Driver Sleep Alert System

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Vinamra Jain(161233)

Under the supervision of

Dr. Pardeep Kumar
Associate Professor, SM-ACM

`

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan-173234, Himachal Pradesh

Certificate

I hereby declare that the work presented in this report entitled **"Driver Sleep Alert System"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from May 20 to June 2020 under the supervision of **Dr. Pardeep Kumar. (Associate Professor, SM-ACM)**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

vinamra jain
……..

(Student Signature)

Vinamra Jain
(161233)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Pardeep Kumar
……

Dr. Pardeep Kumar
Associate Professor, SM-ACM

Department of Computer Science and Engineering and Information Technology

Dated: …24/06/2020…..

Acknowledgement

I would like to take the opportunity to thank and express my deep sense of gratitude to my mentor and project guide Dr. Pardeep Kumar for his immense support and valuable guidance without which it would not been possible for me to reach at this stage of my final year project.

I am also thankful to all our faculty members for their valuable support in their respective fields which helped me in reaching at this stage of my project. My thanks and appreciations also go to my colleagues who had helped me out with their abilities in developing the project.

Date..24/06/2020.. Vinamra Jain (161233)

Table of Contents

List of Figures:

ABSTRACT

Nowadays, many professions requires concentration for long term. Drivers have to keep a close eye on the roads, so that they can react to some events occur on road immediately. Driver drowsiness often becomes a main cause for many traffic accidents. Therefore, there is an urgent need to develop a systems that could easily detect any bad psychophysical condition and notify it to the driver, which could massively decrease the number of car accidents related to fatigueness. However, many difficulties occur during the development of this systems related to fast or proper recognition of a driver's fatigue symptoms. Vision-based approach is one of the technical possibilities to implement driver drowsiness detection systems. This project presents the currently used driver drowsiness detection systems and the technical aspects of using a vision based system to detect a driver drowsiness.

Chapter 1:

Introduction

Chapter 1: Introduction

1.1 Introduction

Development in technology allows advanced solutions for everyday life. This makes work easy for people, and also increases work safety. Vision based systems are getting more popular and more widely used in different applications and can be used in transportation, air security, in industry or in car parking camera Such system could also beaused to detect driver fatigue using vision based solutions. As fatigue is such a psychological condition of a man which doesn't allow the full concentration. It affects the response time, as person reacts slower. So Driver sleeping alert system is a safety technology that could be used so that while driving no accident occurs that may be caused due to drivers felling asleep.

Fig 1.1: Driver sleeping alert

1.2 Problem Statement

With this project, I would develop a device that will detect when the Driver fell sleep. As day and night a countless number of people drive on the highway. Bus drivers, taxi drivers, truck drivers and people who travel for long durations or long-distance may sometimes suffer from lack of sleep. And Feeling sleepy becomes very dangerous to drive.

Drowsiness of the driver causes the majority of accidents. So using Python, OpenCV, and Keras we will build a system that wil prevent these accidents and alert the driver when he feels sleepy.

1.3 Objective

The objective behind this project is to develop a Driver sleeping alert system that could find if a person had closed his/her for a longer period of time. If system find eyes to be closed for some seconds than the system will sound an alarm to alert the driver.

1.4 Methodology

In this Python project, the person's eyes will be classified whether 'Open' or 'Closed' using OpenCV that is used for collecting images from the camera and giving it to the model of Deep Learning.

1.5 Organization of Thesis

In Chapter 2, I go through the literature survey. looked upon the various Deep learning models Each one of them has their own unique features which distinguishes it from the others.

In Chapter 3, I discussed different Deep learning models with their unique features.

In Chapter 4, I explained the Deep learning model that had been used in the development of this project.

In Chaptera5, I thereby concluded my report and discussed the future scope of this project.

Chapter 2:

Literature Survey

Chapter 2: Literature Survey

2.1 Using a vision-based system to detect a driver fatigue: **Damian Salapatek, Jacek Dybala, Pawel Czapski, Pawel Skalski**

Fatigue detection is a hard task. It requires taking many factors into account. Best solution for this purpose can be video system. This would allow for precise detection of a drowsiness in real time.

An important issue in designing the vision-based driver fatigue detection system is the correct selection of the analysed symptoms of fatigue as it is not possible to detect all the potential symptoms, it should be limited to the detection of the most important ones such as: closing of eyelids, slow in the movements of eyes, yawning or drooping a head. Methods based on the vector operations and the pattern classification allow detecting individual facial elements.

Most popular methods are Component Analysis, frequency spatial methods, Neural Networks, Gabor filters. Neural networks are mainly used for identification and classification of pattern data, however they may also be used in face detection and recognition systems.

