IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Integrated Driver Alert System Using Deep Learning and Internet of Things

Rounak Bhadra
School of Computer Science & Engineering, Lovely Professional University, India

Sourabh Kumar
School of Computer Science & Engineering, Lovely Professional University, India

AshutoshVerma
School of Computer Science & Engineering, Lovely Professional University, India

Mukul Suthar
School of Computer Science & Engineering, Lovely Professional University, India

Krishnaveer Singh
School of Computer Science & Engineering, Lovely Professional University, India

Savleen Kaur
School of Computer Science & Engineering, Lovely Professional University, India

***Abstract*— In this paper we have used "Tensorflow object Detection API" for traffic sign detection and "Haar feature based cascade classifier" for face detection. To determine the eye state we have used SVM classifier that is trained on blinking and non-blinking state of eyes. Then we have calculated Eye Aspects Ratio(EAR) to detect whether eyes are closed or open. A dataset suitable for "Tensorflow Object Detection API framework" has been created. Images of traffic signs for the dataset were collected from GTSRB (German Traffic Sign Recognition Benchmark) and Google open image dataset. Then we have trained our model based on the dataset and results were obtained on real time images.**

Keywords—Machine Learning, TensorFlow, Image Analysis, Deep Learning, OpenCV.

I. INTRODUCTION

The majority of car manufacturers are concerned about driver warning systems. We all know how common road accidents are all over the world. As a result, many people can die. There are several measurements that can be taken to prevent this. One of the most important solutions is a driver warning system. As a result, all car manufacturers that are using technology to solve this problem should be applauded. One of the most important solutions is a driver warning system. Thus, all smart car manufacturers using this technology to solve this problem are looking for an integrated platform that will address all issues related to drowsiness, driver exhaustion, and retaining calm while driving while following all traffic laws and regulations. So, in this project, we'll be creating an integrated system with two components.

As we get closer to self-driving cars and smart cars, demand for driver warning systems is growing. Several studies have been conducted in this area. However, no such scheme exists that can inform the United States of each driver's status in the same way that traffic rules can. In this paper, the authors propose a Deep Learning model to analyze the images and use TensorFlow to play with the data.

In the first one we will analyze the face and possibility or occurring of drowsiness or behavior of the driver to detect if there is any fatigue by using OpenCV [10] and Deep Learning. Side by side, the second part of the model will focus on the road-side sign boards which tell all the details traffic rules and regulations like if there was a sign board which tells that speed limit should be below 40 km/h or huge turn ahead. Then if the driver was feeling sleepy then an alarm will be there to wake him up and in the second part we will tell him details about that you have missed this important sign board regarding speed limit or whatever it is. We have used TkinterGUI [11] to implement this process.

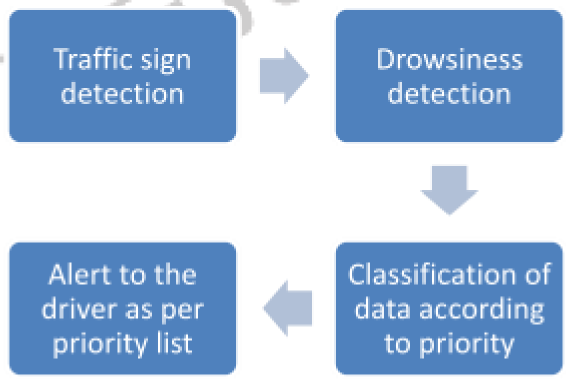


Fig.1- Flowchart of the proposed model

The outline of our research paper is decorated as mentioned below. Section II gives the background of driver state and classification of traffic signs in Deep Learning. Section III contains proposed methodology. Section IV mentions the dataset that is used in this model. Section V contains data preprocessing of the dataset and analysis. In section VI methodology on pre-processed data using Convolution Neural Network is done. Result and outcome analysis is written in Section VII. In Section VIII, Conclusion and Future Work is given, post which we have attached acknowledgement. In the last of this paper we have attached all the references.

