International Journal of Scientific Research in Computer Science and Engineering

Research Paper

Vol., Issue., pp.-, Month (year)

E-ISSN:

Drowsiness Detection System

Amit Divekar[1*], Shrey Srivastava[1], Ved Kulkarni[1]

[1] Department of Computer Science, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, Chennai, India

Email: amitvishvas.divekar2019@vitstudent.ac.in, shrey.srivastava2019a@vitstudent.ac.in, ved.kulkarni2019@vitstudent.ac.in

*Corresponding author: amitvishvas.divekar2019@vitstudent.ac.in, Mob: +91-92841-99286

Available online at: www.isroset.org

Received: dd/mm/yyyy, Accepted: dd/mm/yyyy, Online: dd/mm/yyyy

***Abstract* —** Every day, the number of cars on the road rises and with it, rise the number of road accidents. Deaths occurring in road accidents due to fatigued and drowsy driving account for a significant percentage of the total deaths across the world. So, now more than ever, there is a need to come up with a solution to these problems. Solutions exist, but are only available in high end, top of the line models. Thus, keeping them out of the reach of the majority. This project aims to solve this very problem by developing a non-intrusive, modular, compact, easy-to-deploy yet affordable solution. The proposed system monitors the driver's face through a suitably placed camera. The video feed it put through a Faster-RCNN, for tracing and detecting the driver face. Each region on the face is associated with a point vector. A filter is applied to single out his/her eyes, and Eye Aspect Ratio (EAR) is calculated. It decreases if the driver's eyes close or if he/she looks away i.e. in any situation that he/she is distracted, this ratio decreases. This, in turn increments a score. After a certain set threshold, an alarm is sounded and the driver alerted. This system can be extended in the fields of aviation, online teaching (to make sure students are paying attention) and heavy machine industries.

***Keywords* —** Drowsiness detection, FRCNN.

I. INTRODUCTION

In 2018 alone, over 1.5 lakh people lost their life in a road accident and over 4.4 lakh people were injured. This number accounts to approximately one-third of all accidental deaths caused in India. [14] This fraction is similar all over the world. According to records, only 0.4% of road accidents were due to physical fatigue of drivers. [15] However, the reported number is often incorrect and far lower than the actual number. This is largely due to the fact that in the aftermath of an accident, it is not always possible to tell if the driver was fatigued and the amount sleep, he/she had. [17] A survey conducted in the United States reported that around 16-21% of all fatal accidents are a result of drowsy driving. [25] Drowsy driving has also been compared to drunk driving, with driving for 21 hours being equivalent to 0.15 blood-alcohol concentration, far more than the legal limit. [16] It is particularly a concern for truck and bus drivers and chauffeurs who are either required to or need to drive for long periods of time and during night-time. [3] Although countries have laws preventing their citizens from driving for more than a certain period of time, calculated by experts and generally agreed upon to be no more than 9 hours, with breaks in between. [18] [4]

However, a lack of enforcement, poor awareness and the sheer competition in the transportation business results in driving for more than the legal limit. Along with the duration, the time of the day is also important. Drivers are more likely to fall asleep at night. [24] Another factor at night-time is the glare of headlights from the vehicles moving in the opposite direction. So, while headlights are absolutely essential for you to see and drive safely, they also disrupt drivers coming from the opposite direction. [20] It must also be noted that some drivers don't realise that they're falling asleep, other do but are unable to do anything about it. [22]

As a solution to this, car manufacturers have come up with drowsiness detection features. However, such systems are only found in their top models which are pricier and thus, unaffordable for the general public. Also, a considerable percentage of people buy used (second-hand) cars, which are of an older generation and lack any kind of drowsiness detection feature. So, for the majority of the population, their cars come without any detection system. So, there is a need to build a modular, compact and inexpensive detector that can be fitted easily into any car.

Methods that require attaching equipment to the driver is inconvenient as it might reduce mobility and harm driving experience, especially for longer journeys. Also, technology like infrared cameras [10] [2] is highly specialised, expensive and unnecessary. The human body gives clear signs when feeling drowsy such as yawning and closing eyes for an extended period of time. Based in this, a system was designed to detecting yawning. [1] This can be captured non-intrusively by a small camera placed on the dashboard. This can then be fed to Faster RCNN, one of the most well known object detection neural networks, to analyse facial features like the shape of the mouth. A vector diagram of points attached to different parts of the face is used to calculate the Eye Aspect Ratio (EAR), which decreases as the eyes close. The driver is alerted if his/her eyes remain closed for a certain period of time. This project is implemented in Python, as opposed to previous implementations in MatLab. [8] Python is easy to code in, and is the first choice for machine learning. Our project is not only limited to cars, but can be easily extended and adapted in fields of aviation, heavy machine operation, online classes and any other field where the operator being attentive is important.

II. Literature survey

There is a considerable amount of research work in face monitoring systems that detect drowsiness while you are behind the wheel. For the current project we carried out literature survey to learn different methods that have been employed so far. Most of the systems are based on detection of eye, face, symptoms, face tracking or estimating the state of the driver. Here we will discuss a few important research papers that helped us to carry out this project.

