iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XI **Month of publication:** November 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

Identification of Altered Levels of Consciousness and Countermeasures for Vehicular Safety

P.kavitha[1],
[1]*H.O.D/Department of Mechatronics Engineering, M.A. M. School of Engineering,*

Abstract: Consciousness is the key to performing any activity with precision and accuracy. Driving is a challenging activity that involves both the muscle memory and conscious decisions. Consciousness can be altered to various levels due to numerous psychological and physiological factors. Many of the road accidents occur due to this altered (or) impaired state of driver's consciousness such as through drowsiness, alcohol consumption, medical conditions (illness), etc. This project aims to identify the impaired consciousness using a 5MP camera and provide effective countermeasures by inducing alarm systems to ensure safety of both the subject and other vehicles on road.

The camera placed above the steering, scans and records the image of the driver's eyes, head and face movement. Algorithms are used to extract features from the images captured and signs of tiredness and sleepiness of the driver is detected. Using Raspberry Pi 3, an LED display and alarm system is invoked so as to alert the subject and nearby personnel once the sleepiness/tiredness of the subject is identified.

An approximate of 1.5 lakhs accidents occurs every year in India out of which 70% is known for drunken driving or sleepiness. Therefore this project aims to reduce the accidents rate due to impaired consciousness and to consider the shortcomings of existing methods and improvise on it.

I. INTRODUCTION

Globally over 35% of automobile accidents are the result of the driver driving in an unfit state of consciousness. Driving is a complex process that requires precise motor control and attention from the driver. Any hindrance (or) alteration to the consciousness of the driver can cause fatal accidents, leading to destruction of life and property. Billions of dollars are spent world-wide every year for dealing with medical treatments & compensation for damage to the property by governments, insurance companies and individuals as a result of such accidents.

This project intends to improve public safety via implementing a system that monitors the state of consciousness of the driver on real-time basis and will signal the unsuspecting co-drivers (automobiles in the proximity) about the careless driver so as to maintain precaution and also attempt to wake the subject (driver) via application of mild shock as well as sound a buzzer and also displaying a message accompanied by a voice instruction suggesting the driver to stop and take rest.

Various numbers of attempts have been made over the years to identify sleep/ drowsiness among the drivers. Though few have failed, few others have been successful in detection of sleep but have not taken a step to alert the driver, co-passengers and also the vehicles in the proximity.

A. Components used

1) Power Supply
2) Raspberry Pi 3
3) NoIR Camera Module
4) I Illuminator
5) Output Driver
6) LED Warning Display
7) Alarm