II. BACKGROUND

Previous papers have their own merits and limitations. Either they only detect drowsiness or traffic signals. There are some existing approaches, in the study by Kilic, Irfan, and GalipAydin[1] TensorFlow's Object Detection API is used in Turkey's Traffic Sign Detection and recognition of the traffic signs. A dataset needed for TensorFlow's Object Detection API has been created for traffic signs by different kinds of images of traffic signs in different conditions. The detection of traffic signals is done using CNN algorithm [2].

According to the research paper by Wali, Safat B. et al [3], an efficient and working TSDR system can be categorized into several stages i.e., Preprocessing, Detection, Tracking, and Recognition. In the preprocessing stage, the visual outlook of images has been ameliorated. The drowsiness is detected using Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR)[4]. The process starts by extracting and detecting the face from the video frame using the Support Vector Machine (SVM) face detector. Then another approach for eye and mouth detection and state analysis based on Circular Hough Transform (CHT) is applied on eyes and mouth regions [5].

Deep Learning has an effective role in detecting drowsiness using TensorFlow, OpenCV and TkinterGUI. A number of images of human faces have been used in the experiment. Some pictures are simple or

of those people who are not feeling drowsy but in some images people face are drowsy or eyes are not properly open. Our model can also Traffic signals for which we have used the Faster_RCNN_InceptionV2_COCO [12] model which is used for object detection. For detection of drowsiness of the driver we have used a classifier named Cascaded Adaboost with the Haar like features to detect the face region. A dataset of images have been processed looking for the expression and coordination change of eyes, nose and ears. The coordinates have been used to calculate EAR (Eye Aspect Ratio) and MAR (Mouth Aspect Ratio) which shows the condition of the face i.e. sleepy or not sleepy.

III. PROPOSED METHODOLOGY

Tensor Flow is an open source library consisting of mainly symbolic Math libraries which can perform dataflow programming across different ranges of tasks. This library is also widely used for ML applications like neural networks. In the name TensorFlow, 'Tensor' means multidimensional data arrays. Hence, TensorFlow computations are expressed in the form of graphs of stateful dataflow.[9]

Machine Learning is a special type of programming which enables the machine to have the capability to learn from the data without being programmed so that the machine can react according to that.

Python is undoubtedly one of the best languages for machine learning and there are a lot of inbuilt libraries like SciPy, Pandas and NumPy; great for Machine Learning.

OpenCV stands for Open Source Computer vision. It includes advanced Computer Vision algorithms that can be used for image processing, image manipulation and much more. It can also be used for image blending operation.[10] TkinterGUI is a GUI library for Python which provides fast, easy and powerful interface.[11]

IV. DATASET

In the first part of our Drowsiness detection model we are using real time data [6]. Here data is taken by using Haar feature based cascade classifier. After detecting the face, a further processing is done to detect whether the driver is sleepy or not. Then in the next part which is Traffic Sign Detection, we will use the Google open images dataset [6] where type is "Detection" and category is "Traffic sign".

We will create a dataset of all possible traffic signs and then arrange them according to the priority list. Here we are using the Open Image v6 which contains 16 million boundary boxes for 600 object classes on 1.9 million images with object location annotations.

Fig 2. Examples of traffic signals [6]

V. DATA PREPROCESSING

In this section, we process the data that we obtain from the car and also classify the data that is already presented to us from open images dataset. Data that is presented in these dataset must be classified into several classes for better identification and reaction by our model. Data will be collected by the driver's car. The photo may be taken horizontally or vertically. So we have to reduce the aspect ratio to standard values. Then we will label the images and will make a table of dataset according to that like no passing, speed limit, no entry, bumper ahead, go right, animal crossing, road work, U turn, bumpy road yield. Then we will pass this data to the model and our trained model will detect the images on the basis of priority list. The priority list will be based on the importance of that data in preventing significant loss. Factors like speed limit, turn in road, bumper, and traffic will be given higher priority than others. Such priority type and event will be created and our trained model will react to each event according to this preprocessed list.

Priority List	Event(Traffic sign)
Emergency	Speed Limit
Very High	Any Turn
High	Bumper Ahead
Low	Straight Road
Very Low	Blow Horn

Table 1: Classification of sign according to priority list

Go straight or left	Road work
Go straight or right	Pedestrians
Ahead only	Crosswalk
No, stopping and parking	Wind on hand
Stop	Wild animals crossing
Children crossing	U Turn is not allowed
Bicycles crossing	End of all speed and passing limits
No vehicles	Traffic lights
No entry	End of all speed
90km_h Speed limit	Dangerous curve to the left
80km_h Speed limit	Double curve–Right
70km_h Speed limit	Keep left
60km_h Speed limit	Turn right ahead
50km_h Speed limit	Go from both sides
30km_h Speed limit	No passing
20km/h Speed limit	End of no passing
120km_h Speed limit	Bumpy road
100km_h Speed limit	Slippery road
No passing for vechiles	

Table 2: Types of traffic signs

VI. METHODOLOGY

Now in the first part of this model consisting of Drowsiness Detection, firstly we have trained Haar feature-based cascade classifiers with a lot of positive and negative images. This method was first proposed and explained by "Paul Viola" and "Michael Jones" in the paper "Rapid object Detection using a Boosted Cascade of Simple Features" in 2001. After exploiting the face we can see that the image has been segmented into numbers of rectangular areas. Next, we have detected the eyes position by visualizing landmark coordinates. To determine the eye state we have used an SVM classifier that is trained on blinking and non-blinking state of eye.

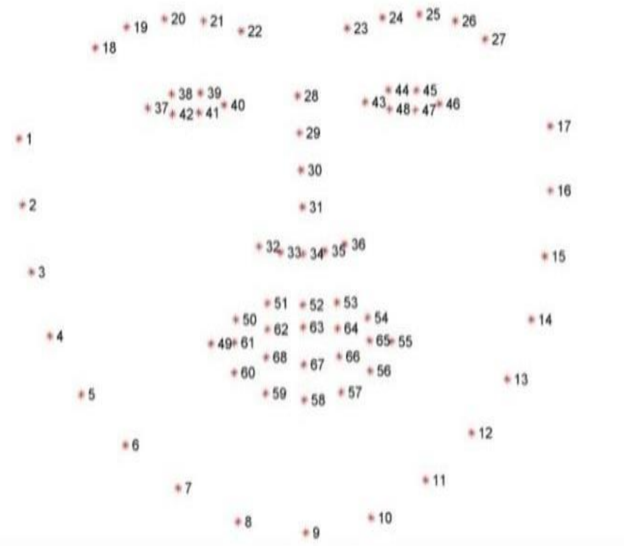
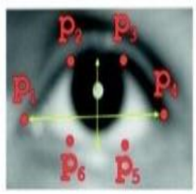
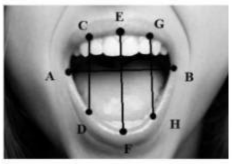


Fig.3: Detection of coordinates of eyes, nose, mouth and face[14]



$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Fig.4. Formula to calculate EAR[4]



$$MAR = \frac{|EF|}{|AB|}$$

Fig.5: Formula to calculate MAR[4]

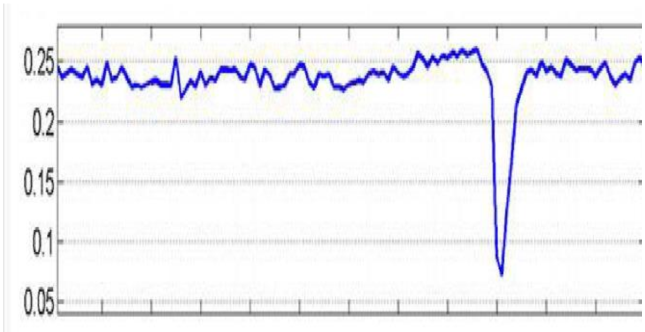


Fig 6. EAR for single blink of eye

We will calculate the Eye Aspect Ratio. Now if Ear value is zero or close to the zero then it will be treated as close otherwise we will treat it as open. That is how we can detect whether the person is sleeping or not, post which proceed to part II of our model which is detection of a traffic sign and inform it to the driver who was sleeping. Now in the second part to determine different location of multiple objects in a single image we have used the Tensor Flow's Object Detection API[9]. The base of this API is done by J. Huang etal. Later on, we have trained the SSD_inception_v2_coco [12] for TensorFlow version 1.x model using the Tensor Flow's Object Detection API framework by GTSRB (German Traffic Sign Recognition Benchmark)[8]. This dataset has 43 classes with corresponding labels of different traffic signs. We have trained the model on the Nvidia GTX 1050 Ti Graphic Card, using Python latest version and TensorFlow 1.14. After training the model, it is tested on test images and later being assigned a random road picture to test its functionality.

