SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

WINTER SEMESTER - 2022

IOT Based Real-Time Drowsiness Monitoring

A Report

submitted by

19BCI0181 - Vishal Haswani
19BCI0187 - Gokul Gopakumar
20BCE0724 - Deepanshu Sharma

CSE3009 – Internet of Things – J Component

B1-18

Faculty Name – Dr.A.Saraswathi Priyadharshini,

Department of Analytics,

SCOPE

TABLE OF CONTENTS

Introduction

Countless accidents have occurred as a result of driver exhaustion, tiresome road conditions, and adverse weather conditions. According to enforcement personnel monitoring the motorways and major roads here, sleep-deprived drivers are still responsible for around 40% of all traffic accidents. "This is especially relevant to exhausted drivers who fall asleep at the wheel on our roadways between midnight and 5 a.m.," stated B. Shefiq, Joint Regional Transport Officer, Perumbavur. Only expert drivers, according to B. G. Sreedevi, Chief Scientist at the National Transportation Planning and Research Centre (Natpac) of the Kerala State Council for Science, Technology, and Environment ensures that they get enough sleep before embarking on nighttime travels.

Advanced image processing, machine learning, and computational intelligence algorithms have been used to develop several methodological strategies for the identification and prediction of driver drowsiness. The goal of this research is to look at the underlying intelligence algorithms that are used to identify driver weariness, and then employ improved ways to properly detect driver exhaustion in both day and night driving situations. Although there are various DDD systems to handle this issue, none of them used the multi-cam technique with hybrid characteristics employing two cameras and heart-beat sensors, according to our limited knowledge.

A recent review article for real-time detection of driver sleepiness was published, but the study concluded that much more work is needed in this area. According to recent trends, the researcher is working on a hybrid system that combines visual and non-visual based information to improve prediction accuracy. However, the cost of hardware will rise as a result of this procedure. Despite these realities, selecting machine learning algorithms is a critical endeavour. The implementation of these driver inattention monitoring devices is always a compromise between time and spatial complexity. The Deep Learning Model suggested in this study addresses these concerns.

Literature Survey

Table 1.1 : Related Works

Title	**Authors**	**Methodology**	**Gaps**
Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection (2019)	Arefnezhad S., Samiee S., Eichberger A., Nahvi A.	This paper presents a novel feature selection method to design a non-invasive driver drowsiness detection system based on steering wheel data. The proposed feature selector can select the most related features to the drowsiness level to improve the classification accuracy. This method is based on the combination of the filter and wrapper feature selection algorithms using adaptive neuro-fuzzy inference system (ANFIS). This method was based on the idea of combining filter and wrapper feature selection methods to improve the performance in comparison with the results when each of the filter and wrapper methods has been employed individually. This combination was	It exploits the advantages of four different filter feature selection methods to improve the reliability of the classification results. The SVM algorithm used is also a robust algorithm which is suitable for real world data sets. As the detection is based on steering wheel data, it can be inaccurate as each person may have different tendencies to how they drive their vehicle.

		performed in the structure of the designed ANFIS. Four different filter indexes were calculated for each feature and used as inputs to ANFIS to produce importance degree of each feature.	
Driver Drowsiness Detection (IJERT) NCAIT – 2020	V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P	Machine learning techniques have been used in order to predict the condition and emotion of a driver to provide information that will improve safety on the road. Viola Jones algorithm uses the following techniques in its algorithm . They are: HAAR based features, Integral Image Formation, AdaBoost Technology and A cascade of classifiers.	The system is based on computer vision. It makes use of the Viola Jones algorithm, AdaBoost classifier and CAMSHIFT algorithm. A low-cost application can be devised by implementing this system using a raspberry-pi module. It is based on the concept of eye-tracking. In order to obtain finer results, a hundred and fifty images of different people have been used. If the state of fatigue has been identified, an alarm system is turned on. In order to detect drowsiness, certain facial features were identified. This system uses the

			concept of video processing. It also mentions certain disadvantages of the proposed system.
Driver Safety Development: Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network (2020)	Hashemi, M., Mirrashid, A. & Beheshti Shirazi	In this system, to detect the falling sleep state of the driver as the sign of drowsiness, Convolutional Neural Networks (CNN) are used regarding the two goals of real-time application, including high accuracy and fastness. The authors also gathered a dataset for driver drowsiness detection, which contains a new state of the eye, named oblique view. The expanded dataset advantage is considering the oblique view, which makes the system work in more varied situations. The authors also proposed three networks to achieve better accuracy and less computational time for drowsiness detection based on eye state.	The authors referred to the benefits of proposed approaches as high accuracy and low computational complexity. However, there is a tradeoff between the size of the dataset and accuracy. Deeper networks with more hidden layers and parameters show better performance and need a bigger dataset. Therefore, there is a compromise between the number of available data and the number of FD-NN parameters.

IoT-Based Smart Alert System for Drowsy Driver Detection (2020)	Anil Kumar Biswal, Debabrata Singh,Binod Kumar Pattanayak, Debabrata Samanta,and Ming-Hour Yang	In this proposed paper, they have addressed a drowsy driver alert system that has been developed using a technique in which the Video Stream Processing (VSP) is analyzed by eye blink concept through an Eye Aspect Ratio (EAR) and Euclidean distance of the eye. They have also added a collision detection system to alert authorities nearby if any mishap occurs. Face landmark algorithm is also used as a proper way to eye detection. When the driver's fatigue is detected, the IoT module issues a warning message along with impact of collision and location information, thereby alerting with the help of a voice speaking through the Raspberry Pi monitoring system.	This proposed system has provided an efficient or successful drowsy detection using the facial landmark method as well as another interface for detection of collision due to drowsiness or unconscious state of driving with a 97% accuracy when tested on 10 distinct situations. This process takes a little bit more time to load at deep night vision and causes poor performance due to the eyes not being detected clearly. For industrial-scale use, especially for ordinary drivers, is costly and not practical for everyday use.
IOT BASED DRIVER DROWSINESS AND HEALTH MONITORING SYSTEM	Ashwini, Veda M , Smitha S , Divya Krishna, Pooja Suresh Talekar	In this proposed paper, they have implemented a drowsiness detection system by using a heart rate monitoring	This proposed system has provided an efficient solution with a high success rate, owing to the extensive number of

(2020)		sensor and an alcohol sensor. The system is locally placed on the driver's wrist with another module integrated within the car. When alcohol is detected, the speed of the vehicle goes down, ultimately coming to a halt. A USB Camera is provided to continuously monitor the position of the driver's eyes. When a driver's eyes are closed for more than a particular number of seconds, a buzzer goes off, to alert the driver. The driver's data is also sent to the health monitoring system in the server, and his condition is informed to his friends/relatives through an SMS.	sensors The driver needs to wear the wristband every time for the car to function,which is not practical for daily use. The speed control module integrated with the car can be of a risk. The cost of the system can be pretty high with all its sensors.
Driver Drowsiness Detection (2020)	V B Navya Kiran,Raksha R, Varsha K N, Anisoor Rahman, Dr. Nagamani N P	The paper presents an arithmetic based method to solve the problem related to the detection of drowsiness. Three stages were involved. They are Face detection, Eye position detection and Eye tracking. This paper provides an	Within the framework of the definitions provided above, the data delivered by the eye tracker are direct measures, which are objective. The video data is also an objective direct measure. The coding of the

		efficient method for the detection of the state of the driver. This framework uses the motion of the eyes to detect the state of the driver and gives an alert within 0.5 seconds. The performance of the driver is transcribed in the form of a graph	two raters have a subjective share. Even though they are highly correlated, they do not match a hundred per cent. The project leader decided to use rater A's results, because of her higher grade of experience. This decision influences the results slightly.
A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability (2017)	Muhammad Awais, Nasreen Badruddin, Micheal Drieberg	The proposed method demonstrated that combining EEG and ECG has improved thesystem's performance in discriminating between alert and drowsy states, instead of using themalone. Their channel reduction analysis revealed that an acceptable level of accuracy (80%) could be achieved by combining just two electrodes (one EEG and one ECG), indicating the feasibility of a system with improved wearability compared with existing systems involving many electrodes.	However, only EEG is discussed in depth as they label this as the most "promising and feasible method". Furthermore, limitations of described studies are mentioned, but not protocols including diversity between subjects and the number of subjects

Proposed Work

Countless accidents have occurred as a result of driver exhaustion, tiresome road conditions, and adverse weather conditions.
According to enforcement personnel monitoring the motorways and major roads here, sleep-deprived drivers are still responsible for around 40% of all traffic accidents.
By focusing on the driver's eye movements, an IoT-based solution is aimed to prevent innumerable accidents caused by fatigued drivers' behavioral and psychological changes.
Our project's goal is to assist in the cost-effective solution of real-world problems. The buzzer is sounded if the driver becomes tired and shuts his eyes for longer than a second.

<u>Methodology</u>

In this project we will be using Transfer Learning to train the model on eye detection for drowsiness. We first train the model using the dataset using the MRL Eye Dataset, which is a large scale dataset of human eye images. We then classify each of the images in the dataset as open and close and train the model to detect the same. We will be using the keras application architecture for transfer learning processes as it provides architectures which are trained for 1000 classes and importing their knowledge will improve the accuracy of the algorithm.

The architecture used is the Inception v3 which is an image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset.The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using Softmax as the project is a categorical model project.

The main file then detects the face first and further detects the eyes and then classifies whether the eyes are open or closed. For this, haarcascade_frontalface and haarcascade_eye will be used for categorization.

We then Capture the video from the ESP 32 WEBCAM, detect the face and eyes and report the current eye status by using an alarm which is played if the eye is found to be closed for a certain short period of time.

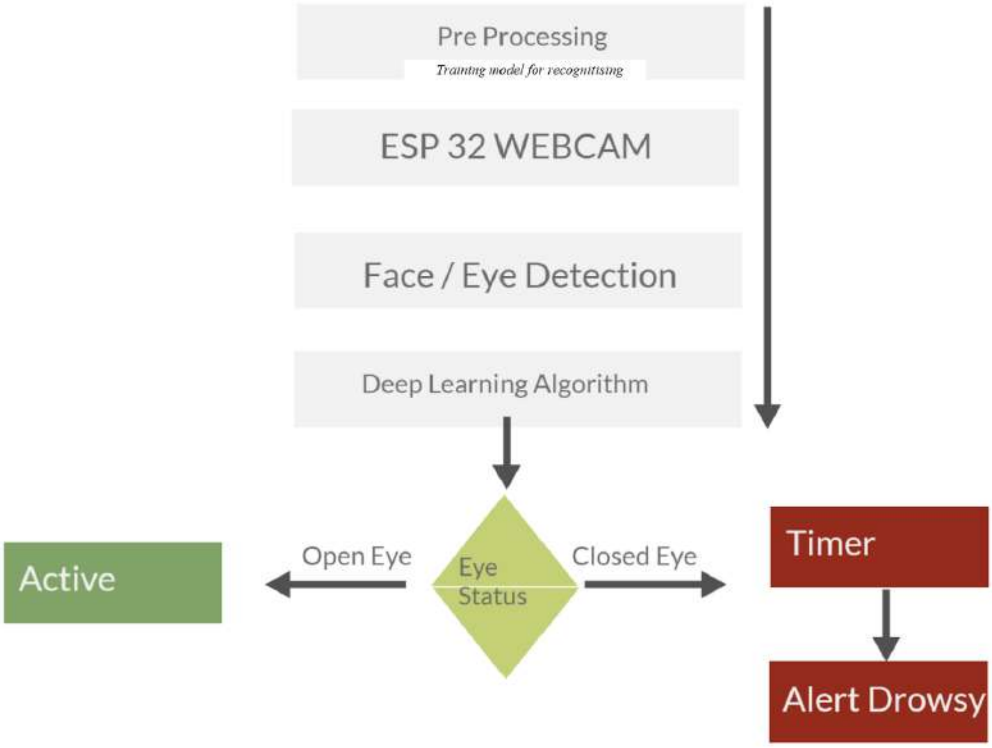


Fig 1 : Methodology of the proposed system

<u>Transfer Learning</u>

Transfer learning is a machine learning technique in which a model created for one job is utilized as the basis for a model on a different task.

Given the vast compute and time resources required to develop neural network models on these problems, as well as the huge jumps in skill that they provide on related problems, it is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks.

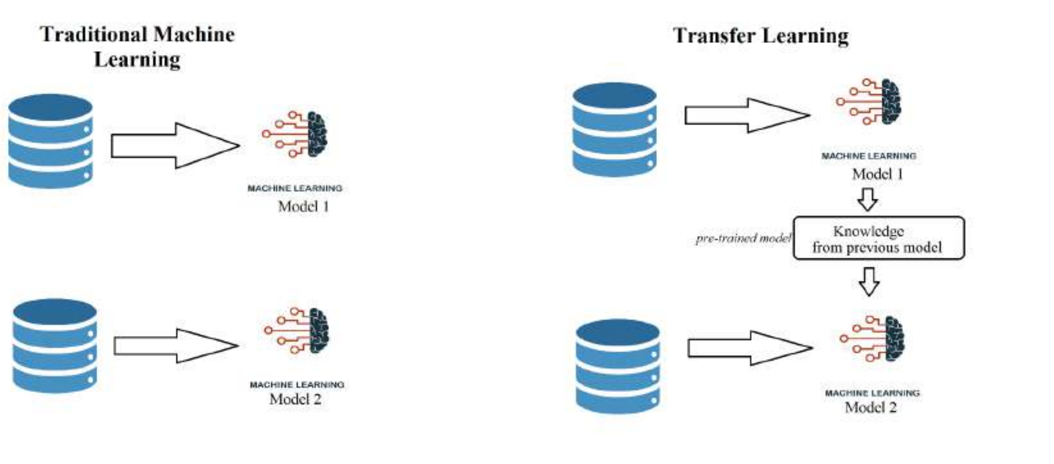


Fig 2 : Transfer Learning

As we can see, Traditional Machine Learning models have to train their dataset from scratch every time a new model comes to the surface. But in the case of Transfer Learning, it learns from the pre-trained model from before and utilizes the information from the previous model to enhance the accuracy of the new model.

<u>Hardware Setup:</u>

Components:

- Arduino UNO
- ESP32 camera module
- A wifi network, like mobile hotspot
- Jumper cables and bread board
- Arduino IDE
- Python 3.9 (OpenCV, …)

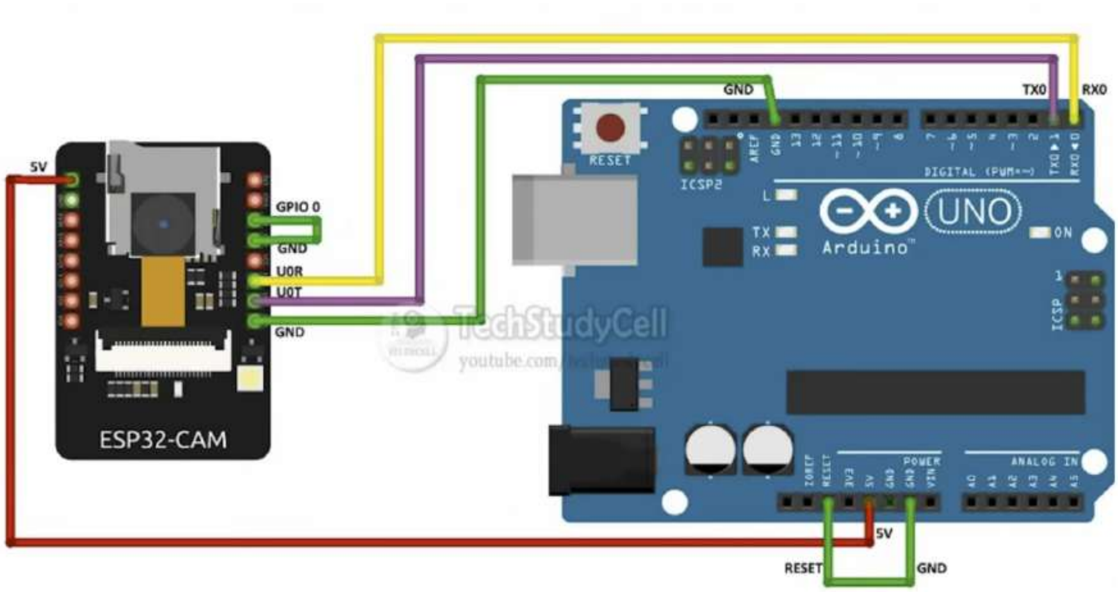
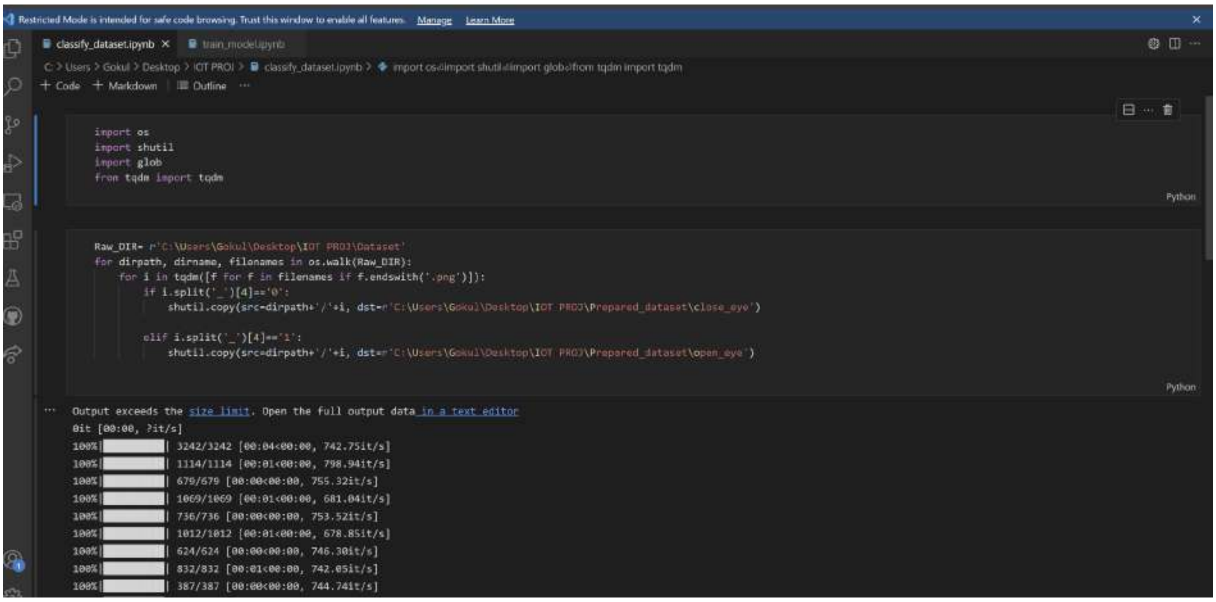