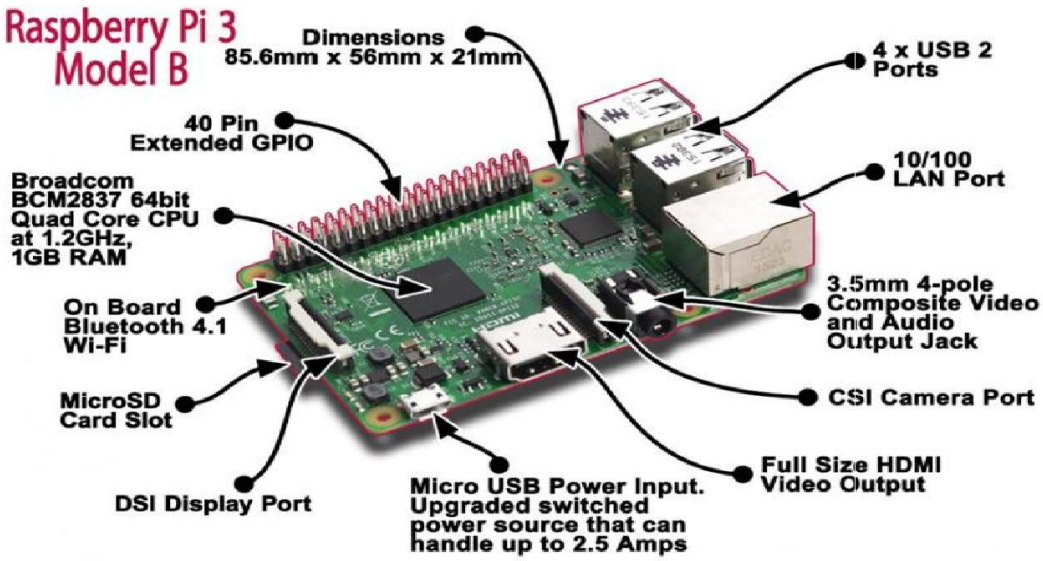
B. Basic Components of Raspberry Pi 3

Here are the various components on the Raspberry Pi board:

1) *Arm Cpu/Gpu:* This is a Broadcom BCM2837 System on a Chip (SoC) that's made up of an ARM central processing unit (CPU) and a Video core 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
2) *GPIO:* These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
3) *RCA:* An RCA jack allows connection of analog TVs and other similar output devices.
4) *Audio out :* This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers.

There is no audio in.

5) *LEDs :* Light-emitting diodes, for all of your indicator light needs.
6) *USB:* This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your *keyboard if it has its own USB port.*
7) *HDMI :* This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
8) *Power:* This is a 5v Micro USB power connector into which you can plug your compatible power supply.
9) *SD card slot:* This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.
10) *Ethernet* -- This connector allows for wired network access and is only available on the Model B. Many of the features that are missing, such as WiFi and audio in, can be added using the USB port (s) or a USB hub as needed.



II. CAMERA MODULE (N OIR)

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs .The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90.

The camera used is a 5MP sensor with 1080p video support. It is of a very compact and portable size (25 x 20 x 9 mm). Unlike the normal camera module, this is a NOIR camera which has no IR filter.

A. Output Driver

The output driver is a circuit that activates various output devices based on the control signals received from Raspberry Pi.

It is used to avoid direct contact of high voltage and current with the GPIO interface of the Raspberry Pi. The required power to run the peripherals is supplied by the power supply unit to the output driver.

It uses electromechanical relays that act as switches, controlled by the digital control signals from the Raspberry Pi's GPIO interface.

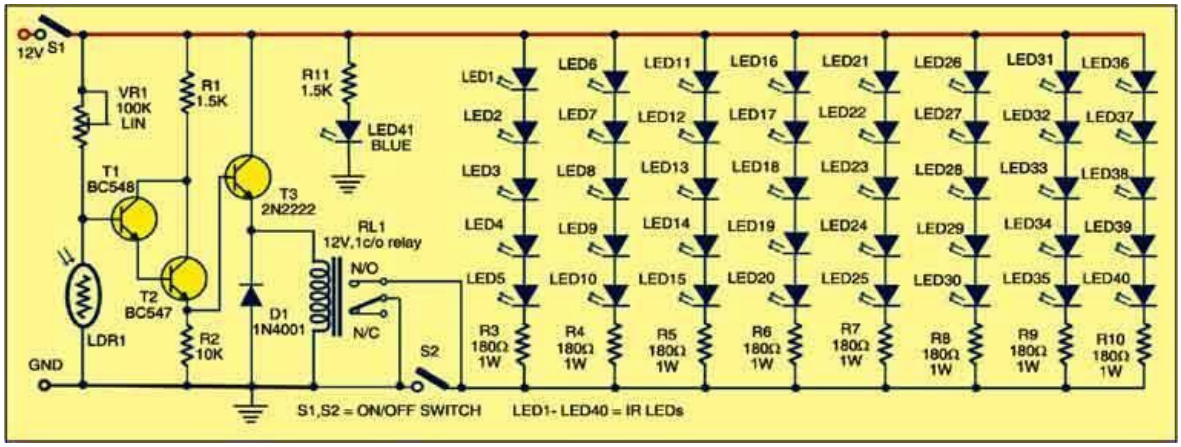
B. led display

LEDs (or) Light Emitting Diodes are devices made out of semiconductors. They emit light when forward biased. They are preferred over the traditional light sources because of their low power consumption, high luminosity and long life.