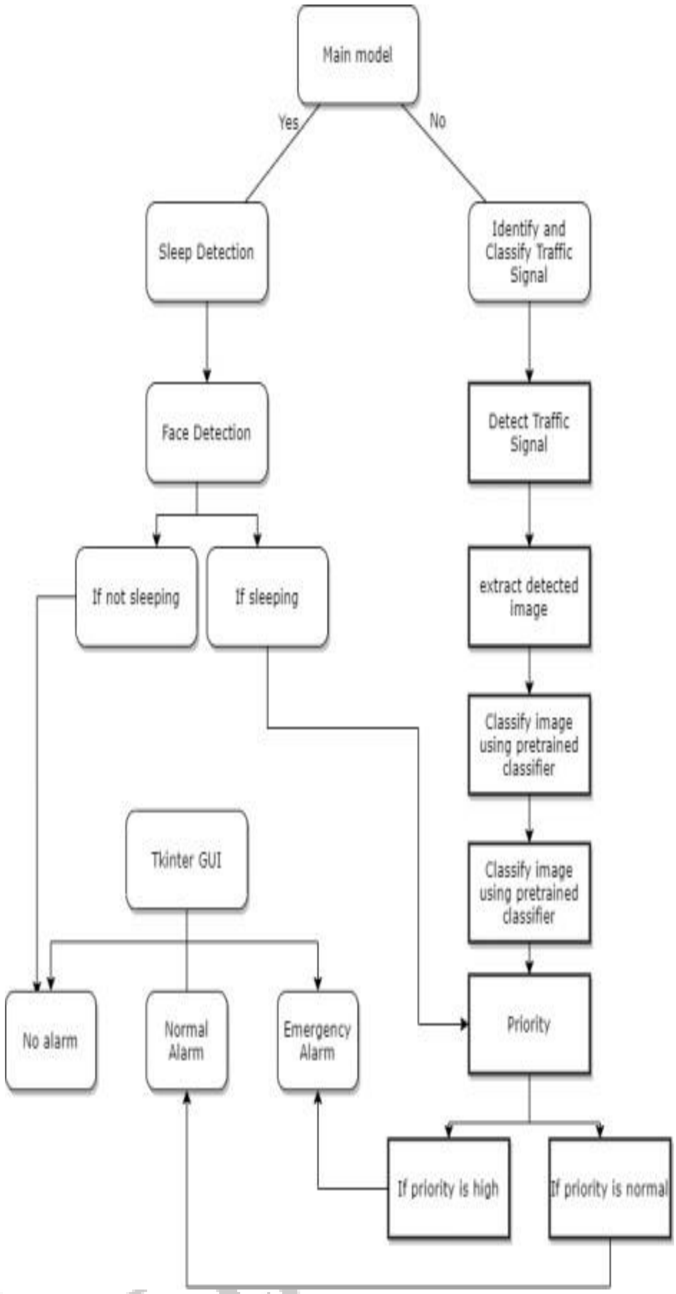


Fig. 7: Model architecture of our proposed work

VII. IMPLEMENTATION AND RESULT ANALYSIS

As we have integrated two parts of our model in the TkinterGUI interface, at first we were given access to the camera and after doing all the process. It will prompt an alarm if the driver is sleeping and he misses some important sign. Here the priority and intensity of the alarm is variable. For example, if the priority of the sign is high then there is an alarm with maximum intensity and so on. Also if the driver was sleeping but there was no important sign he missed then there will be a light alarm.

Fig 8.: Detection of traffic sign stop[2]

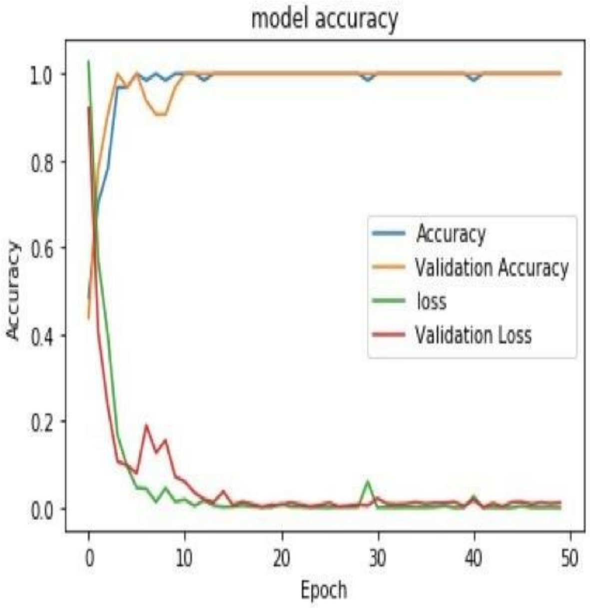


Fig.9. Model Accuracy with respect to Epochs

VIII. CONCLUSION AND FUTURE WORK

Till now we have just implemented this on a Tkinter based GUI[11]. Over a period of time, this can also be applied to a car where we can integrate it with the main car structure. In the car we will use a webcam which helps to capture real time images of the driver and send this to the model. We have trained our model on the basis of GRSE [8] but in near future we will train it on many other databases by which it will be capable of usage in many countries.

In the output section, presently our model is only alarming, however in the near future; we will show the sign on an integrated along with some additional information. To take an example, the model will also be sufficient enough to inform the driver periodically of any signboard information or any upcoming journey details, such as an accident on the road or abnormal weather conditions. Thus in this way, we can provide an integrated system that will enhance the security of the smart car to further level.

X. REFERENCES

1. Kilic, Irfan, and GalipAydin. "Traffic Sign Detection and Recognition Using TensorFlow's Object Detection API With A New Benchmark Dataset." In 2020 International Conference on Electrical Engineering (ICEE)*)*, pp. 1-5. IEEE, 2020.

.2.Wen, Lihua, and Kang-Hyun Jo. "Traffic sign recognition and classification with modified residual networks." IEEE/SICE International Symposium on*System Integration (SII)*, pp. 835-840.IEEE.

3. Wali, Safat B., Mahammad A. Hannan, AiniHussain, and Salina A. Samad. "An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and svm." *Mathematical Problems in Engineering* 2015 (2015).

4. Mehta, Sukrit, SharadDadhich, SahilGumber, and ArpitaJadhav Bhatt. "Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio." In *Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, Jaipur-India*. 2019.

5. Alioua, Nawal, Aouatif Amine, Mohammed Rziza, and DrissAboutajdine. "Driver's fatigue and drowsiness detection to reduce traffic accidents on road."In *International Conference on Computer Analysis of Images and Patterns*, pp. 397-404.Springer, Berlin, Heidelberg, 2011.

6."https://storage.googleapis.com/openimages/web/visualizer/index.html?set=train&type=detection&c=%2Fm%2F01mqdt"

7. "Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," Neural Networks,

2018, doi: 10.1016/j.neunet.2018.01.005."

8. "The German Traffic Sign Detection Benchmark." http://benchmark.ini.rub.de/?section=gtsdb&subsection=news (accessed Apr. 15, 2015)".

9."https://www.tensorflow.org/federated/tutorials/federated_learning_for_image_classification".

10."
https://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/".

10. "https://docs.opencv.org/master/d9/df8/tutorial_root.html".
11." https://docs.Python.org/3/library/Tkinter.html#".
12."faster_rcnn_inception_v2_coco."http://download.tensorflow.org/models/object_detection/faster_rcnn_inception_v2_coco_2018_01_28.tar.gz.

13. A. Shustanov and P. Yakimov, "CNN Design for Real-Time Traffic Sign Recognition," Procedia Eng., vol. 201, pp. 718–725, Jan. 2017, doi: 10.1016/J.PROENG.2017.09.594.

14."https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-Python/".