iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35966

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: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Intelligent Drowsy Drive Alert using OpenCV and Convolutional Neural Network

Somojit Banerjee[1], Manisha Ghosh[2], Somnath Chatterjee [2], Soumen Mondal[4], Rohan Chandra Das[5], Laboni Maji[6], Amit Kumar Rai[7]

[1, 2, 3, 4, 5, 6]*Under Graduate Scholar, Asansol Engineering College, Asansol, India*

[7]*Assistant Professor, Asansol Engineering College, Asansol, India*

Abstract: ***The objective of this paper is to propose a non-intrusive system that can detect fatigue of any human body and issue a warning in time.Driver sleep detection is an automobile safety feature that helps to avoid accidents when the driver falls asleep behind the wheel. According to various researches, roughly 25% of traffic accidents are caused by weariness. Without taking regular breaks, drivers become drowsy and they may fail to recognize the risks. This project can give drivers utmost safety by preventing him from getting asleep while driving.***

Keywords: CNN, OpenCV, Keras, Tensor Flow, Region of Interest (ROI).

I. INTRODUCTION

Even in the year 2021, India is still classified as a third-world country! & of course due to some specific, vivid, and valid reasons; like road condition, claustrophobic traffic, and above all dangerous road mishaps all contribute to INDIA's "third world country" status.

To sustain performance over time, drivers must fully utilize their physiological and cognitive resources. According to the National Transportation Safety Board, any loss of theseResources might result in severe repercussions, including accidents. The dangers of drowsy driving are quite real, and they can result in serious injuries, deaths, and major financial losses. The most terrifying aspect is that drowsy driving isn't merely slipping off to sleep behind the wheel, when a driver is not concentrating solely on the road, drowsy driving can be as simple as a momentary episode of unconsciousness.

The primary goal of the Drowsy Driver Detector is to create a system that can reduce the number of accidents caused by vehicle sleep driving. A more precise detection and accuracy can be delivered with the help of this project. The approach to this problem is to create a detection mechanism that recognizes key characteristics of drowsiness and sends a warning when someone becomes drowsy before it's too late.

II. MODELLING AND ANALYSIS

The Basic requirement for this program is a webcam and a system with Python (3.6.x/3.7.x recommended) installed.

The system must have some needed packages installed. For modelling and analysis, four major packages are used in this project: OpenCV, Tensor Flow, Keras, and Pygame.

1) *OpenCV:* OpenCV (Open-Source Computer Vision Library) is an extensive open source computer vision library which comprise of 2500 optimized algorithm. It has an entire module dedicated to Deep Learning inference (face detection, object tracking) [1].
2) *Tensor Flow:* Tensor Flow is a Python-friendly open-source framework for numerical computation that accelerates and simplifies machine learning [2].
3) *Keras:* Keras is a python based Deep Learning framework, which is official high level API of Tensor Flow. It runs on top of Tensor Flow, Theano or CNTK [3].
4) *Pygame:* The Python Pygame Mixer Library adds audio and sound playback capabilities. The nice thing about it is that it isn't only for developing games. Programmer can use the Mixer library as a stand-alone library to play sound and music in any Python programme [4].

The project uses a behavioural technique which is developed in this research. It is convenient and straightforward implementation procedures and is more dependable. In the suggested methodology, the system would take images as input from a webcam mounted in front of the driver or near the vehicle's dashboard. As a result, the webcam records a sequence of photographs of the driver and transmits them into the system on a regular basis. Once the photos have been supplied into the system, they must be identified.

With the use of **haar cascade classifiers** (Haar Cascade Classifier, is used to categorise depending on object characteristic of interest. The haar cascade classifier's XML files can be used to extract features of the required part) [5]. It is configured according to our region of interest and works accordingly. There are 3 haar cascade files used which are frontal_face, left_eye, and right_eye. A CNN algorithm has been constructed. The application is separated into two sections: the main programme and the model programme. It has a personalised CNN Algorithm to classify the collected images as open or closed.