A research work by S Abtahi et al. proposes a method of yawning detection which is based on the geometry of the face of the driver. The paper reports using an algorithm to more than 500 images that have different facial characteristics. The images are from different lighting conditions. The method rules out any false detections due to changes in facial geometry and is hence accurate. [1]

Another method by Hashemzadeh F. et al. reviewed for the current project work involves image processing for drowsiness detection. In this paper the authors have used simple algorithms to detect the changes in face and pupil. The method is implemented with the help of a microcontroller. The method was calibrated for each driver and MATLAB was used to get simulation results.) [8]

Another method proposed by Mohsen Poursadehgiyan studies the blinking of eyes as a signal to detect drowsiness. In this paper by MLP neural system was used for detection. It determines two features, eyes closed in order to get the features of the eyes and second was rate and duration of blinking. The image processing was done based on the Violla- Jones algorithm. [19]

In this model by M. Karchani it was proposed that model be evaluated with objective variables in a driving simulator That would include factors like Standard Deviation of Lane Position (SDLP), Steering Wheel Movement (SWM), and other subjective criteria, such as the driver's self-reported measures of drowsiness on the Karolinska Sleepiness Scale (KSS) and the Observer Rating of Drowsiness (ORD). [11]

O. Khunpisuth et al developed a method to calculate the level of drowsiness in drivers. The head tilting frequency and eye blinking were used as two parameters to determine the level of drowsiness. The authors have used a Raspberry Pi device with a novel algorithm that allows the camera to detect face and eye movement. 9O. [12]

We also studied work by Wang et al where they proposed where they proposed a method known as PATECP (Percentage and Time that Eyelids Cover the Pupils) and PATMIO (Percentage and Time that Mouth Is Open) to detect whether the driver is drowsy by setting a specific threshold. [27]

Similarly, another study by Zhao et al. proposed anew scheme in their paper for recognition of the expressions of drowsiness. The method is based on a combination of facial dynamic fusion information and a deep belief network. (DBN). [28]

An advanced application of this can be seen in the work done by Q Ji et al. where they used visual cues with remotely located camera that had infrared illuminators so that system is more robust in different light conditions. [10]

A method by M. Assari uses an infrared camera to capture videos streams of the driver. This helps in resolving the limitations faced due to presence of beard on face or lighting conditions, wearing glasses etc. Template matching is then used to recognize face by extracting the stream and comparing the facial components with drowsy person. [2]

III. Proposed Method

The flow of data starts from image capturing that happens at the camera. The camera captures frames which are sent to the image processing module. This module consists of 3 levels. The first level identifies the face in the frame that arrives from the image capturing module and sent to the second level. Here facial landmarks are acquired, most importantly the eyes using EAR (Eye Aspect Ratio). The information of the eyes is then sent to the third level that identifies whether the eyes are closed or not and calculates the time for which the eyes have been shut. If that time exceeds 15 seconds, the alerting module provides an audio and visual alert to the user.

The image processing module is the main module of the system. The first level isolates the face of the user from the frames that are received from the camera and sends that data to the second level where features of the face are highlighted using 68 points on the face of the user. Then the eyes are isolated using the Eye Aspect Ratio and this data is further sent to the third level. Here if the eyes are identified to be closed, A timer starts to count the seconds for which the eyes have been closed and send this as “Score” to the next decision box that determines whether or not to give an alert.

First, we’ll set up a camera that monitors a stream for faces.
When a faced is detected, points on it like the outline of the eyes, corners of the mouth are marked and tracked.
Once we get outline of the eyes, we can easily calculate the Eye Aspect Ratio (EAR). The EAR is linked to the state of the eyes. If eyes are open, it is high. As they close, it drops. (Range: 0-1)
Once it drops below a set threshold, we increment the drowsiness score. When this score exceeds a set value, the alarm is sounded to alert the driver. The score is decremented as the driver’s eyes open.

