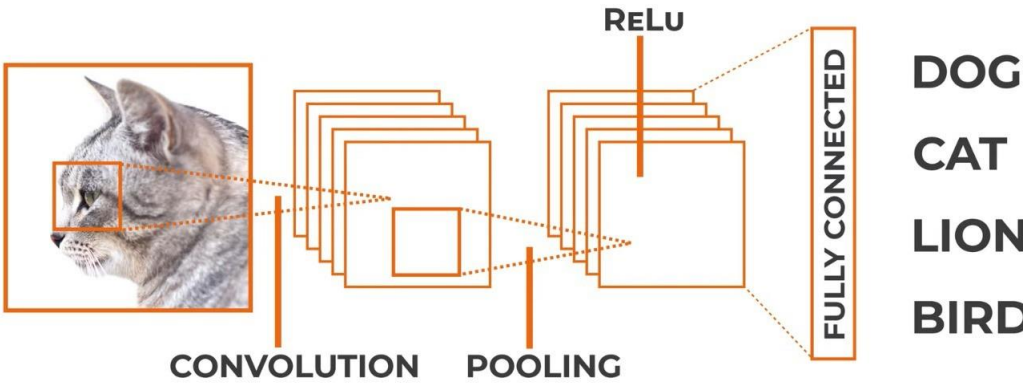


Figure :CNN Diagram

2) PoolingLayer: The poolinglayer minimises the features maps dimensions. In the CNN's hidden neurons, there are going to be several activation as well as pooling layers(pl).

3) FullyConnectedlayer: FullyConnectedLayers are the networks end layer. Which give us the very output result. Fnal Convolutional or Pooling Layers, and compressed and fended on to the fully auto connectedd layers, is the result of the fullyconnected layer(FCL).

4.2 Dataset

Captured the raw dataset source from kaggle. Data preprocessing is performed so as to make the dataset compatible to the program. Two categories of data which helps in determining the difference which is open and close. Equal number of images in the data so as to avoid uneven biasness.

Figure :data set Example

4.3 TECHNIQUE USED

Gray scaling:

An image may be a matrice or a two-dimensional array with a row and column number. The pixels of the image are represented by the cells. As a result, a 10x10 image will have 100 pixels. Each pixel in an 8-bit colour image comprises three data points for the Red, Green, and Blue (RGB) components, each of which spans from 0 to 255. Each pixel during a grayscale image has only 1 component and ranges from 0 to 255.

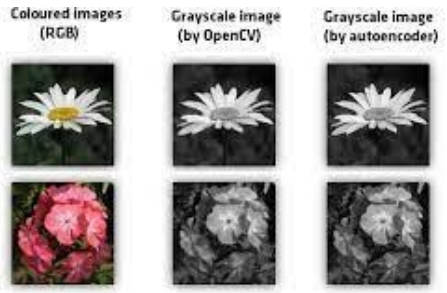


Figure :Gray Scaling

Each pixel in an exceedingly black and white image has only 1 component, which is either 1 or 0. These equations make it simple to work out the quantity of knowledge points which will be displayed or that has got to be processed in each scenario. Thus, converting a color image to grayscale reduces data by thrice, while converting to black-and-white image reduces data by 128 times. As a result, greater processing speed is critical.

Grayscaling is the process of converting an image to shade of grey out of various colour spaces suchas CMYK, HSV, RGB, etc. It can be fully white or fully black.

*All three planes are combined in a Gray picture (R,G,B) As a result, you'll obtain a "2-D" image with only two spatial coordinates. An RGB image, on the other hand, has a third dimension in addition to the two dimensions in a grey image, which distinguishes it from a grey image. RGB is made up of three components: a third component called intensity, as well as two spatial coordinates. When converting an RGB image to a grey image, we merge three components into one, i.e. Gray image = 0.29 * r(component)+ 0.59 * g(component)+ 0.11 * b(component)*

4.4 Data Processing

Data processing is the process in which data turns from one format to another which is more useable as well as informative. Machine Learning techniques, mathematical modelling, and statistical knowledge can all help with this.

The 5 major steps of data pre-processing are as follows:

Data cleaning. ,Data quality assessment. ,Data reduction. ,Data transformation.

Before its practical use, pre processing of data has to be done. The study of data pre processing is transforming of data set which is raw set in a data set which is clean. Just before actually using the method, we have to preprocesses the dataset so as to look out for king of data that is noisy or is missing values, and some other irregularity. 3 main dataa processing methods are as follows

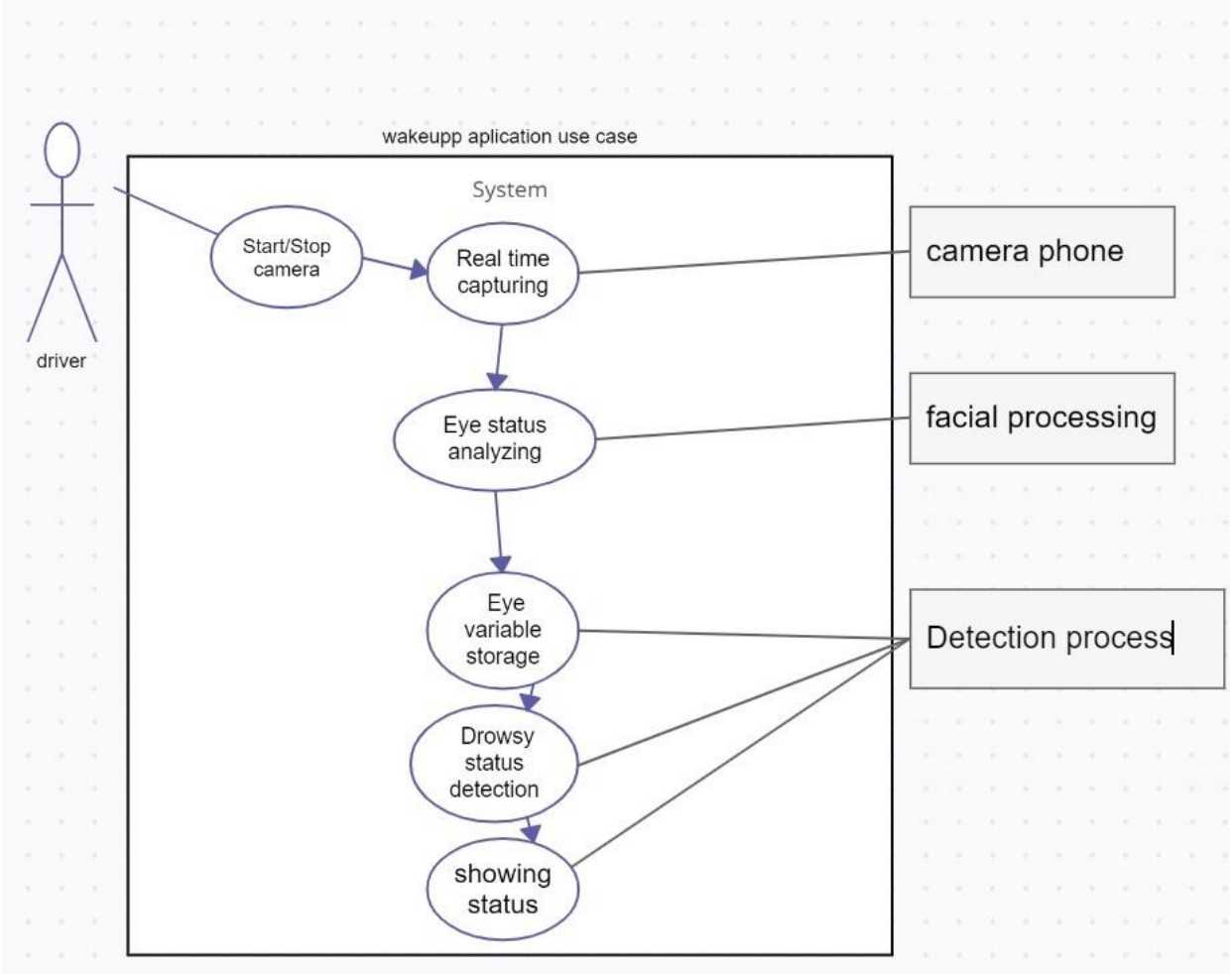
• Electronic Data Processing : in this data is processed by computers involving electric communication

• Manual Data Processing : in this data processing is done manually.

• Mechanical Data Processing : in this Data is processed mechanically with the help of

machines and devices.

4.5 ARCHITECTURE



4.6 TECHNOLOGY USED

Open CV is a real-time computer vision focused programming library. Intel designed it, and Willow Garage and Itseez sponsored it after that it was taken by "Intel". Underlying the opensource Apache2 Licenses, the libraries are free to use and cross-platform.

Figure :OpenCV

It's a massive opensource library which has computer visions, as well as machine learing also image processing that is playing a crucial component of real-time activities, which are essential in today's modern systems It can recognise objects, faces, and sometimes even human handwriting in photographs and movies. When Python3.9 is used in conjunction with other modules like NumPy, it may analyse the OpenCV arrays structure. We are usually employing vector spaces and are executing mathematical operation on the feature;s to find different features of visual patterns.

As we know Open CV was made with C++ and its primary interface is in c++,. The C++ interfaces is used to show all the latest breakthrougs and algos. "Python", "Java", and "MATLAB/OCTAVE" all are having some bindings. The online documents contain the APIs of such interface. They are still significantly older than C interface Wrappers for different types of programming language are beaning developed to encourage more and more usage.

Keras: Keras is a Python3.9 based openn-source high-level neural network library thatthat works with Theanoa, TensorFlow, and CNTKl. Which was made by Francois Chollet. It was developed by Francois Chollet, a Google developer. To allow for faster deep neural network testing, it has been made user-friendly, extendable, and modularIt notionly supports Convolutional and R.N on their own, butialso mixture with tandem. The backend storage library covers the low-level API in a very high level API, making it possible to run it on TensorFlowa, CNTk, or Theano. It had more than 4600 contributors when it was launched, and it today has over 250,000 developers. It has risen at a rate of 2X every year since its inception. Microsoft, Google, NVIDIA, and Amazon have all made significant contributions to developments of Kerass. Having a fantastic industry's reaction as well as interaction and is employed in creation of well-known companies like as Netflix, Uber, Google,*

Expedia, and others. Keras can be written in R or Python, and the execution of the

code can be done with the help of CNTK ,TensorFlow, MXNet, or Theano depending on needs. Keras may run on several different systems., including CPU's, NVIDIAs GPUs, AMD GPUs, TPUs, etc . It gives guarante that building model with the help of Keras is a breeze, as it can fully support Android(TF, TF Lite), iOS(Native CoreML),Raspberry Pi and TensorFlow serving, GPU acceleration (WebKeras, Keras.js). The one draw-back is that Keras is having some pre- configured layers of its own, and it won't let you build an abstract layer since it can't work with low-level API's. It only works with high-level APIs that work in conjunction with the backend engine (Theano, TensorFlow, and CNTK).

Figure :Tensorflow

TensorFlow:

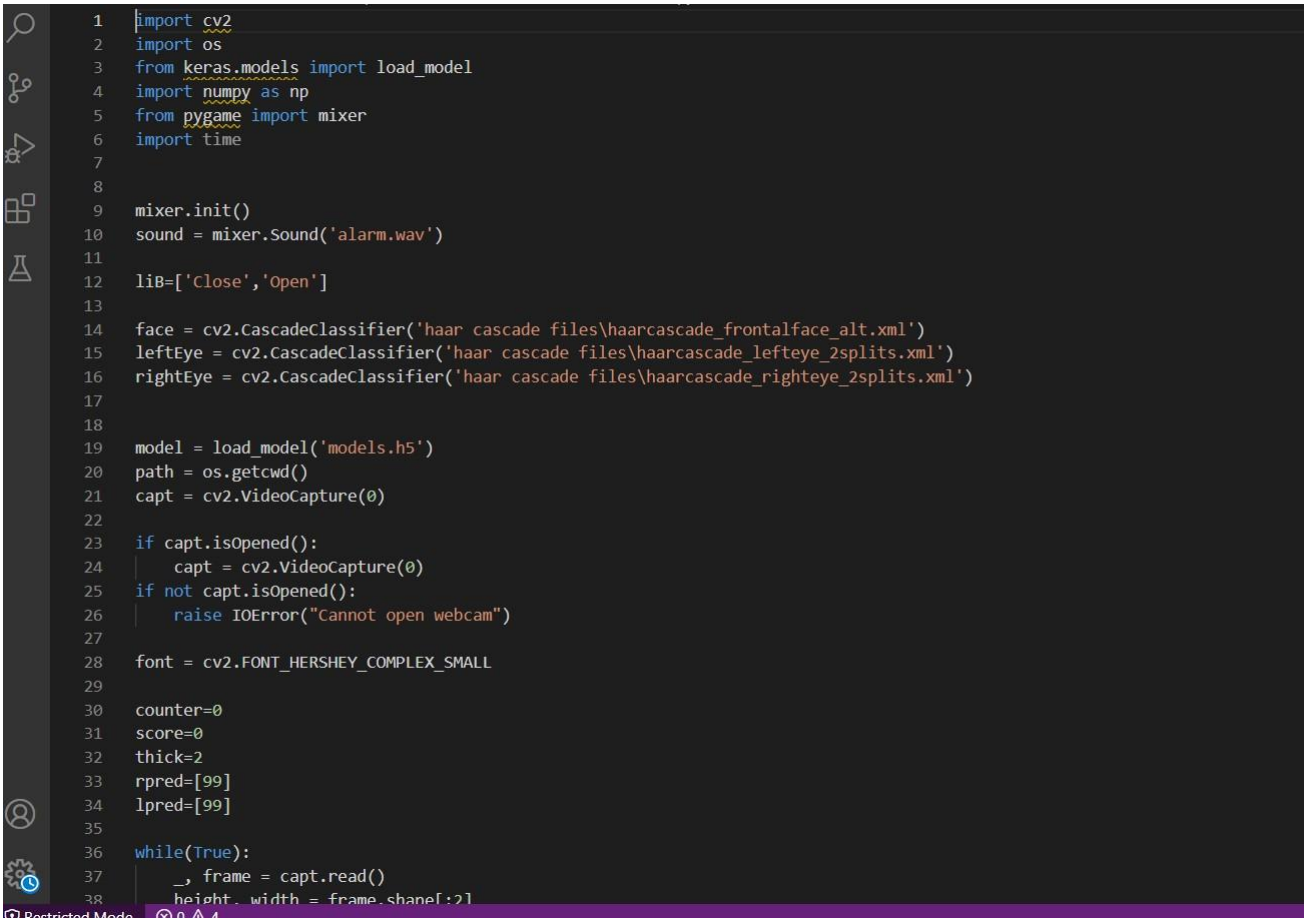
TensorFlow could be a machine learning and computing based softwares libraries that is free to use and is open source. They can be utilised for spread of many application, but are mainly focused in deep-neural networks inference as well as training . The Google's Brain office whole team developed Tensor-Flow only for Google's use in production and research. In 2015, the primary version was launched underneath the Apache License 2.0. In September 2018, Google launched TensorFlow 2.0, an improved version of TensorFlow. TensorFlow is compatible with a large range of

programming languages, including Python, C++, Java, and JavaScript. This ver-satilitylend's itself to a big variety of application in a very style of industries.

Google Brains 2ndgen. structure is known as TensorFlow. TensorFlow may operate on several CPUsiand GPUsi, though the ref. model, which runs on a single device (along options as SYCL and CUDA extension for general-purpose on graphic process-ing units). TensorFlow is there for macOS, 64-bit Lnx, Windows, and Android and iOS

mobile platforms. Its adaptable architecture enables computing to be released on a large scale of platforms (CPU's, GPU's, TPU's), from PCs to server cluster to mobiles and edge device. Stateful dataflow graphs are used to represent TensorFlow calculations. TensorFlow gets its name from computations that those neural network execute on multi-dimensional data array.

CODING, TESTING

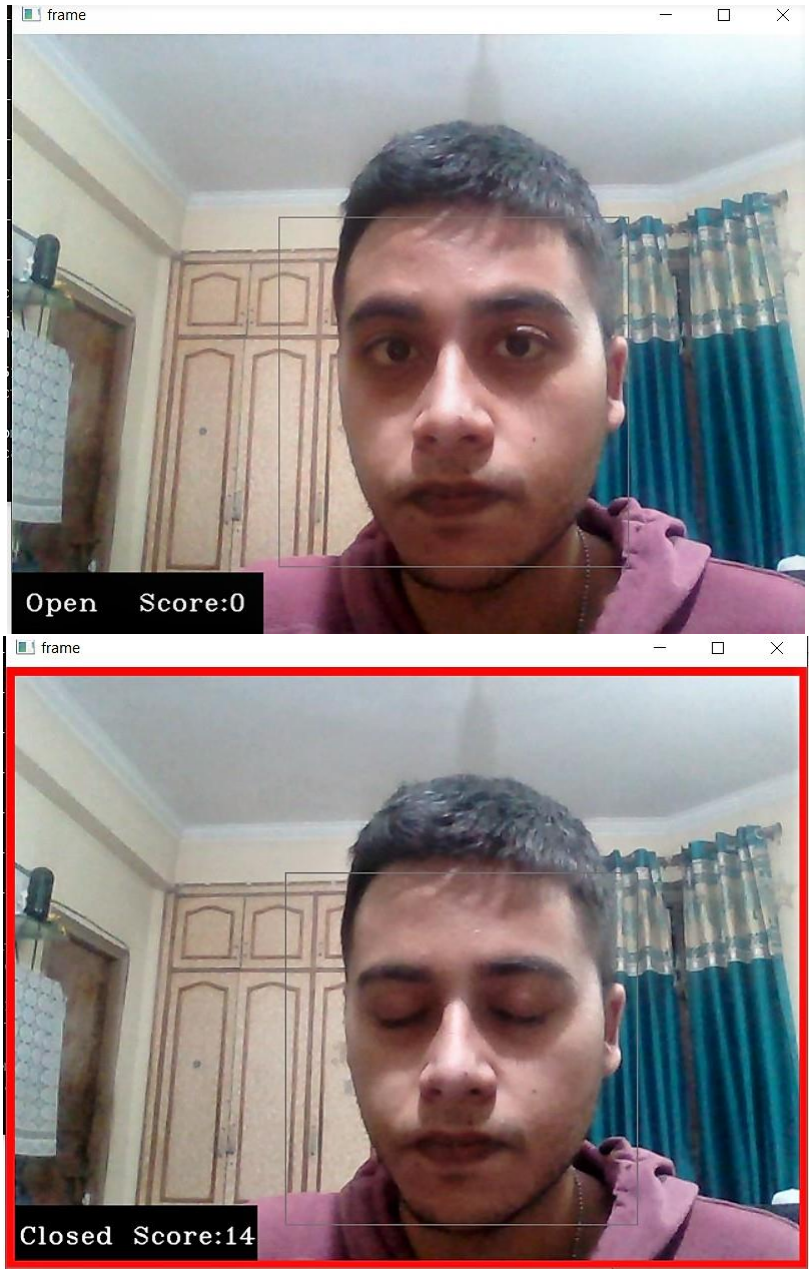


```python
import cv2
import os
from keras.models import load_model
import numpy as np
from pygame import mixer
import time


mixer.init()
sound = mixer.Sound('alarm.wav')

liB=['Close','Open']

face = cv2.CascadeClassifier('haar cascade files\haarcascade_frontalface_alt.xml')
leftEye = cv2.CascadeClassifier('haar cascade files\haarcascade_lefteye_2splits.xml')
rightEye = cv2.CascadeClassifier('haar cascade files\haarcascade_righteye_2splits.xml')


model = load_model('models.h5')
path = os.getcwd()
capt = cv2.VideoCapture(0)

if capt.isOpened():
    capt = cv2.VideoCapture(0)
if not capt.isOpened():
    raise IOError("Cannot open webcam")

font = cv2.FONT_HERSHEY_COMPLEX_SMALL

counter=0
score=0
thick=2
rpred=[99]
lpred=[99]

while(True):
    _, frame = capt.read()
    height, width = frame.shape[:2]
```

```
        height, width = frame.shape[:2]

        #convert the image to grayscale
        graysc = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        #this will return x,y coordinates , height , width of the boundary boxes object
        faces = face.detectMultiScale(graysc,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
        left_eye = leftEye.detectMultiScale(graysc)
        right_eye = rightEye.detectMultiScale(graysc)

        cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) , thickness=cv2.FILLED )

        #drawing boundary boxes over each face while iterating
        for (x,y,w,h) in faces:
            cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )


        for (x,y,w,h) in right_eye:
            r_eye=frame[y:y+h,x:x+w]
            counter=counter+1
            r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
            r_eye = cv2.resize(r_eye,(24,24))
            r_eye= r_eye/255
            r_eye=  r_eye.reshape(24,24,-1)
            r_eye = np.expand_dims(r_eye,axis=0)
            rpred = np.argmax(model.predict(r_eye))
            if(rpred==1):
                liB='Open'
            if(rpred==0):
                liB='Closed'
            break

        for (x,y,w,h) in left_eye:
            l_eye=frame[y:y+h,x:x+w]

            l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
            l_eye = cv2.resize(l_eye,(24,24))
```

```
            if(lpred==0):
                liB='Closed'
            break

        if(rpred==0 and lpred==0):
            score = score+1
            cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)

        else:
            score = score-1
            cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)


        if (score < 0):
            score = 0
        cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
        if (score > 10) :
            #person is feeling sleepy so we beep the alarm

            cv2.imwrite(os.path.join(path,'image.jpg'),frame)
            try:
                sound.play()

            except:  # isplaying = False
                pass
            if(thick<16):
                thick= thick+2
            else:
                thick=thick-2
                if(thick<2):
                    thick=2
            cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thick)
        cv2.imshow('frame',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    capt.release()
    cv2.destroyAllWindows()
```

5.1 OUTPUT



CONCLUSION

The Drowsiness road accidents , historically viewed as a necessary problem on normal roads, highways or any other vehicular area ,so to avoid those type of road accidents that can be completely automated through an efficient software program. The benefits of implementing this technology would help in achieving unhesitating state where all the problems or bug's are resolved . The system extracts hidden knowledge from data and helps in implementing the process.

This system can be further enhanced and expanded.

FUTURE ENHANCEMENT

The future works may focus on the utilization of outer factors such as vehicle states, sleeping hours, weather conditions, mechanical data, etc, for fatigue measurement. Driver drowsiness pose a major threat to highway safety, and the problem is particularly severe for commercial motor vehicle operators. Twenty-four hour operations, high annual mileage, exposure to challenging environmental conditions, and demanding work schedules all contribute to this serious safety issue

REFERENCES

1. *G Chandan, Ayush Jain, Harsh Jain, et al. Real time object detection and tracking using deep learning and opencv. In 2018 International Conference on inventive research in computing applications (ICIRCA), pages 1305–1308. IEEE, 2018.* https://ieeexplore.ieee.org/abstract/document/8597266

2. *Pramod Kaler. Study of grayscale image in image processing. International Journal of Recent and Innovation Trends in Computing and Communication, 4(11):309–311, 2016*

3. *Ramandeep Kaur and Ankit Guleria. Digital eye strain detection system based on svm. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pages 1114–1121. IEEE, 2021.* (https://ieeexplore.ieee.org/abstract/document/9453085)

4. *Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1867–1874, 2014.*

5. *Hankil Kim, Jinyoung Kim, and Hoekyung Jung. Convolutional neural network based image processing system. Journal of information and communication convergence engineering, 16(3):160–165, 2018.* https://www.koreascience.or.kr/article/JAKO201831854885756.page

6. *Tereza Soukupova and Jan Cech. Eye blink detection using facial landmarks. In 21st computer vision winter workshop, Rimske Toplice, Slovenia, 2016.*

7. *About Supervised Learning* https://www.ibm.com/cloud/learn/supervised-learning

8. *About Deep Learning* https://in.mathworks.com/campaigns/offers/deep-learning-with- matlab.html.

9. *About OpenCV* https://opencv.org/about/

10. *About Digital image processing* https://sisu.ut.ee/imageprocessing/book/