Drowsiness detection system

A PROJECT REPORT

Submitted by

BATHALA VISHAL [Reg No: RA1911003010495]
ADIT MAHAJAN [Reg No: RA1911003010497]

Under the guidance of

Mrs. Jayapradha J
(Assistant Professor, Department of Computing Technologies)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTING TECHNOLOGIES

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**Drowsiness Detection System**" is the bonafide work of "**Bathala Vishal [Reg No: RA1911003010495], Adit Mahajan [Reg No:RA19110030497]**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

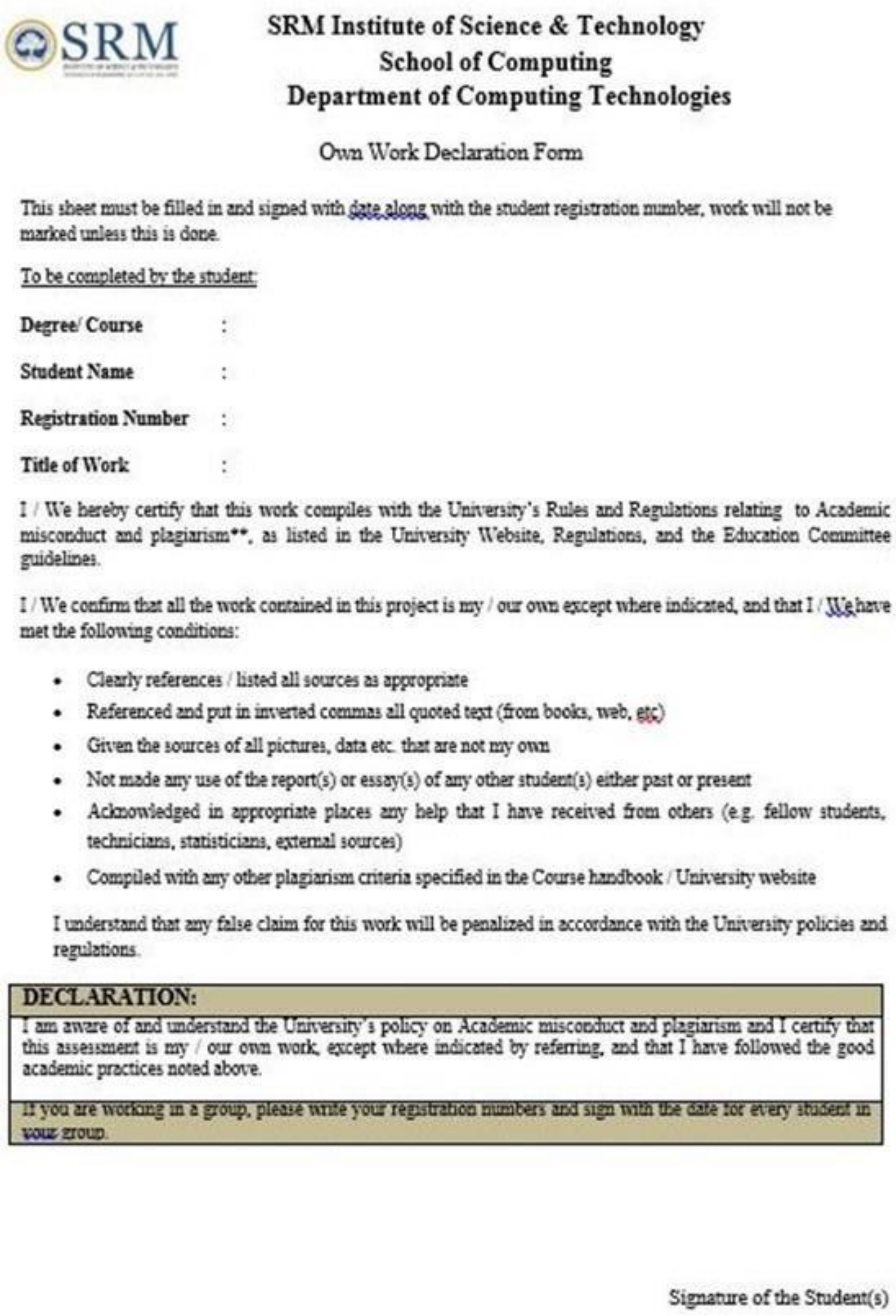
GUIDE
Mrs. J. Jayapradha
Assistant Professor
Dept. of Computing Technologies

PANEL HEAD
Dr.M.Baskar
Associate Professor
Dept. of Computing Technologies

HEAD OF THE DEPARTMENT
Dr. M. Pushpalatha
Professor
Dept. of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER



SRM Institute of Science & Technology
School of Computing
Department of Computing Technologies

Own Work Declaration Form

This sheet must be filled in and signed with date along with the student registration number, work will not be marked unless this is done.

To be completed by the student:

Degree/ Course :

Student Name :

Registration Number :

Title of Work :

I / We hereby certify that this work compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this project is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

Signature of the Student(s)

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We are indebted to **Dr. T.V. Gopal**, Dean, College of Engineering & Technology, SRM Institute of Science and Technology, for all the help provided to carry out the Project Work. We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her invaluable support. We wish to thank **Dr. M. Pushpalatha**, Professor & Head, Department of Computing Technologies, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Panel Head, **Dr.M.Baskar**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for his inputs during the project reviews. We register our immeasurable thanks to our Faculty Advisor, **Dr.K.Alice**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Mrs. Jayapradha J**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us an opportunity to pursue our project under her mentorship. She provided us the freedom and support to explore the research topics of our interest.

We sincerely thank staff members and students of the Computing Technologies Department, SRM Institute of Science and Technology, for their help during our project work. Finally, we would like to thank our parents, our family members and our friends for their unconditional love, constant support and encouragement.

Bathala Vishal
Adit Mahajan

ABSTRACT

In today's world, one of the most sought-after methods to reduce the likelihood of automobile collisions is improved driver safety. To ensure the integrity of the safety system is the primary focus of this project. We are now detecting the eye blinks of the driver in order to improve safety. Based on this information, we are assessing the driver's status and controlling the vehicle accordingly. The primary objective of this project is to design a non-invasive device that is capable of identifying weariness in any human and is able to sound an alert at the appropriate time. When travelling long distances, drivers who do not take breaks at regular intervals run a significant danger of becoming drowsy, which is a state that they frequently fail to notice in a timely manner. This device will use an eye-tracking camera to keep an eye on the driver, and a custom algorithm, we will be able to detect symptoms of driver weariness early enough to prevent the individual from falling asleep while behind the wheel. As a result, the results of this project will be useful for identifying driver fatigue in advance as well as generate an audible beep as a warning signal. The Drowsiness Detection Device, also known as DDS, is an effective system that is able to determine the condition of a person while they are seated in a vehicle. A Drowsiness Detection System is a piece of automotive safety technology that assists drivers avoid getting into accidents that are brought on by the driver becoming sleepy.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In today's world, one of the most sought-after methods to reduce the likelihood of automobile collisions is improved driver safety. To ensure the integrity of the safety system is the primary focus of this project. We are now detecting the eye blinks of the driver in order to improve safety. Based on this information, we are assessing the driver's status and controlling the vehicle accordingly. The primary objective of this project is to design a non-invasive device that is capable of identifying weariness in any human and is able to sound an alert at the appropriate time. When travelling long distances, drivers who do not take breaks at regular intervals run a significant danger of becoming drowsy, which is a state that they frequently fail to notice in a timely manner. This device will use an eye-tracking camera to keep an eye on the driver, and a custom algorithm, we will be able to detect symptoms of driver weariness early enough to prevent the individual from falling asleep while behind the wheel. As a result, the results of this project will be useful for identifying driver fatigue in advance as well as generate an audible beep as a warning signal. The Drowsiness Detection Device, also known as DDS, is an effective system that is able to determine the condition of a person while they are seated in a vehicle. A Drowsiness Detection System is a piece of automotive safety technology that assists drivers avoid getting into accidents that are brought on by the driver becoming sleepy.

CHAPTER 2

LITERATURE REVIEW

[1]: In their research paper titled "A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration," Feng you, xiaolong li, unbo gong, haiwei wang, and hongyi li make a The authors propose an algorithm for detecting drowsiness in drivers in real time. that takes into account the unique characteristics of each individual driver. They employed Dlib toolkit in order to detect the face region after constructing a CNN with deep layers. This network was used to detect the face region. They only worked during the day, which had an impact on the accuracy during the night.

[2]: Capacitive sensors were used by Mohammad Amin Assari and Mohammad Rahmati in their paper "Driver Drowsiness Detection Using Face Expression Recognition." Despite the fact that this method requires a lot of hardware as well as makes travelling uncomfortable due to sensors, the study discovered that their method had a very high level of accuracy.

[3]: In their research paper titled "Driver Fatigue Detection Using Situation Recognition Learning Approach," Jongmin Yu, Sangwoo Park, Sangwook Lee, and Moongu Jeon used techniques such as transfer learning, self - learning, and convolutional neural network; however, the accuracy they obtained was poor.

[4]: In their research work titled "A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers," the authors Anirban Dasgupta, Daleef Rahman, and aurobinda routray employed perclos and calculation algorithm and verification. However, it takes a long time to execute.

[5]: Dian artanto, prayadi sulistyanto, deradjad pranowo and ervan erry pramesta in their paper "Drowsiness Detection System based on Eye-closure using A Low-Cost EMG and ESP8266" they used Low-cost EMG

ESP8266 the accuracy of this model is good but it cannot run in realtime.

[6]: In their research work titled "Sleepy Eye's Recognition for Drowsiness Detection," Shinfeng D. Lin, Jia-Jen Lin, and Chih-Yao Chung employed a using Classifier with cascading Adaboost Haar-like features and an Active Model to detect drowsiness in the subjects (ASM) This model does not include any form of training, hence the results it produces are not very accurate.

[7]: Anilkumar c, mansoor ahmed, sahana R, thejashwini R and Anisha p.s in their paper " Design of Drowsiness, Heart Beat Detection System and Alertness Indicator for Driver Safety" They used MATLAB, R-peak detection algorithm, Image processing toolbox, Computer vision cascade Toolboxit also detects heartbeat but the hardware is huge makes the driver uncomfortable.

[8]: Detection of Drowsiness from Facial Images in Real-Time Video Media using Nvidia Jetson Nano is a paper written by Petchara Inthanon and Surasak Mungsing. In this paper, they used Nvidia Jetson Nano, which is a tool that accurately evaluates images by tracking closing eye motions for more than 35 frames per second (FPS) or 1.5 seconds.

An infrared camera is essential for face detection, but finding one can be challenging.

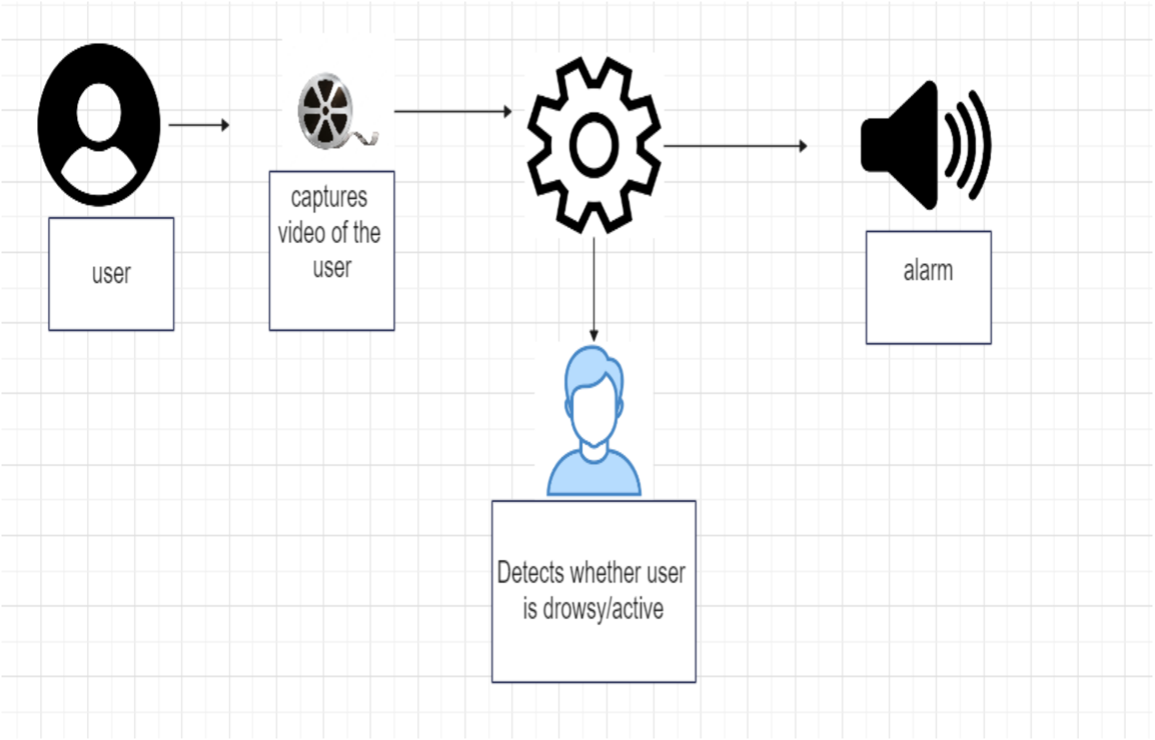
[9]: Ghulam hyder, bhawanui shankar Chowdary and khuhed memon in their paper " THE SMART AUTOMOBILE (SAM): An Application Based on Drowsiness Detection, Alcohol Detection, Vital Sign Monitoring and Lane based Auto Drive to avoid Accidents" they used Raspbian Operating System

OpenCV, Python, Dlib library and it also detects vitals.

[10]: In their paper "Drowsiness Detection using Photoplethysmography Signal," Deepu Kurian, Johnson Joseph P., Krishnaja Radhakrishnan, and Arun A.balakrishnan used Acquisition of signal using DAQ and PPG technique. They The algorithm for detecting peaks in data sets developed by got works nearly flawlessly, but it becomes difficult to carry and maintain physical things like ppg and electrodes.

CHAPTER 3

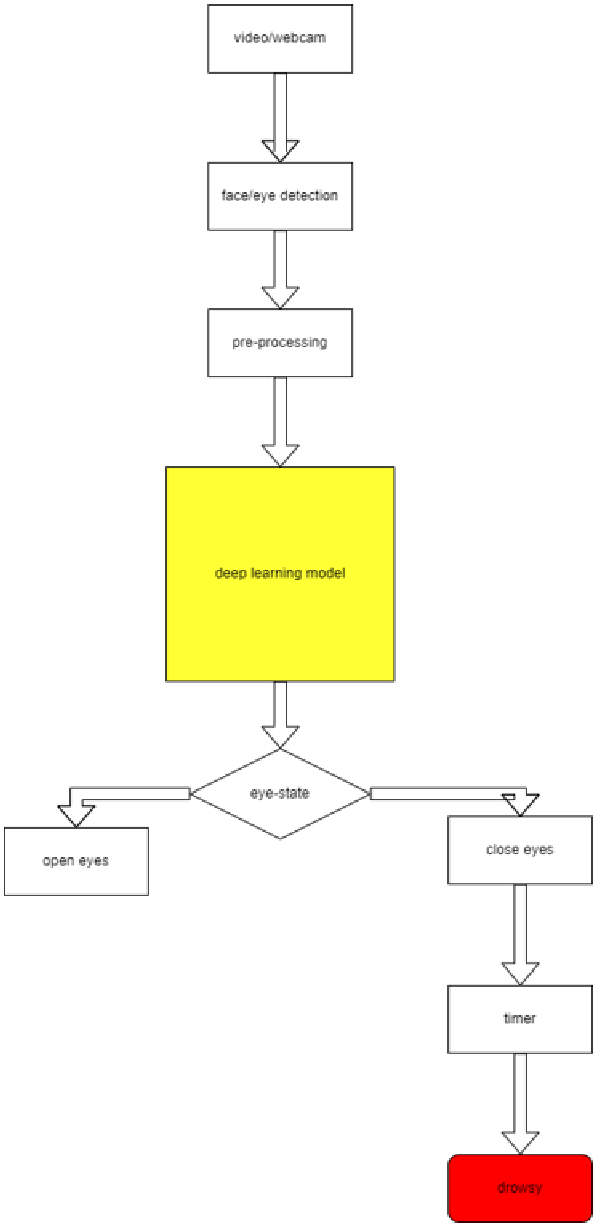
ARCHITECTURE AND DESIGN


user
captures video of the user
Detects whether user is drowsy/active
alarm

When this occurs, first a video of the user is gathered, then it is decided whether or not the user is fatigued, and finally, an alarm is activated in accordance with that result if it is discovered that the user is sleeping.

FLOWCHART DIAGRAM:

Here first the videocam is enabled to capture the drivers face, after that it detects the users eyes there pre-processing takes place it sends the results i.e whatever it detects will send it to the deep learning model we have used.the deep learning model detects the users eye state ,if the eyes are closed only then the timer starts and if the timer exceeds its time limit then the alarm rings to warn the driver.



CHAPTER 4

METHODOLOGY

We are breaking the project down into two phases: the first is model building, and the second is real-time face identification. For model building, we make use of keras and transfer learning (a deep learning method). OpenCV is what we use for real-time facial detection, and an alarm sound is what we utilise to inform the driver.

In order to train our model, we made use of the transfer learning method, and we used OpenCV for the actual implementation in real time.

Modules used in this project are

1. OS:-

The Python OS module gives the user the ability to communicate with the operating system by providing the facilities necessary to do so. It provides a large number of helpful OS functions that may be used to carry out OS-based activities and get information about the operating system that is connected to those actions. Python's operating system is covered by its basic utility modules.

2. SHUTIL:-

The Python module known as shutil offers a wide variety of high-level functions for performing operations on files and groups of files. It is part of Python's standard collection of utility modules. This module contributes to the automation of the process of copying files and directories as well as removing them.

The data set consits of data images in single folder, to create different set of folders for open eyes and closed eyes based on annotations on the image we use shutil to split the data into different folders.

3. GLOB:-

Through the use of the glob command, we can look for a specific file pattern, or more efficiently, we can look for files whose filenames match a given pattern by using wildcard characters. Each of these is a viable option.

4. RANDOM:-

The Python Random module is an integral part of the Python programming language and may be used to produce random numbers. The fact that these numbers are just "pseudo-random" indicates that they are not actually random. This module may be used to carry out arbitrary tasks like as the generation of random numbers, the printing of a random value for a list or string, and other similar activities.

5. TQDM:-

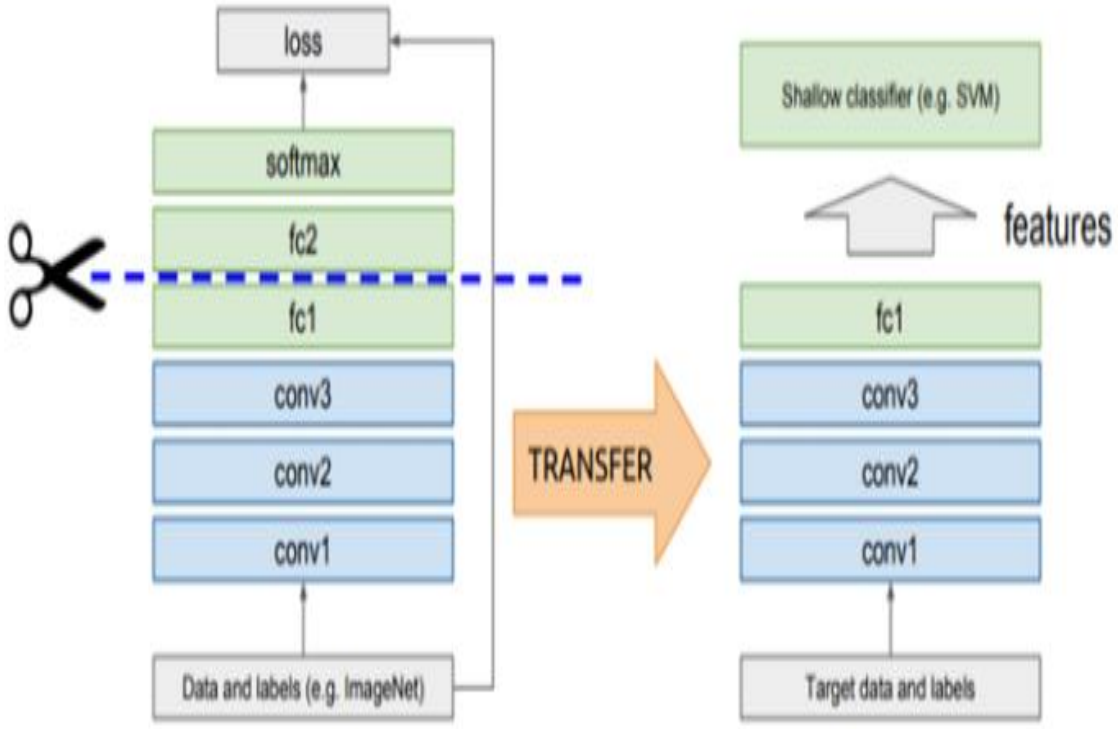
Python's tqdm library is responsible for the creation of progress metres and bars, and it may also be called a progress metre. The word "progress" comes from the Arabic word "taqaddum," which is where the term "tqdm" originated from.

In this whole project we used this module in different parts to show the status of process(eg- downloading and coping files).

6. TRANSFER LEARNING:-

Ensemble learning is a type of artificial neural network where a model that's been created for one job is utilised again as the starting point for a model that is being built for a various task.

Considering the enormous processing power and duration assets taken to create CNN architecture on these difficulties but from the huge jumps in pre-trained models' ability provide on problems related, it is a popular strategy in transfer learning to use which was before models as the starting place on pattern recognition and machine learning processing tasks. This is because pre-trained models significantly outperform untrained models when it comes to resolving



The insceptionv3 model served as the foundation for our work.

First, we flattened the inceptionV3 model, performed a rectified activating, took out a level, and implemented the softmax to the model. Afterwards when, humans deleted the top of the framework and used the softmax layer.

7. OpenCV:-

OpenCV is a library of code whose main purpose is to facilitate real-time computer vision.. It's a public library that can be accessed from various devices without cost..

In order to identify the user's face and eyes in real time, we integrated the OpenCV module. To locate the user's face in an image, we used the face cascade from the cv2.cascadeclassifier (haarcascade frontalface default.xml). For the detection of the user's eyes in a frame from the face, we used the eye cascade from the cv2.cascadeclassifier (haarcascade eye.xml).

