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.2021.35526

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 Jun 2021- Available at www.ijraset.com

Drowsy Face Detection using Deep Learning Algorithms

Anish Chandrasekaran[1], Shreyas Bhat[2], Charmika Tankala[3], Manasi Tambade[4],
Prof. Shraddha Khonde[5], Prof. Deepali Ahir[6]
[1,2,3,4]*Student,* [5,6]*Professor, Modern Education Society's College Of Engineering Affiliated to Savitribai Phule Pune University Pune, India*

Abstract - An important aspect of machine vision and image processing could be drowsiness detection system due to its high significance. In recent years there have been many research projects reported in the literature inthis field.In this paper unlike the conventional drowsiness detection methods using machine learning we used deep learning techniques.Driver drowsiness results in many car crashes and fatalities worldwide.Whereas drowsiness in onlineattendees results in less attention span and decrease in the learning capabilities, such as meetings, lectures, webinars held. The advancement in computing technology has provided the means for building intelligent face detection systems.Faces contain information that can be used to interpret levels of drowsiness.Here we employ deep learning to determine actual human behavior during drowsiness episodes targeting the facial features.

Keywords- Drowsy face Detection, Eye Feature Extraction, Deep learning, Transfer Learning, CNN

I. INTRODUCTION

Drowsiness influences mental alertness, decreasing an individual's capability of basic understanding of one's surroundings and expanding the possibility of a human errors.As per the data received from the police department of States/UTs in India, 1,50,785 persons and 1,47,913 persons were killed in road accidents during the calendar years 2016 and 2017 respectively[1].It has been estimated that drowsiness causes between 10 % and 20 % of traffic accidents, causing both fatalities dead [2] and injuries [3], whereas within the trucking industry 57 % of fatal truck accidents are caused by this problem [4],[5].30 % of all traffic accidents have been caused by drowsiness[6].In the USA, drowsiness is responsible for 100000 traffic accidents yearly producing costs of close to 12.000 million dollars [7]. In Germany, one out of four traffic accidents originate from drowsiness, while in England 20 % of all traffic accidents are produced by drowsiness [8], and in Australia 1500 million dollars has been spent on fatalities resulting from this problem [9]. Drowsiness or sleepiness can be described as a biological state where the body is in-transition from an awake state to a sleeping state. At this stage, a driver can lose concentration and be unable totake actions such as avoiding head-on collisions or braking timelessly.

II. DATASET

A. MRL Dataset

MRL eye dataset is a large scale dataset of Human eye images. This dataset contains infrared images in low and high resolution all captured in different lighting conditions with varying intensities and by using different devices. The dataset is suitable for testing several features or trainable classifiers.[10]

B. Properties

The properties of this dataset are as follows:

1) *subject ID :* In the dataset, we collected the data of 37 different persons (33 men and 4 women).
2) *image ID :* The dataset consists of 84,898 images.
3) *gender [0 - man, 1 - woman] :* The dataset contains the information about gender for each image (man, woman).
4) *glasses [0 - no, 1 - yes] :* The information if the eye image contains glasses is also provided for each image (with and without the glasses).
5) *eye state [0 - closed, 1 - open] :* This property contains the information about two eye states (open, close).

6) *reflections [0 - none, 1 - small, 2 - big] :* We annotated three reflection states based on the size of reflections (none, small, and big reflections)
7) *lighting conditions [0 - bad, 1 - good] :* Each image has two states (bad, good) based on the amount of light during capturing the videos.
8) *sensor ID [01 - RealSense, 02 - IDS, 03 - Aptina]:* At this moment, the dataset contains the images captured by three different sensors (Intel RealSense RS 300 sensor with 640 x 480 resolution, IDS Imaging sensor with 1280 x 1024 resolution, and Aptina sensor with 752 x 480 resolution)[10]

III. PROPOSED MODEL

The system would gather the images from webcam and feed them into a Deep Learning model The approach we used forthis system is as follows :

A. Take image as input from a camera.
B. Detect the face in the image and create a Region of Interest (ROI).
C. Extract various features from the ROI and feed it to the classifier.
D. The Deep Learning framework will classify the features to train and further test the model
E. Calculate score to check whether the person is drowsy and display the Percentage of Drowsiness.