In vision-based systems it is very important to identify the specific elements as well as to analyse their movement. Differential and gradient are the common methods used to detect a motion in video systems.

It determines the brightness level in the grayscale or the intensity of color of the pixel during the frame changes to detect the movement of the object. One of the limitations of this method is to have an object still, background lighting should be constant, and there should be minimum noise in the film.

To make an effective use of this method, some conditions should be met: invariability of light, a slight displacement of movement in objects in one. Two most popular gradient algorithms are Lucas-Canada and Horn-Schunk algorithm. Over the time the principle operation of the first algorithm is a characteristic assumption: the brightness of the dots in the image is unchanged , constant movement of the frames .

Intention of this method is for the methodological purposes, that is, devoted to the area at the section of traffic in the area. Pyramid could improve the performance of this algorithm The only disadvantage of the
algorithm is to be more susceptible to interference as compared to the local methods.

Problem with vision based system operating in real time are also apply to the hardware used for video and processing. One of the issue is implementing the number of cameras infront the driver to record the object of the interest. More cameras gives better result that generate a 3D

image and then implement the appropriate data analysis methods but that may disqualify it for the industrial scale.

The quality of the recording equipment is also of great importance. Using an equipment of poor quality recording video signal may become noisy, affecting the performance of the algorithms for video processing and analysing which thing could result in the miscalculation of the result.

2.2 Intelligent Driver Drowsiness Detection system Method: Devendra Singh, **Ugra Mohan Kumar, Pankaj Punia, Sudhir Jugran, Vinay Negi**

This drowsiness detection is based on Artificial Neural Network (ANN) learning and acquired data fusion of a few advanced pointers From vision system driver bodily pointers get information.

The camera are set infront the driver to obtain pictures of the driver freely ignoring surroundings. In bad illumination cases NIR consequently remain on and vice versa. Picture handling method to detect the eye movement and tracking the parameters to estimate the pieces of information.

They are taken through accompanying enlisted motions from vehicle like controlling wheel developments, parallel site .

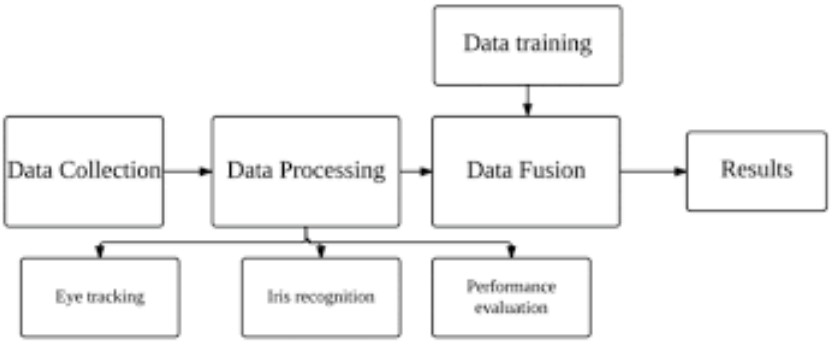

Fig 2.1 Flowchart

ANNs are used to gather informations through vision without any earlier learning of the examples. At that point, the best performance in driver drawsiness considers, as indicated by the best in class. For example, in a portion of the previous works this system recovers review rates in the vicinity of 55% and 90% contingent upon the data sources. Ideal mixes of an assortment in the writing assessed through advantageously trained method .

A driver's conduct is perplexing, variable and non-direct. One of the principle points of interest of training method is that they find arrangements based on earlier information. This trademark is helpful due to driver conduct examines, correct information yield connection are difficult to build up. ANNs likely can sum up, which is extremely useful, because true data is loud,amutilated and frequently fragmented. ANNs are nonlinear, which enables them to take care of some mind boggling issues more accurately than straight strategies.

In this work, a standout amongst the most types of ANNs, is utilised, where respectively layer comprises of neurons accepting contributions straightforwardly beneath and gives specifically over the unit. For situation, information gone through

the layers. Yield signals from neurons were gone through a solitary gets neuron, thus producing the yield.

In this paper a method is developed that help in the detection strategies if there may rise an occurrence of any imperfection in one technique, the rest of the strategies to identify the drowsiness, effectively. Tests works on twelve test cases driving test system. This comes about accompanying and finishing up comments can be dawn. The planned system can work in the event of the flag misfortune. In such conditions, generally performance is corrupted of around 6.85%.

Exactness of recognition in most scenarios was 87.78% . Estimated data inside a 20 seconds time are adequate for the proposed system. In Future we need to design more accurate system as basic management unit with improved algorithms.

Chapter 3:

System Development

Chapter 3: System Development

3.1 System Requirements

In order to run this project , these following Softwares And Hardware requirements are necessary to meet:-

3.1.1 Software Requirements:

1. Window 10 (64-bit) or Window (8/8.1)

2. I had used Python language for this project as Pythonis a general purpose programming language which contains many unique libraries such as Pygame or NumPy that makes it more reliable to create model based upon linearaalgebrass. This language is easy to use as it has simple syntax relatively and as it is designed with certain features to facilitate data analysis and visualization.

3.1.2 IDLE

IDLE is Python's Integrated Development and Learning Environment.

IDLE has the following features:

- Using the tkinter GUI toolkit it is coded in 100% pure Python
- Works same on all platforms Windows, Unix, and macOS
- It has an interactive interpreter that colorizes the input, output code and also the error messages
- It also has text editor that support multi-window with multiple undo call tips and other features
- It search through multiple files (grep) within any window and replaces within editor windows.

- It debugs with breakpoints that are persistant, viewing of global and local namespaces and stepping.

Packages Required:

i. OpenCV package is used to detect face and eye.
ii. Pygame package is used to play the sound alarm.
iii. Keras package is used in building our classification model.
iv. Tensorflow package is used as a backend.

3.1.3Hardware Requirements:

- Window 10/8/8.1, Mac OSX, LINUX
- Intel Core i3, i5, or i7 .
- 1.6 ghz or faster processor.
- Min. of 10 GB free memory space

3.2 Existing Methodologies

Supervisedavs Unsupervised Models

There are various features that distinguish these two, but how these models are trained is the most integral point of difference. While supervised models are trained through examples of a particular set of data, unsupervised models are only given input data and don't have a set outcome they can learn from So in an unsupervised model that y-column that we're always trying to predict is not there. While supervised models have tasks such as regression and classification and will produce a formula, unsupervised models have clustering and association rule learning.

3.2.1 Supervised Models

i. Classic Neural Networks (Multilayer Perceptrons)

ii. Convolutional Neural Networks (CNNs)

iii. Recurrent Neural Networks (RNNs)

3.2.1.1: Classic Neural Networks (Multilayer Perceptron)

Classic Neural Networks can also be referred to as Multilayer perceptron. The perceptron modelawas created in 1958 by American psychologist Frank Rosenblatt. It adapt basic binary patterns through aseries of inputs through its singular nature, simulating the learning patterns of human-brain A classic neural network model is the Multilayer perceptron consisting of more than 2 layers.

When to use :

- Tabular dataset formatted in rows and columns (CSV files)
- Classification and Regression problems where a set of real value ais given as the input.
- A higher level of flexibility is required in your model. ANNs can be applied to different types of data.

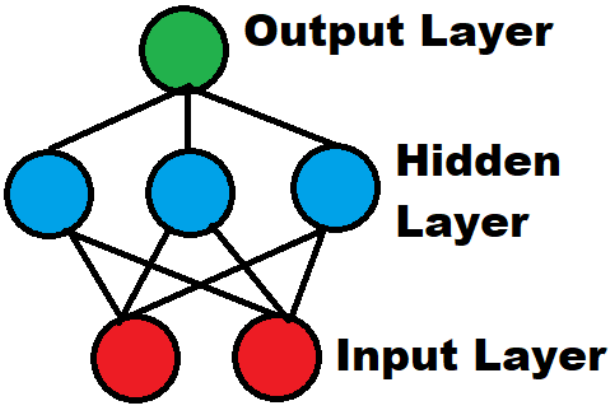

Fig: 3.1 Classic Neural Networks (Multilayer Perceptron)

3.2.1.2 Convolutional Neural Networks

A more capable and advanced variation of classic artificial neural networks, a Convolutional Neural Network (CNN) is built to handle a greater amount of complexity around pre-processing, and computation of data .

CNNs are the most efficient and flexible model and were designed for image data for image classification problems. Although they do not particularly built to work with data that do not have images, they can achieve stunning results with non-image data as well

After the data is imported into the model, there are 4 parts to building the CNN:

1. Convolution: a processain which feature maps are created out of our input data.
A function is then applied to filter maps.
2. Max-Pooling: which enables CNN to detect an image when presented with modification.
3. Flattening: Flatten the data into an array so CNN can read it.
4. Full Connection:, The loss function for our model is calculated using the hidden layer.

When to use:

- Image Datasets (including OCR document analysis).

  For faster processing 2-dimensional Input data but can be converted to 1-dimensional internally.

- When the model may be required for great calculating the output in great complexity.

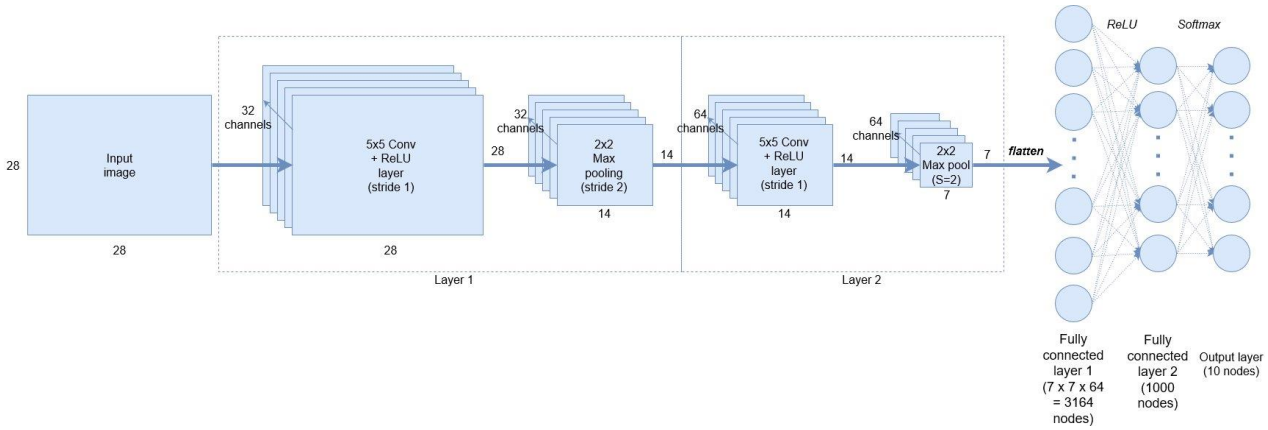

FIG: 3.2 Convolutional Neural Networks

3.2.1.3 Recurrent Neural Networks

Recurrental Neural Networks(RNNs) were invented to be used around predicting sequences LSTM (Long-term memory) is a popular RNN algorithm with many possible use cases:

When to use:

1. **One to one:** a single input mapped to a single output.
   e.g. — Image Classification

2. **One to many:** a single input mapped to a sequence of outputs.
   e.g. — Image captioning (multiple words from a single image)

3. **Many to one:** A sequence of inputs produces a single output.
   e.g. — Sentiment Analysis (binary output from different words)

4. **Many to many:** A sequence of inputs produces a sequence of outputs.
   for e.g. — Video classification by labeling each frame separately by splitting the video into frames.

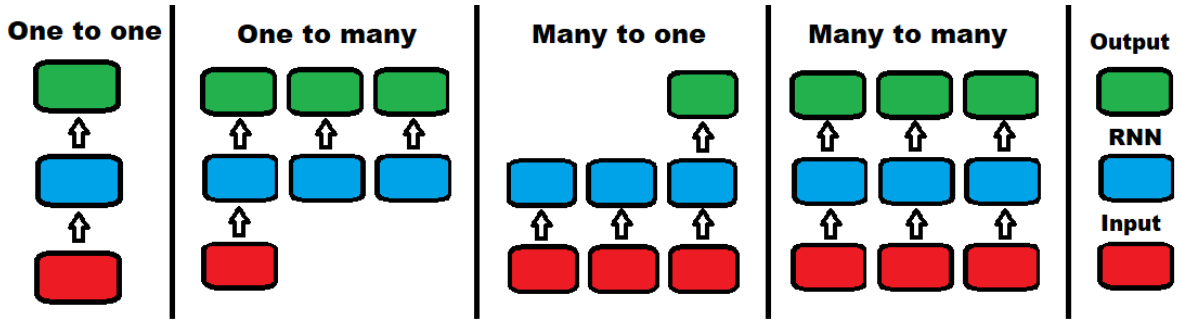

Fig: 3.3 Recurrent Neural Networks

3.2.2 Unsupervised Models

i. Self-Organizing Maps (SOMs)

ii. Boltzmann Machines

iii. Auto Encoders

3.2.2.1 Self-Organizing Maps

Self-Organizing Maps or SOMs deals with data which is unsupervised and help with dimensionality reduction (reducing how many random variables you have in your model). For self-organizing map the output dimension is always 2-dimensional. So if there are more than 2 input features, the output will be reduced to 2 dimensions. Each synapse making connection between input and output nodes having a weight given to them. In model each data point competes for representation. The node which is closest is called the BMU (best matching unit), and the SOM update its weights to move closer to the BMU.a The neighbours of the BMU keep decreasing as the model progresses. The closer to the BMU a node is, the more its weights would change.

Note: node characteristic are weights itself, which represent where the node lies in the input space. No activation functionais here (weights are different from what they were in ANNs).

When to use:

- When an output or a Y column is not present into provided data.
- To understand the framework behind a dataset explore projects.
- Creative projects (Music/Text/Video produced by AI).
- Dimensionality reduction for feature detection.

3.2.2.2 Boltzmann Machinesa

In the 4 models above, there's one thing in common. These models work in a certain direction. SOMs being unsupervised works same as supervised models work in a particular direction. By direction, I mean:
Input → Hidden Layer → Output.

Boltzmann machines don't follow a certain direction. All nodes are connected to each other in a circular kind of hyperspace like in the image.
A Boltzmann machine rather than working with fixed input parameters, could generate all parameters of the model.
Such models are referred as stochastic and are different from other deterministic models mentions above. Restricted Boltzmann Machines are more practical.

When to use:

- When monitoring asystem (since the BM will learn to regulate)
- Building a binary recommendation system
- When working with a very specific set of data

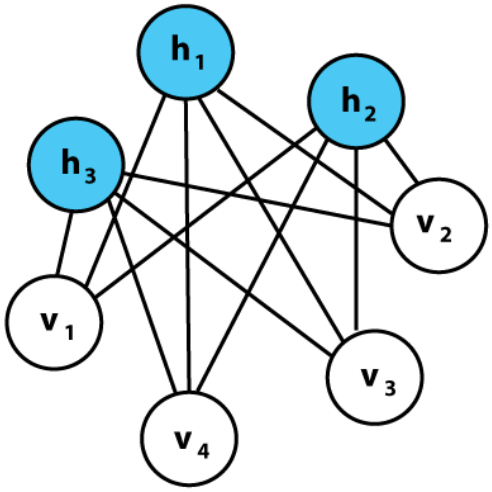

Fig: 3.4 Boltzmann Machines

3.2.2.3 Auto Encoders

Auto encoders work by automatically encoding data based on input values, then performing an activation function, and finally decoding the data for output. some sort of bottleneck imposed on the input features, compressing them into fewer categories. Thus, if some inherent structure exists within the data, the auto encoder model will identify and leverage it to get the output.

Types/Variations of Auto Encoders:

- **Sparse Auto Encoders**: Where the hidden layer is greater than the input layer but a regularization technique is applied to reduce overfitting. Adds a constraint on the loss function, preventing the autoencoder from using all its nodes at a time.

- **Denoising Auto Encoders:** Another regularization technique in which we take a modified version of our input values with some of our input values turned in to 0 randomly.

- **Contractive Auto Encoders :** Adds a penalty to the loss function to prevent overfitting and copying of values when the hidden layer is greater than the input layer.

- **Stacked Auto Encoders:** When you add another hidden layer, you get a stacked autoencoder. It has 2 stages of encoding and 1 stage of decoding.

When to use :

- Dimensional reduction/Feature detection
- Building a recommendation systems which is powerfull(more powerful than BM)
- Encoding features in massive datasets

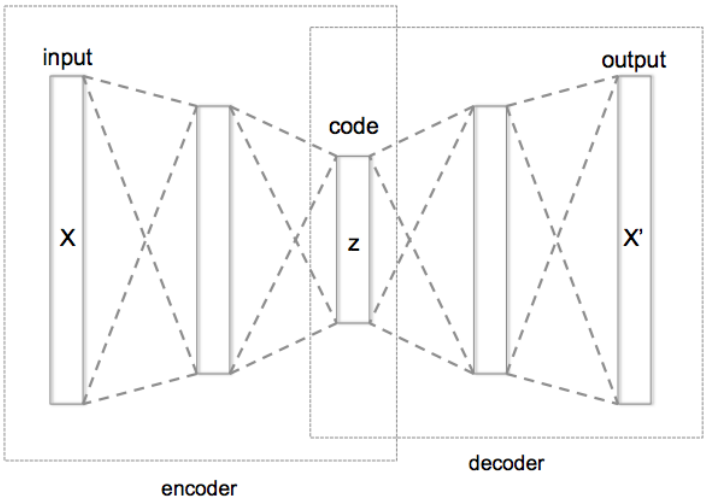

Fig: 3.5 Auto Encoders

Chapter 4:

Implementations and Results

Chapter 4: Implementations and Results

4.1 Dataset Used

The dataset is build by myself which is used in this model. We wrote a script to create the dataset that collect images from camera and detect eyes from those images.. Than we separate them in two categories labeled as ‘Open’ or ‘Closed’. For building the model I removed the images that we do not want and were not necessary. In this data we collect different people images in different conditions of light which are around 7000 in number. After the model is trained, the final weights were attached in the “models/cnnCat2 .h5 ”.model architecture file.

4.2 Architecture of Model

Using **Convolutional Neural Networks (CNN)** we build a model with Keras. A special type of neural network of deep learning which could perform significantly good in purpose like image classification is convolutional neural network. In CNN there is layers like input, output layer and another layer which could have layer in multiple numbers is hidden layer. on the layer and the filter that could do the multiplication of matrix in 2d form a operation called convolution is performed.
These layers are there in a CNN model architecture:

- 32 nodes and kernel of size 3
- 64 nodes and kernel again of size 3
- 128 nodes nad a completely connected layer

final layer has 2 node and is completely connected. We used an activation function called Relu function only in the output layer we used different function called Softmax.

4.2.1 Score

Basically score is just a value which could be used to examine the period of time the person has closed his/her eyes. If both the eyes are detected as closed, the score will increase and when the eyes are detected as open, the score will descent. I had defined a threshold so if the score gets more than 15 that will conclude that the person's eyes were closed for a longer period of time. This is when the alarm will start beeping.

4.3 Implementation

Step 1 – we will take images from camera as input.
Step 2 – than we detect the face in the image and create a Region of Interest (ROI).
Step 3 – than from the ROI eyes will be detected and would feed it to the classifier.
Step 4 – Classifier would categorize whether the eyes are open or closed.
Step 5 – The score would be calculated to check whether the personais drowsy.

4.4Results

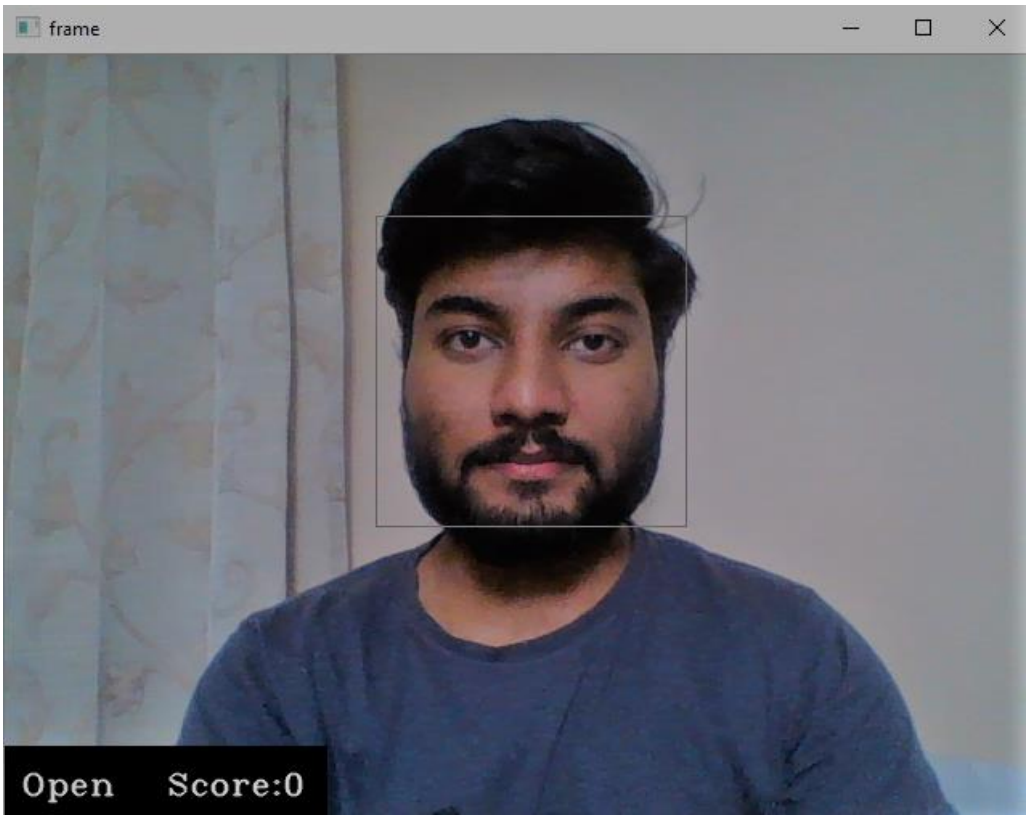

Fig 4.1 Open Eye Detection

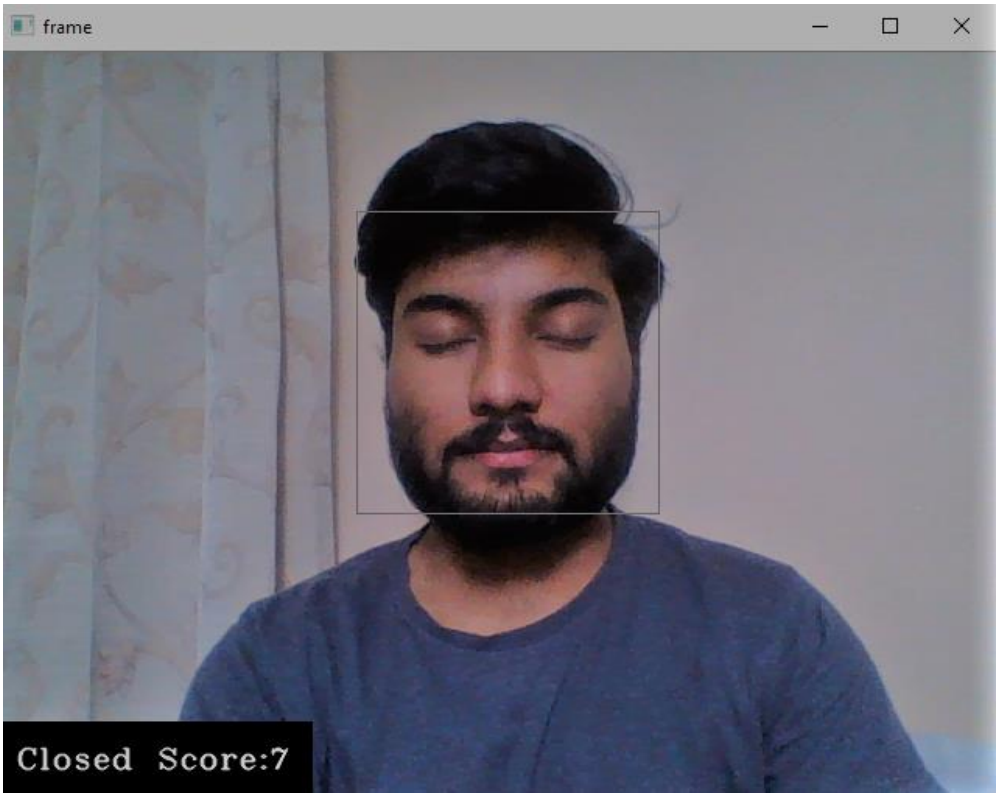

Fig 4.2 Closed Eye Detection

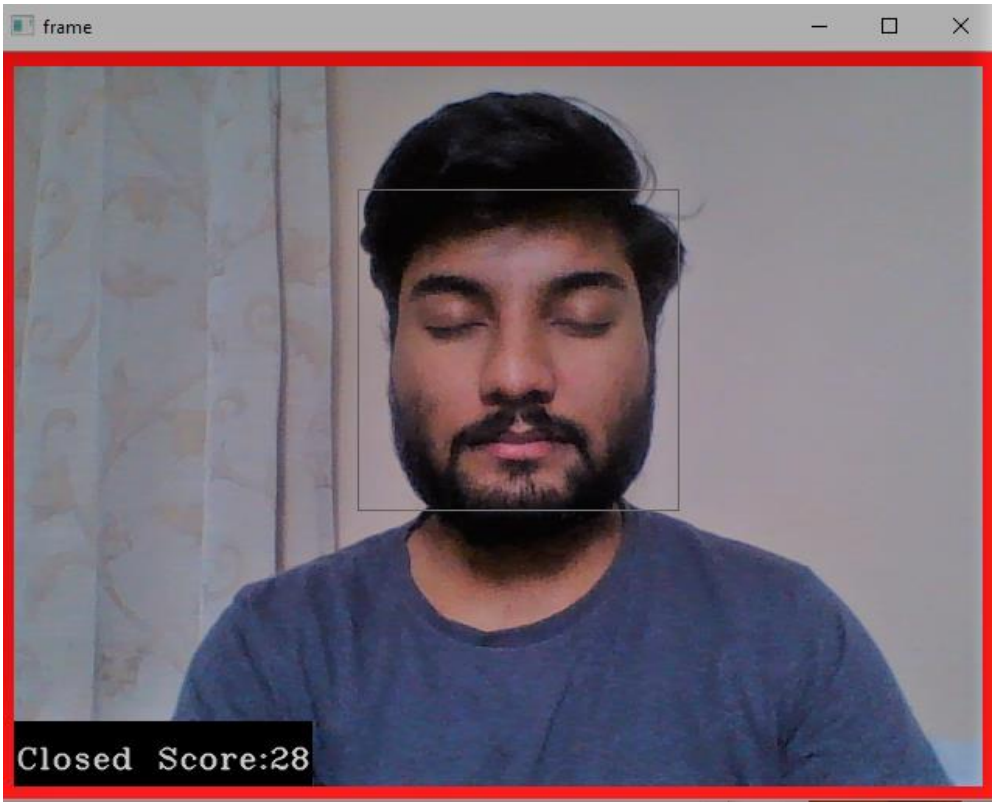

Fig:4.3 Sleep Alert

Chapter 5:

Conclusions

5.1 Conclusions

In order to reduce the number of road accidents which occurs due to the driver drowsiness, it is Of immense importance to introduce this system in a automotive industry that would immediately detect the first signs of the drowsiness and notify the driver by sounding an alarm. A system based on real-time face analysis could be one of the most effective approach to detect fatigue symptoms. There are number of problems associated with this design and one of them is uneven algorithms. Current technologies advances in recording and processing video to help in reducing and eliminating such problems. It is for sure that if these systems are integrated with other on-board on car, this driving system would definitely increase road safety.

5.2 Future Scope

The model could be improved incrementally by adding more parameters than only vision based like blinking eye rate, no. of yawning, state of the car, etc. If all these parameters are gathered and used it could immensely improve the accuracy of the system.

Further we could work upon this project by adding sensors which could help us to track the heart rate in order to prevent accidents caused due to sudden cardiac attacks to drivers.

Same model and techniques could be used for various other purposes like on OTT platforms like Netflix, youtube and other streaming services to detect when the viewer is asleep and stop playing video. It could also be used in other application that prevents users from sleeping.

References

1. Using a vision-based system to detect a driver fatigue: **Damian Salapatek, Jacek Dybala, Pawel Czapski, Pawel Skalski**

2. **Intelligent Driver Drowsiness Detection system Method: Devendra Singh,** Ugra Mohan Kumar, Pankaj Punia, Sudhir Jugran, Vinay Negi

3. Driver Drowsines Detection system : Damian Salapatek, Jacek Dybala, Pawel Czapski, Pawel skalski

4. List of Deep Learning Models: Annamaria R. Varkonyi- koczy

APPENDICES

4.5 Code

```
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')

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



lbl=['Close','Open']

model = load_model('models/cnncat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]

while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye =  reye.detectMultiScale(gray)
```

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

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]
    count=count+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 = model.predict_classes(r_eye)
    if(rpred[0]==1):
        lbl='Open'
    if(rpred[0]==0):
        lbl='Closed'
    break

for (x,y,w,h) in left_eye:
    l_eye=frame[y:y+h,x:x+w]
    count=count+1
    l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
    l_eye = cv2.resize(l_eye,(24,24))
    l_eye= l_eye/255
    l_eye=l_eye.reshape(24,24,-1)
    l_eye = np.expand_dims(l_eye,axis=0)
    lpred = model.predict_classes(l_eye)
    if(lpred[0]==1):
        lbl='Open'
    if(lpred[0]==0):
        lbl='Closed'
    break

if(rpred[0]==0 and lpred[0]==0):
    score=score+1
    cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
# if(rpred[0]==1 or lpred[0]==1):
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>15):
    #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(thicc<16):
            thicc= thicc+2
        else:
            thicc=thicc-2
            if(thicc<2):
                thicc=2
        cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

VJJ1

ORIGINALITY REPORT

6%	1%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Ugra Mohan Kumar, Devendra Singh, Sudhir Jugran, Pankaj Punia, Vinay Negi. "A System on Intelligent Driver Drowsiness Detection Method", International Journal of Engineering & Technology, 2018 Publication	1%
2	Sajid Ali Khan, Shariq Hussain, Sun Xiaoming, Shunkun Yang. "An Effective Framework for Driver Fatigue Recognition Based on Intelligent Facial Expressions Analysis", IEEE Access, 2018 Publication	1%
3	Submitted to Liverpool John Moores University Student Paper	1%
4	Submitted to University of KwaZulu-Natal Student Paper	1%
5	docs.python.org Internet Source	<1%
6	Submitted to INTI International University Student Paper	<1%

7	Submitted to Icon College of Technology and Management Student Paper	<1%
8	Submitted to University of Newcastle upon Tyne Student Paper	<1%
9	pt.scribd.com Internet Source	<1%
10	Submitted to University of Sheffield Student Paper	<1%
11	cs231n.stanford.edu Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 14/07/2020

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report ✓ | Paper

Name: Vinamra Jain **Department:** C.S.E **Enrolment No** 161233

Contact No. 9627243639 **E-mail.** vinamravipin@gmail.com

Name of the Supervisor: Dr. Pardeep Kumar Khokhar

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): Driver Sleep Alert System

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 31
- Total No. of Preliminary pages = 7
- Total No. of pages accommodate bibliography/references = 1

vinamra jain

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……6……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Pardeep Kumar

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com