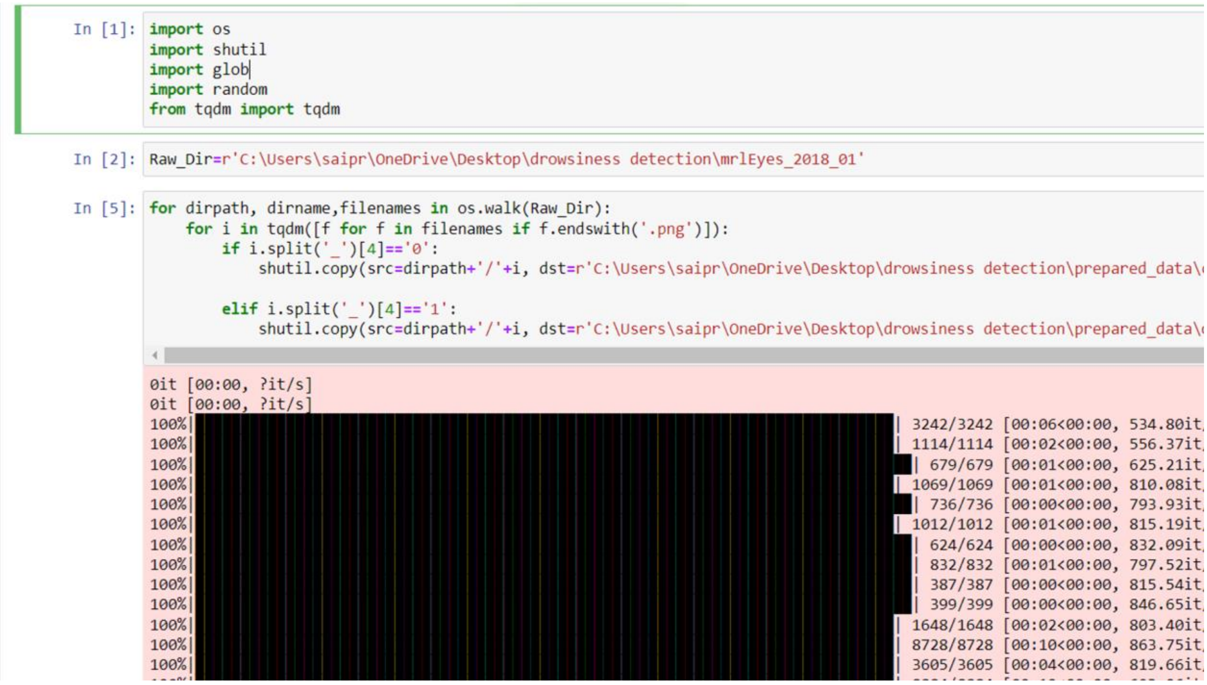
The model will forecast the state of an eye (that is, whether it is open or closed) once it has received the input from openCV, and it will then set off the alarm in accordance with its prediction.

(alarm has a timer of 5 seconds, which means that if the user's eye is detected to be closed for 5 seconds, the alarm will go off).

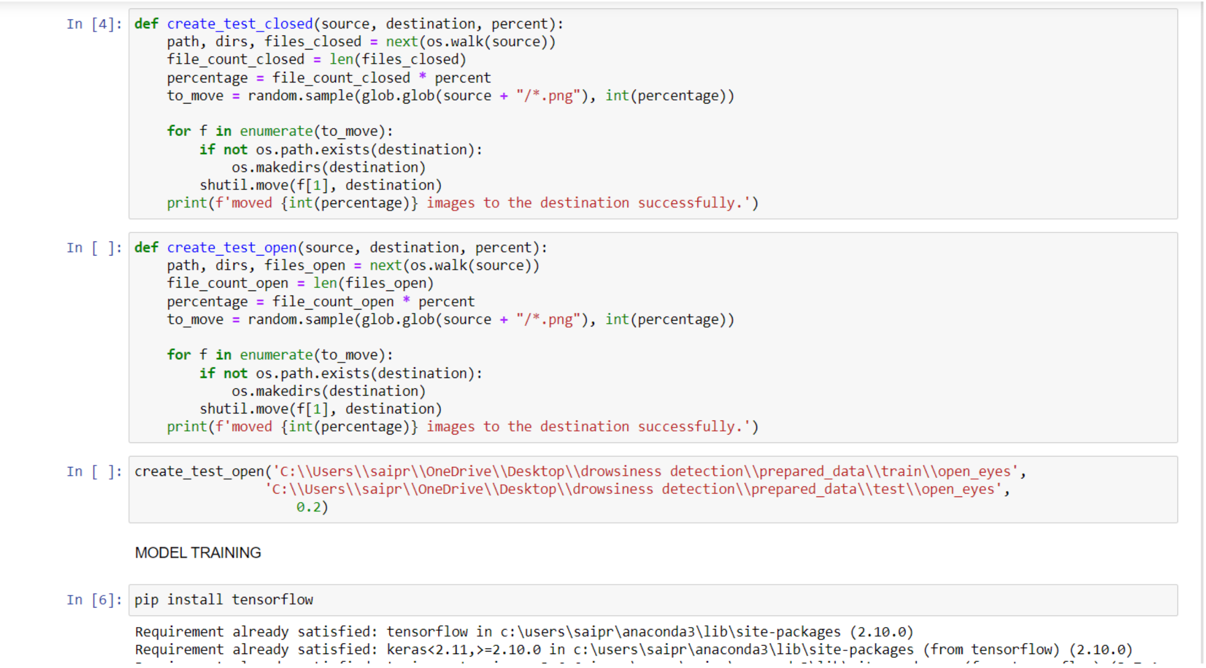
CHAPTER 5

CODING AND TESTING

CODE:



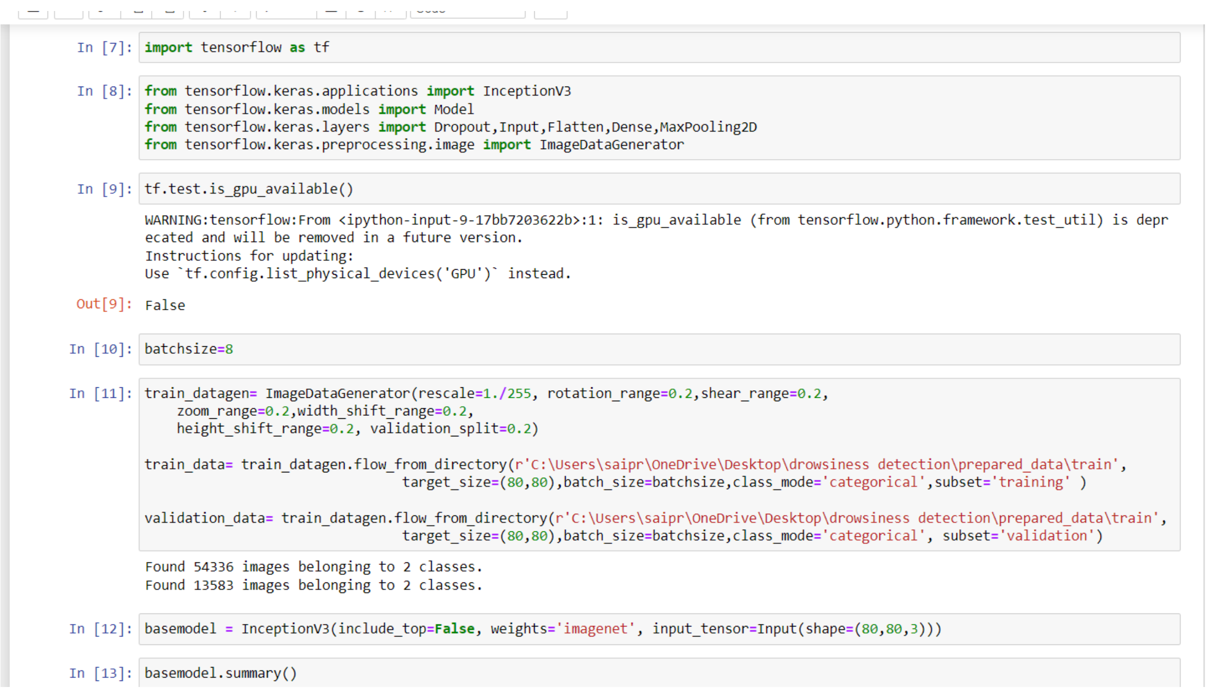
```
In [1]: import os
        import shutil
        import glob
        import random
        from tqdm import tqdm

In [2]: Raw_Dir=r'C:\Users\saipr\OneDrive\Desktop\drowsiness detection\mrlEyes_2018_01'

In [5]: 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'C:\Users\saipr\OneDrive\Desktop\drowsiness detection\prepared_data\

                elif i.split('_')[4]=='1':
                    shutil.copy(src=dirpath+'/'+i, dst=r'C:\Users\saipr\OneDrive\Desktop\drowsiness detection\prepared_data\

        0it [00:00, ?it/s]
        0it [00:00, ?it/s]
        100%|████████████████████████████████████████| 3242/3242 [00:06<00:00, 534.80it
        100%|████████████████████████████████████████| 1114/1114 [00:02<00:00, 556.37it
        100%|█████████████████████████████████████████| 679/679 [00:01<00:00, 625.21it
        100%|████████████████████████████████████████| 1069/1069 [00:01<00:00, 810.08it
        100%|█████████████████████████████████████████| 736/736 [00:00<00:00, 793.93it
        100%|████████████████████████████████████████| 1012/1012 [00:01<00:00, 815.19it
        100%|█████████████████████████████████████████| 624/624 [00:00<00:00, 832.09it
        100%|█████████████████████████████████████████| 832/832 [00:01<00:00, 797.52it
        100%|█████████████████████████████████████████| 387/387 [00:00<00:00, 815.54it
        100%|█████████████████████████████████████████| 399/399 [00:00<00:00, 846.65it
        100%|████████████████████████████████████████| 1648/1648 [00:02<00:00, 803.40it
        100%|████████████████████████████████████████| 8728/8728 [00:10<00:00, 863.75it
        100%|████████████████████████████████████████| 3605/3605 [00:04<00:00, 819.66it
```



```
In [4]: def create_test_closed(source, destination, percent):
            path, dirs, files_closed = next(os.walk(source))
            file_count_closed = len(files_closed)
            percentage = file_count_closed * percent
            to_move = random.sample(glob.glob(source + "/*.png"), int(percentage))

            for f in enumerate(to_move):
                if not os.path.exists(destination):
                    os.makedirs(destination)
                shutil.move(f[1], destination)
            print(f'moved {int(percentage)} images to the destination successfully.')

In [ ]: def create_test_open(source, destination, percent):
            path, dirs, files_open = next(os.walk(source))
            file_count_open = len(files_open)
            percentage = file_count_open * percent
            to_move = random.sample(glob.glob(source + "/*.png"), int(percentage))

            for f in enumerate(to_move):
                if not os.path.exists(destination):
                    os.makedirs(destination)
                shutil.move(f[1], destination)
            print(f'moved {int(percentage)} images to the destination successfully.')

In [ ]: create_test_open('C:\\Users\\saipr\\OneDrive\\Desktop\\drowsiness detection\\prepared_data\\train\\open_eyes',
                        'C:\\Users\\saipr\\OneDrive\\Desktop\\drowsiness detection\\prepared_data\\test\\open_eyes',
                            0.2)
```

MODEL TRAINING

```
In [6]: pip install tensorflow

        Requirement already satisfied: tensorflow in c:\users\saipr\anaconda3\lib\site-packages (2.10.0)
        Requirement already satisfied: keras<2.11,>=2.10.0 in c:\users\saipr\anaconda3\lib\site-packages (from tensorflow) (2.10.0)
```



```
In [7]: import tensorflow as tf
```

```
In [8]: 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.preprocessing.image import ImageDataGenerator
```

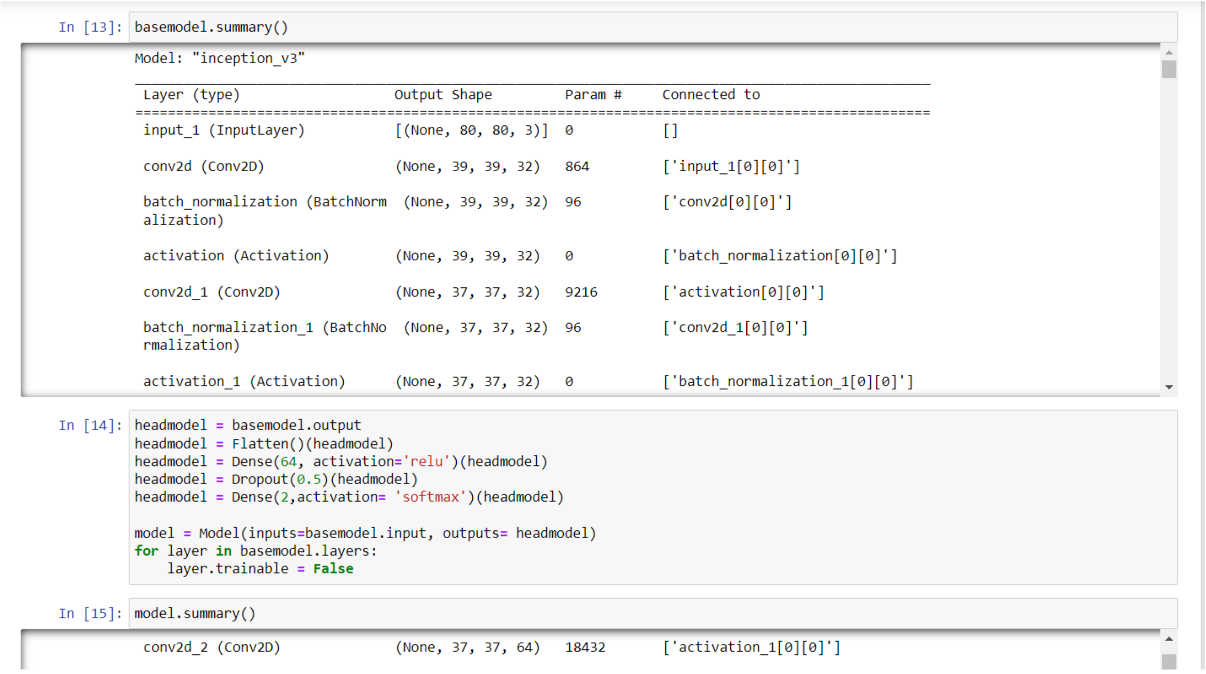
```
In [9]: tf.test.is_gpu_available()

        WARNING:tensorflow:From <ipython-input-9-17bb7203622b>:1: is_gpu_available (from tensorflow.python.framework.test_util) is depr
        ecated and will be removed in a future version.
        Instructions for updating:
        Use `tf.config.list_physical_devices('GPU')` instead.

Out[9]: False
```

```
In [10]: batchsize=8
```

```
In [11]: train_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'C:\Users\saipr\OneDrive\Desktop\drowsiness detection\prepared_data\train',
                                         target_size=(80,80),batch_size=batchsize,class_mode='categorical',subset='training' )

         validation_data= train_datagen.flow_from_directory(r'C:\Users\saipr\OneDrive\Desktop\drowsiness detection\prepared_data\train',
                                         target_size=(80,80),batch_size=batchsize,class_mode='categorical', subset='validation')

         Found 54336 images belonging to 2 classes.
         Found 13583 images belonging to 2 classes.
```

```
In [12]: basemodel = InceptionV3(include_top=False, weights='imagenet', input_tensor=Input(shape=(80,80,3)))
```

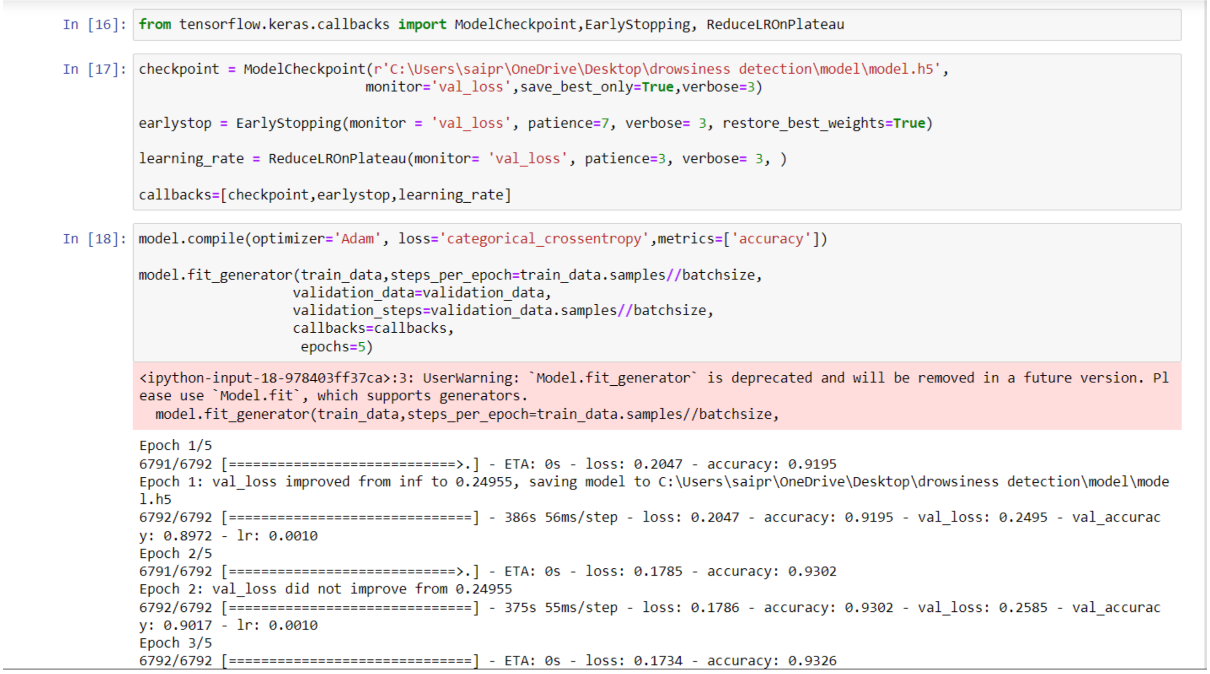
```
In [13]: basemodel.summary()
```



```
In [13]: basemodel.summary()

         Model: "inception_v3"
         __________________________________________________________________________________________________
          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]']
```

```
In [14]: headmodel = basemodel.output
         headmodel = Flatten()(headmodel)
         headmodel = Dense(64, activation='relu')(headmodel)
         headmodel = Dropout(0.5)(headmodel)
         headmodel = Dense(2,activation= 'softmax')(headmodel)

         model = Model(inputs=basemodel.input, outputs= headmodel)
         for layer in basemodel.layers:
             layer.trainable = False
```

```
In [15]: model.summary()

          conv2d_2 (Conv2D)              (None, 37, 37, 64)   18432       ['activation_1[0][0]']
```



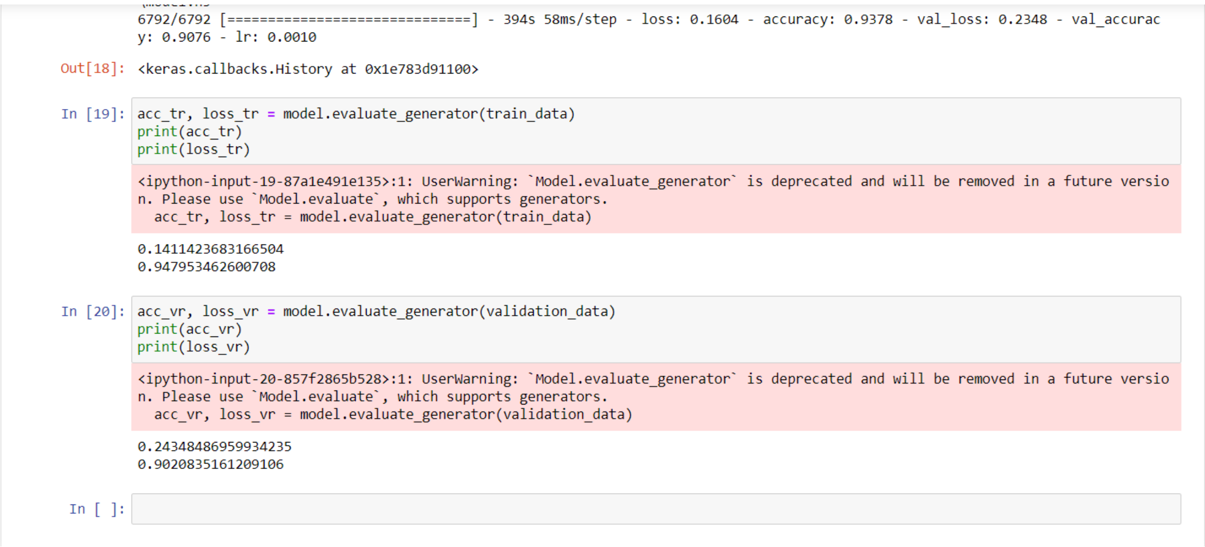
```
In [16]: from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping, ReduceLROnPlateau
```

```
In [17]: checkpoint = ModelCheckpoint(r'C:\Users\saipr\OneDrive\Desktop\drowsiness detection\model\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]
```

```
In [18]: model.compile(optimizer='Adam', loss='categorical_crossentropy',metrics=['accuracy'])

         model.fit_generator(train_data,steps_per_epoch=train_data.samples//batchsize,
                             validation_data=validation_data,
                             validation_steps=validation_data.samples//batchsize,
                             callbacks=callbacks,
                              epochs=5)
```

```
<ipython-input-18-978403ff37ca>:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Pl
ease use `Model.fit`, which supports generators.
  model.fit_generator(train_data,steps_per_epoch=train_data.samples//batchsize,
```

```
Epoch 1/5
6791/6792 [============================>.] - ETA: 0s - loss: 0.2047 - accuracy: 0.9195
Epoch 1: val_loss improved from inf to 0.24955, saving model to C:\Users\saipr\OneDrive\Desktop\drowsiness detection\model\mode
l.h5
6792/6792 [==============================] - 386s 56ms/step - loss: 0.2047 - accuracy: 0.9195 - val_loss: 0.2495 - val_accurac
y: 0.8972 - lr: 0.0010
Epoch 2/5
6791/6792 [============================>.] - ETA: 0s - loss: 0.1785 - accuracy: 0.9302
Epoch 2: val_loss did not improve from 0.24955
6792/6792 [==============================] - 375s 55ms/step - loss: 0.1786 - accuracy: 0.9302 - val_loss: 0.2585 - val_accurac
y: 0.9017 - lr: 0.0010
Epoch 3/5
6792/6792 [==============================] - ETA: 0s - loss: 0.1734 - accuracy: 0.9326
```



```
6792/6792 [==============================] - 394s 58ms/step - loss: 0.1604 - accuracy: 0.9378 - val_loss: 0.2348 - val_accurac
y: 0.9076 - lr: 0.0010
```

```
Out[18]: <keras.callbacks.History at 0x1e783d91100>
```

```
In [19]: acc_tr, loss_tr = model.evaluate_generator(train_data)
         print(acc_tr)
         print(loss_tr)
```

```
<ipython-input-19-87a1e491e135>:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future versio
n. Please use `Model.evaluate`, which supports generators.
  acc_tr, loss_tr = model.evaluate_generator(train_data)
```

```
0.1411423683166504
0.947953462600708
```

```
In [20]: acc_vr, loss_vr = model.evaluate_generator(validation_data)
         print(acc_vr)
         print(loss_vr)
```

```
<ipython-input-20-857f2865b528>:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future versio
n. Please use `Model.evaluate`, which supports generators.
  acc_vr, loss_vr = model.evaluate_generator(validation_data)
```

```
0.24348486959934235
0.9020835161209106
```

```
In [ ]:
```

CHAPTER 6

RESULTS AND OBSERVATIONS

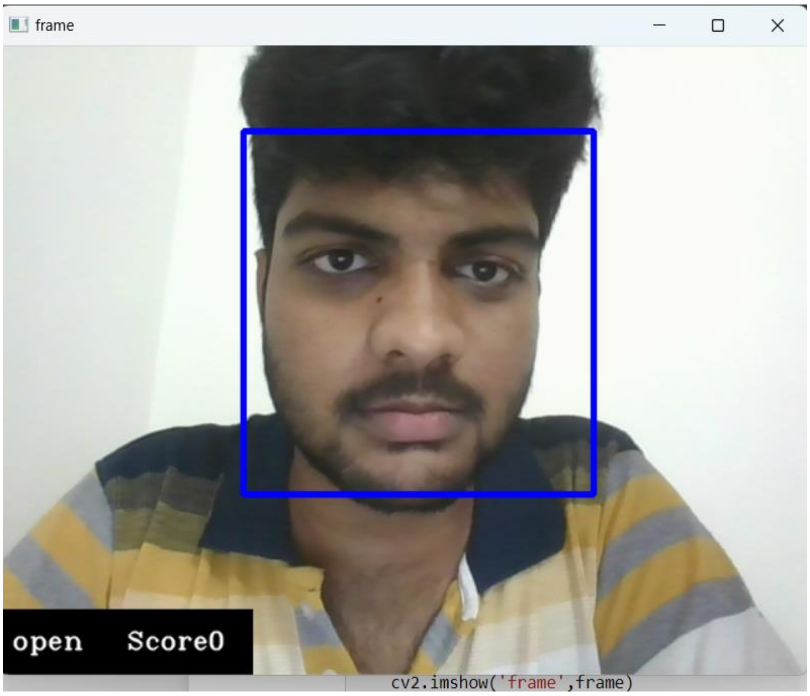
RESULTS:

The predictions that we obtained from the model are accurate in comparison to those obtained from many other models that came before it. The main advantages of this model are that it is highly compatible with the driver and does not irritate them in any way, in contrast to other models that are composed of conductive materials throughout the driver's body.

The outputs we got from the model are

PERSON1:

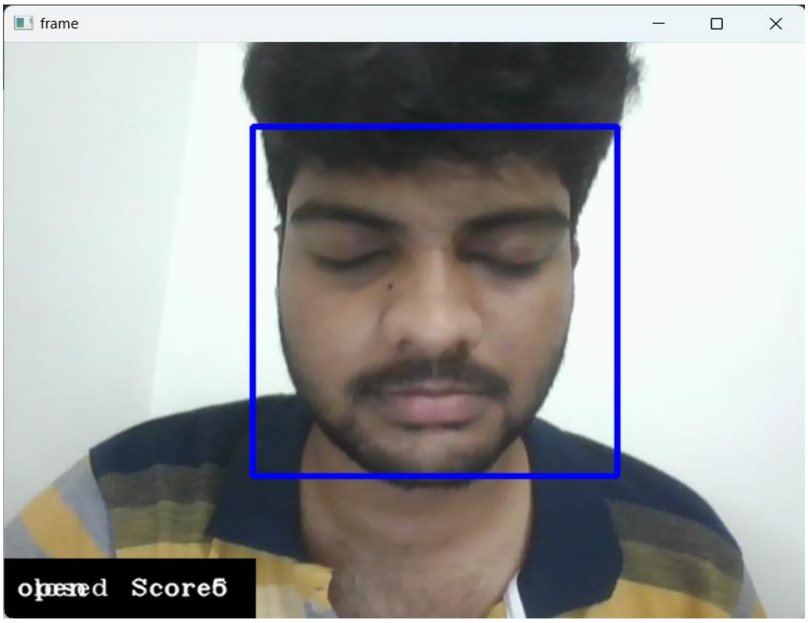
EYES OPEN:



```
open    Score0
```

```
cv2.imshow('frame',frame)
```

EYES CLOSE:

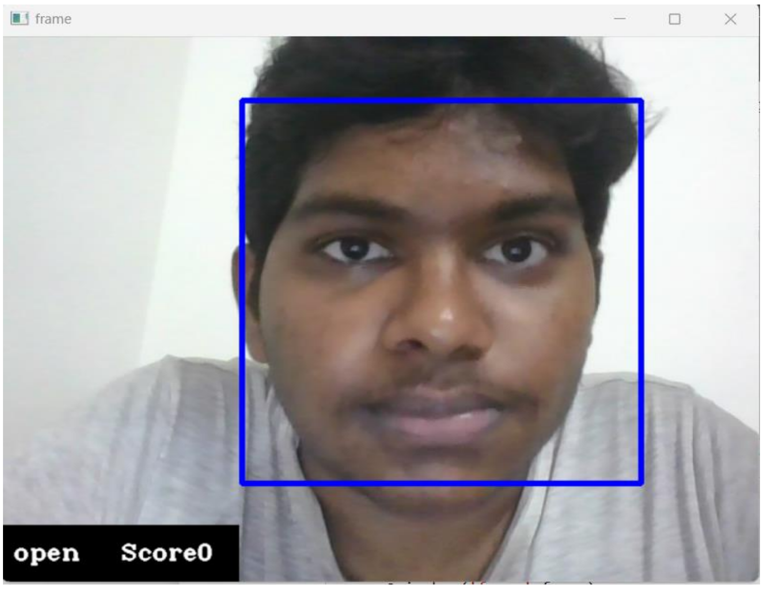


Here in the first picture the person has open eyes so the model detected it and gave a score of 0 upon closing eyes in the second picture the model detected the closed eyes with a score of 6.

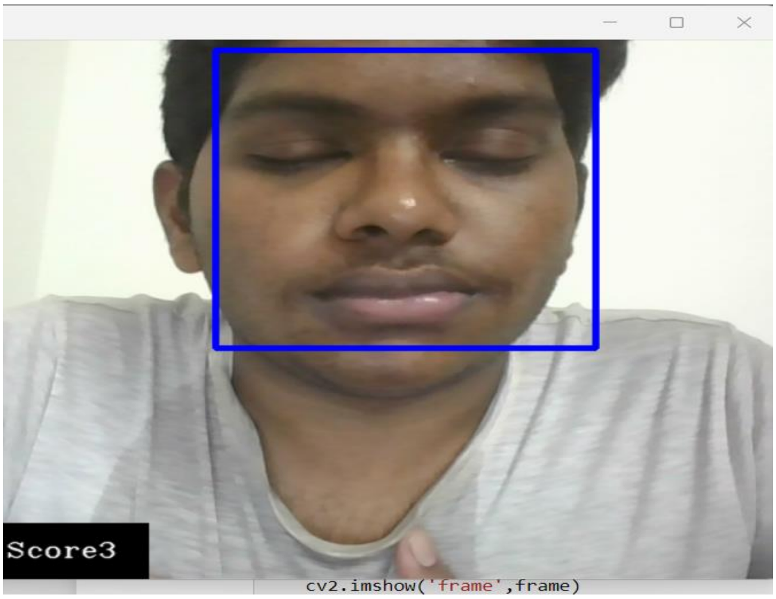
Here Score is the amount of time the person is closing his eye.

PERSON2:

EYES OPEN:

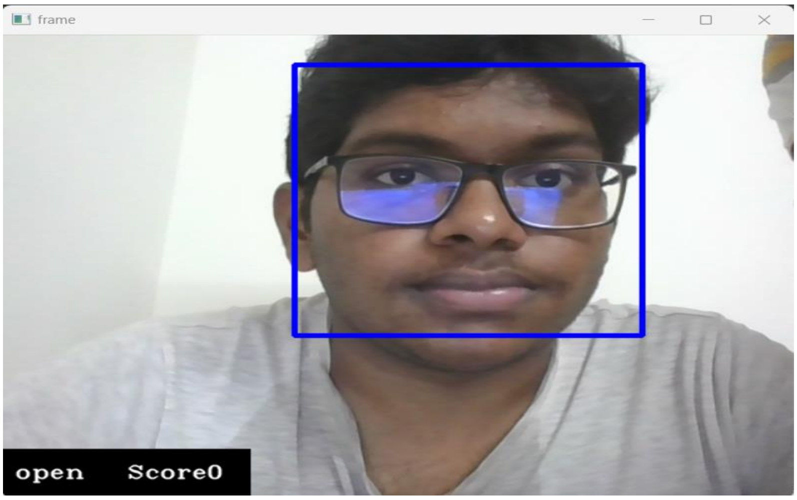


EYE CLOSE:



```
cv2.imshow('frame',frame)
```

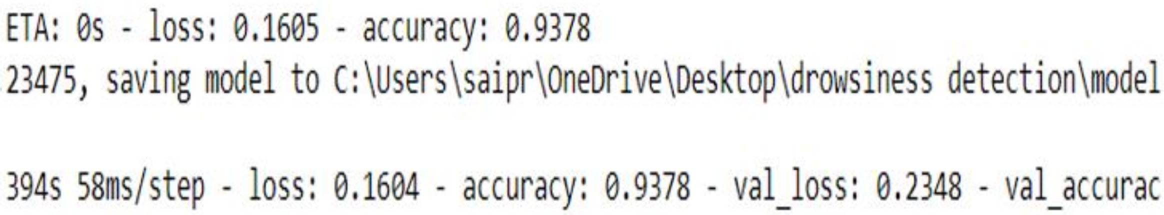
WITH GLASSES:



Here in this picture, the model detected even when the person was wearing glasses and gave the output as open eyes with a score of 0.

ACCURACY:

The accuracy obtained for this model:



```
ETA: 0s - loss: 0.1605 - accuracy: 0.9378
23475, saving model to C:\Users\saipr\OneDrive\Desktop\drowsiness detection\model

394s 58ms/step - loss: 0.1604 - accuracy: 0.9378 - val_loss: 0.2348 - val_accurac
```

CHAPTER 7

CONCLUSION

Research on an algorithm that is able to recognise drowsy driving is one of the most important things that can be done to reduce the number of people who are injured or killed in automobile accidents.It is general known that individuals may be recognised from one another in significant ways, notably in terms of the size of their eyes. One of these ways is that people can be distinguished from one another in substantial ways. When doing research on an algorithm that is based on computer vision, it is important to take into consideration the differences that might be found between different people. In this article, we provide an algorithm for reducing the amount of accidents that are brought about by drivers who are sleepy.

We were successful in attaining an accuracy score of 93.7%, which is higher than the majority of the other models. While previous models were either constructed on hardware or based on some other models with less accuracy, we were able to attain this level of accuracy by using the ideas of deep learning. Because the hardware is mounted to the driver itself, using capacitive sensors may be exceedingly unpleasant and aggravating for the user. Earlier versions, on the other hand, were constructed using resistive sensors. In addition, several of the models were unsuccessful in a variety of situations, such as poor lighting.

7.1 Future Enhancement

via the implementation of additional functionalities such as facial recognition for the purpose of driver identification or the use of a phone while driving with the intention of reducing the likelihood of unintentional dangers on the road.

CHAPTER 8

REFERENCES

[1] Verma, A., Cabrera, S., Mayorga, A., & Nazeran, H. (2013, May). “A robust algorithm for derivation of heart rate variability spectra from ECG and PPG signals.” In *2013 29th Southern Biomedical Engineering Conference* (pp. 35-36). IEEE.

[2] Yacin, S. M., Manivannan, M., & Chakravarthy, V. S. (2009). “Effect of gastric myoelectric activity on photoplethysmographic signals”. *International Journal of Recent Trends in Engineering*, *2*(7), 27.

[3] Vanisree, K., & Singaraju, J. (2011). “Automatic Detection of ECG RR Interval using Discrete Wavelet Transformation 1”.

[4] Vengatesan, K., Kumar, A., Karuppuchamy, V., Shaktivel, R., & Singhal, A. (2019, December). “Face recognition of identical twins based on support vector machine classifier”. In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 577-580). IEEE.

[5] Kavitha, P. K., & Saraswathi, P. V. (2019, December). “An Efficient Image Retrieval Using Fuzzy Based Clustering.” In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 551-555). IEEE.

[7] Tang, I., & Breckon, T. P. (2010). “Automatic road environment classification*”. IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 476-484.

[8] Jansen, P., Van Der Mark, W., Van Den Heuvel, J. C., & Groen, F. C. (2005). “Colour based off-road environment and terrain type classification*”* (pp. 61-66). Piscataway, NJ: IEEE.

[9] Choi, I., Han, S., & Kim, D. (2011, July). “Eye detection and eye blink detection using adaboost learning and grouping.*”* In 2011 *Proceedings of 20th International Conference on Computer Communications and Networks* (ICCCN) (pp. 1-4). IEEE.

[10] Liu, X., Tan, X., & Chen, S. (2012, October). “Eyes closeness detection using appearance-based methods.” *In International Conference on Intelligent Information Processing* (pp. 398-408). Springer, Berlin, Heidelberg.

vis

ORIGINALITY REPORT

6%	5%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Petchara Inthanon, Surasak Mungsing. "Detection of Drowsiness from Facial Images in Real-Time Video Media using Nvidia Jetson Nano", 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2020 Publication	1%
2	royallib.com Internet Source	1%
3	www.wwrf.ch Internet Source	1%
4	ebin.pub Internet Source	1%
5	Deepu Kurian, Johnson Joseph P.L., Krishnaja Radhakrishnan, Arun A. Balakrishnan. "Drowsiness Detection Using Photoplethysmography Signal", 2014 Fourth International Conference on Advances in Computing and Communications, 2014 Publication	<1%

6	iopscience.iop.org Internet Source	<1 %
7	Raja Mariatul Qibtiah, Zalhan Mohd Zin, Mohd Fadzil Abu Hassan, Siti Salwa Md Noor. "An Innovative Anomaly Driving Detection Strategy for Adaptive FCW of CNN Approach", 2021 IEEE 17th International Colloquium on Signal Processing & Its Applications (CSPA), 2021 Publication	<1 %
8	"Table of Contents", 2013 International Symposium on Biometrics and Security Technologies, 2013 Publication	<1 %
9	iris.unige.it Internet Source	<1 %
10	dblp.dagstuhl.de Internet Source	<1 %
11	Anirban Dasgupta, Daleef Rahman, Aurobinda Routray. "A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers", IEEE Transactions on Intelligent Transportation Systems, 2019 Publication	<1 %
12	Anshul Pinto, Mohit Bhasi, Durvesh Bhalekar, Pradyoth Hegde, Shashidhar G. Koolagudi. "A Deep Learning Approach to Detect Drowsy	<1 %

Drivers in Real Time", 2019 IEEE 16th India Council International Conference (INDICON), 2019

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		