IV. METHODOLOGIES USED

A. Local Binary Pattern

Local Binary Pattern (LBP) is a fast texture operator that labels pixels in an image by thresholding each pixel's vicinity and converting the result to a binary number. In its most basic form, the LBP feature vector is constructed as follows:

1) Divide the examined window into cells (e.g. 16x16 pixels for each cell).
2) Compare each pixel in a cell to each of its eight neighbours (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels in a clockwise or counterclockwise direction around a circle.
3) By altering the radius of the circle surrounding the pixel, R, and the quantisation of the angular space P in the preceding step, the neighbours considered can be varied.
4) Write "0" where the value of the centre pixel is greater than the neighbor's value. If not, write "1." This yields a binary number of eight digits (which is usually converted to decimal for convenience).
5) Calculate the histogram of the frequency of each "number" occurring in each cell (i.e., each combination of which pixels are smaller and which are greater than the center). A 256-dimensional feature vector can be observed in this histogram.
6) Optionally normalize the histogram.
7) Histograms of all cells concatenated (normalised). This results in a feature vector for the whole window.

B. Haar Cascades

Haar Cascades is a well-known mathematical model for object detection in machine learning. The goal of the procedureis to identify and categorise the image's features. There are three types of features: two rectangles, three rectangles, and four rectangles. The difference in the number of pixels in two rectangular areas equals the characteristic value of two rectangles. These zones can be both horizontal and vertical in nature.

The sum of the pixels in the two side rectangles minus pixels in the central rectangle is used to determine the feature of the image of the three rectangles. Finally, this is the difference between the rectangles on the diagonals for the four rectangles.

The stage of training the classifier using the AdaBoost approach begins after the features from positive and negative images have been defined. The goal is to combine "weak" classifiers to build a "strong" classifier. In this scenario, a "weak" classifieris constructed using only a few features. AdaBoost is also used to identify the most important attributes for categorization.

Sliding windows are classified throughout the detecting process. The decision tree approach or the cascade approach are used to classify the data. To be discovered, the window must pass through a series of weak classifiers. If one of the classifiers rejects the window, the following classifier is not engaged, and the window is not recognised.

C. Transfer Learning

The basic principle of transfer learning is straightforward: take a model that has been trained on a large dataset and applyit to a

smaller dataset. We freeze the early convolutional layers of the network for object recognition and just train the lastfew levels that produce a prediction. The concept is that the convolutional layers extract broad, low-level properties that apply across images — such as edges, patterns, and gradients — whereas the later layers recognise individual aspects within animage, such as eyes or wheels.Because there are universal, low-level properties shared by photos, we can leverage a network trained on unrelated categories in a big dataset (typically Imagenet) and apply it to our own problem.

V. IMPLEMENTATION

A. Data Pre-processing

- Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to aclean data set.
- In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis
- Steps involved in pre-processing:-
  1) Applying transformations on data such as resizing and converting into different formats like Color image to RGBimage.
  2) Shuffling of data to avoid over-fitting conditions.
  3) Normalising the data.

Tech stack for the module: MRL Eye Dataset

B. Training and Testing

- In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the modelbuilt. Data points in the training set are excluded from the test (validation) set. We created training Data by convertingthe images into an Array.
- We Later, applied the resultant training dataset into a Deep Learning Framework. In this project we be used TensorFlow with Keras which acts as a interface for TensorFlow library.
- We have used Transfer Learning for building the Neural Network. It is a method that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.The pre-trained model used is MobileNet.

Tech stack for the module: TensorFlow, Keras, MobileNet.

C. Evaluation

- Firstly, we evaluated our Model with respect to an Image to check if the model bodes well with the image provided.
- We extracted only those features from the facial image that are necessary to us. For drowsy face detection eyes are considered the most important factor.
- For the purpose of extraction of face we used Local Binary Pattern(LBP) and for the further extraction of eyes Haar Cascade algorithm was used.
- From the model the Accuracy Score is checked.When the Accuracy turned out to be satisfactory then the model was evaluated on Real-Time Video stream from the Webcam.
- For the purpose of Real-Time video detection we used an open Source Python library called OpenCV. It is a library of programming functions mainly aimed at real-time computer vision application.

Tech stack for the module: OpenCV

D. User Interface

- Users cannot directly interact with the DL Model.Though theoretically it is possible but not user-friendly to do so. Therefore, we need an User Interface so that users can use the developed model easily , without any hassle.
- For this GUI is we used a python library called Tkinter. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI.
- Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.
- GUI is created by generating a frame with specific buttons mapping to their corresponding functions of recording and detection.

Tech stack for the module: Tkinter

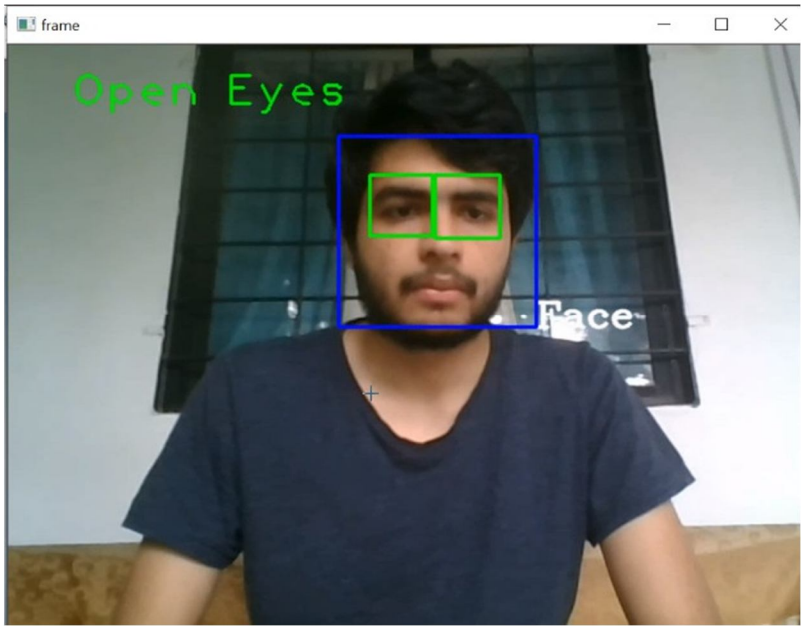
VI. TEST CASES AND RESULTS

A. Test Cases

TABLE I TEST CASES

ID	Attribute	State	Expected	Actual	Result
TC1	Eyes	Open	Open Eyes	Open Eyes	PASS
TC2	Eyes	Closed	Closed Eyes	Closed Eyes	PASS
TC3	Eyes	Closed > 3 sec	Drowsy	Drowsy	PASS
TC4	Eyes	Open > 15 sec	Error: Static Image	Error: Static Image	PASS