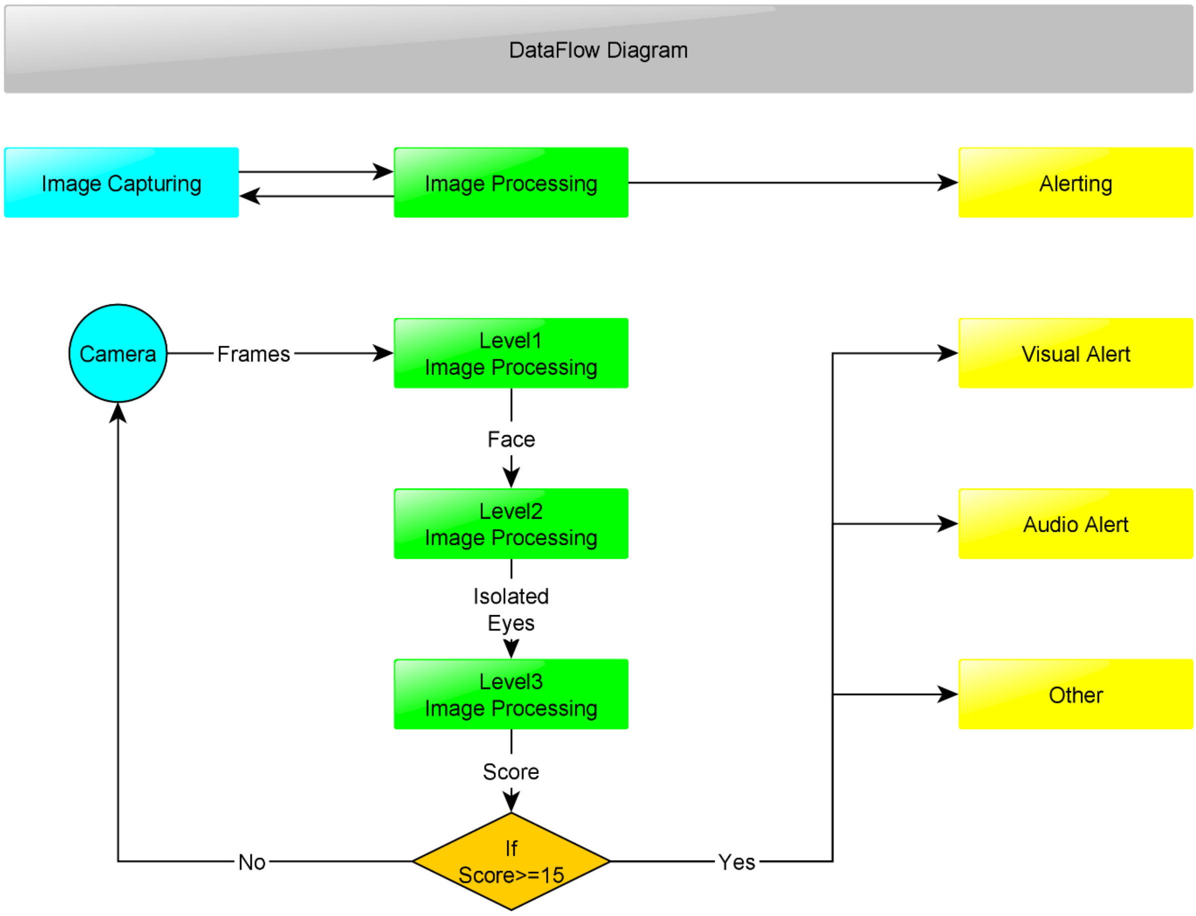

Figure 1

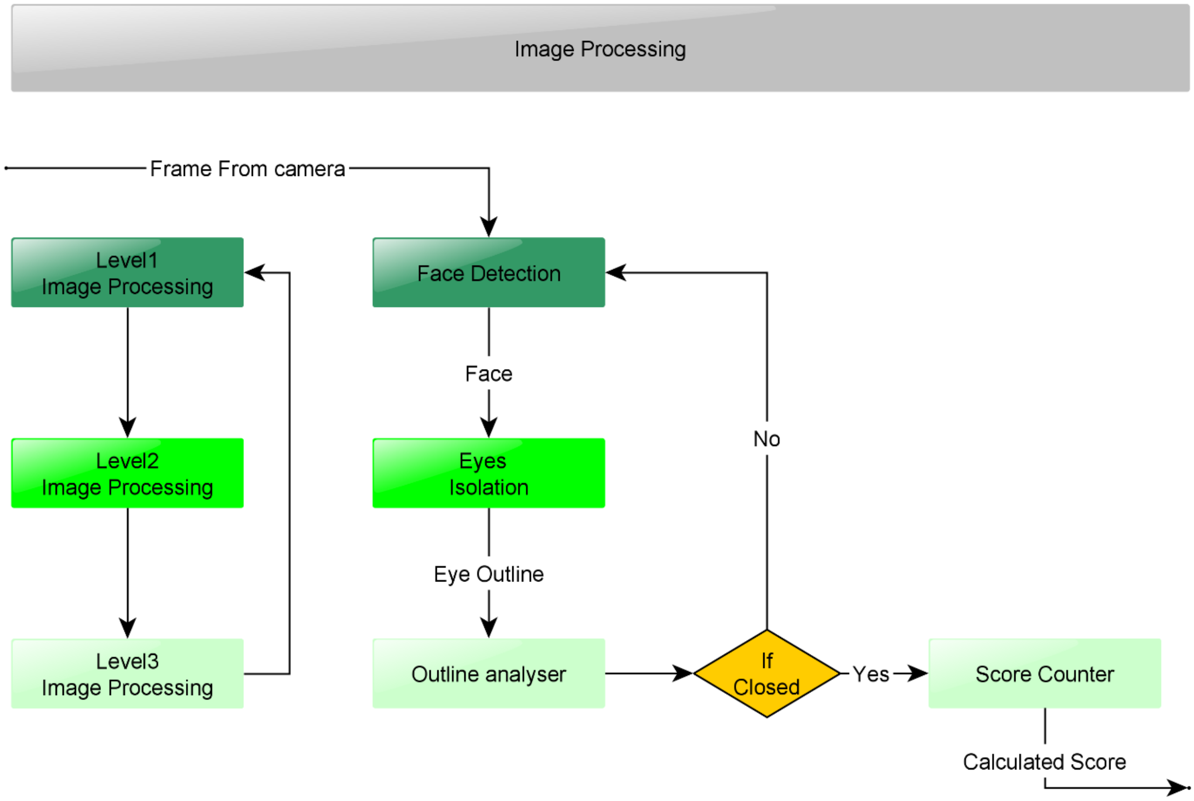

Figure 2

IV. THE FASTER RCNN ALGORITHM

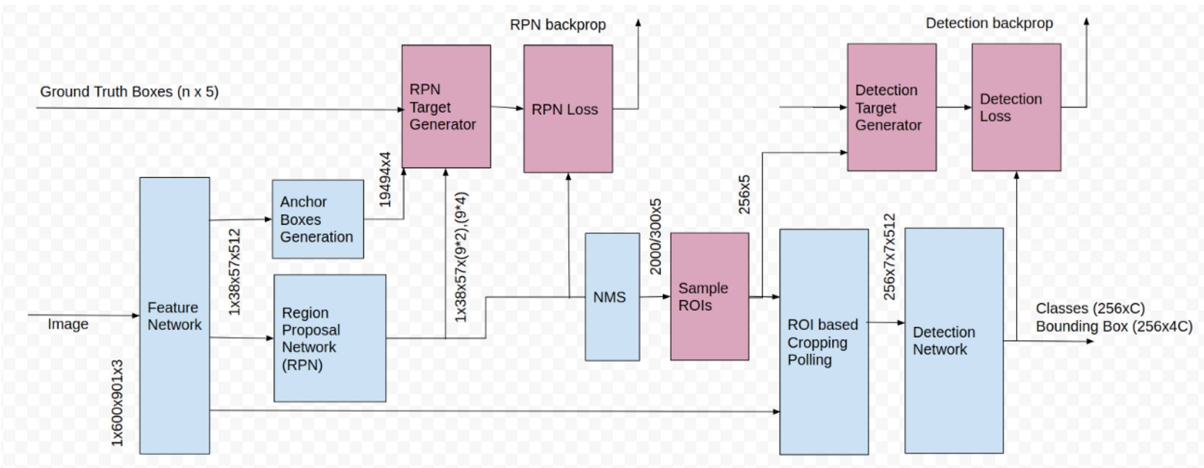

Figure 3

A. *Previous algorithms*

RCNN and Fast RCNN algorithms were introduced in 2014 and 2015 respectively. RCNN, when it was introduced was 30% better than other methods at the time. [6] It stands for "Regions with CNN features". Fast RCNN performs 9 times better on the very deep VGG16 network as compared to RCNN and was 2 times faster at test time. [6]. It was able to do this by evaluating the network and extracting features for the whole image once, instead of extracting features from each Region of Interest (RoI), which are cropped and rescaled every time in R-CNN.

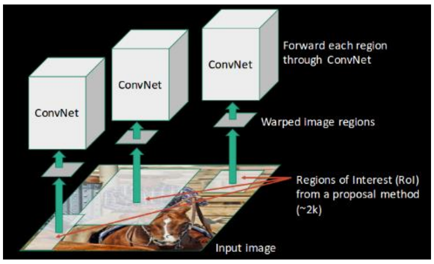

Figure 4

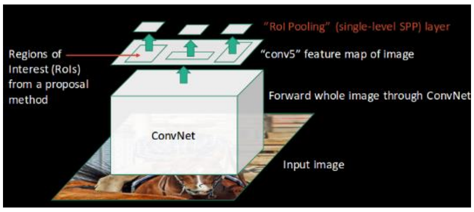

Figure 5

However, Fast RCNN continued using selective search for object detection. Selective search is slow and time consuming. The algorithms so far, have treated the object detection part of the algorithm separate from the region proposals. Thus, they relied on a classifier to interpret the results. This was the bottleneck in the process.

B. Improvements

Faster RCNN applies a unified approach for both the tasks in hand — which meant that both the region proposal and classifier shared the same convolutional features. For this, it makes use of Region Proposal Network (RPN) instead of selective search. This resulted in it being over 10 times faster than Fast RCNN.

The RPN outputs the objective-ness score which indicates whether the piece of image in question (anchor) contains a background or a foreground object, i.e. a binary classification task which outputs 'object' or 'no object'. This is how the RPN functions:

First, RPN receives feature maps from FRCNN. It uses a sliding window over these feature maps, and at each window, it generates *k* Anchor boxes of different shapes and sizes:

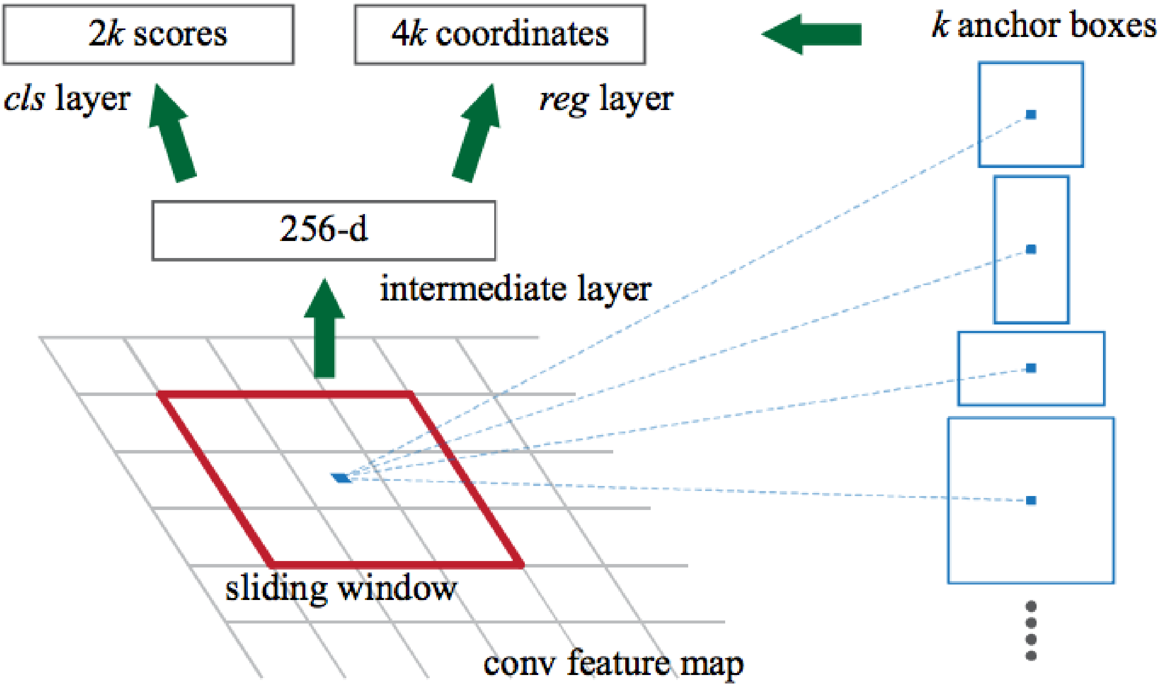

Figure 6

Anchor boxes are fixed sized boundary boxes that are placed throughout the image and have different shapes and sizes. For each anchor, RPN predicts two things:

The first is the probability that an anchor is an object (it does not consider which class the object belongs to)

Second is the bounding box regressor for adjusting the anchors to better fit the object.

V. Code Implementation

The entire project will be coded using Python due to its highly versatile and dynamic nature. For the implementation we will be using several Python libraries such as the SciPy package. It is a stack that is used to calculate the Euclidean distance. It is the distance between facial landmarks which inturn is useful in the calculation of aspect ratio. The OpenCV package, which is our primary image processing library. The imutils package, which provides a series of computer vision and image processing functions to make working with OpenCV easier. A class named Thread is thus used for playing a sound wave file and in our case we are going to use it for playing the alarm. The dlib library, that contains pretrained face recognition models that speed up and reduce cost of development. Important functions:

dlib.get_frontal_face_detector(): This is an inbuilt function that returns an array of values, each value corresponding to a certain property of the driver's face. By slicing this array, we can get the values pertaining to the eyes specifically.

eye_aspect_ration(): The sliced array is passed to this function which calculates the aspect ratio of the eye. A closed eye has a smaller aspect ratio than an open one and hence, we can define a threshold value of 0.3. If the ratio is below 0.3 for 48 consecutive image frames, the alarm is sounded.

dlib.shape_predictor(): This object is a tool that takes in an image region containing some object and outputs a set of point locations that define the pose of the object. The best example would be the process of facial landmarking, where we take an image of a face and we landmark facial features like corners of eyes, mouth and tip of the nose and jawline.

cv2.convexhull(): Finds the convex hull of a point set. The function is used to establish a convex hull of a 2D set of points using an algorithm called Sklansky's algorithm which has time complexity of O(N logN) in our current implementation.

Figure 7

Figure 8

cv2.putText(): The function cv::putText renders the specified text string in the image. There are certain symbols that do not have the capacity or the range to be used in the process so they are not rendered. As a mitigation practice, they are replaced by question marks to complete the process.

cv2.drawContours(): This is a double check function which ensures that the boundaries and landmarks we have got are accurate. Especially the regions of eyes and mouth and also the function debugs our script and checks the detection and localisation of facial features.

A. Image Capturing

The most important part of drowsiness detection is image capturing where we have many alternatives like amazon Rekognition, Google Cloud Vision API, Microsoft Computer Vision API etc. But we have chosen OpenCV because it is easy to implement and other architectures have various variables that have no use in drowsiness detection.

Steps for finding motion-detection using OpenCV:

i. Capture the video from the camera

```
CvCapture* capture = cvCaptureFromCAM(CV_CAP_ANY);
```

ii. Capture the frame 1

```
IplImage* frame1 = cvQueryFrame(capture);
```

iii. Capture the frame 2

```
blur (frame1, frame1_blur, Size(4, 4));
blur (frame2, frame2_blur, Size(4, 4));
```

iv. Find the absolute difference

absdiff (frame2_blur, frame1_blur, result_frame);

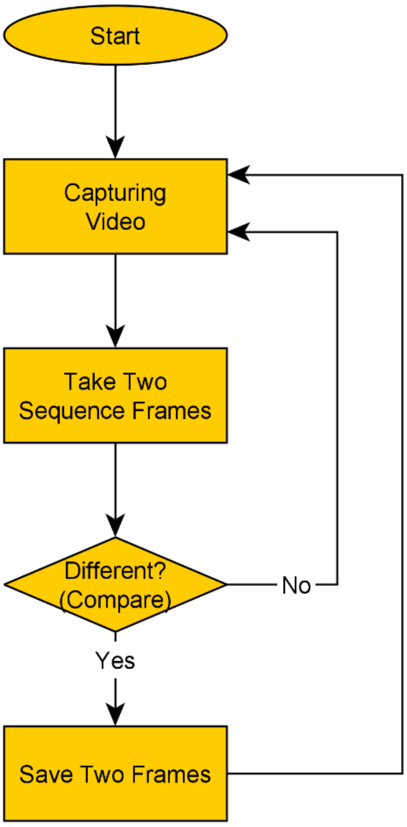

Figure 9

Flow chart illustrating motion detection

B. *Image Processing*

1) *Facial Landmarking*

Next is rect_to_bb: This is primarily utilized function and it takes one argument. From that function it is taken to be the bounding box rectangle and that is an outcome of detector called dlib which is a face detector. Facial graphical points are the features of the aforementioned argument.

As in OpenCV the coordinates are a 4-character list, having the points as well as height and width as a factor, as a matter of easement the previously mentioned functions process the inputs and transform it into desirable form.

A shape or object containing 68(x,y)- points of the markable frontal face and its features regions are a product of the detector.

With the help of a function, the tuple is converted to a numpy array which incorporates in a better way with the rest of the operation.

With the help of the functions now we can follow through with the facial landmarking in the images. Now the detections are analysed one by one with the use of loops.

A detailed tuple with 68 graphical points is generated through the processes of facial landmarking. This is done for every detection and the generated data is transformed into a more processable array with 68x2 shape.

Then, we sketch boundary boxes that enclose the detected face from the image. After that comes the indexing.

Finally, the code goes over every detection and draws the features individually. The output image is ready to display.

2) *Authentication System*

a) *Face Detection*

Face Detection it is the place where an input image is probed to determine a face, followed by image processing that involves cropping and extracting of the face. Haar-Cascade classifier is a module used for facial detection which is a part of the Open CV module. OpenCV is a highly used and functional module for computer vision. When given an input that might a photo or video or maybe even a live feed, the step-by-step binary determination whether a face is present or not takes place by the help of functions and examination and location involved. An Xml file thus is made use of for classifying the location of each image. This xml file is stored in harrcascades file in the data directory which in turn is in sources file.

b) *Face Recognition*

This might sound like detection but is a totally different sphere where the taken picture after detection and other processes is tried to be matched with any picture in their pre-set database. OpenCV has this as the main feature and the accuracy is in the mid-90s for clear images and slowly dies down as the clarity decreases. It is slightly difficult to detect a face if a person wearing glasses or an image is blurring. But we have used the latest version of CNN called Faster R-CNN for facial detection and landmarking.

C. *Alerting*

For the alerting mechanism the determinant factor is the eye A.R.(Aspect Ratio). It is calculated by dividing the “blink/closed” ratio and is then compared to the predetermined eye threshold. If the calculated ratio is below the threshold then we consider that the eyes are collapsed. We also maintain a counter which is incremented for each consecutive frame where the eyes are collapsed. There is pre-set limit for the counter boundaries and if the live counter in the code exceeds the boundary limit then it is assumed that the driver starts to feel sleepy. As a checker there is a feature that sets the alarm off if the alarm is physically turned off. Alarm sound is played if a path to the file is passed when the code is being run. Another important feature is where the driver doesn’t have the authority to switch the alarm off. Also, there exists an independent thread for the alarm system so the main program doesn’t interfere with the duration of the alarm. Finally, a projection of alert message takes place outside of the frame for the following reason. They are used for debugging when library named ‘playsound’ is temporarily not in use.