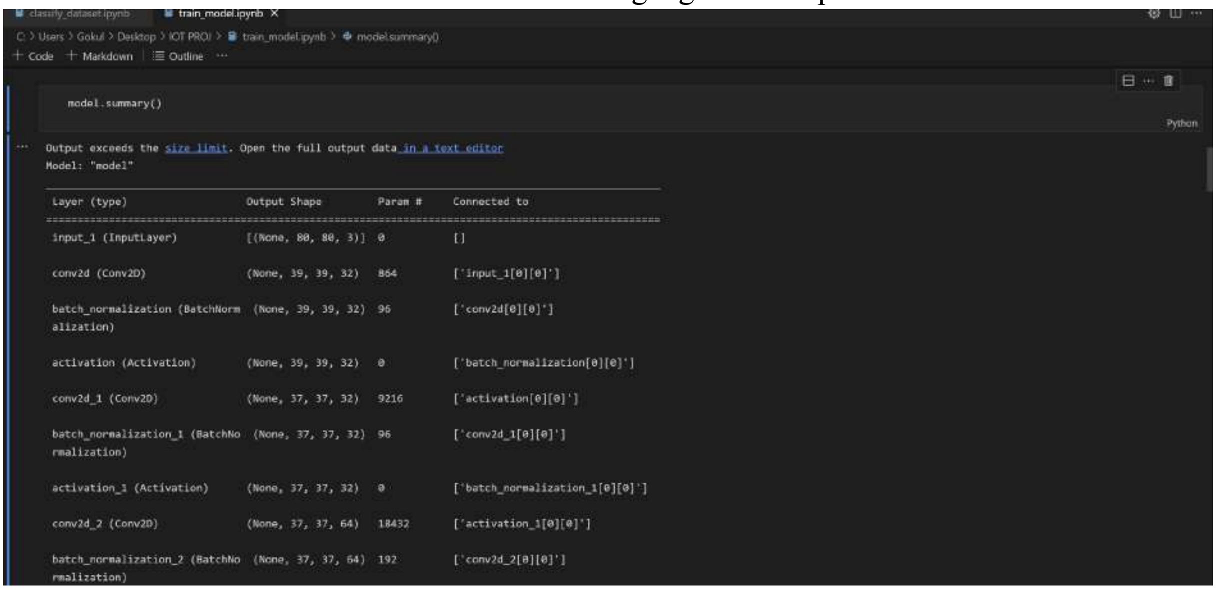
Fig 3 : Programming ESP32 CAM using Arduino Uno

Software Setup:
OpenCV: Tool which will be used for image processing and visual tasks.
Python 3.9 : Coding platform where we will be teaching the algorithm on detecting drowsiness using machine learning.
Machine Learning Algorithm : Transfer Learning for Eye Status Detection.

In the algorithm we first split the dataset from the MRL eye dataset into open and closed eye folders.



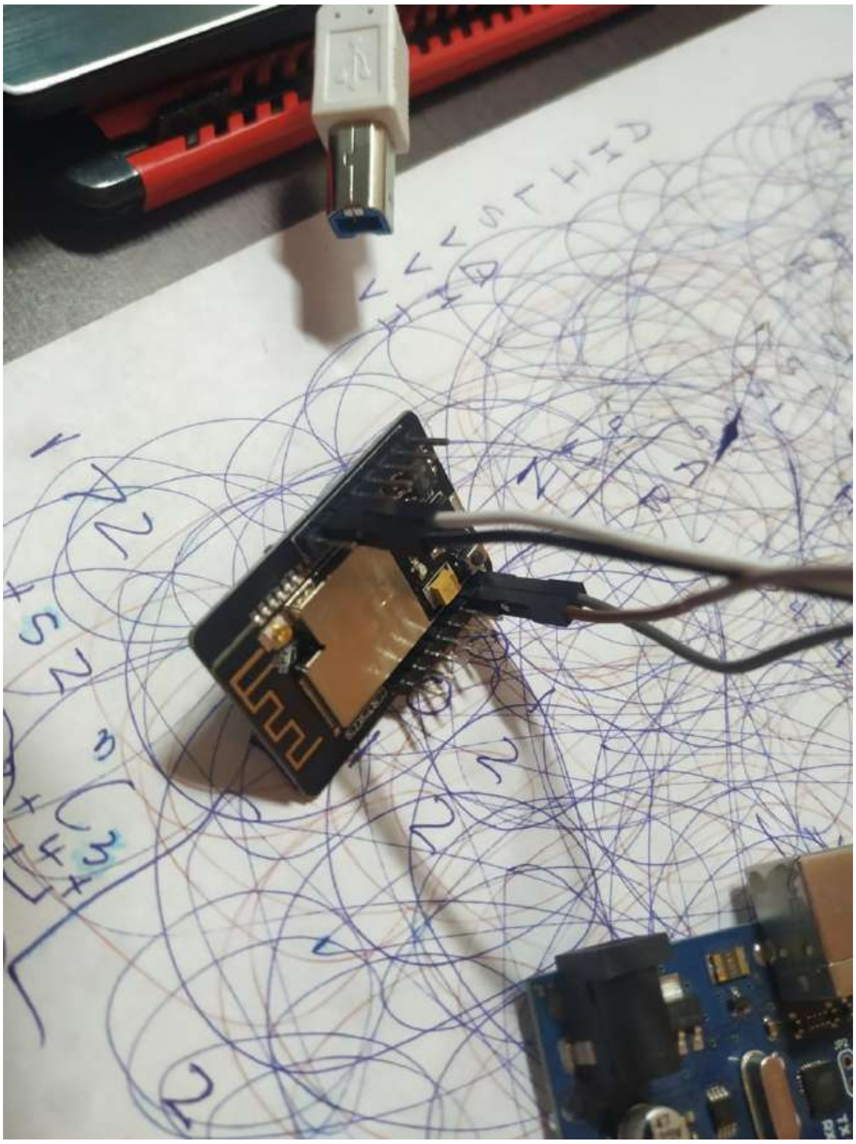
Now we train the model with the Transfer learning algorithm explained above.



