ISSN No: 2456-2156

# Real Time Productivity Analyser

Himanshu Goyal [1] , Kanika Bagri [2] , Eakansh Gupta [3] , Vishal Kumar Singh [4] , Nishi Sharma [5]
[1,2,3,4]Dept. of Computer Science, Dr. Akhilesh Das Gupta Institute of Technology and Management
[5] Assisstant Professor, Dept. of Computer Science, Dr. Akhilesh Das Gupta Institute of Technology and Management
GGSIPU, Delhi, India

**Abstract:- The purpose of this Python project is to develop a real time productivity analyser system that will enhance the productivity in various industries by minimizing the unaffordable productivity losses as much as possible. The system will work by detecting whether a person's eyes are closed for a sufficient amount of time and whether he is yawning or not. This device will alert him when fatigue is detected. Fatigue poses a major risk of life-threatening injuries. We used OpenCV to gather images from a camera and feed them to our CNN binary Classifier that will determine whether the person's eyes are Open' or 'Closed' along with the yawning score. The classifier we developed is using Convolutional Neural Networks (CNN). A convolutional neural network is a sort of deep neural network that is particularly effective for image categorisation.[8]**

**This system has been designed in order to reduce accidents in most of the industries be it construction sites, roads, etc. as most of the accidents happen because of fatigue. Hence this a system will alert the person before it gets too late. The system is capable of calculating the accurate position of the eye. [5]**

## I. INTRODUCTION

Analysis of productivity is a technology capable of avoiding unbearable productivity losses that can affect the name and fame of an organization because of exertion. Also, road accidents and poor workplace results are often caused by exhaustion. More and more occupations need long-term focus nowadays. Drivers must keep an open eye on the lane, so they can quickly respond to unexpected events. In many traffic accidents, driver exhaustion also becomes a direct cause. There is also a need to improve systems to identify and warn a driver of a bad neuropsychological situation, which could dramatically reduce the number of car accidents associated with exhaustion. [1]

The aim of this Python project is to develop a productivity analyser system that will detect that whether the eyes of an individual is closed and if he is yawning or not. This device will warn you when it senses fatigue. Exhaustion presents a serious risk of causing injuries that are potentially fatal.

Works previously attempted to address the issues of productivity losses by detecting fatigue was done using feature extraction-based techniques like Har cascade classifiers. This technique provides faster detection, but the problem is that they are more sensitive to illumination conditions as well as imaging resolution. This problem can be solved using machine learning algorithms that are well-trained as they provide high precision and robustness in fatigue recognition. A simple CNN-based productivity analyser has been developed with an average accuracy of 83.74% across multiple subjects using their customized training data sets.

## II. PRODUCTIVITY MONITORING SYSTEMS THAT ARE COMMONLY USED

The system is one instance of analysing the productivity by detecting the exhaustion of a person. The system also works well by reducing the accidents happening on road because of fatigue of the driver. Enforced in Ford cars into the Driver Assistant. This analyses fast steering Movements, walking, erratic and sudden braking or acceleration on lines dividing lanes. The system collects and analyses this data, assigns one of the 5-degree concentration levels to the driver (5- the driver is centred, drives correctly, 1- the driver is very tired, should stop driving and rest immediately) [2].

PERCLOS is a somnolence detection technology that measures the percentage of closed eyes over the pupil over time, rather than blinks, and reflects these closures or drops. [3] Different real-time operator somnolence detection systems use PERCLOS evaluation and specially designed algorithms to determine the onset of fatigue. Each techno uses a unique set-up and combination of hardware.

Electroencephalography is a tool that non-invasively reports electrical brain activity. It was discovered in 1924 by Hans Berger and developed to the advanced technology of today over more than 90 years [4]. A drastic decrease in EEG instrumentation scale, weight and cost, and the ability to interact wirelessly with other digital systems paved the way for the technology to be applied to previously unsuspected areas, such as entertainment, bio-feedback, and learning and memory support.

To identify and monitor the operator's facial features, the computer vision system uses an unobtrusive dashboard-mounted camera and two infra-red illumination sources. Eye closes and head poses are analysed by the machine to assess the early onset of exhaustion and diversion.

AVECLOS is determined by the fatigue sensing algorithm. This is the percentage of time over one minute that the eyes are completely closed.[5]

The fatigue detection system software has recently been altered to run on Android cell phones. The technology uses a cell phone camera mounted on a stand on the cab dashboard to control the movement of the operator's eyes. The system's developers opted to use the technique of eyelid motion.[6] The robust system is capable of monitoring fast head movements and facial expressions. External illumination, which reduces operator interference, is minimal.

## III. PROPOSED SOLUTION

### 3.1 Network Architecture

In our proposed exhaustion detection system, we created a model that had three double convolutional layers, followed by max-pooling layers, and ended with three fully linked hidden layers and a Softamx layer.

Proposed exhaustion detection network architecture

1) Convolutional Layer: It is the first layer consisting of multiple learning filters which produces an activation map. We are converting RGB pictures to greyscale in order to emphasise face characteristics rather than skin tone.. We use double convolutional layers because this has experimentally produced a balance between accuracy and computations.
2) Max-Pooling Layer: It is a non-linear down-sampling method to reduce the spatial size of the representation so that number of parameters could be decreased, which reduces the computation time as well as control over-fitting.
3) Dropout: This layer can be used as an image noise reduction technique. When it is used with the fully connected layers, it deactivates a part of the neurons in order to improve the generalization by allowing the layer to learn by itself with different neurons.
4) Fully-Connected Layer: They are dense hidden layers that take input volume from the last max-pooling layer's output to generate an N-dimensional vector, where N is the number of classes. In our case, the model has to choose from two classes (exhausted and controlled) for classification.

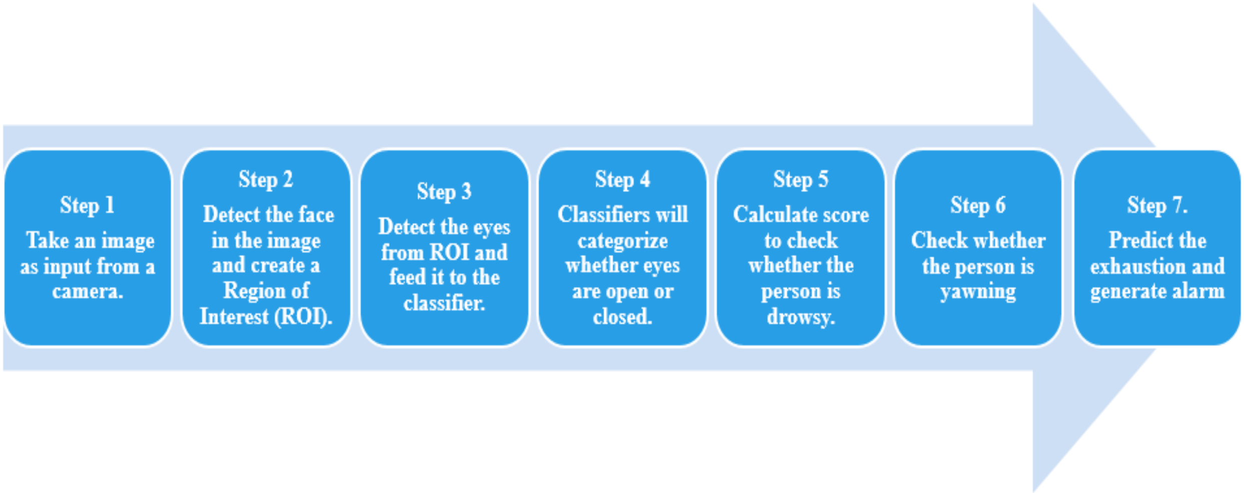


Fig 1: Steps involve

**Step 1 – Face Detection**

Taking input from our camera is the first sub step in this. So, an infinite loop has been made to access the webcam that will catch each frame. Then we have used a learning-based technique for face detection on the basis of Har-like features of Viola and Jones and cascade classifiers.

**Step 2 – Detect Face in the Image and Create a Region of Interest (ROI)**

An integral image is used to reduce the initial image processing required for face detection and compute the rectangle features correctly and efficiently. The value of integral image at point (x, y) can be calculated in one pass over the original image which will be equal to the sum of all pixels above and to the left of it. This will return an array of detections with the x, y, and height coordinates of the object's border box width. We can now go through the faces and draw boundary boxes for each one [8].

**Step 3 – Detect the eyes from ROI and feed it to the classifier**

To detect the eyes, the same technique is used to detect faces. First, in leye and reye, we set our CNN binary classifier for eyes. We only need to extract the eye info from the entire image now. This may be accomplished by deleting the eye's border box, and then using this code, we can extract the eye image from the frame. l eye solely includes the eye's picture data [9]. This data will be sent into our CNN classifier, which will predict whether or not the eyes are open. Similarly, we will be extracting the right eye into r_eye[7][8].

**Step 4 – Classifier will Categorize whether Eyes are Open or Closed**

To predict eye status, we use our CNN Binary classifier. We must execute such procedures in order to feed our image into the model, as the model requires accurate measurements to begin with. With our model, we are now predicting each eye lpred = model.predict(l_eye) classes [8].

ISSN No: 2456-2156

If the value of lpred[0] = 1 means that the eyes are open, then if the value of lpred[0] = 0, the eyes are closed.

**Step 5 – Start Calculating Score after mouth localization to Check whether Person is Exhausted**

The score is simply a value that we are going to use to assess how long the individual has closed his eyes. So, if both eyes are closed, we will continue to raise the score, and we will decrease the score when the eyes are open. Using the cv2.putText() feature, which will show the person's real-time status, we draw the result on the screen. For example, if the score is greater than 15, a threshold is established, which means that the eyes of the individual are closed for a long period of time. This is when we use sound.play() to beep the alarm [8].

**Step 6 – Detecting Mouth Localization**

(i)Detect facial edge using gradient edge detector .

(ii)Compute vertical projection on the half lower face edge by Vproj(j) = ∑i grad(i,j), where grad symbolises the image gradient and represent the ith row and jth column, respectively. This phase seeks to identify the borders of the right and left mouth regions.

(iii)Compute horizontal projection to resulting gradient region according to Hproj(j) = ∑i grad(i,j), to obtain the upper and lower limits of the mouth and then the mouth localized region

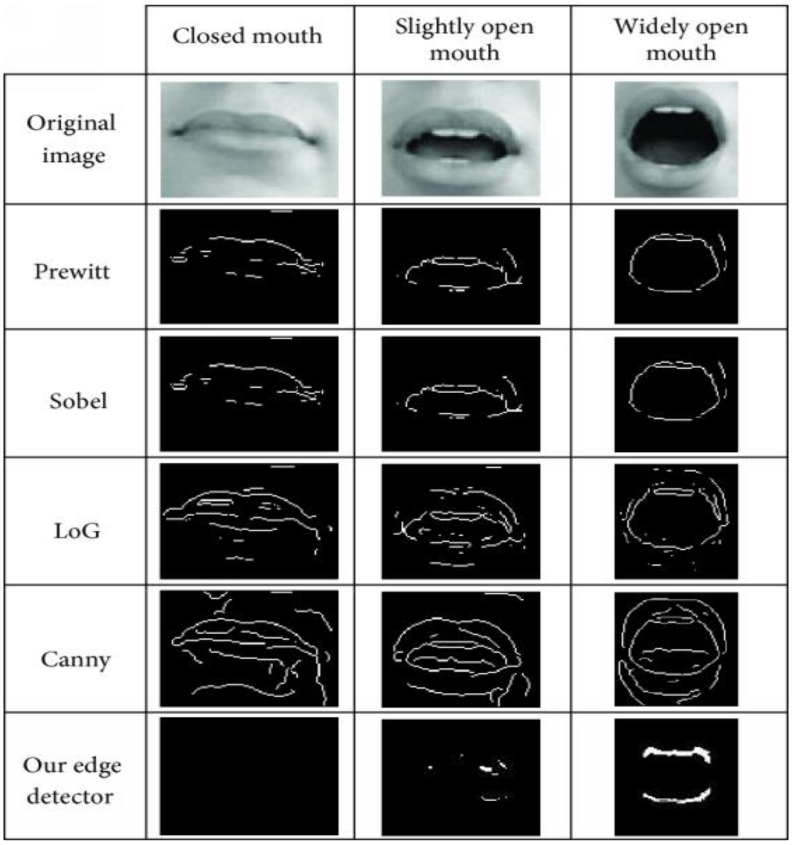


Fig 2: Detection of Mouth

**3.3. Proposed architecture of application**

This section describes and illustrates the architecture of the implemented solution in Fig.2. [9] The user's face is captured by the camera. It will run the ML model after shooting the photo and decide if the user is weary or not depending on the score generated.

The model recognises and extracts facial landmarks from the image, then passes the data to the trained model, which analyses the user's state or level of weariness. Finally, if the result reveals that the user is exhausted, the application will alert the user. Finally, if the result reveals that the user is exhausted, the programme will notify the user.

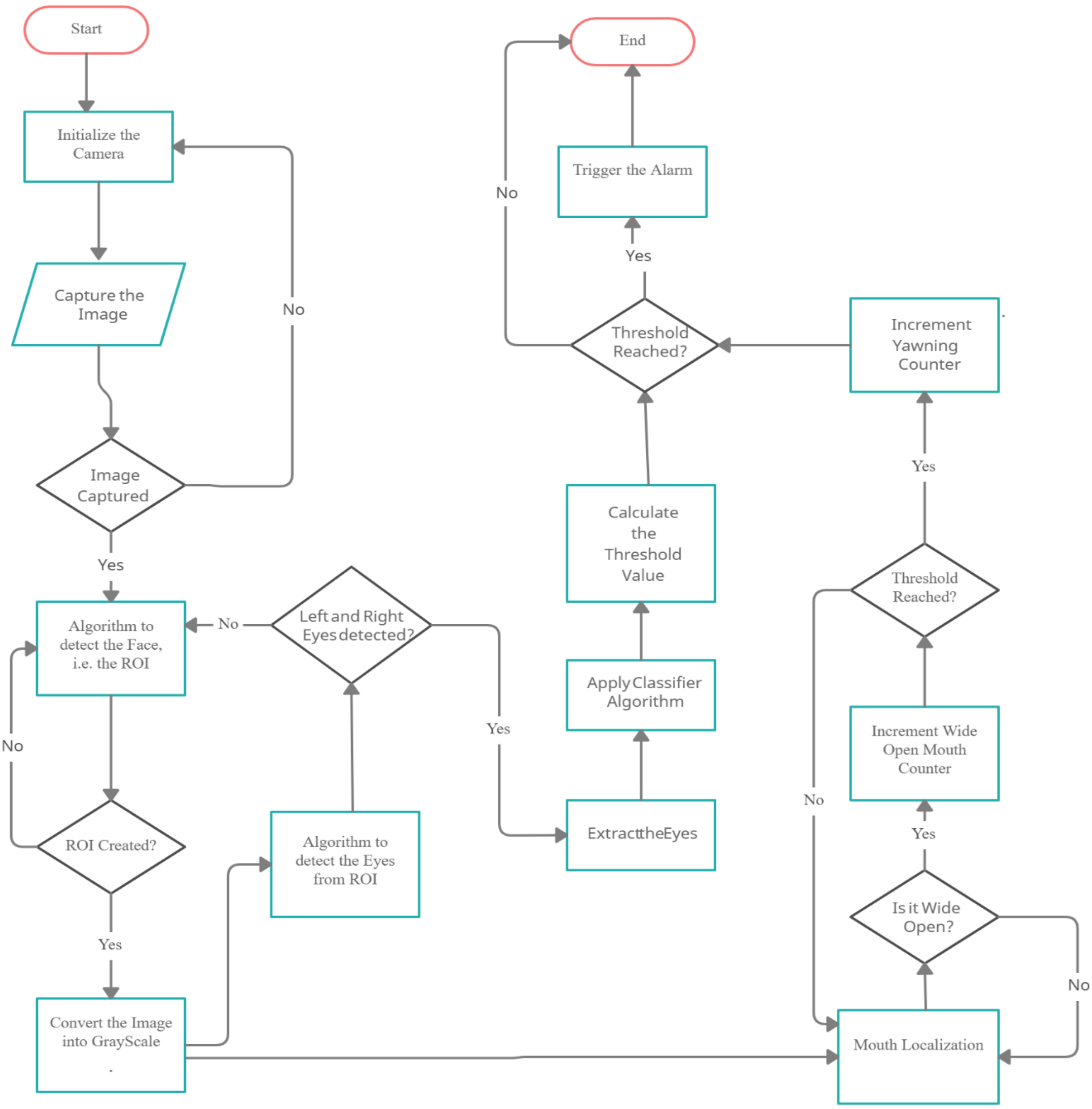


Fig. 3. Flowchart of the application

## IV. RESULT AND ANALYSIS

Cross validation accuracy was reported to be 93.8 percent during training. We also examined the performance of our suggested system on a variety of subjects and found that it has an average accuracy of 95%. We achieved testing accuracy by putting the model to the test on subjects that were not part of our training data. If the testing and training accuracy curves begin to diverge in a regular manner, we must halt the training. Also, we have faced the problem of overfitting when the model is over-learning the training data set. We can observe from the graph that 15 epochs in our model are sufficient for accurate results.

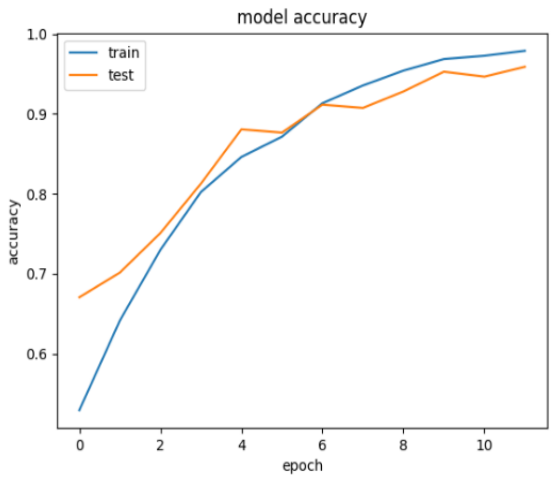


Fig. 4. Accuracy with our CNN Classifier

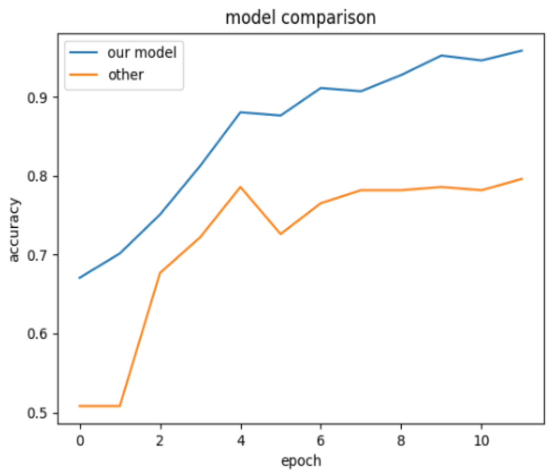


Fig. 5. Accuracy with Har-cascade Classifier

### 4.1. Real Time Video Acquisition from Camera

We have used beep sound for alerting whenever our model detects some exhaustion it will instantly produce the beep sound.

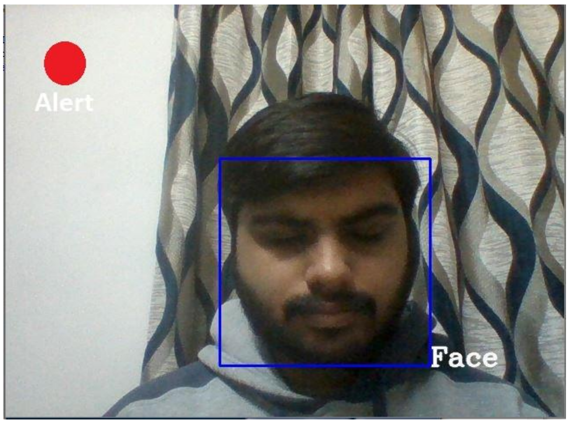


Fig. 6. When eyes are closed

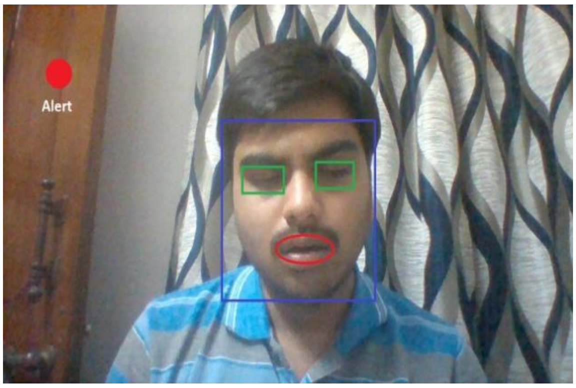


Fig. 7. Detecting Closed Eyes and Yawn

| S. No | Category | Accuracy |
|---|---|---|
| 1 | Good light with glasses | 88.241 |
| 2 | Low light without glasses | 81.35 |
| 3 | Low light with glasses | 77.262 |
| 4 | Good light without glasses | 88.137 |
| | **Average** | **83.747** |

Table 1. Accuracy w.r.t. different condition

## V. CONCLUSION

This system has been designed in order to reduce unaffordable productivity losses happening because of the exhaustion of people at workplaces be it construction sites, offices, cockpits, etc. and also to reduce accidents as in most of the industries accidents happen because of fatigue. Hence this a system will alert the person before it gets too late.

A system for continuously measuring a person's tiredness under various conditions has been described. In addition, using multilayer perceptron classifiers, this research develops a method for detecting exertion. The system's purpose is to recognise the face of the person sitting in front of the camera and feed the information to a trained model for determining the person's condition.

According to the experimental results, the size of the model used is small. Around the same time that the accuracy rate is 82.782%. It can also be incorporated into advanced productivity enhancer systems and smartphone applications. However, improvement can still be enhanced in results. Further work will be undertaken to detect the distraction and the yawn of the person.

## REFERENCES

[1]. Damian Sałapatek, Proceedings of the Institute of Vehicles 3(112)/2017

[2]. PawełCzapski, Ph.D., Eng.; Institute of Aviation, Center of Space Technologies

[3]. Federal Highway Administration (1998). "PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance" (PDF). Washington, DC: US Department of Transportation.

[4]. www.smartcaptech.com. EdanSafe. 2015. Retrieved April 30, 2017.

[5]. Q.; Lan, P.; Zhu, Z. (2004). "Real-time nonintrusive monitoring and predition of driver fatigue". IEEE Transactions on Vehicle Technology. 55 (3): 1052–1068.

[6]. Abulkair, M.; Alsahli, A.H.; Alzahrani, F.M; Alzahrani, H.A; Bahran, A.M; Ibrahim, L.F; Taleb,K.M (2015). "Mobile platform detect and alerts system for driver fatigue"

[7]. Rateb Jabbar*, Khalifa Al-Khalifa, Mohamed Kharbeche, Wael Alhajyaseen, MohsenJafari, Shan Jiang "Real-time Driver Drowsiness Detection for Android Application Using Deep Neural Networks Techniques" (2018)

[8]. https://data-flair.training/blogs/python-project-driver-drowsiness-detection-system/
[9]. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 2012;1097-1105.
[10]. Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt"Real-Time Driver Drowsiness Detection System Using Eye AspectRatio and Eye Closure Ratio- 2019SUSCOM". Hosting by Elsevier SSRN
[11]. Nora Kamarudin1 , Nur Anida Jumadi1,2,*, Ng Li Mun1 , Ng Chun Keat1 , Audrey Huong Kah Ching1 , Wan MahaniHafizah Wan Mahmud1 , Marlia Morsin1 , Farhanahani Mahmud –"Implementation of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness Detection Using Raspberry Pi Universal Journal of Electrical and Electronic Engineering" 6(5B): 67-75, 2019 http://www.hrpub.org DOI: 10.13189/ujeee.2019.061609
[12]. https://towardsdatascience.com/drowsiness-detection-system-in-real-time-using-opencv-and-flask-in-python-b57f4f1fcb9e
[13]. https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/
[14]. Franklin Silva, Eddie Galarza "Real time driver drowsiness detection based on Driver's Face Image Behaviour using a system of Human Computer interaction implemented in a smartphone." https://www.researchgate.net/publication/322250457_Real_Time_Driver_Drowsiness_Detection_Based_on_Driver's_Face_Image_Behavior_Using_a_System_of_Human_Computer_Interaction_Implemented_in_a_Smartphone
[15]. Ajeer, sharija and Mathew, Akhila, Driver Exhaustion Detection Systems (March 12, 2020). Available at SSRN: https://ssrn.com/abstract=3553366 or http://dx.doi.org/10.2139/ssrn.3553366
[16]. Li, Z.; Chen, L.; Peng, J.; Wu, Y. Automatic Detection of Driver Fatigue Using Driving Operation Information for Transportation Safety. Sensors 2017, 17, 1212.
[17]. J. Antunes, N. F. Neves and P. J. Veríssimo, "Detection and Prediction of Resource-Exhaustion Vulnerabilities," 2008 19th International Symposium on Software Reliability Engineering (ISSRE), Seattle, WA, 2008, pp. 87-96, doi: 10.1109/ISSRE.2008.47.
[18]. Pia M. Forsman, Bryan J. Vila, Robert A. Short, Christopher G. Mott, Hans P.A. Van Dongen, Efficient driver drowsiness detection at moderate levels of drowsiness, Accident Analysis & Prevention, Volume 50, 2013, Pages 341-350, ISSN 0001-4575, https://doi.org/10.1016/j.aap.2012.05.005. (http://www.sciencedirect.com/science/article/pii/S0001457512001571)
[19]. Srijayathi and M. Vedachary, "Implementation of the Driver Drowsiness Detection System," International Journal of Science, Engineering and Technology Research (IJSETR), vol. 2, no. 9, pp. 1751–1754, 2013.
[20]. S. Nagargoje and D. S. Shilvant, "Drowsiness Detection System for Car Assisted Driver Using Image Processing," International Journal of Electrical and Electronics Research ISSN, vol. 3, no. 4, pp. 175–179, 2015.