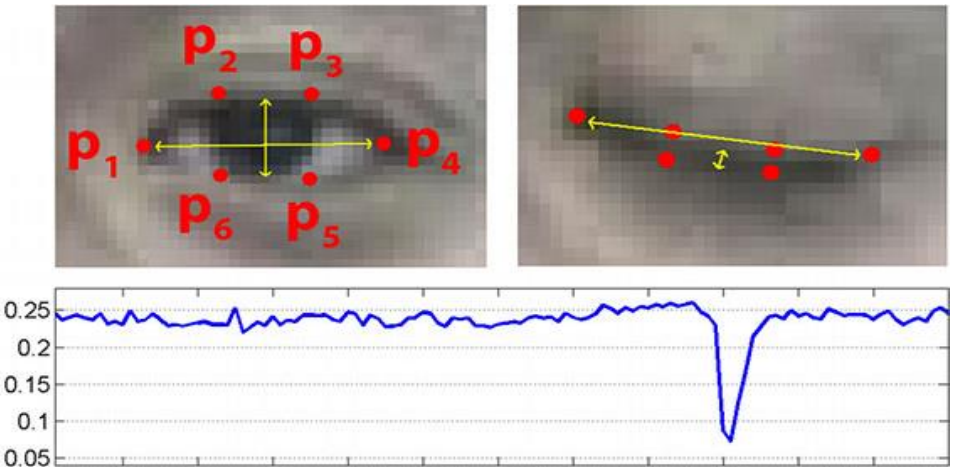

Figure 10

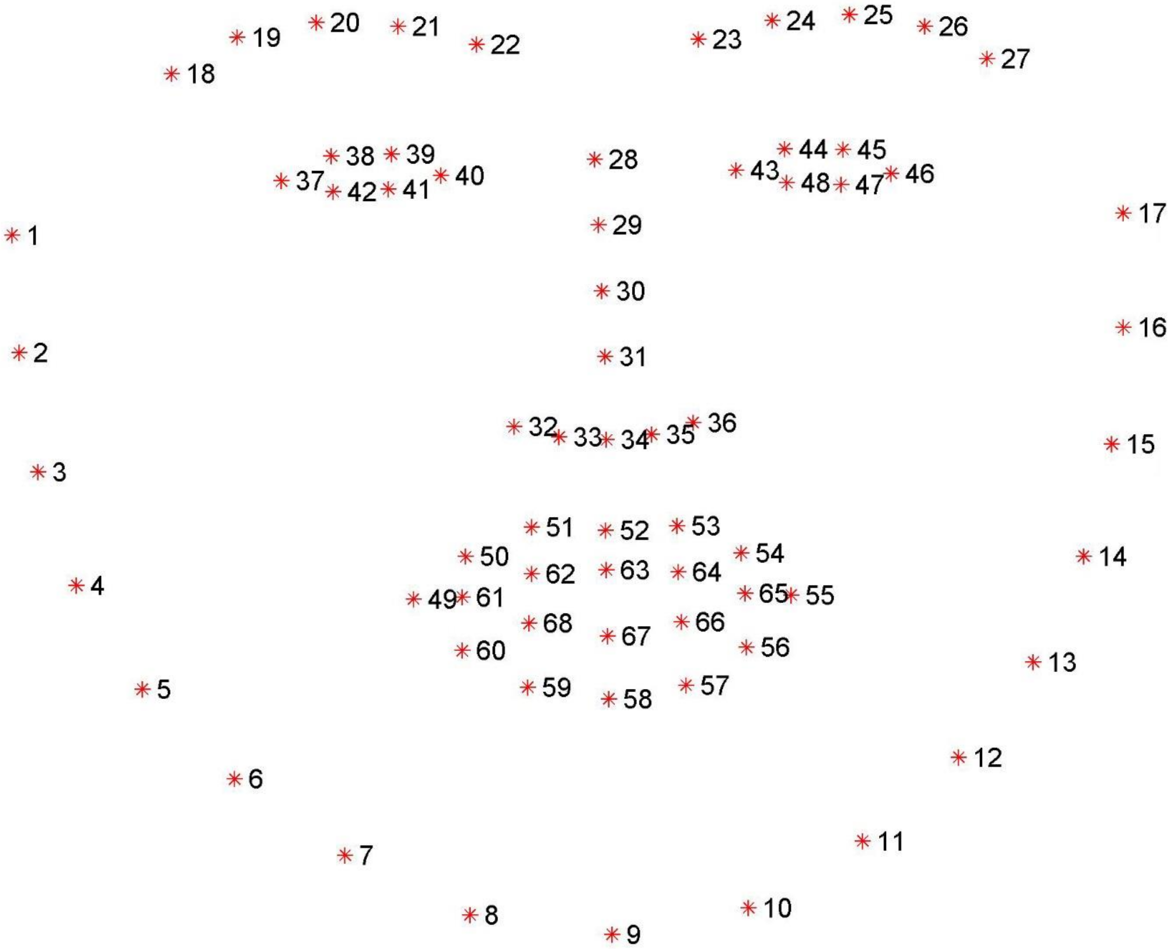

Figure 11

VI. Requirements

A. Hardware

- 1 GB RAM
- < 100 MB Storage
- Intel i5 processor
- Camera module / Webcam

B. Software

- Windows / Linux / Mac OS

- Python 3.x and its libraries:
- NumPy
- SciPy
- Keras
- MatPlotLib
- PyGame
- Tensorflow
- OpenCV

VII. RESULTS

Light (bright-light) | Dark (low-light)

Eyes OPEN. NO alert.

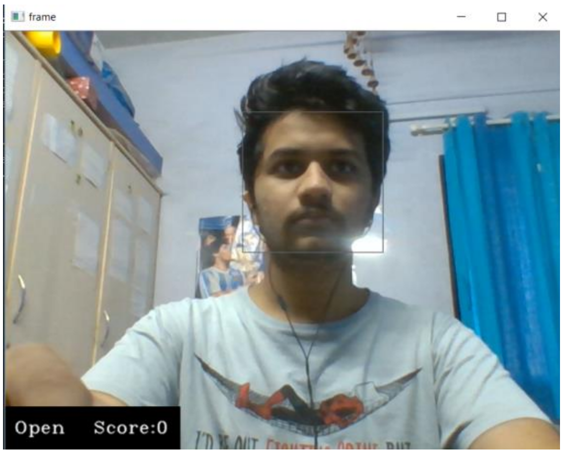

Figure 12

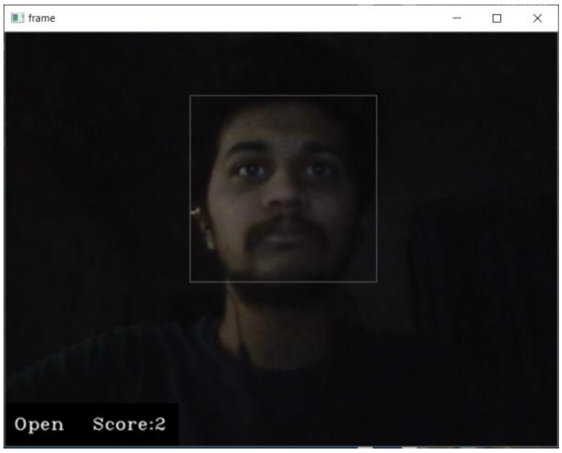

Figure 13

Eyes CLOSED, ALERT!

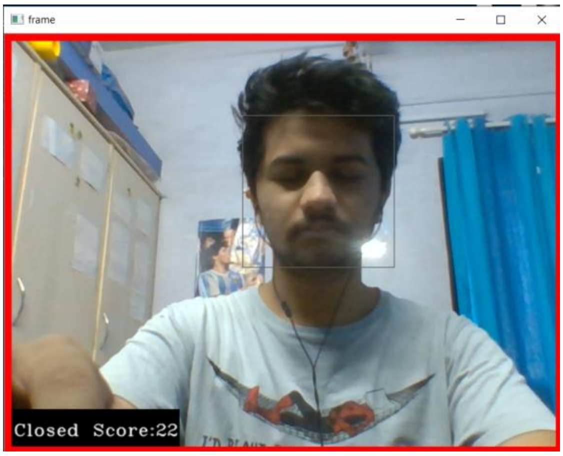

Figure 14

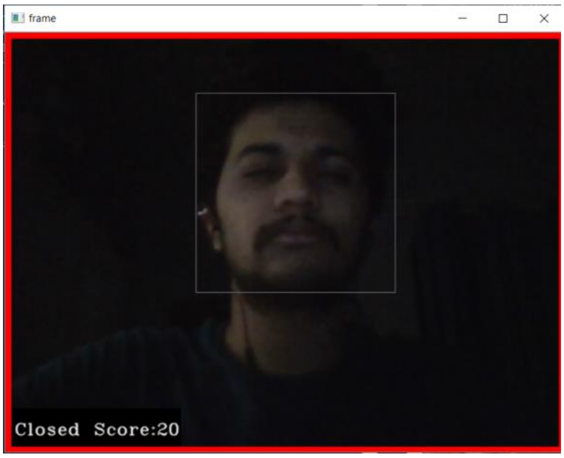

Figure 15

Table 1

VIII. CONCLUSION AND FUTURE SCOPE

The absence of an in-built drowsiness detection system in most vehicles is evident. Therefore, a need for modular, easy to install and inexpensive detection system arises. An application written in Python, taking advantage of the many machine-learning libraries available, that processes a live input from a webcam with the highly accurate Faster RCNN algorithm is a suitable choice. Such an application can be easily ported to a Raspberry Pi to create a compact system.

IX. ACKNOWLEDGEMENT

We express our deep and sincere gratitude to our professor, Dr. Sindhia Lingaswamy (sindhia.lingaswamy@vit.ac.in) for her invaluable guidance throughout this project. Her expertise, comments and inputs have helped us immensely throughout the project.

X. REFERENCES

[1] Abtahi, S., Hariri, B., & Shirmohammadi, S. (2011). Driver Drowsiness Monitoring based on Yawning Detection. *2011 IEEE International Instrumentation and Measurement Technology Conference*, (pp. 1-4). doi:10.1109/IMTC.2011.5944101

[2] Assari, M. A., & Rahmati, M. (2011, November). Driver drowsiness detection using face expression recognition. *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, (pp. 337-341). doi:10.1109/ICSIPA.2011.6144162

[3] CDC. (2020, May 28). *Drowsy Driving: Asleep at the Wheel*. Retrieved from Centers for Disease Control and Prevention: https://www.cdc.gov/sleep/features/drowsy-driving.html

[4] EU Legislature. (2006). *Driving time and rest periods*. Retrieved from European Commission: https://ec.europa.eu/transport/modes/road/social_provisions/driving_time_en

[5] Gao, H. (2017, September 27). *Faster R-CNN Explained*. Retrieved from Medium: https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8

[6] Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1440-1448). IEEE. doi:10.1109/ICCV.2015.169

[7] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580-587). IEEE. doi:10.1109/CVPR.2014.81

[8]Hashemzadeh, F., Ostadi, M., & Mohammadi-ivatloo, B. (2017, March 1). A Fast and Simple Drowsiness Detection System Based on ARM Microcontrollers. *Intelligent Industrial Systems, 3*(1), 23-28. doi:10.1007/s40903-017-0069-x

[9] Jabbar, R., Shinoy, M., Kharbeche, M., Al-Khalifa, K., Krichen, M., & Barkaoui, K. (2020). Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, (pp. 237-242). Doha, Qatar. doi:10.1109/ICIoT48696.2020.9089484

[10] Ji, Q., Zhu, Z., & Lan, P. (2004, July). Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Transactions on Vehicular Technology, 53*(4), 1052-1068. doi:10.1109/TVT.2004.830974

[11] Karchani, M., Mazloumi, A., Saraji, G. N., Gharagozlou, F., Nahvi, A., Haghighi, K. S., . . . Foroshani, A. R. (2015, April 1). Presenting a model for dynamic facial expression changes in detecting drivers' drowsiness. *Electronic Physician, 7*(2), 1073-1077. doi:10.14661/2015.1073-1077

[12] Khunpisuth, O., Chotchinasri, T., Koschakosai, V., & Hnoohom, N. (2016). Driver Drowsiness Detection Using Eye-Closeness Detection. *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, (pp. 661-668). doi:10.1109/SITIS.2016.110

[13] Mahamkali, N., & Vadivel, A. (2015). OpenCV for Computer Vision Applications. *National Conference on Big Data and Cloud Computing (NCBDC'15).* Trichy. Retrieved from https://www.researchgate.net/publication/301590571_OpenCV_for_Computer_Vision_Applications

[14] NCRB. (2018). *Causes-wise Number of Accidents, Persons Injured and Died due to Other Causes during 2018 (All India).* Retrieved from National Crime Records Bureau: https://ncrb.gov.in/sites/default/files/table-1.6.pdf

[15] NCRB. (2018). *Cause-wise Distribution of Road Accidents and Unmanned Railway Crossing Accidents during 2018 (All India).* Retrieved from National Crime Records Bureau: https://ncrb.gov.in/sites/default/files/table-1A.8.pdf

[16] NRSPP. (2016). *Drowsiness And Fatigue Risk*. Retrieved from National Road Safety Partnership Program: https://www.nrspp.org.au/resources/nrspp-quick-fact-drowsiness-fatigue/

[17] NSC. (2020). *Drivers are Falling Asleep Behind the Wheel*. Retrieved from National Safety Council: https://www.nsc.org/road-safety/safety-topics/fatigued-driving

[18] Parliament of India. (1961). *THE MOTOR TRANSPORT WORKERS ACT, 1961.* Retrieved from India Code: https://www.indiacode.nic.in/bitstream/123456789/1662/1/a1961-27.pdf#search=Motor%20Transport%20Workers%20Act,%201961

[19] Poursadeghiyan, M., Mazloumi, A., SARAJI, G., Niknezhad, A., AKBARZADEH, A., & Ebrahimi, M. (2017, January 25). Determination the Levels of Subjective and Observer Rating of Drowsiness and Their Associations with Facial Dynamic Changes. *Iranian Journal of Public Health, 46*, 93-102. Retrieved from https://ijph.tums.ac.ir/index.php/ijph/article/view/8922

[20] Pulling, N. H., Wolf, E., Sturgis, S. P., Vaillancourt, D. R., & Dolliver, J. J. (1980, February 1). Headlight Glare Resistance and Driver Age. *Human Factors, 22*(1), 103-112. doi:10.1177/001872088002200111

[21] Ren, S., He, K., Girshick, R., & Sun, J. (2015, June 4). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137-1149. doi:10.1109/TPAMI.2016.2577031

[22] Reyner, L. A., & Horne, J. A. (1998, April). Falling asleep whilst driving: are drivers aware of prior sleepiness? *International Journal of Legal Medicine, 111*(3), 120-123. doi:10.1007/s004140050131

[23] Sharma, P. (2018, October 11). *A Step-by-Step Introduction to the Basic Object Detection Algorithms (Part 1)*. Retrieved from Analytics Vidya: https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

[24] Stutts, J. C., Wilkins, J. W., Osberg, J. S., & Vaughn, B. V. (2003). Driver risk factors for sleep-related crashes. *Accident Analysis & Prevention, 35*(3), 321-331. doi:10.1016/S0001-4575(02)00007-6

[25] Tefft, B. C. (2016). *Acute Sleep Deprivation and Risk of Motor Vehicle Crash Involvement.* Washington DC: AAA Foundation for Traffic Safety. Retrieved from https://aaafoundation.org/wp-content/uploads/2017/12/AcuteSleepDeprivationCrashRisk.pdf

[26] Uijlings, J. R., van de Sande, K. E., Gevers, T., & Smeulder. (2013, September 1). Selective Search for Object Recognition. *International Journal of Computer Vision, 104*(2), 154-171. doi:10.1007/s11263-013-0620-5

[27] Wang, P., & Shen, L. (2012). A method of detecting driver drowsiness state based on multi-features of face. *2012 5th International Congress on Image and Signal Processing*, (pp. 1171-1175). doi:10.1109/CISP.2012.6469987

[28] Zhao, L., Wang, Z., Wang, X., & Liu, Q. (2018, MArch). Driver drowsiness detection using facial dynamic fusion information and a DBN. *IET Intelligent Transport Systems, 12*(2), 127-133. doi:10.1049/iet-its.2017.0183

[29] Zhong, G., Ying, R., Wang, H., Siddiqui, A., & Choudhary, G. (2019, December 14). *Drowsiness Detection with Machine Learning*. Retrieved from Towards Data Science: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a

Authors Profile

All the authors are 2nd year students at Vellore Institute of Technology, Chennai enrolled for B. Tech CSE with specialization in Artificial Intelligence and Machine Learning.