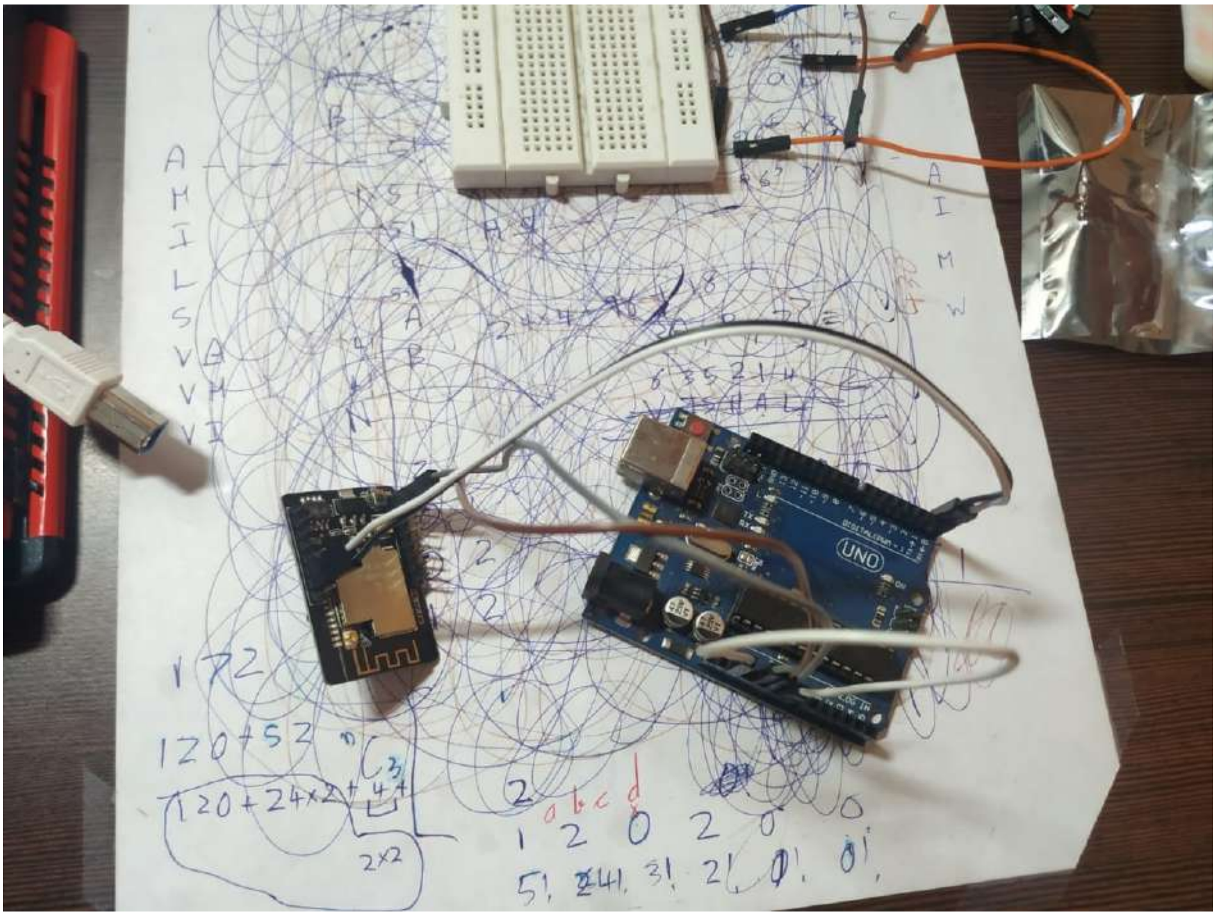
Note : In the algorithm we used 5 Epochs

Results and Discussion

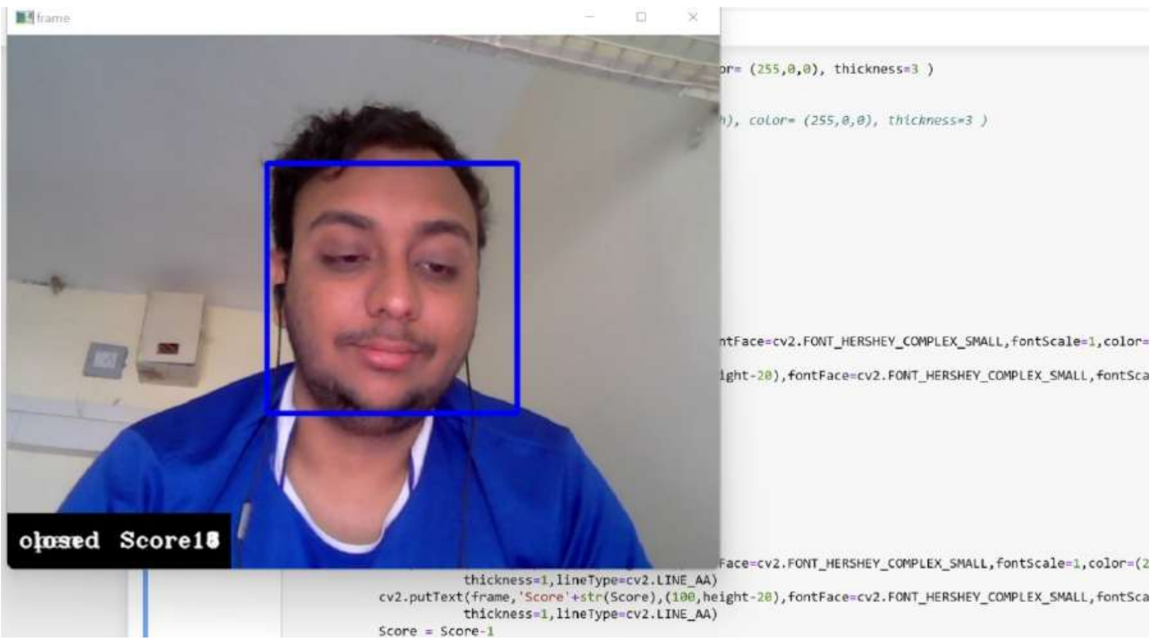
As we can see we first setup the camera with Arduino Uno



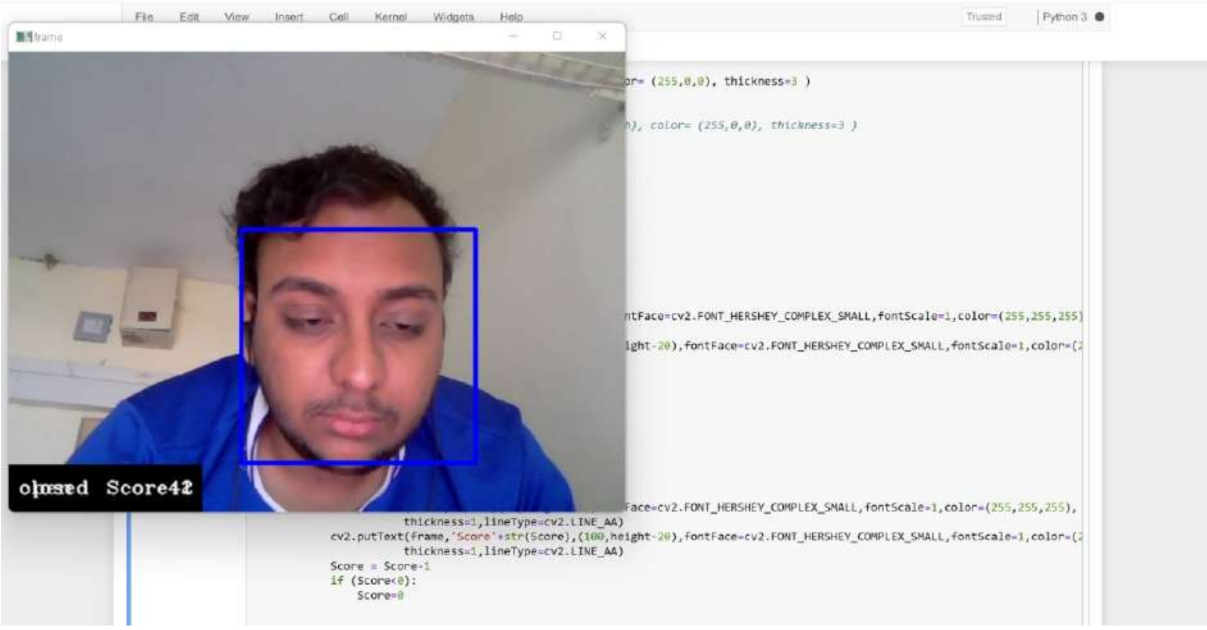
We connect it to the ESP32 CAM



The full connection



This is the image of the student who appears to not be in the best conditions to drive and appears to be sleepy though he is still awake and aware but as we can see the score is increasing giving it a good chance that he might feel drowsiness while driving.



As we can see, the student shown in the figure appears to be sleepy which makes the score go high. As the score reaches a benchmark (in this case 20), it start to alert the user using an alarm.

```
classify_dataset.ipynb    train_model.ipynb ×
C: > Users > Gokul > Desktop > IOT PROJ > train_model.ipynb > model.summary()
+ Code  + Markdown | Outline ...

acc_tr, loss_tr = model.evaluate(train_data)
print(acc_tr)
print(loss_tr)

8490/8490 [==============================] - 278s 33ms/step - loss: 0.1304 - accuracy: 0.9495
0.13040095567703247
0.9495126605033875

acc_vr, loss_vr = model.evaluate(validation_data)
print(acc_vr)
print(loss_vr)

2123/2123 [==============================] - 70s 33ms/step - loss: 0.2462 - accuracy: 0.9025
0.24615120887756348
0.9024677276611328

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

325/325 [==============================] - 10s 29ms/step - loss: 0.2955 - accuracy: 0.8813
0.13040095567703247
0.9495126605033875
```

From the image above, we can see that the accuracy when using trained data is 94.95%, when using the validation data is 90% and the test data which we had taken different images in different lighting compared to the training data and got 88% accuracy

Conclusion and Future work

This study presents a reliable approach for detecting driver tiredness and a collision impact (severity) system in the current era. As a result, the suggested system is used to develop a non-intrusive technique for assessing the driver's drowsiness in relation to the severity of a collision caused by braking or a mistake.
The Arduino Uno module and the ESP32 Camera module are the major components of this system, which are employed for continual face recording and detection of eye pupils.
If the calculated score value exceeds the threshold range, the system sends a warning buzzer to the user. If the estimated score value falls below the threshold range, the eyes remain open and the system's status remains unchanged.
In this project we have successfully been able to detect if the driver is in a state of complete control or if he/she is feeling drowsy. We have used Transfer Learning algorithms to achieve an accuracy of 90% which significantly reduces the risk of drowsy driving.

This paper's future work might focus on the use of external elements to measure exhaustion and sleepiness.
Weather conditions, vehicle condition, and sleeping time are examples of external variables. Driver sleepiness is one of the most serious risks to road safety, and it is especially problematic for commercial motor vehicle operators. Twenty-four-hour services, unpredictable environmental conditions, and an increase in demanding work schedules are all elements that contribute to this major safety concern. Continuously monitoring the driver's sleepiness condition is a key step in the preventative steps needed to remedy this problem. More work can be done in the future to automate the zoom on the eyeballs after they have been localized.

References

- *Arefnezhad, S.; Samiee, S.; Eichberger, A.; Nahvi, A. Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection. Sensors 2019, 19, 943. https://doi.org/10.3390/s19040943*
- *V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P, 2020, Driver Drowsiness Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15),*
- *Hashemi, M., Mirrashid, A. & Beheshti Shirazi, A. Driver Safety Development: Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network. SN COMPUT. SCI. 1, 289 (2020).*
- *Anil Kumar Biswal, Debabrata Singh, Binod Kumar Pattanayak, Debabrata Samanta, Ming-Hour Yang, "IoT-Based Smart Alert System for Drowsy Driver Detection", Wireless Communications and Mobile Computing, vol. 2021, Article ID 6627217, 13 pages, 2021.*
- *Ashwini, Veda, M., Smitha, S., Krishna, D.H., & Talekar, P. (2020). IOT BASED DRIVER DROWSINESS AND HEALTH MONITORING SYSTEM.*
- *V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P, 2020, Driver Drowsiness Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15),*
- *Awais M, Badruddin N, Drieberg M. A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability. Sensors (Basel). 2017 Aug 31;17(9):1991. doi: 10.3390/s17091991. PMID: 28858220; PMCID: PMC5620623.*

Annexure – Code/Implementation details if any

Note: learning_model

```
import tensorflow as tf
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dropout,Input,Flatten,Dense,MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
tf.test.is_gpu_available()
batchsize=8
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\Gokul\Desktop\IOT
PROJ\Prepared_dataset\train',

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

validation_data= train_datagen.flow_from_directory(r'C:\Users\Gokul\Desktop\IOT
PROJ\Prepared_dataset\train',
                        target_size=(80,80),batch_size=batchsize,class_mode='categorical',
subset='validation')
test_datagen = ImageDataGenerator(rescale=1./255)

test_data = test_datagen.flow_from_directory(r'C:\Users\Gokul\Desktop\IOT
PROJ\Prepared_dataset\test',
                        target_size=(80,80),batch_size=batchsize,class_mode='categorical')
bmodel = InceptionV3(include_top=False, weights='imagenet',
input_tensor=Input(shape=(80,80,3)))
hmodel = bmodel.output
hmodel = Flatten()(hmodel)
hmodel = Dense(64, activation='relu')(hmodel)
hmodel = Dropout(0.5)(hmodel)
hmodel = Dense(2,activation= 'softmax')(hmodel)

model = Model(inputs=bmodel.input, outputs= hmodel)
for layer in bmodel.layers:
    layer.trainable = False
model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping,
ReduceLROnPlateau
checkpoint = ModelCheckpoint(r'C:\Users\Gokul\Desktop\IOT PROJ\models\model.h5',
                    monitor='val_loss',save_best_only=True,verbose=3)
```

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

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

callbacks=[checkpoint,earlystop,learning_rate]

model.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)
#evaluation of model
acc_tr, loss_tr = model.evaluate(train_data)
print(acc_tr)
print(loss_tr)
acc_vr, loss_vr = model.evaluate(validation_data)
print(acc_vr)
print(loss_vr)
acc_test, loss_test = model.evaluate(test_data)
print(acc_tr)
print(loss_tr)
```

Note: after learning the model

```
import cv2
import tensorflow as tf
from tensorflow.keras.models import load_model
import numpy as np
from pygame import mixer
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
model = load_model(r'C:\Users\Gokul\Desktop\IOT PROJ\models\model.h5')

mixer.init()
sound= mixer.Sound(r'C:\Users\Gokul\Desktop\IOT PROJ\alarm.wav')
cap = cv2.VideoCapture(0)
Score = 0
while True:
    ret, frame = cap.read()
    height,width = frame.shape[0:2]
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces= face_cascade.detectMultiScale(gray, scaleFactor= 1.2, minNeighbors=3)
```

```
    eyes= eye_cascade.detectMultiScale(gray, scaleFactor= 1.1, minNeighbors=1)

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

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,pt1=(x,y),pt2=(x+w,y+h), color= (255,0,0), thickness=3 )

    for (ex,ey,ew,eh) in eyes:
        #cv2.rectangle(frame,pt1=(ex,ey),pt2=(ex+ew,ey+eh), color= (255,0,0), thickness=3 )

        # preprocessing steps
        eye= frame[ey:ey+eh,ex:ex+ew]
        eye= cv2.resize(eye,(80,80))
        eye= eye/255
        eye= eye.reshape(80,80,3)
        eye= np.expand_dims(eye,axis=0)
        # preprocessing is done now model prediction
        prediction = model.predict(eye)

        # if eyes are closed
        if prediction[0][0]>0.30:

cv2.putText(frame,'closed',(10,height-20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SM
ALL,fontScale=1,color=(255,255,255),
                    thickness=1,lineType=cv2.LINE_AA)

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

        # if eyes are open
        elif prediction[0][1]>0.75:

cv2.putText(frame,'open',(10,height-20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMA
LL,fontScale=1,color=(255,255,255),
                    thickness=1,lineType=cv2.LINE_AA)

cv2.putText(frame,'Score'+str(Score),(100,height-20),fontFace=cv2.FONT_HERSHEY_CO
MPLEX_SMALL,fontScale=1,color=(255,255,255),
                    thickness=1,lineType=cv2.LINE_AA)
            Score = Score-1
```

```
            if (Score<0):
                Score=0


        cv2.imshow('frame',frame)
        if cv2.waitKey(33) & 0xFF==ord('q'):
            break

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

IOT Based Real-Time Drowsiness Monitoring System

19BCI0181 - Vishal Haswani
19BCI0187 - Gokul Gopakumar
20BCE0724 - Deepanshu Sharma

Group B1 - 18

Introduction

Countless accidents have occurred as a result of driver exhaustion, tiresome road conditions, and adverse weather conditions.

According to enforcement personnel monitoring the motorways and major roads here, sleep-deprived drivers are still responsible for around 40% of all traffic accidents.

"This is especially relevant to exhausted drivers who fall asleep at the wheel on our roadways between midnight and 5 a.m.," stated B. Shefiq, Joint Regional Transport Officer, Perumbavur.

Only expert drivers, according to B. G. Sreedevi, Chief Scientist at the National Transportation Planning and Research Centre (Natpac) of the Kerala State Council for Science, Technology and Environment, ensure that they get enough sleep before embarking on nighttime travels.

Proposed Statement

By focusing on the driver's eye movements, an IoT-based solution is aimed to prevent innumerable accidents caused by fatigued drivers' behavioral and psychological changes.

Our project's goal is to assist in the cost-effective solution of real-world problems. The buzzer is sounded if the driver becomes tired and shuts his eyes for longer than a second.

Literature Survey

Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection

Arefnezhad S., Samiee S., Eichberger A., Nahvi A.

(18 February, 2019)

This paper presents a novel feature selection method to design a non-invasive driver drowsiness detection system based on steering wheel data. The proposed feature selector can select the most related features to the drowsiness level to improve the classification accuracy. This method is based on the combination of the filter and wrapper feature selection algorithms using adaptive neuro-fuzzy inference system (ANFIS).
This method was based on the idea of combining filter and wrapper feature selection methods to improve the performance in comparison with the results when each of the filter and wrapper methods has been employed individually. This combination was performed in the structure of the designed ANFIS. Four different filter indexes were calculated for each feature and used as inputs to ANFIS to produce importance degree of each feature.

- ➢ It exploits the advantages of four different filter feature selection methods to improve the reliability of the classification results.
- ➢ The SVM algorithm used is also a robust algorithm which is suitable for real world data sets.
- As the detection is based on steering wheel data, it can be inaccurate as each person may have different tendencies to how they drive their vehicle.

Literature Survey

Driver Drowsiness Detection

V B Navya Kiran, Raksha R, Anisoor Rahman, Varsha K N, Dr. Nagamani N P, 2020, Driver Drowsiness Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15),

Machine learning techniques have been used in order to predict the condition and emotion of a driver to provide information that will improve safety on the road. Viola Jones algorithm uses the following techniques in its algorithm . They are: HAAR based features, Integral Image Formation, AdaBoost Technology and A cascade of classifiers.

- The system is based on computer vision. It makes use of the Viola Jones algorithm, AdaBoost classifier and CAMSHIFT algorithm. A low-cost application can be devised by implementing this system using a raspberry-pi module.
- It is based on the concept of eye-tracking. In order to obtain finer results, a hundred and fifty images of different people have been used. If the state of fatigue has been identified, an alarm system is turned on.

➢ In order to detect drowsiness, certain facial features were identified. This system uses the concept of video processing. It also mentions certain disadvantages of the proposed system.

Literature Survey

Driver Safety Development: Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network

Hashemi, M., Mirrashid, A. & Beheshti Shirazi, A. Driver Safety Development: Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network. *SN COMPUT. SCI.* **1,** 289 (2020). https://doi.org/10.1007/s42979-020-00306-9

In this system, to detect the falling sleep state of the driver as the sign of drowsiness, Convolutional Neural Networks (CNN) are used with regarding the two goals of real-time application, including high accuracy and fastness.

- The authors also gathered a dataset for driver drowsiness detection, which contains a new state of the eye, named oblique view. The expended dataset advantage is considering the oblique view, which makes the system works in more varied situations. The authors also proposed three networks to achieve better accuracy and less computational time for drowsiness detection based on eye state.
-

➢ he authors referred to the benefits of proposed approaches as high accuracy and low computational complexity. However, there is a tradeoff between the size of the dataset and accuracy. Deeper networks with more hidden layers and parameters show better performance and need a bigger dataset. Therefore, there is a compromise between the number of available data and the number of FD-NN parameters.

Literature Survey

IoT-Based Smart Alert System for Drowsy Driver Detection

Anil Kumar Biswal, Debabrata Singh,Binod Kumar Pattanayak, Debabrata Samanta,and Ming-Hour Yang

(29 December, 2020)

In this proposed paper, they have addressed a drowsy driver alert system that has been developed using a technique in which the Video Stream Processing (VSP) is analyzed by eye blink concept through an Eye Aspect Ratio (EAR) and Euclidean distance of the eye. They have also added a collision detection system to alert authorities nearby if any mishap occurs.

Face landmark algorithm is also used as a proper way to eye detection. When the driver’s fatigue is detected, the IoT module issues a warning message along with impact of collision and location information, thereby alerting with the help of a voice speaking through the Raspberry Pi monitoring system.

- ➢ This proposed system has provided an efficient or successful drowsy detection using the facial landmark method as well as another interface for detection of collision due to drowsiness or unconscious state of driving with a 97% accuracy when tested on 10 distinct situations.
- This process takes a little bit more time to load at deep night vision and causes poor performance due to the eyes not being detected clearly.
- For industrial-scale use, especially for ordinary drivers, is costly and not practical for everyday use.

Literature Survey

IOT BASED DRIVER DROWSINESS AND HEALTH MONITORING SYSTEM

Ashwini, Veda M , Smitha S , Divya Krishna, Pooja Suresh Talekar

(June, 2020)

In this proposed paper, they have implemented a drowsiness detection system by using a heart rate monitoring sensor and a alcohol sensor. The system is locally placed on the driver's wrist with another module integrated within the car. When alcohol is detected, the speed of vehicle goes down, ultimately coming to a halt.

USB Camera is provided to continuously monitor the position of driver's eyes. When driver's eyes are closed for more than a particular number of seconds, a buzzer goes off, to alert the driver. The driver's data is also sent to the health monitoring system in the server, and his condition is informed to his friends/relatives through an SMS.

➢ This proposed system has provided an efficient solution with a high success rate, owing to the extensive number of sensors

- The driver needs to wear the wristband every time for the car to function,which is not practical for daily use.
- The speed control module integrated with the car can be of a risk.
- The cost of the system can be pretty high with all its sensors.

Thank You

IOT Based Real-Time Drowsiness Monitoring

19BCI0181 - Vishal Haswani
19BCI0187 - Gokul Gopakumar
20BCE0724 - Deepanshu Sharma

Team : B1 - 18

Proposed Statement

Countless accidents have occurred as a result of driver exhaustion, tiresome road conditions, and adverse weather conditions.

According to enforcement personnel monitoring the motorways and major roads here, sleep-deprived drivers are still responsible for around 40% of all traffic accidents.

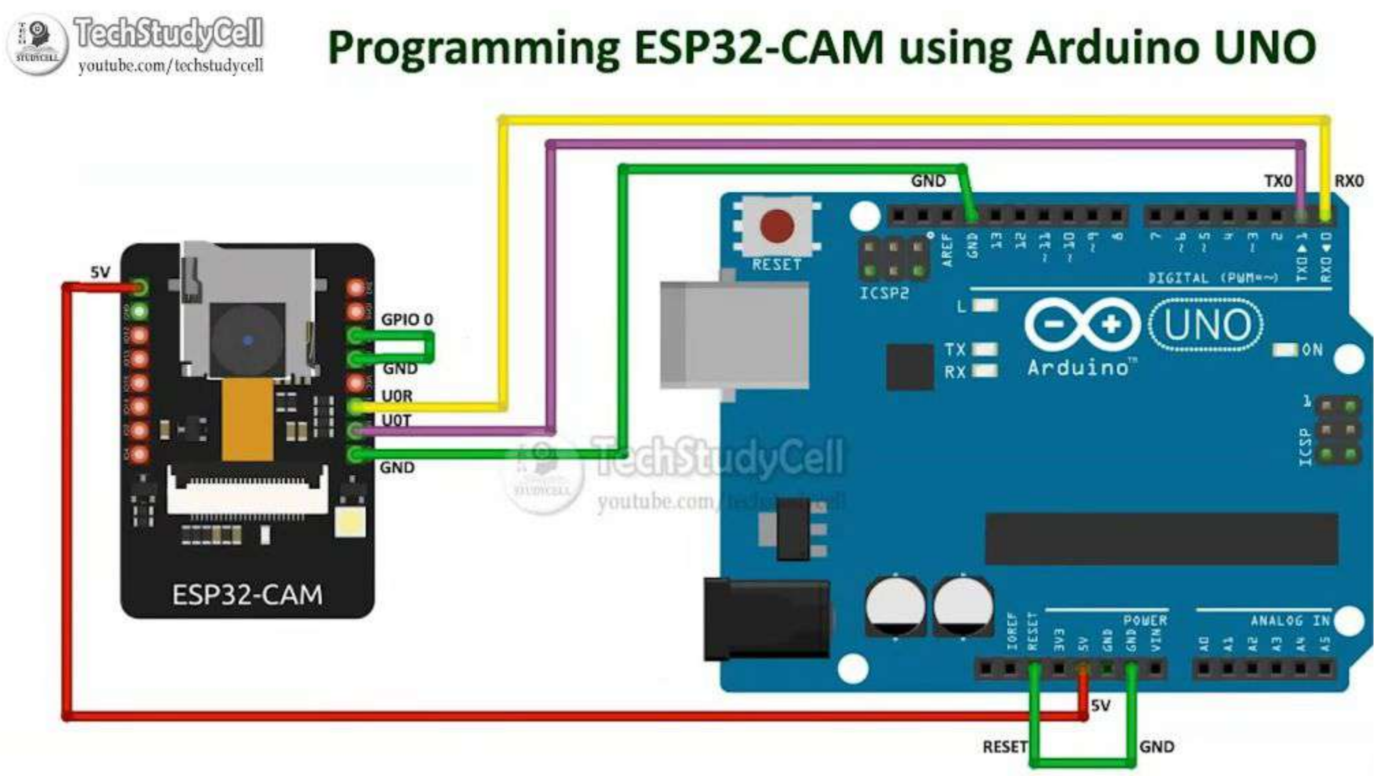
By focusing on the driver's eye movements, an IoT-based solution is aimed to prevent innumerable accidents caused by fatigued drivers' behavioral and psychological changes.