Fig. 1. Open



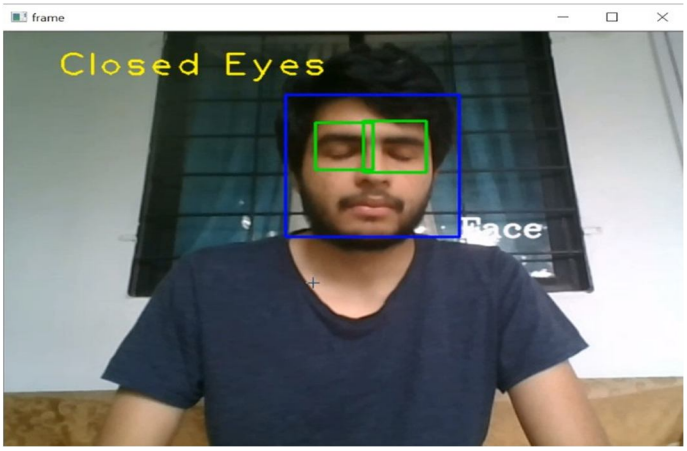


Fig. 2. Closed

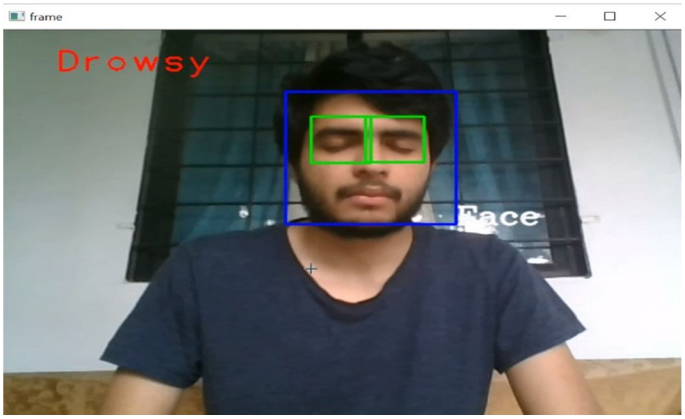


Fig. 3. Drowsy

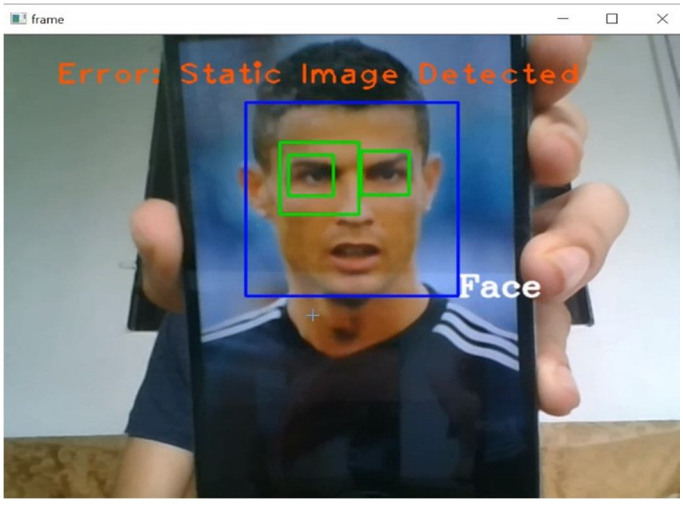


Fig. 4. Static Image

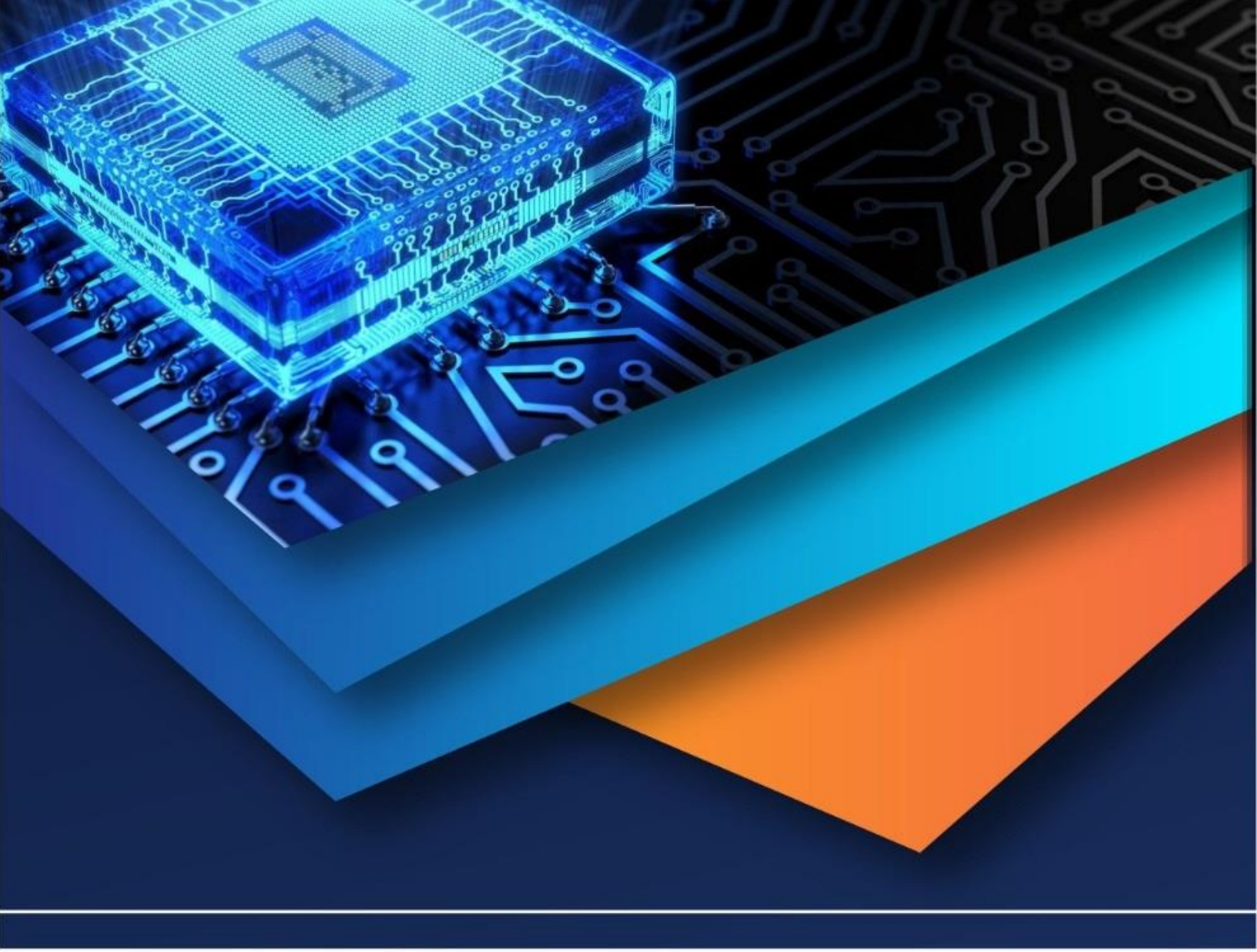
VII. CONCLUSION

There are many techniques that support behavioral methods and machine learning which will be utilized for the purpose of drowsiness detection.

In this project, we developed an image-based classifier for Drowsiness Detection using Deep Learning. We opted for the Transfer Learning approach for our model and obtained an Accuracy of 96.99%. So our model is able to successfully classify whether a person is Drowsy or not.

REFERENCES

[1] www.motorindiaonline.in/driver-welfare/drowsy-driving-a-safety-challenge/
[2] Z Tian, H Qin, Real-time driver's eye state detection. Proceedings of the IEEE International Conference on VehicularElectronics and Safety, October 2005, 285–289
[3] W Dong, X Wu, Driver fatigue detection based on the distance of eyelid. Proceedings of the IEEE International Workshopon VLSI Design and Video Technology (IWVDVT '05), May 2005, Suzhou-China, 397–400
[4] Q Ji, X Yang, Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. Real-Time Imaging 8(5), 357–377(2002).
[5] LM Bergasa, J Nuevo, MA Sotelo, M Vázquez, Real-time system for monitoring driver vigilance. Proceedings of the IEEE Intelligent Vehicles Symposium, June 2004, 78–83
[6] L Fletcher, L Petersson, A Zelinsky, Driver assistance systems based on vision in and out of vehicles. Proceedings of theIEEE Symposium on Intelligent Vehicles, 2003, 322–327
[7] NHTSA, Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management(National Highway Traffic Safety Administration, Washington, DC, USA, 1998)
[8] L Hagenmeyer, in Development of a multimodal, universal human-machine-interface for hypovigilance-managementsystems,Ph, ed. by . D. thesis (University of Stuttgart, Stuttgart, Germany, 2007)
[9] G Longhurst, Understanding Driver Visual Behaviour (Seeing Machine, Canberra, Australia)
[10] http://mrl.cs.vsb.cz/eyedataset



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY