International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 04 | April -2020 ISSN: 2582-3930

DRIVER DROWSINESS DETECTION SYSTEM WITH OPENCV AND KERAS

PIYUSH SANKLHA , KATHIRESAN V. (PROFESSOR)
Dept of Master Of Computer Applications
JAIN UNIVERSITY

ABSTRACT

Every year many human beings lose their lives due to fatal street accidents around the world and drowsy driving is one of the foremost causes of avenue accidents and death.Driving controls are frequently the root reason of serious accidents.It can be detected earlier than a vital scenario arises and therefore, detection of driver's fatigue and its indication for the research paper.Most of the standard strategies to detect drowsiness are based on behavioural aspects while some are intrusive and may additionally distract drivers.Therefore, in this paper, a light-weight, actual time driver's drowsiness detection device is developed and applied on web application.Videos and pictures detects driver's face in each body with the aid of employing photograph processing techniques.So,we will build a system using Python, OpenCV, and Keras which will alert the driver when he feels sleepy.

Keywords:-drowsiness detection, ROI, CNN,pygame,keras,opencv,cascade classifiers

1. INTRODUCTION

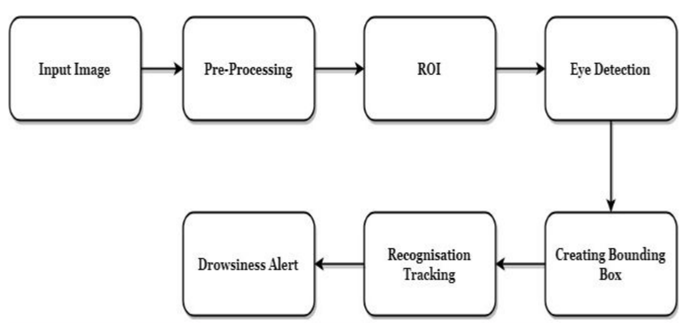
Drowsiness detection is a protection technology that can forestall accidents that are triggered by means of drivers who fell asleep whilst driving.The goal of this intermediate Python venture is to construct a drowsiness detection device that will detect that a person's eyes are closed for a few seconds. This gadget will alert the driver when drowsiness is detected.In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into a Deep Learning model which will classify whether the person's eyes are 'Open' or 'Closed'.

2. THE PROPOSED SYSTEM

In this section, we discuss our proposed system which detects driver drowsiness. The overall flowchart and the steps of our system is as follows:-

Step 1 – Take Image as Input from a Camera

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable.



\

Figure 1 - Block Diagram

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 04 | April -2020 ISSN: 2582-3930

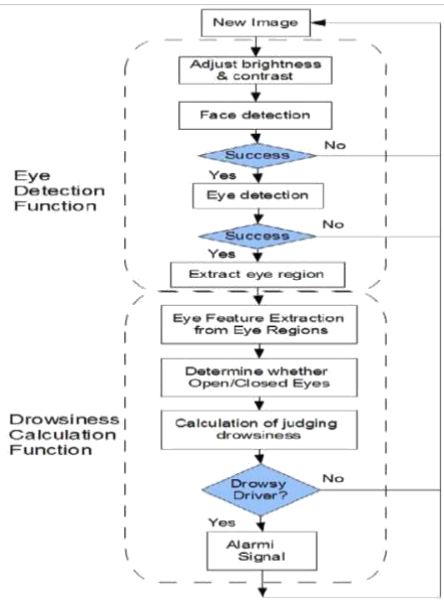


Figure 2 - Flow Chart

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

To detect the face in the image, we need to first convert the picture into grayscale as the OpenCV algorithm for object detection takes gray pix in the input. We don't need coloration facts to discover the objects. We will use the haar cascade classifier to realize faces. This line is used to set our classifier face = cv2.CascadeClassifier(' course to our haar cascade xml file'). Then we function the detection using faces = face.detectMultiScale(gray). It returns an array of detections with x,y coordinates, and height, the width of the boundary container of the object. Now we can iterate over the faces and draw boundary containers for every face.

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

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in leye and reye respectively then detect the eyes using left_eye = leye.detectMultiScale(gray). Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code.

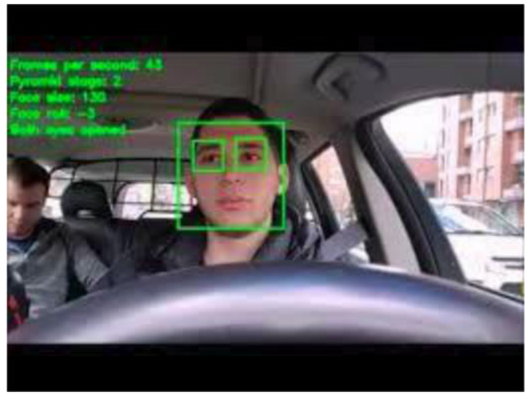


Figure 3 - Detection of face and eyes using ROI

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

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to operate certain operations because the mannequin desires the correct dimensions to begin with. First, we convert the colour photo into grayscale.Then, we resize the picture to 24*24 pixels and we normalize our statistics for better convergence.Expand the dimensions to feed into our classifier. Now we predict each eye with our model,if the fee = 1, it states that eyes are open, if fee=0 = then, it states that eyes are closed.

Step 5 – Calculate Score to Check whether Person is Drowsy

The rating is basically a fee we will use to decide how lengthy the person has closed his eyes. So if each eyes are closed, we will maintain on growing score and when eyes are open, we reduce the score. We are drawing the result that displays real time reputation of the person.

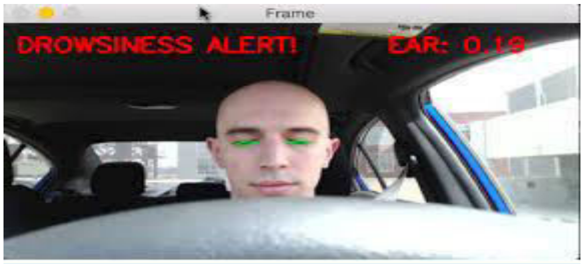


Figure 4 - Calculates whether eyes close or opened and give the results accordingly

3. CONCLUSION

In this work, a real time device that monitors and detects the loss of interest of drivers of vehicles is proposed. The face of the driver has been detected through shooting facial landmarks and warning is given to the driver to keep away from real time crashes.

Non-intrusive techniques have been preferred over intrusive techniques to forestall the driver from being distracted due to the sensors attached on his body. The proposed approach uses Eye AspectRatio and Eye Closure Ratio with adaptive thresholding to observe driver's drowsiness in real-time.

This is useful in situations when the drivers are used to strenuous workload and force always for long distances. The proposed system works with the accrued records sets below different conditions. The facial landmarks captured through the gadget are saved and laptop gaining knowledge of algorithms have been employed for classification.

4. FUTURE WORK

The future work can include integration of the proposed machine with globally used functions.The system, if integrated,can minimize the variety of casualties and injuries that appear regularly due to these drowsy states of the drivers.

This experiment can run as a part of pilot diagram i.e. for a few days/months in different regions of the world the place such incidents occur regularly. Thus, our proposed approach also gives the identical accuracy for the humans carrying spectacles. Accuracy of our proposed gadget improves with the extend in brightness of the surrounding environment. The work can be extended for specific sorts customers such as bike riders or in distinct domains like railways, airways etc

REFERENCES

[1]A. Malla, P. Davidson, P. Bones, R. Green and R.J/ones, "Automated Video-based Measurement of Eye Closure for Detecting Behavioral Microsleep", in 32nd Annual International Conference of the IEEE, Buenos Aires, Argentina,2010.

[2] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.

[3] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", inProceedings of the IEEE International Conference on Image Processing, 2002.

[4] OpenCV. Open Source Computer Vision Library Reference Manual, 2001.

[5]Driver Drowsiness Detection System-2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)

[6]S. Nagargoje and D. S. Shilvant, “Drowsiness Detection System for Car Assisted Driver Using Image Processing,” International Journal ofElectrical and Electronics Research ISSN, vol. 3, no. 4, pp. 175–179, 2015.

[7]S. Podder and S. Roy, “Driver’s drowsiness detection using eye status to improve the road safety,” International Journal of Innovation Research in Computer and Communication Engineering, vol. 1, no. 7, 2013.

[8]Omidi and G. Nasleseraji, “Non-intrusive Methods used to Determine the Driver Drowsiness: Narrative Review Articles,” International Journal of Occupational Hygiene, vol. 8, no. 3, pp. 186–191, 2016.

[9]Gopal and V. . Vineeth, “Driver Drowsiness Detection System,” International Journal of Advances in Electronics and Computer Science, vol.4, no. 3, pp. 101–104, 2017.