Our project's goal is to assist in the cost-effective solution of real-world problems. The buzzer is sounded if the driver becomes tired and shuts his eyes for longer than a second.

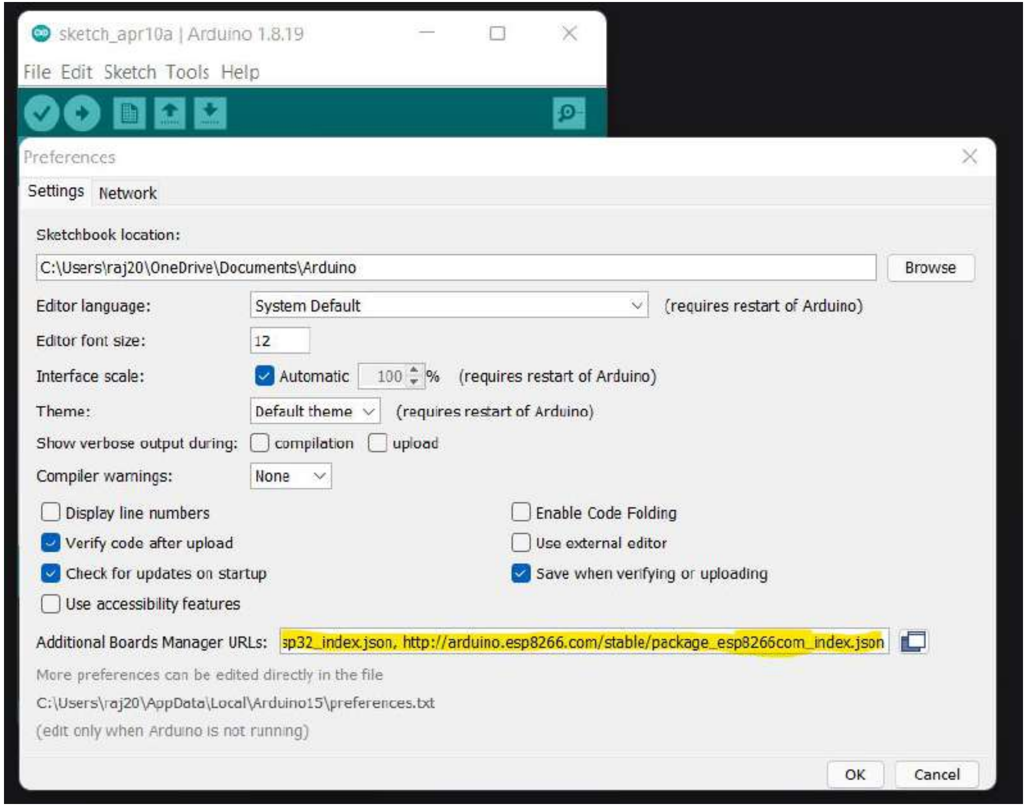
Components Used

- Arduino UNO
- ESP32 camera module
- A wifi network, like mobile hotspot
- Jumper cables and bread board
- Arduino IDE
- Python 3.9 (OpenCV, …)

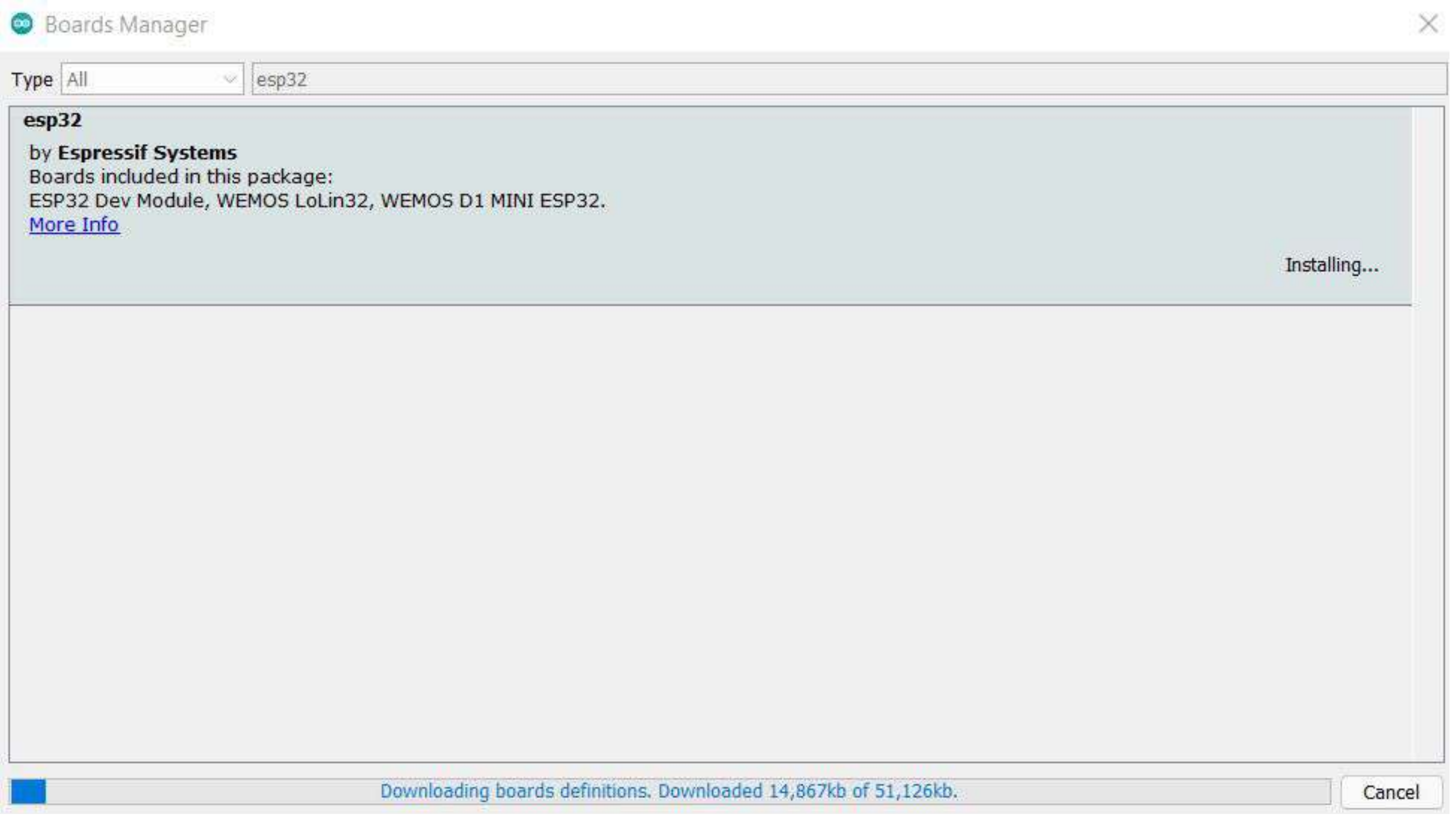
Hardware Setup

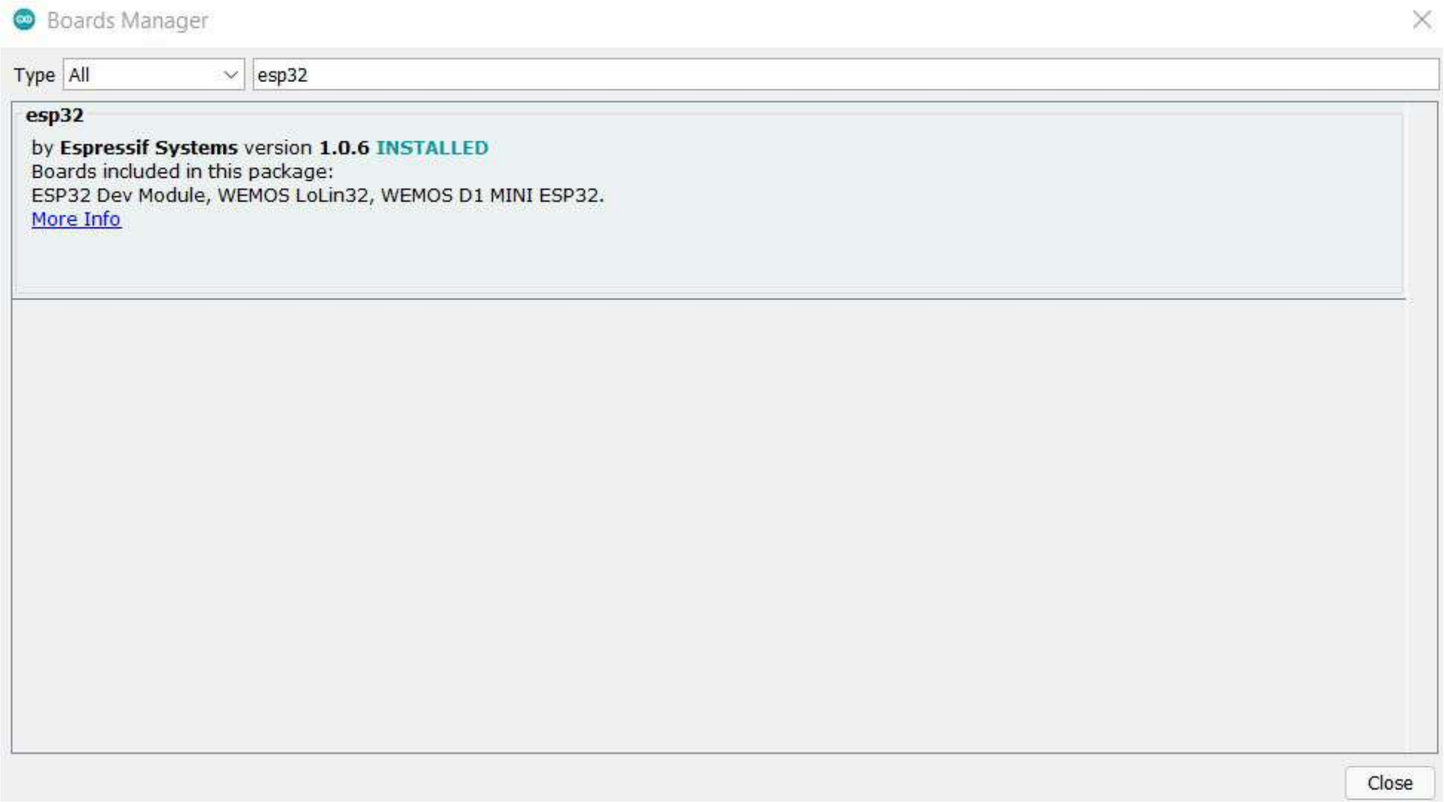


Setup


sketch_apr10a | Arduino 1.8.19
File Edit Sketch Tools Help
Preferences
Settings
Network
Sketchbook location:
C:\Users\raj20\OneDrive\Documents\Arduino
Browse
Editor language:
System Default
(requires restart of Arduino)
Editor font size:
12
Interface scale:
Automatic
100 %
(requires restart of Arduino)
Theme:
Default theme
(requires restart of Arduino)
Show verbose output during:
compilation
upload
Compiler warnings:
None
Display line numbers
Enable Code Folding
Verify code after upload
Use external editor
Check for updates on startup
Save when verifying or uploading
Use accessibility features
Additional Boards Manager URLs:
sp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json
More preferences can be edited directly in the file
C:\Users\raj20\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)
OK
Cancel

Setup (cont…)

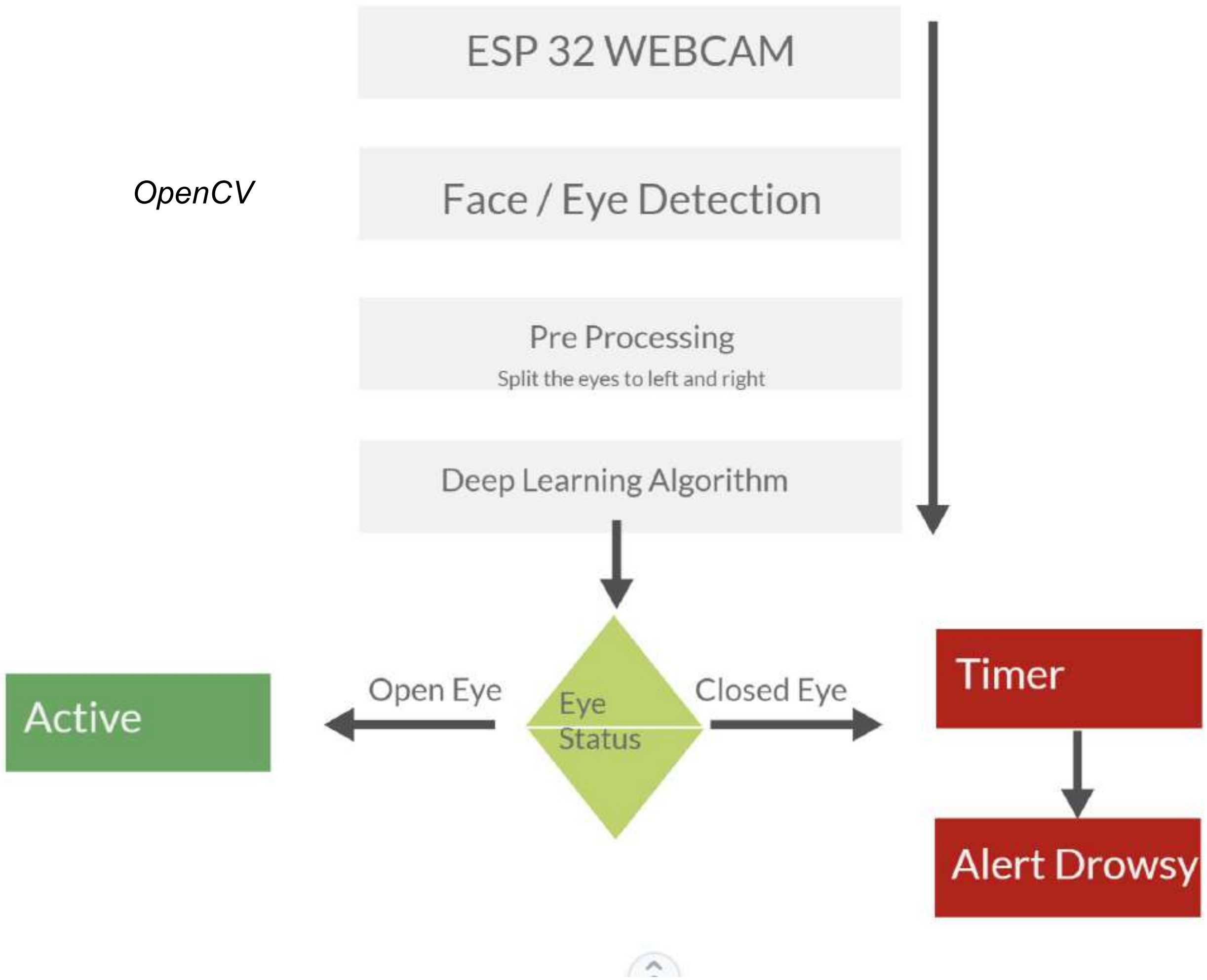

Boards Manager
Type All
esp32
esp32
by Espressif Systems
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info
Installing...
Downloading boards definitions. Downloaded 14,867kb of 51,126kb.
Cancel


Boards Manager
Type All
esp32
esp32
by Espressif Systems version 1.0.6 INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info
Close

Software Setup

Components

- OpenCV: Tool which will be used for image processing and visual tasks.
- Python 3.9 : Coding platform where we will be teaching the algorithm on detecting drowsiness using machine learning.
- Machine Learning Algorithm : Transfer Learning for Eye status Detection.



OpenCV
ESP 32 WEBCAM
Face / Eye Detection
Pre Processing
Split the eyes to left and right
Deep Learning Algorithm
Eye
Status
Open Eye
Active
Closed Eye
Timer
Alert Drowsy

Thank You