C. IR Illuminator

The Infrared Illumination is required to light up the subject's face in low-light conditions, mainly during night. The IR

illumination is provided with help of Infrared emitting diodes or IR LEDs.



Typically, IR LEDs run at around 1.3 to 1.7 volts, depending on the LED current (typically 10 to 30 mA). However, this may vary with the type and manufacturer. Practically, IR illuminators may have 6 or 60 to 100 or more LEDs, depending on the output needed. Variable resistor VR1 and the 10mm encapsulated LDR are used to sense the ambient light. As light fall on the surface of LDR1, its resistance changes. The amount of minimum light needed to actuate the relay through driver transistor T3 can be varied by adjusting VR1. Diode 1N4001 eliminates any back voltage when the relay re-energizes. Switch S1 is the mains power on/off switch and switch S2 is added to bypass the ambient light detection function.

This mechanism prevents unnecessary IR exposure such as during daytime when the ambient light is enough to highlight the details of the subject's face. The excessive IR illumination during ambient light availability can cause noise in the captured data, making it impossible to process therefore such an intelligent setup is required.

D. Alarm

The alarm system basically consists of a speaker and a pair of electrodes (optional).

1) *Audio Alarm - Speaker:* The speaker emits high volume, high pitched noise that is intended to wake up a sleepy person.
2) *Minor Electric Shock – Wake Up System:* This is an optional feature that applies a minor electric shock to the subject if the subject fails to respond to the audio alarm. This shock induces a tingling sensation that should wakeup most humans from REM phase of Sleep.

This system basically consists of a capacitor, two reverse coupled switches (if one is ON, the other will be OFF) and a resistor. The output driver supplies 12 V and 500mA - 1A power as input to the circuit. The switch I closes and opens the contact to the capacitor. When Switch I is closed, the capacitor is charged. When the circuit is required to apply shock, Switch I is opened and the Switch II is closed thus supplying power to the electrodes that are in contact with the subject's skin. Thus it applies a mild shock to the skin and will wake up the subject immediately.

E. Power Supply Unit

The power supply unit is a dedicated system that converts the available power supply, 12V DC in this case, to different voltage and current as required by the devices.

The power supply unit uses voltage regulators, resistors and capacitors to achieve the different power requirements.

It supplies

1) 2.1A 5V DC to Raspberry Pi 3
2) 12V DC to Output Driver

The Power Supply Unit is connected directly to the vehicle's battery circuit, in series with the ignition switch so that system is online as soon as the key is turned.

III. METHODOLOGY IMPLEMENTED

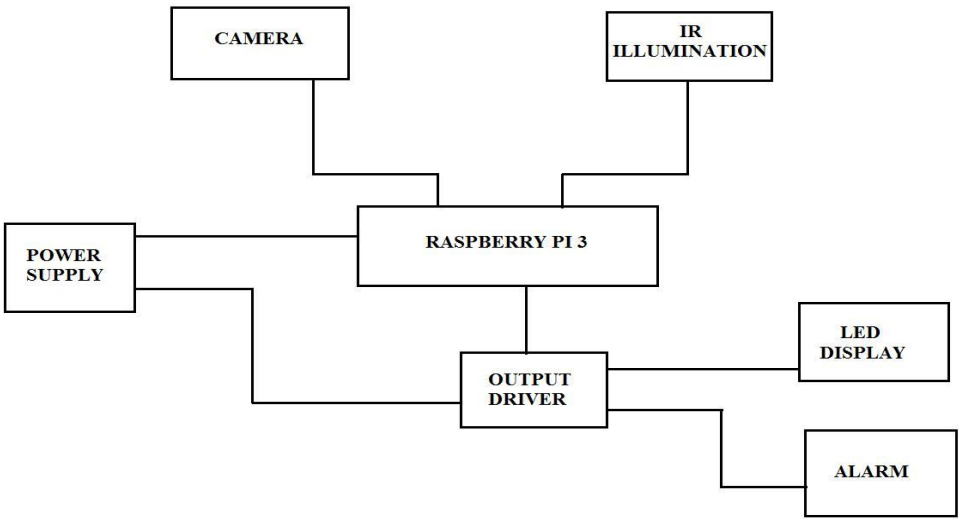
The visual data of the subject's face and actions is obtained via an IR Illumination supported 5MP Camera (with IR Filter Removed). The captured visual data is streamed to Raspberry Pi 3 at ~20FPS. The Raspberry Pi 3 is setup and configured to extract the significant features from facial expressions and eye blink rate of the subject using algorithms.

The program extensively uses Open CV library along with Numpy and Dlib for the image feature extraction assisted by Haar Cascade for facial land-marking.

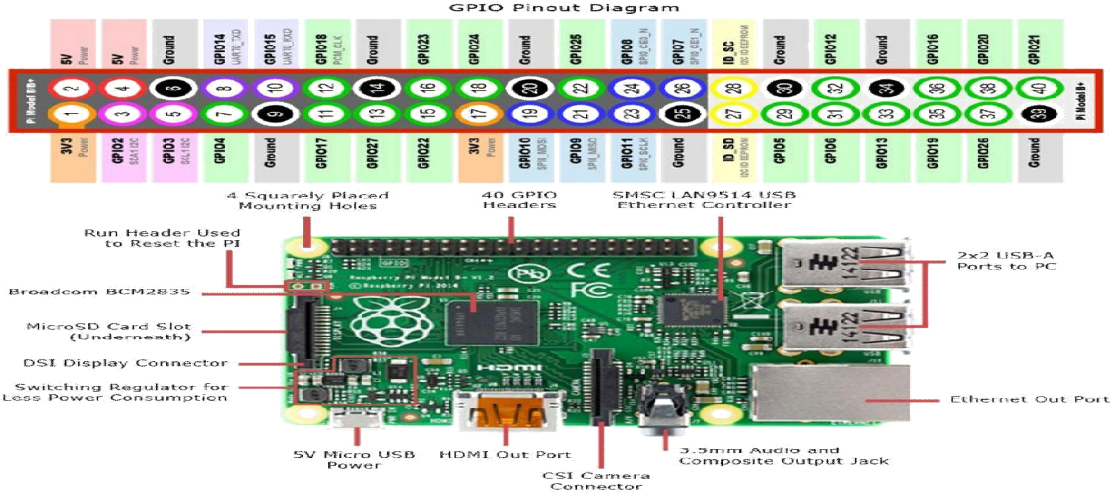
International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

The program keeps running uninterrupted while the engine is on and if it detects the driver is showing symptoms of unfit driving conditions such as dozing off, it executes the scripts to activate the countermeasures such as sounding an alarm and activating the warnings for personnel in proximity.
The program is written in Python 3 with wrappers and functions from Dlib to convert the CPU intensive parts of the program to C++ to achieve maximum processing speed.
The system also suggests the driver to take rest before proceeding for safety purpose. Once the engine is stopped or driver seems to be in condition for driving, the alarm stops else it continues till the subject responds accordingly.
The Raspberry Pi 3, Camera and other peripherals are powered by a 2.1 Amps 5V DC supply. The camera is placed in such a way that it neither obstructs the driver's view nor interrupts the visual data acquisition by camera.

A. Block diagram of system



The camera and IR illumination is connected to the Raspberry Pi 3 board. A 1.2A to 2.5A, 5V power supply is provided to run the Raspberry Pi and the output driver. The output of this output driver is used as inputs to both LED warning display, which is used to alert the co-passengers and the nearby personnel, and an alarm system to wake the driver up and create alertness.



B. PROGRAM

Python detect_drowsiness.py --shape-predictor shape_predictor_68_face_landmarks.dat
Detect_drowsiness.py --shape-predictor

```
Shape_predictor_68_face_landmarks.dat --alarm alarm.wav
import the necessary packages
From scipy.spatial import distance as dist
From imutils.video import videostream
From imutils import face_utils
From threading import thread
Import numpy as np
Import playsound
Import argparse
Import imutils
Import time
Import dlib
Import cv2
Defsound_alarm(path):
Play an alarm sound
Playsound.playsound(path)
Defeye_aspect_ratio(eye):
Compute the euclidean distances between the two sets of
Vertical eye landmarks (x, y)-coordinates
A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])
Compute the euclidean distance between the horizontal
Eye landmark (x, y)-coordinates
C = dist.euclidean(eye[0], eye[3])
Compute the eye aspect  ratio ear = (a + b) / (2.0 * c)
Return the eye aspect ratio return ear construct the argument parse and parse the
arguments ap = argparse. argumentparser()
Ap.add_argument("-p", "--shape-predictor", required=true,
Help="path to facial landmark predictor")
Ap.add_argument("-a", "--alarm", type=str, default="",
Help="path alarm .wav file")
Ap.add_argument("-w", "--webcam", type=int, default=0,
Help="index of webcam on system")
Args = vars(ap. parse_args())
Define two constants, one for the eye aspect ratio to indicate
Blink and then a second constant for the number of consecutive
Frames the eye must be below the threshold for to set off the
Alarm
Eye_ar_thresh = 0.3
Eye_ar_consec_frames = 48
Initialize the frame counter as well as a boolean used to
Indicate if the alarm is going off
Counter = 0
Alarm_on = false
Initialize dlib's face detector (hog-based) and then create
The facial landmark predictor
Print("[info] loading facial landmark predictor...")
Detector = dlib.get_frontal_face_detector()
Predictor = dlib.shape_predictor(args["shape_predictor"])
Grab the indexes of the facial landmarks for the left and
Right eye, respectively
(lstart, lend) = face_utils.facial_landmarks_idxs["left_eye"]
(rstart, rend) = face_utils.facial_landmarks_idxs["right_eye"]
```

```
Start the video stream thread
Print("[info] starting video stream thread...")
Vs = videostream(src=args["webcam"]).start()
Time.sleep(1.0)
Loop over frames from the video stream
While true:
Grab the frame from the threaded video file stream, resize
It, and convert it to grayscale
Channels)
Frame = vs.read()
Frame = imutils.resize(frame, width=450)
Gray = cv2.cvtcolor(frame, cv2.color_bgr2gray)detect faces in the grayscale frame rects = detector(gray, 0)
Loop over the face detections
For rect in rects:
Determine the facial landmarks for the face region, then
Convert the facial landmark (x, y)-coordinates to a numpy
Array
Shape = predictor(gray, rect)
Shape = face_utils. shape_to_np(shape)
Extract the left and right eye coordinates, then use the
Coordinates to compute the eye aspect ratio for both eyes lefteye = shape[lstart:lend]
Righteye = shape[rstart:rend] leftear = eye_aspect_ratio(lefteye)
Rightear = eye_aspect_ratio(righteye) average the eye aspect ratio together for both eyes ear = (leftear + rightear) / 2.0
   Compute the convex hull for the left and right eye, then
Visualize each of the eyes
Lefteyehull = cv2.convexhull(lefteye)
Righteyehull = cv2.convexhull(righteye)
Cv2.drawcontours(frame, [lefteyehull], -1, (0, 255, 0), 1)
Cv2.drawcontours(frame, [righteyehull], -1, (0, 255, 0), 1)
Check to see if the eye aspect ratio is below the blink threshold, and if so, increment the blink frame counter if ear
<eye_ar_thresh:
Counter += 1
If the eyes were closed for a sufficient number of
            Then sound the alarm
If counter >= eye_ar_consec_frames:
if the alarm is not on, turn it on
If not alarm_on:
Alarm_on = true
Check to see if an alarm file was supplied,
And if so, start a thread to have the alarm
Sound played in the background
If args["alarm"] != "":
                                    T = thread(target=sound_alarm,
                                          Args=(args["alarm"],))
                                    T.deamon = true
                                    T.start()
Draw an alarm on the frame
Cv2.puttext(frame, "drowsiness alert!", (10, 30),
V2.font_hershey_simplex, 0.7, (0, 0, 255), 2)
Otherwise, the eye aspect ratio is not below the blink
threshold, so reset the counter and alarm
else:
COUNTER = 0
```

```
ALARM_ON = False
draw the computed eye aspect ratio on the frame to help
with debugging and setting the correct eye aspect ratio
thresholds and frame counters
cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
show the frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
if key == ord("q"):
break
cv2.destroyAllWindows()
vs.stop()
```

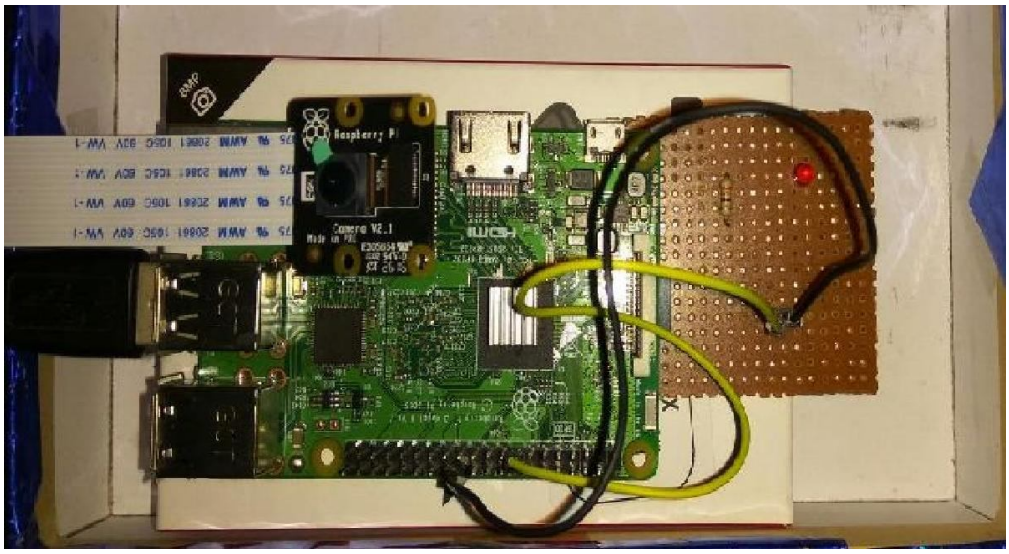
IV. CONCLUSION

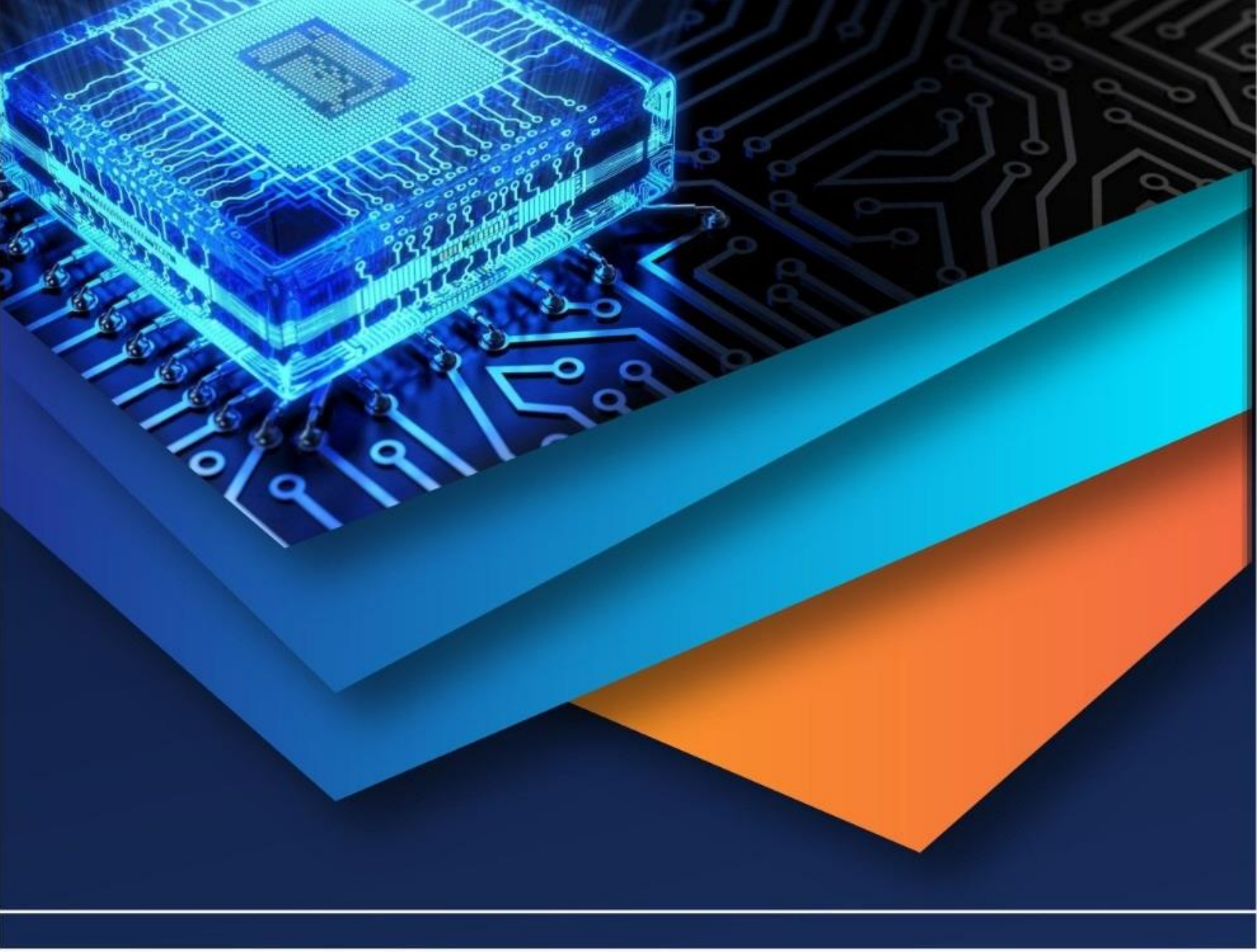
This system is thus found to have an effective warning system to alert the driver and to warn personnel in the proximity about the ‘unfit driving condition’ of the subject. This system is commercial and easy to install and run. Unlike earlier systems, involving EEG sensors, this system is more cost effective and does not involve physical contact with the subject.
In future, this system can be improvised by interfacing it with the Engine Control Unit (ECU) to prevent crashing.

REFERENCES

[1] Sleep detection and driver alert apparatus, James Russell Clarke,Sr., Phyllis Maurer Clarke- Nov 18, 1997
[2] EEG-based Drowsiness Detection for Safe Driving Using Chaotic Features and Statistical Tests, J Med Signals Sens. 2011 May-Aug;1(2): 130–137, Zahra Mardi, SeyedehNaghmehMiriAshtiani, and Mohammad Mikaili.
[3] Measurement of Driver's Consciousness by Image Processing -A Method for Presuming Driver's Drowsiness by Eye-Blinks coping with Individual Differences, Systems, Man and Cybernetics, 2006.SMC '06. IEEE International Conference on 16 July 2007, Mai Suzuki ; Nozomi Yamamoto ; Osami Yamamoto ; Tomoaki Nakano ; Shin Yamamoto
[4] Detecting Driver Drowsiness Based on Sensors: A Review, 7 December 2012- Sensors 2012, 12(12), 16937-16953
[5] Drowsiness Detection System And Method, Harry Zhang, Gerald Witt,Matthew Smith, May 13, 2004.
[6] Vision-based method for detecting driver drowsiness and distraction in driver monitoring system, Jaeik Jo, Sung Joo Lee, Ho Gi Jung, KangRyoung Park and Jaihie Kim.
[7] A Review on Sleep Detection Using EEG Signals, SuchiRawat and Virendra K. Verma.
[8] Real Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance, QiangJi and XiaojieYang ,Real-Time Imaging 8,2002, 357-377.

PHOTOGRAPHY





10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY