ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 1, Issue 1, (Jan-Jun) 2024

Drowsiness Detection and Prevention: A Deep Learning Solution for Safer Roads and Engaged Learning Environments

Khushi Pastaria
Department of CSIT
Acropolis Institute of Technology and Research
Indore, India

Duhita Jagdeva
Department of CSIT
Acropolis Institute of Technology and Research
Indore, India

***Abstract*—Drowsiness is phenomena of feeling sleepy on abnormal times. It may include drowsiness during the driving car, attending the sessions or any important activity. Drowsiness for driver may lead to accidents on road. Students feeling drowsiness could not pay attention in class and trainer may get irritated. It has been observed that drowsy driving caused 100,000 crashes, 71,000 injuries, and 1,550 fatalities in India only. We have proposed a deep learning model for drowsiness detection developed using python, OpenCV and Dlib packages. Drowsiness detected by monitoring aspect ratios of eyes and mouth. Performance evaluation of the proposed system designed is carried out by testing videos from a standard public dataset as well as real-time video captured in our lab. The proposed system gave a maximum recognition accuracy of 96.71% for dataset video input**

I. Introduction

A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers, and people traveling long distances suffer from lack of sleep. Drowsiness detection is a safety technology that can prevent accidents caused by drivers who fall asleep while driving. The objective of this intermediate Python project is to build a drowsiness detection system that detects when a person's eyes are closed for a few seconds.

The drowsiness detection is based on a deep learning model, which records facial behavior during any activity such as a trip or online classes. It then recognizes changes over the course of long trips and monitors the drowsiness level of a person during the activity, popping an alarm. Nowadays, due to COVID, online classes are gaining popularity, and people prefer them. Our drowsiness detector helps them stay attentive during their activities. Another use of the detector is in the office to monitor the productivity of employees, including office employees and security guards, in every place. This system will alert the driver when drowsiness is detected.

Improving technologies for recognizing or avoiding laziness at the wheel could be a major challenge in the field of accident evasion frameworks. Due to the risk that laziness presents on the street, strategies need to be developed to neutralize its influences. Driver diversion occurs when an object or event draws a person's attention away from the driving task. Unlike driver diversion, driver laziness involves no triggering event but is characterized by a gradual withdrawal of attention from the road and traffic demands. Both driver tiredness and diversion, however, may have the same effects, such as reduced driving performance, longer reaction time, and an increased risk of crash involvement. Based on the acquisition of video from the camera in front of the driver, real-time processing of an incoming video stream is performed to assess the driver's level of attention. Laziness is detected through changes in the pattern of the eyes. We have used PyCharm, Python, and Django to develop the model. The hardware required for the model includes a

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 1, Issue 1, (Jan-Jun) 2024

camera for capturing the person's moments along with a high-configuration system.
In the remaining sections, we will discuss the literature review in Section 2, the proposed model in Section 3, results and discussion in Section 4, and conclusions in Section 5.

II. Literature Review

Drowsiness detection poses a consistent challenge for researchers, primarily due to the intricate movements of the eyes. Wei et al. (2012) [5] presented a non-intrusive drowsiness acknowledgment technique utilizing eye-tracking and image processing. Their method introduced a robust eye detection algorithm capable of addressing challenges arising from variations in eye movements, yawning, and driver pose. In 2013, Elal et al.[6] proposed a module for Advanced Driver Assistance System (ADAS) aimed at reducing the occurrence of accidents This system utilizes automatic driver sleepiness identification based on visual data and Artificial Intelligence. Shibo and Xiaojie (2013)[7] enhanced the Histograms of Oriented Gradients used for representing the edge of picture data. Their approach incorporated background deduction detection with Histograms of Oriented Gradients to achieve the necessary accuracy and meet continuous demand. Manu (2016) [8] outlined an efficient method for drowsiness detection consisting of three distinct stages: facial features detection using Viola Jones, eye tracking, and yawning detection. Feng et al. (2019)[6] developed a real-time driving laziness detection algorithm that accounts for individual driver differences. Their approach involved constructing a deep-felled convolutional neural network to recognize the face region, thus avoiding accuracy issues associated with artificial feature extraction. Using the Dlib toolbox, landmarks of frontal driver facial features in a frame were identified. Feng et al. also introduced another parameter called Eyes Aspect Ratio, derived from eye landmarks, to assess driver tiredness. Bruno et al. (2019)[9] devised a drowsiness level recognition system integrating image processing with the utilization of Raspberry Pi3, OpenCV library, and sensors. Still there is scope of improvement in the methods .

III. Proposed Method

In most of the other application in a few ways relate to highlights of the eye (ordinary reflections from the eye) inside a video picture of the driver. The first point of this venture was to first identifying the face with proper outlining of that and differentiate eyes and mouth part very accurately to work on that utilize the retinal reflection as a implies to finding the eyes on the confront, and after that utilizing the nonappearance of this reflection as a way of identifying when the eyes are closed as same as on mouth. Applying this calculation on sequential video outlines may help within the calculation of the eye closure period. The eye closure period for lazy drivers are longer than ordinary blinking. It is additionally exceptionally longer time may result in an extreme crash. So, we'll caution the driver immediately as a closed eye is identified.

A. Take Picture as Input from a Camera

With a webcam, we can take picture as input to get admission to the webcam, we made countless loops so that it will capture each frame of object. We use the approach supplied through OpenCV, cv2.VideoCapture(0) to get admission to the digital camera and set the seize object (cap). Cap.Read() will read each frame and we store the photograph in a frame variable.

B. Image face detection and creation of a region of interest (ROI)

To identify the face within the picture, we ought, to begin with, change over the image into grayscale as the Dlib Library calculation for protest location takes gray images within the input Here we used CNN for extracting the gray image of person in dimension of 32*32,. We don't require color data to identify the objects. By using dlib library we mark landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points as we know that These points are identified from the pre-trained model where the Dataset was used going be utilizing a haar cascade

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 1, Issue 1, (Jan-Jun) 2024

classifier to identify faces. This line is utilized to set our classifier to confront

```
\begin{lstlisting}[language=Python]
hog_face_detector =
dlib.get_frontal_face_detector()
dlib_facelandmark =
dlib.shape_predictor(
"shape_predictor_68_face_landmarks
.dat")
\end{lstlisting}
```

Presently able to emphasize over the faces and draw boundary boxes for each ROI.

C. ROI detects the eyes and feeds them to the classifier

The same strategy to identify faces is utilized to distinguish eyes and mouth To begin with, we set the c classifier for eyes in l-eye and r-eye separately by giving them the rage of ordinates of face and mouth at that point distinguish the eyes and mouth landmarks by utilizing

```
\begin{lstlisting}[language=Python]
x = face_landmarks.part(n).x
y = face_landmarks.part(n).y
cv2.circle(frame, (x, y), 1,
(0, 255, 255), 1)
\end{lstlisting}
```

Presently we have to extract only the eyes and mouth information from the complete image. This may be accomplished by extricating the boundary box of the eye and after that able to drag out the eye image from the outline with code.

D. Categorizes that eyes are open or closed

We are utilizing classifier for anticipating the eye status. To bolster our picture into the demonstration, we got to perform certain operations since the model needs the right measurements, to begin with. To begin with, we change over the color picture into grayscale utilizing

```
\begin{lstlisting}[language=Python]
```

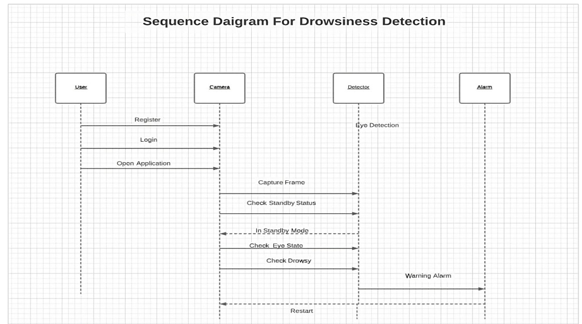


Fig. 1: Sequence Diagram For Drowsiness Detection

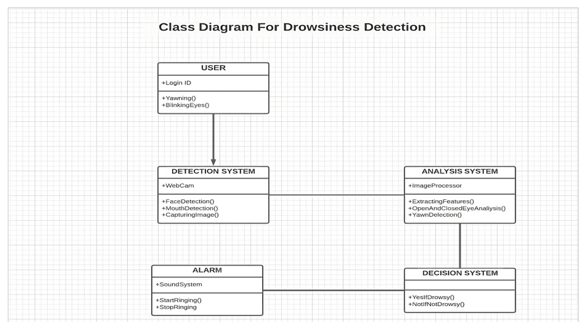


Fig. 2: Class Diagram For Drowsiness Detection

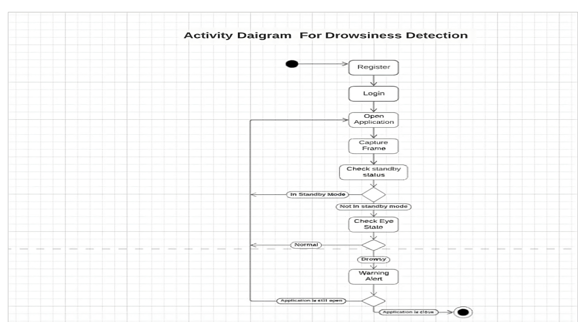


Fig. 3: Activity Diagram For Drowsiness Detection

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 1, Issue 1, (Jan-Jun) 2024

```
def calculate_EAR(eye):
    A = distance.
        euclidean(eye[1], eye[5])
    B = distance.
        euclidean(eye[2], eye[4])
    C = distance.
        euclidean(eye[0], eye[3])
    ear_aspect_ratio =
    (A + B) / (2.0 * C)
    # Assuming (1) is a comment
   # return ear_aspect_ratio
\end{lstlisting}
```

It states that eyes are open if the value of lpred[0] equals 1; otherwise, it states that eyes are closed.

E. Categorizes that mouth are open or closed

We are utilizing classifier for anticipating the mouth status. To bolster our picture into the demonstration, we got to perform certain operations since the model needs the right measurements, to begin with. To begin with, we change over the color picture into grayscale utilizing

```
\begin{lstlisting}[language=Python]
def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip =
    np.concatenate(
    (top_lip, shape[61:64])
    )
    low_lip = shape[56:59]
    low_lip =
    np.concatenate(
    (low_lip, shape[65:68])
    )
    top_mean =
    np.mean(top_lip, axis=0)
    low_mean =
    np.mean(low_lip, axis=0)
    yawn =
    abs(top_mean[1] - low_mean[1])
    return yawn
\end{lstlisting}
```

F. Check whether Person is Drowsy or not

The score is fundamentally esteem we'll utilize to decide how long the individual has closed his eyes. So if both eyes are closed, we are going to keep on expanding the score and when eyes are open, we diminish the score We are drawing the result on the screen $cv2.putText()$ work which can display the real-time status of the individual. A limit is characterized for the case if score becomes more noteworthy than the person's eyes are closed for a long period of time. This is often when we beep the caution utilizing sound.play() and if after sound play person couldn't wake up or come under screen all the set of their meeting will automatically closed and meet will throw his/her out of the meet. we also and apply limit of blinking eyes so it could not beep if person is normally blinking there eye, at the same time after capturing and giving result it will throw as garbage so that is good by space complexity and user window couldn't fill with trash images by

```
\begin{lstlisting}[language=Python]
cap.release()
cv2.destroyAllWindows()
\end{lstlisting}
```

G. Results and Experimental Analysis

The developed driver anomaly detection framework is capable of identifying instances of laziness, intoxication, and carelessness in drivers within a short span of time. The Laziness Detection Framework, based on the analysis of driver eye closures, effectively distinguishes between normal eye blinking and signs of tiredness, thereby detecting instances of laziness during driving. This proposed system significantly contributes to preventing accidents caused by driver sleepiness. Notably, the system operates reliably even when drivers are wearing spectacles and under low-light conditions, provided the camera delivers high-quality output. Head and eye position data are acquired through a series of custom-built image processing algorithms. During monitoring, the system assesses whether the driver's eyes are open or closed. If the eyes remain closed for an extended period, a warning signal is issued, and the driver's alertness level is evaluated based on the frequency of eye closures.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 1, Issue 1, (Jan-Jun) 2024

The threshold value is set to 0.25. Here are the scenarios based on the calculated Eye Aspect Ratio (EAR):

- **Case I:** If EAR = 0.37, which is clearly above the EAR threshold value (0.25) set for the system, indicating that the driver is alert and there are no signs of drowsiness.
- **Case II:** If EAR = 0.20, which falls below the EAR threshold of the Drowsiness Detection system, triggering an alarm as the driver is in a semi-drowsy state.
- **Case III:** If EAR = 0.15, significantly below the EAR threshold (0.25) of the system, a "DROWSINESS ALERT!" warning is displayed accompanied by a loud alarm sound.

The Mean Squared Error (MSE) or Mean Squared Deviation (MSD) of the estimator measures the average squared difference between the estimated and true values of the detector based on its EAR value. It serves as a risk function, corresponding to the expected value of the squared error loss.
X-AXIS = REAL DATA
Y-AXIS = PREDICTED DATA
To write the Mean Squared Error (MSE) formula in LaTeX, you can use the following code:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (1)$$

In this formula:

- MSE represents Mean Squared Error.
- n is the number of data points.
- y_i represents the actual (real) data.
- $\hat{y}_i$ represents the predicted data.

$MSE = 0.8$

IV. Experimental Results

A. Experimental Dataset

The experimental dataset used in our project is the Dataset, which consists of 800 indoor and as many outdoor images. The dataset covers a large variety of identities, face size, lighting conditions, pose, etc. The images in the dataset cover more expressions than the common neutral smile such as surprise or scream. Annotations on the image were

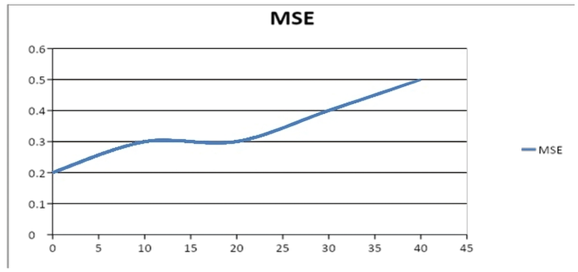
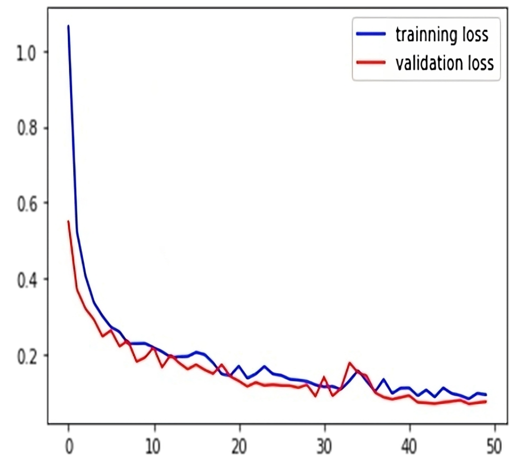
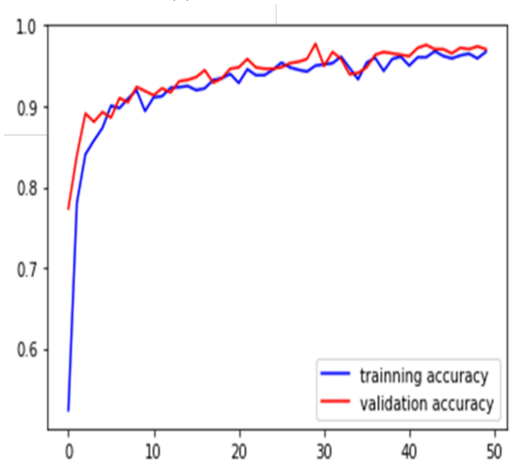


Fig. 4: Loss Graph



(a) Validation Loss



(b) Training Accuracy

Fig. 5: Validation Loss and Training Accuracy

done using 68-point markup covering all the key facial landmarks including eyes, mouth, nose, etc. But with regards to this project our region of interest (ROI) includes only the eyes and lips region which can be extracted using Python slicing techniques.

TABLE I: Performance Analysis

Total Image	Number of Annotated faces involved
295	1
56	2
56	[3,7]

TABLE II: System Testing

Test case	**Eye detect**	**Mouth detect**	**Result**
Case 1	No	No	No Result
Case 2	Yes	No	No Result
Case 3	Yes	Yes	Voice Alert
Case 4	Yes	No	Voice Alert
Case 5	Yes	Yes	Voice Alert

According to the methodology of the system, when the eyes are closed for more than the set threshold number of frames or when the driver is yawning, then the driver is feeling tired. Hence, one of these distinguished cases arises, and the corresponding result happens. The accuracy as measured in the performance analysis phase is almost found to be 100% in the case when the face is properly aligned and no wearable obstacle is present. Accuracy drops down a little when an obstacle is present (e.g., Hat). Ambient lighting conditions are very essential for getting proper results. In case, the user's eye closure and yawn occur simultaneously, a voice alert is raised but the system behaves in an erroneous and unsynchronized fashion. Hence, such a situation should be avoided to prevent any inconsistent results.

V. Conclusions

The study has shown promising results in applying vehicular driver surveillance based on artificial vision techniques and implemented in a smartphone. The implemented system allows efficient detection of indicators that appear in drowsiness, as long as the measurements are carried out under the established conditions. The correct functioning of the system depends on these conditions. The increase in processing characteristics in smartphones made it possible to develop an application of artificial vision capable of detecting the face and visual indicators present in a person who suffers from drowsiness such as yawning, head movements, and the state of the eyes. The symptoms that people present during the transition between awake and asleep appear as the intensity of drowsiness increases. The greater intensity of drowsiness means a higher loss of concentration and a lower ability of driver reaction. In developing this work, the implementation of 3 levels of sleepiness allows the system to alert the driver about their condition, not necessarily at a critical level where it may have serious repercussions, but at early levels where drowsiness is just emerging. An HCI could be implemented using smartphones like shown in this work, which would allow the massification of their use and therefore provide greater solutions improving the quality of life of the people even if has special skills.

References

[1] A. Williamson and T. Chamberlain, "Review of on-road driver fatigue monitoring devices," NSW Injury Risk Management Research Centre, University of New South Wales, July 2013.

[2] S. S. Wreggit, C. L. Kim, and W. W. Wierwille, "Fourth Semi-Annual Research Report, Research on Vehicle-Based Driver Status Performance Monitoring," Virginia Polytechnic Institute and State University, ISE Department, Blacksburg, VA, January 2013.

[3] H. Singh, J. S. Bhatia, and J. Kaur, "Eye tracking based driver fatigue monitoring and warning system," in Proc. IEEE IICPE, New Delhi, India, Jan. 2014.

[4] B. Fleming, "New Automotive Electronics Technologies," in Proc. International Conference on Pattern Recognition, pp. 484-488, December 2012.

[5] Zhang, W., Cheng, B., Lin, Y. 2012. Driver drowsiness recognition based on computer vision technology. Tsinghua Science and Technology, vol. 17, no. 3, pp. 354-362.

[6] You, F., Li, X., Gong, Y., Wang, H., Li, H. 2019. A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration. IEEE Access, vol. 7, pp. 179396-179408.

[7] Zhang, S., Wang, X. 2013. Human detection and object tracking based on Histograms of Oriented Gradients. 2013 Ninth International Conference on Natural Computation (ICNC)

[8] Manu, B. N. 2016. Facial features monitoring for real time drowsiness detection. 2016 12th International Conference on Innovations in Information Technology (IIT).

[9] Eraldo, B., Quispe, G., Chavez-Arias, H., Raymundo-Ibanez, C., Dominguez, F. 2019. Design of a control and monitoring system to reduce traffic accidents due to

drowsiness through image processing. 2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)