The model which the program utilised was created with Keras and Convolutional Neural Networks (CNN). A convolutional neural network is a sort of deep neural network that works exceptionally well for image categorization. Convolution is used in CNNs to choose the major important attributes from an input image. To process the image, it includes various learning filters [6].

Normally CNN comprises of 3 main layers- Input, Output & Numerous hidden layers, within which the convolution operation is performed using a filter (kernel or feature-detector) that performs 2d matrix multiplication with the layers.

4 layers are there in our model architecture:

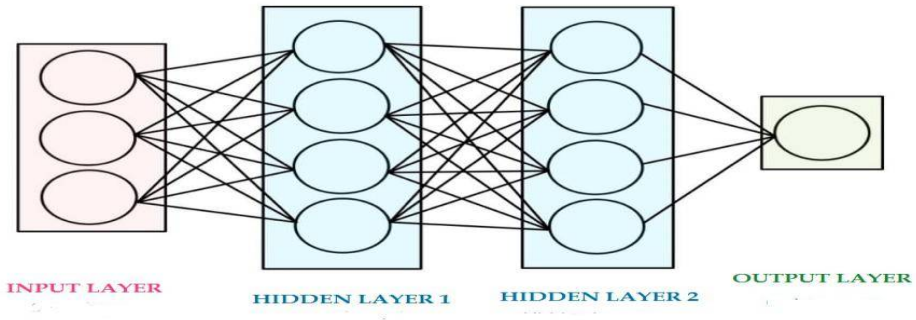


Figure 1: CNN Layers

a) Convolutional layer having 32 nodes and Feature Detector3(kernel/filter)
b) Convolutional layer having 32 nodes and Feature Detector3(kernel/filter)
c) Convolutional layer having 64 nodes and Feature Detector3(kernel/filter)
d) Fully connected layer having 128 nodes

In each layer the program applies ReLu Activation Function[7], which stands for *Rectified Linear Function*& it removes all the negative values making it 0, while the values greater than 0 remains unchanged [R(z)=max (0, z)], except for the outer layer, where we use SoftMax function [8](*it is more generalised logistic activation function which is used for multiclass classification in output layer*) for multinomial probability distribution, which must add up to 1.0.In this case the program is selecting the maximum value within the matrix with the help of MaxPool function [8]. Many types of pooling are there- Maximum, Average, Minimum, Sum & others. Through max-pooling it is reducing the image further which in turn adds flexibility to the image by reducing the number of parameters (features are preserved but information is lessoned)& preventing it from over fitting.

FEATURE **Max Pooling** POOLED
MAP FEATURE MAP

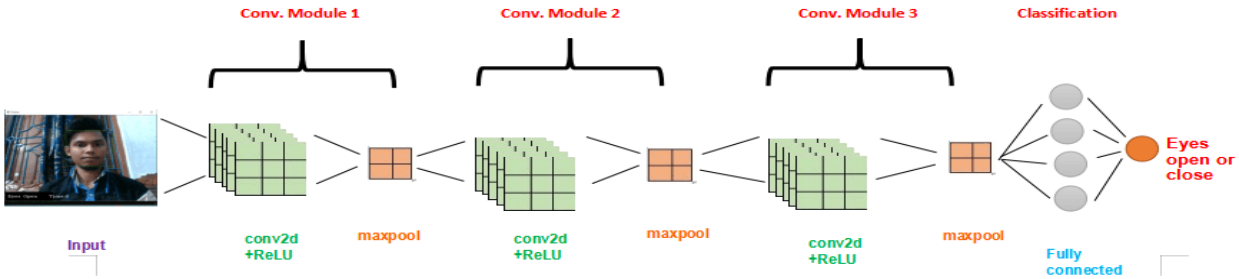


Figure 2: Functioning of CNN Algorithm

III. TEST METHODOLOGY

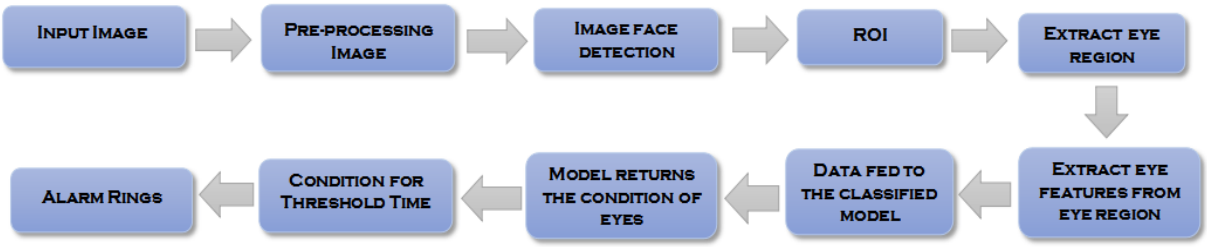


Figure 3: Program Methodology

1) Step 1- Input image from camera and Image pre-processing

Images are taken from a camera as input and are used to test the functioning status of the built system with the help of OpenCV, where cv2.VideoCapture(0) is used to access the camera. The camera collects a sequence of photos continuously and stores every frame with the command cap.read(). Then the images are converted from colour (BGR) to grayscale with the command cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY) to calculations as it contains only one layer image (0-255).

2) Step 2- Face detection and creation of region of interest (ROI)

After the image is being processed, then the face is detected from the frame with the help of the haarcascade files and return the co-ordinates of the face (face.detectMultiScale(gray)). With these co-ordinates we then create a bounding box over the face which is our Region of Interest(ROI).

3) Step 3- Extract eye region, eye features and feed to the classifier model

In the same way, the eye region of both the eyes are detected using code leye.detectMultiScale(gray) and reye.detectMultiScale(gray) for left and right eye respectively. Then the eye features are acquired from the frame (r_eye=frame[y:y+h,x:x+w]) and are fed to the classifier(model.predict_classes(r_eye)).

4) Step 4- Eyes are opened or closed

By executing the model software, the system then retrieves the Region of Interest of the Eyes in order to evaluate the eye status, i.e., open or closed, using the "if conditional check".

5) Step 5- Alarm sound

After checking the eye statuses, if both the eyes are closed, that too for more than 4 seconds (each second will score 5 units) the value will reach 4*5=20, which is the threshold value, the alarm will set on. And as soon as the eyes are opened, the value will reinitialised to zero and the alarm will be set off.

IV. RESULTS AND DISCUSSION

The constructed driver anomaly observation framework is capable of detecting lethargy, laziness, and reckless driving activities in a short period of time. The Laziness Detecting Framework, which is based on the driver's eye closure, can discern between normal eye flicker and exhaustion, as well as drowsiness when driving. The suggested technology is capable of preventing accidents caused by tiredness while driving. Because eyes in Figure 4 are open, the system has shown the status as open and the Time is 0. As a result, the individual does not appear to be sleepy.

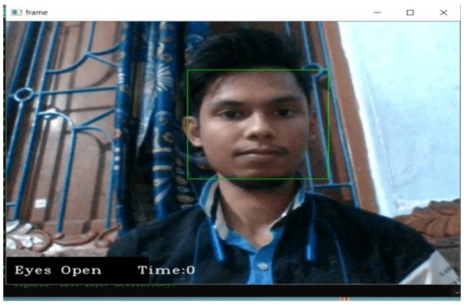


Figure 4: Eyes Open (Not feeling sleepy)

Consider Figure 5: The individual has closed one eye, but because both eyes are open, it's possible that he or she is scouring his or her eye.

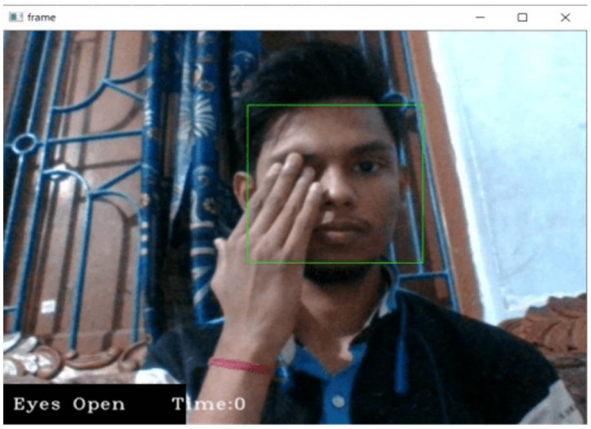


Figure 5: One eye closed

Consider Figure 6, where an eye has been closed for a long time and the Time value has climbed to the point where it has crossed the threshold value, thus the system has sent a warning to wake him up from his drowsiness.

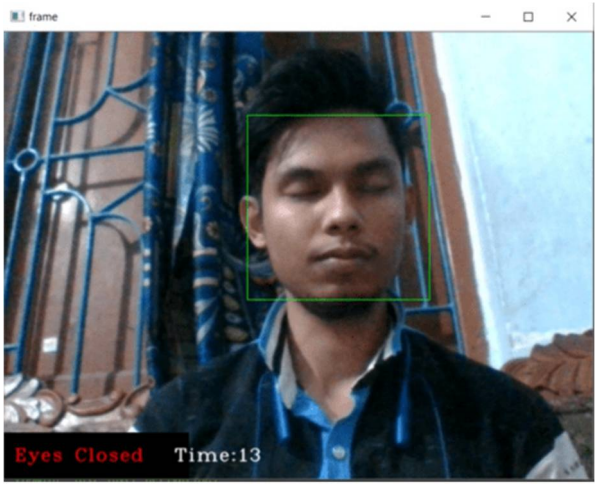


Figure 6: Eyes Closed: Risk of feeling drowsy

The technology can detect whether the eyes are open or closed during the monitoring. A warning indication appears when the eyes are closed for an extended period of time. On the basis of continual eye closures, processing determines the driver's attention level.

V. CONCLUSION & SCOPE OF FUTURE WORK

Vehicle safety is a function which should be maximised to make sure that as many lives as possible can be saved. Here, in our project, a drowsy-driver detection system has been developed, which encompasses idea of the safe driving for all the drivers, be it in private or in commercial section, by alerting the driver at the hour of their need! This drowsy alert system can be executed in numerous possible ways, not only the drivers but also for late night workers like watchmen, factory workers, policemen can be warned of their condition(fatigue, sleepiness etc.). This project can be most safest and feasible technology available that can avert an accident or any undesired incident.

Something further is planned in the upcoming days, for the improvement of our work, both in terms of upgrading the features & as keeping the social benefit as our utmost priority.

A. Firstly,

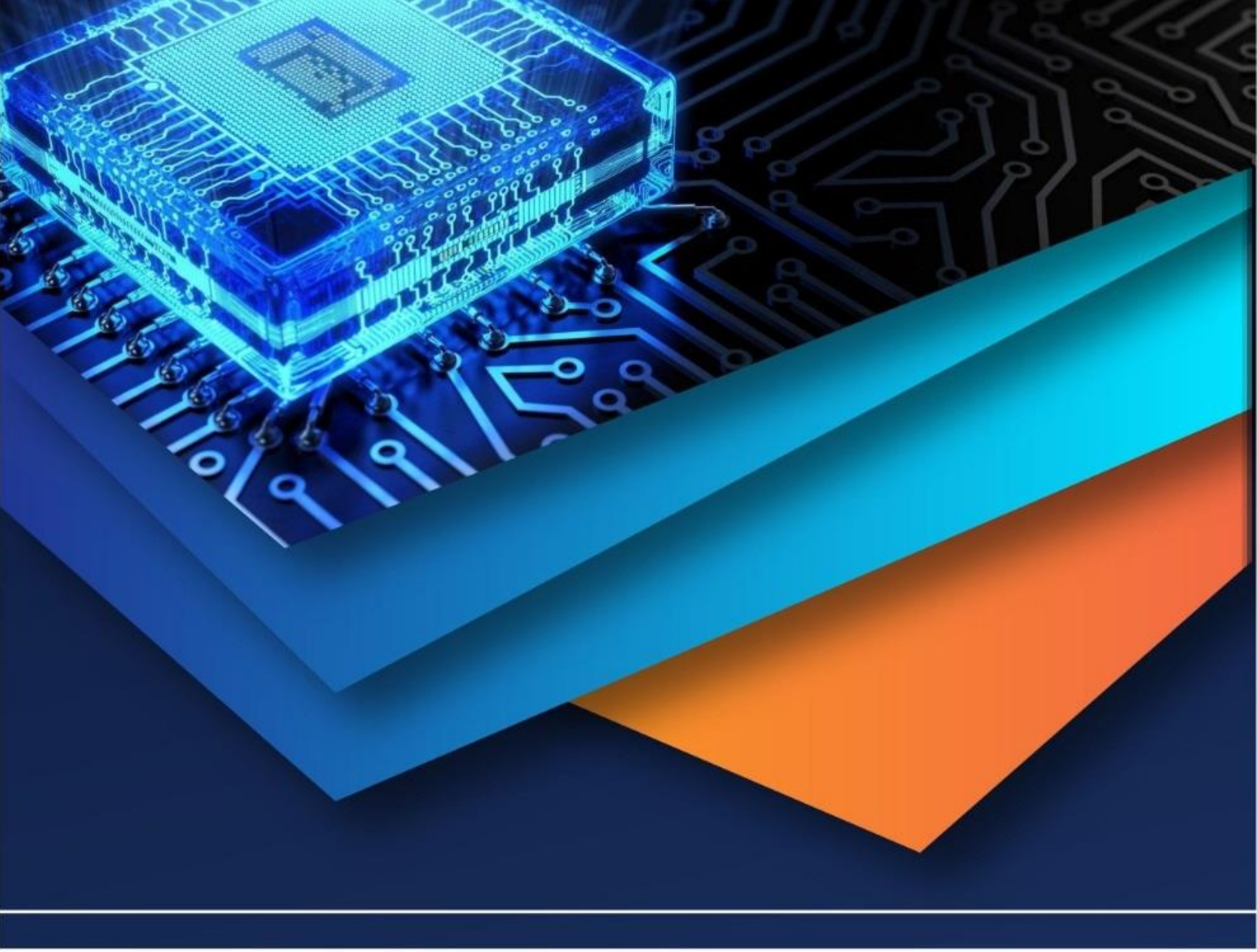
Our Project work will come even more handy & user friendly, if this idea is deployed into an android/ios or in web application. In this way the users can easily classify if they are in a drowsy state just by using their own cell phones attached to a stand or holder!

B. Secondly,

Changing the time-limit from static (which is as of now 4 sec) to dynamic is taken into consideration, depending upon the cruising velocity of the vehicle. Because if the driver is plying the vehicle in highway at higher speed, suppose of 80-100 km/hr, then mishaps can occur within the blink of an eye, so 6 sec seems way much. In this kind of scenarios, to acclimatize the system with various situations, adding up the instantaneous speed as a parameter is also been planned!

REFERENCES

[1] https://opencv.org/
[2] https://www.tensorflow.org/
[3] https://keras.io/about/
[4] https://pypi.org/project/pygame/
[5] https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html
[6] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[7] https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
[8] https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/
[9] Pushpa Narayan, "IIT-M's Wake Up Call for Tired Drivers", Available at: timesofindia.indiatimes.com
[10] B. Verma and A. Choudhary, "Framework for Dynamic Hand Gesture Recognition using Grassmann Manifold for Intelligent Vehicles", IET Intelligent Transport Systems.
[11] Manu B.N,Facial Features Monitoring for Real Time Drowsiness Detection by, 2016 12thInternational Conference on Innovations in Information Technology.
[12] Smart Eye, "Smart Eye", Available at: https://smarteye.se/, Accessed at 2018.
[13] K. Srijayathi, International Journal ofScience, "Implementation of the Driver Drowsiness Detection", System Engineering and Technology Research (IJSETR) on 9th September 2013, Volume 2, Issue 9
[14] "Driver Drowsiness Detection Using Convolutional Neural Networks", Published in: 2020 Cybernetics & Informatics (K&I), Feb. 2020



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY