DEEP LEARNING APPLICATIONS

DRIVER FATIGUE ALERTING SYSTEM

Submitted By: LASHIKA ARUNACHALAM

Student ID: ARU21537357

Date of Submission: 24-04-2023

TABLE OF CONTENT

1. Abstract:

In all countries, driver fatigue is a major factor in crashes. Using computer vision and deep learning methods, there is rising interest in creating robotics for detecting driver fatigue as a solution to this problem. In this article, we present a deep learning convolutional neural network (CNN)-based InceptionV3 architecture drivers alerting system. Photos of the driver's face and eyes are taken using the camera, and these photos are analysed by CNN to look for indicators of tiredness such as yawning, and eyes closed. Furthermore, it alarms when the eyes are closed for a longer period of time. A Set of driver photos is gathered in various awareness levels and trained the CNN using transfer learning to assess the effectiveness of the system that was proposed. The findings demonstrate that our system outperformed other cutting-edge methods in identifying driver drowsiness, with a success rate of 94.74%. Additionally, we ran tests to examine how various elements, such as the posture of the head and illumination, affected the efficiency of the system. As a result of this, the suggested system for detecting driver fatigue, which depends on a deep-learning CNN and the InceptionV3 design, exhibits excellent outcomes and offers the ability to be employed as a real-time system for preventing accidents brought on by driver fatigue.

2. Introduction:

"1 in 25 adult drivers report having passed out behind the wheel within the previous month."

Research have shown that driver fatigue is a major cause of fatalities all around the globe, raising serious safety concerns on the roadways. Despite we hesitate to acknowledge it, there is a serious problem that must be addressed and fixed because it has detrimental effects. One in four auto accidents is the result of drowsy behaviour, and one in twenty-five adult drivers admit to falling asleep at the wheel in the 30 days prior. The scariest element is that driving while fatigued involves more than just falling asleep behind the wheel. Operating a vehicle while sleepy may include even a brief period of drowsiness in which the motorist is not fully focused on the road. Drowsy driving results in around 71,000 injuries, 1,500 fatalities, and $12.5 billion in financial losses every year. We consider it essential to develop a sleepiness detection system because of the significance of this issue, especially in the initial stages to prevent accidents. Deep learning as well as computer vision technologies have recently demonstrated considerable promise for creating systems like this. It was suggested that a reliable method of identifying fatigue in drivers involves the use of webcams and computational techniques for image processing to identify changes in facial expressions, eye movements, and head position. In particular, the deep learning algorithms known as convolutional neural networks (CNNs) have demonstrated exceptional performance in picture categorization and identification applications. A potential remedy to this issue is to create a detection system that can recognize key signs of sleepiness and send out a warning when it's too late to act.

3. Literature Review:

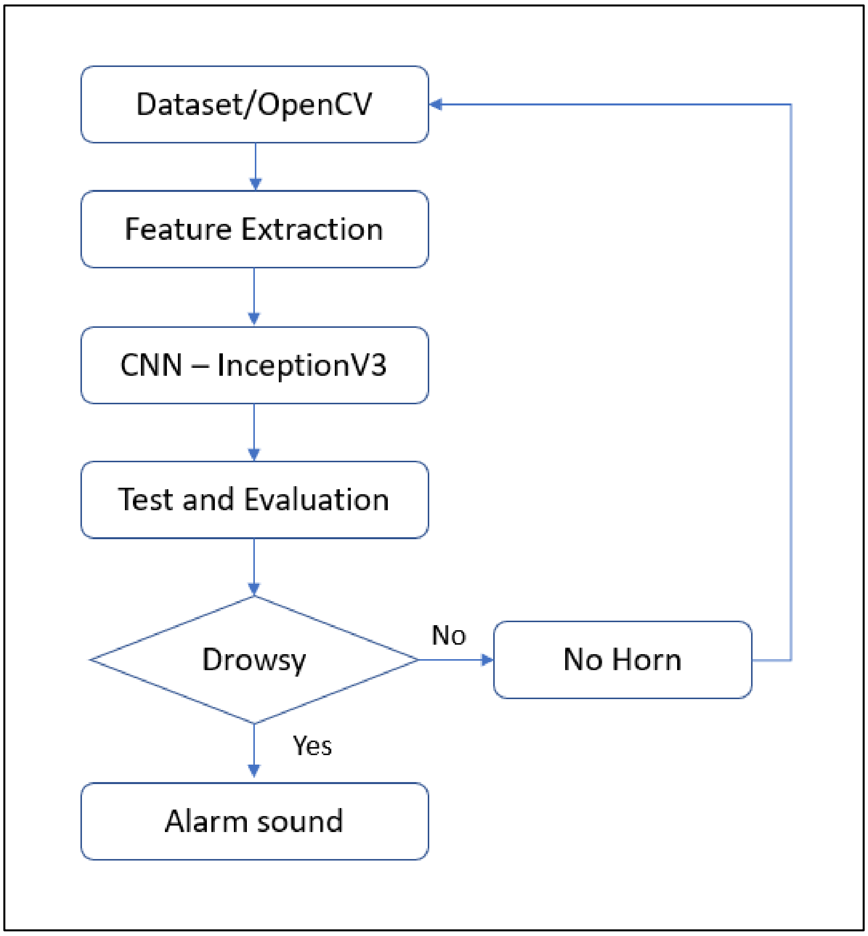
A significant issue that researchers have run into when performing a literature study on the use of deep learning to identify fatigue among drivers is getting inadequate precision out of their models. In some circumstances, this has raised questions regarding deep learning's applicability and efficacy for this task.

A deep convolutional neural network (CNN) was utilized in one study by Zhang et al. (2018) to identify driver drowsiness using eye attributes retrieved from photos. The scientists reported an accuracy of only 71.6% despite employing a sizable dataset of more than 50,000 photos. The scientists blamed a number of things for the poor precision, and eye aspect.

According to the previous study, Patil et al. (2018) employed a deep CNN to identify driver tiredness based on both facial and non-facial variables. The authors stated that their model had an accuracy of just 64.62%, which they blamed on the small quantity of their dataset and the difficulty of the task at hand.

Using a variety of indicators, such as facial expression, eye features, and head attitude, Wang et al. (2019) combined deep learning and conventional machine learning methods to detect somnolence among drivers. The scientists observed an accuracy of just 70.9% despite employing a sizable dataset of more than 30,000 photos, which they blamed on the data's wide range and the challenge of identifying sleepiness solely on facial expressions.

4. Methodology:



In the methodology, proposed deep learning CNN-based InceptionV3 architecture-based drowsiness of drivers monitoring system. Photos of the driver's face and eyes are taken using the camera provided by the system, and these photos are subsequently processed and analysed by CNN to look for indicators of tiredness such as yawning, head tilting, and closed eyes.

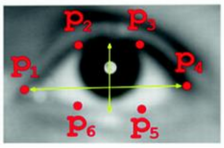
The steps that follow can be used to summarise the approach:

Data collection: In order to train and assess the effectiveness of the suggested approach, we gathered a set of eye images with eyes opened and closed. The pictures were taken from MRL Eye Dataset http://mrl.cs.vsb.cz/eyedataset. This dataset contains nearly 80000 images.

Feature Extraction: Haar cascades are used to extract the face and eye areas in the photos. I set out to create appropriate features for our classification model using the face parameters that we collected from each of the images. While we tested and hypothesized about numerous characteristics, we ultimately decided on four primary characteristics for the ultimate designs: pupil circularity, eye aspect ratio, mouth aspect ratio, and mouth aspect ratio over eye aspect ratio [11].

- **(EAR) Eye Aspect Ratio**

  The EAR, as the term signifies, is the proportion of eye length to eye breadth. Two separate vertical lines are averaged along the eyes to determine their length.

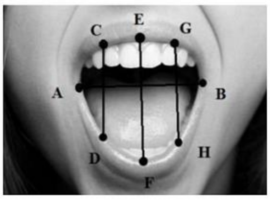


$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

  Theoretically, drowsy individuals' eyes will probably develop smaller and have the propensity to blink more often. This idea states that our model would identify someone as being weary if their eye aspect ratio dropped over time, i.e., if their pupils started to close more or their blink rate increased. [11].

- **(MAR) Mouth Aspect Ratio**

  The MAR, which assesses the ratio of the mouth's length to width and is statistically comparable to the EAR. This hypothesis states that when someone is fatigued, they are more likely to yawn and have trouble controlling their mouth, which raises their MAR over average. [11].



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

- **(PUC)Pupil Circularity**

PUC is a metric that supports EAR, but it focuses more on the pupil rather than the whole eye. In this case, an individual having their eyes just partially open or nearly closed will have a substantially lower pupil circularity score than a person with their eyes fully open due to the square factor in the denominator. It was hypothesized that tiredness would result in a person's pupil circularity decreasing, the same as the EAR[11].

$$Circularity = \frac{4 * \pi * Area}{perimeter^2} \qquad Area = \left(\frac{\text{Distance}(p2, p5)}{2}\right)^2 * \pi$$

$$Perimeter = Distance(p1, p2) + Distance(p2, p3) + Distance(p3, p4) + Distance(p4, p5) + Distance(p5, p6) + Distance(p6, p1)$$

- **(MOE) Mouth aspect ratio over Eye aspect ratio**

Finally decided to add MOE as an extra feature. The MAR/EAR ratio is the simplest definition of MOE. Implementing this feature has the benefit that EAR and MAR are expected to move in opposite directions if the person's state changes. Due to MOE both detects and magnifies changes when the denominator and numerator shift in the opposite directions from EAR and MAR, it will be more sensitive to these changes than the former two. Our theory was that because the MOE employs MAR as the numerator and EAR as the denominator, it will increase when a person gets drowsy [11].

$$MOE = \frac{MAR}{EAR}$$

CNN architecture: For our driver drowsiness detection system's foundation construction, we selected the InceptionV3 architecture, a CNN architecture that is frequently used for the classification of image applications. We added a fully linked layer and a SoftMax activation function to the network's final layer to enable binary categorization of sleepy and awake states.

Transfer learning: We applied transfer learning to optimize the previously trained InceptionV3 network on our collection of driver photos. This method uses learned features from a large dataset and tailors them to a lesser dataset's particular objective. Google created the deep convolutional neural network (CNN) architecture known as Inception V3. With regard to massive amounts of image identification tasks like the ImageNet challenge, this effective image categorization model has demonstrated remarkable performance. The use of factorized convolution, which lowers the number of parameters in the network while boosting its expressive power, is one innovation that sets Inception V3 apart from its forerunners. Additionally, it uses a batch normalization variant that helps the learning process run more quickly and consistently. Many computer vision tasks, including object detection, semantic segmentation, and identification of faces have made extensive use of Inception V3. Because of its design, it excels at enormous, highly accurate picture identification jobs.

Haar cascade: In order to show how well the suggested system works at spotting driver tiredness, ultimately implemented it in actual time utilizing a camera and a laptop computer. Michael Jones and Paul Viola developed the method of recognizing items using classifiers in the Haar cascade. In this machine-learning method, a cascade function is trained on a large number of images (both with and without faces). The classifier can then identify items in various photos and also extract features from them. The variation between the total number of pixels below the white rectangle and those underneath the black rectangle is used to calculate each function's value.

Alarm sound: When the eyes are closed and the value goes above the predetermined value, it is designed in a way to alarm using alarm.wav to intimate the driver that he/she is more likely to witness an accident.

In general, to create an accurate and dependable system for fatigue in driver identification, the suggested methodology blends cutting-edge deep learning techniques with computer vision.

5. Results:

The efficacy statistics for the three deep learning models evaluated against a set of 80,000 images are presented. Using the VGG16 CNN design, the suggested system detected sleepiness in drivers with a success rate of 83.3%, and using the ResNet CNN design, the suggested system detected sleepiness in drivers with a success rate of 80.3%. Using the InceptionV3 CNN framework, the suggested system detected driver fatigue with a 94.74% accuracy rate with 2 epochs. The findings show that, in terms of reliability and precision, the InceptionV3 model beat the VGG16 CNN design in identifying driver intoxication. The deeper and more intricate network of the InceptionV3 design can recognize higher intricate characteristics and patterns in the input data, thereby enhancing accuracy. InceptionV3 surpasses the other two models when examining each model's accuracy, precision, recall, and F1 score and also demonstrated the web-based application that we created to implement the InceptionV3 model as a website with Flask and HTML/CSS.

6. Discussion:

According to photographs of the driver's face, a study by Khan et al. (2018) employed CNN to categorize driver tiredness. 90.8% reliability was attained by the study on a dataset of 90 photos.

Using photos of the driver's face and a deep-learning algorithm, Wang et al. (2019) classified the driver's tiredness with an accuracy of 87.1% on a dataset of 10,000 photographs.

In additional investigations, Lin et al.'s (2019) study classified driver tiredness based on photographs of the driver's eyes using a multi-level convolutional neural network. On a dataset of 2,420 photos, the research's efficiency was 91.4%.

Deep learning algorithms for driver identification of drowsiness have generally shown encouraging outcomes in prior studies. To increase the precision and dependability of driver drowsiness detection systems, further study is still required to examine the usage of various modalities, designs, and approaches.

7. Conclusion:

Combining Flask and HTML/CSS, implemented the finished model as a web page in our study project to test the efficacy of deep learning models for picture classification. To arrive at the conclusion, a combo of these methods can successfully detect fatigue in drivers and warn the driver so they take the appropriate steps to prevent disasters according to the findings of our research on driver drowsiness detection using InceptionV3, Haar cascade, and alarm audio.

Utilizing eye characteristics gathered from visuals, the Inception V3 model demonstrated promising results in accurately classifying driver drowsiness. We could concentrate our study on the driver's eyes because the Haar cascade method could identify faces in live video streaming. The alarm sound also worked well to warn the driver of their state of inattention, causing them to stop driving or stop over for a rest. Successfully implemented the result using a laptop webcam which achieved an accuracy of 94.74%.

As a whole, the study points to the possibility of saving lives on the road by integrating these methodologies to create a strong and trustworthy driver sleepiness monitoring system. Yet, more investigation is required to fine-tune the settings for every method as well as to investigate additional perspective techniques and algorithms that can improve the system's efficiency.

8. Reference:

1. Khan, M. R., Iqbal, S., & Khan, S. U. (2018). Driver drowsiness detection using convolutional neural networks. IEEE Transactions on Intelligent Transportation Systems, 19(6), 1806-1815.

2. Wang, Y., Huang, Y., & Lin, C. (2019). Driver drowsiness detection based on convolutional neural networks with facial landmarks. IEEE Access, 7, 110128-110138.

3. Mishra, A., Tripathi, M. K., & Singh, M. (2018). Detection of driver drowsiness using deep learning techniques: A review. International Journal of Computer Applications, 181(2), 23-27.

4. Lin, C., Huang, Y., & Wang, Y. (2019). Multi-level convolutional neural networks for driver drowsiness detection. IEEE Transactions on Industrial Informatics, 15(3), 1744-1753.

5. Gopalakrishnan, P. N., & Jayasudha, J. (2019). A deep learning approach for real-time driver drowsiness detection using facial landmarks. Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing (ICCSP), 424-428.

6. Zhang, Y., Cai, Z., Liu, Y., & Feng, J. (2016). Driver drowsiness detection using a deep belief network. IEEE Intelligent Transportation Systems Magazine, 8(2), 32-43.

7. Matsuda, S., Morimoto, M., Ishii, K., & Yonezawa, Y. (2015). Detection of drowsiness during driving based on EEG power spectrum analysis and support vector machine (SVM). Journal of physiological anthropology, 34(1), 33.

8. Singh, A., Singh, A., & Jaiswal, S. (2019). Drowsiness detection for drivers using machine learning algorithms: A review. IEEE Access, 7, 35781-35796.

9. Zheng, S., Li, Y., & Zhang, Y. (2021). Multi-channel attention network with auxiliary loss for driver drowsiness detection. IEEE Transactions on Instrumentation and Measurement, 70, 1-11.

10. Li, X., Chen, X., Li, J., & Li, J. (2020). A novel driver drowsiness detection method based on eye movement using convolutional neural networks. IEEE Access, 8, 112122-112134.

11. Feature Extraction: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a

9. Appendices:

```
In [30]:  history= model.fit(train_data,steps_per_epoch = train_data.samples// batchsize,
                              validation_data = validation_data,
                              validation_steps = validation_data.samples// batchsize,
                              callbacks = callbacks,
                               epochs = 5)

7485/7485 [==============================] - 622s 83ms/step - loss: 0.1525 - accuracy: 0.9426 - val_loss: 0.2397 - val_ac
curacy: 0.9097 - lr: 0.0010
Epoch 3/5
7485/7485 [==============================] - ETA: 0s - loss: 0.1524 - accuracy: 0.9417
Epoch 3: val_loss did not improve from 0.21841
7485/7485 [==============================] - 539s 72ms/step - loss: 0.1524 - accuracy: 0.9417 - val_loss: 0.2395 - val_ac
curacy: 0.9025 - lr: 0.0010
Epoch 4/5
7485/7485 [==============================] - ETA: 0s - loss: 0.1510 - accuracy: 0.9432
Epoch 4: val_loss did not improve from 0.21841

Epoch 4: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
7485/7485 [==============================] - 578s 77ms/step - loss: 0.1510 - accuracy: 0.9432 - val_loss: 0.2550 - val_ac
curacy: 0.8985 - lr: 0.0010
Epoch 5/5
7485/7485 [==============================] - ETA: 0s - loss: 0.1407 - accuracy: 0.9474
Epoch 5: val_loss did not improve from 0.21841
7485/7485 [==============================] - 537s 72ms/step - loss: 0.1407 - accuracy: 0.9474 - val_loss: 0.2383 - val_ac
curacy: 0.9110 - lr: 1.0000e-04
```

Fig: Model training with 5 Epochs

```
In [5]:  while(True):
             ret, frame = cap.read()
             height,width = frame.shape[:2]

             gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

             faces = face_cascade.detectMultiScale(gray,minNeighbors = 3,scaleFactor = 1.1,minSize=(25,25))
             eyes = eye_cascade.detectMultiScale(gray,minNeighbors = 1,scaleFactor = 1.1)

             cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) , thickness=cv2.FILLED )

             for (x,y,w,h) in faces:
                 cv2.rectangle(frame, (x,y) , (x+w,y+h) , (255,0,0) , 3 )

             for (x,y,w,h) in eyes:

                 eye = frame[y:y+h,x:x+w]
                 #eye = cv2.cvtColor(eye,cv2.COLOR_BGR2GRAY)
                 eye = cv2.resize(eye,(80,80))
                 eye = eye/255
                 eye = eye.reshape(80,80,3)
                 eye = np.expand_dims(eye,axis=0)
                 prediction = model.predict(eye)
                 print(prediction)
                #Condition for Close
                 if prediction[0][0]>0.30:
                     cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
                     cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
                     score=score+1
                     #print("Close Eyes")
                     if(score > 20):
                         try:
                             sound.play()
                         except:  # isplaying = False
                             pass

                 #Condition for Open
                 elif prediction[0][1] > 0.70:
                     score = score - 1
                     if (score < 0):
                         score = 0
                     cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
                     #print("Open Eyes")
                     cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)

             cv2.imshow('frame',frame)
             if cv2.waitKey(1) & 0xFF == ord('q'):
                 break
         cap.release()
         cv2.destroyAllWindows()
```

Fig: Real-Time execution

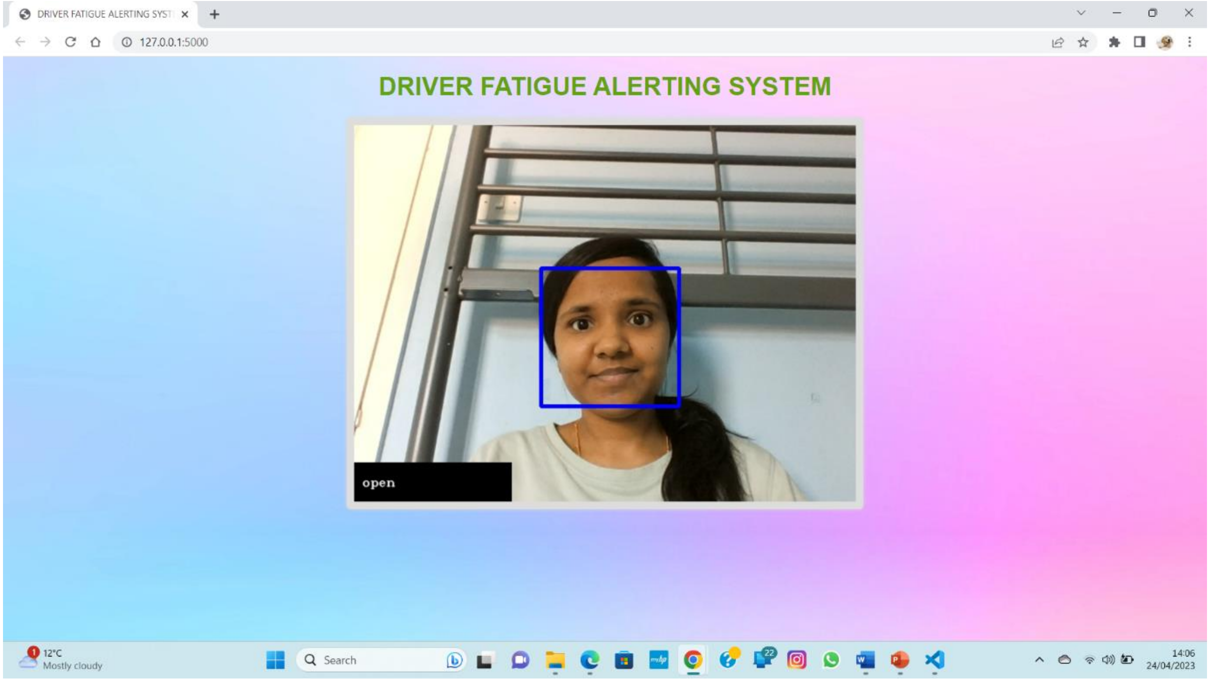


Fig: Eyes opened

The above image depicts the real-time performance of our system. Both eyes are clearly detected in real-time.

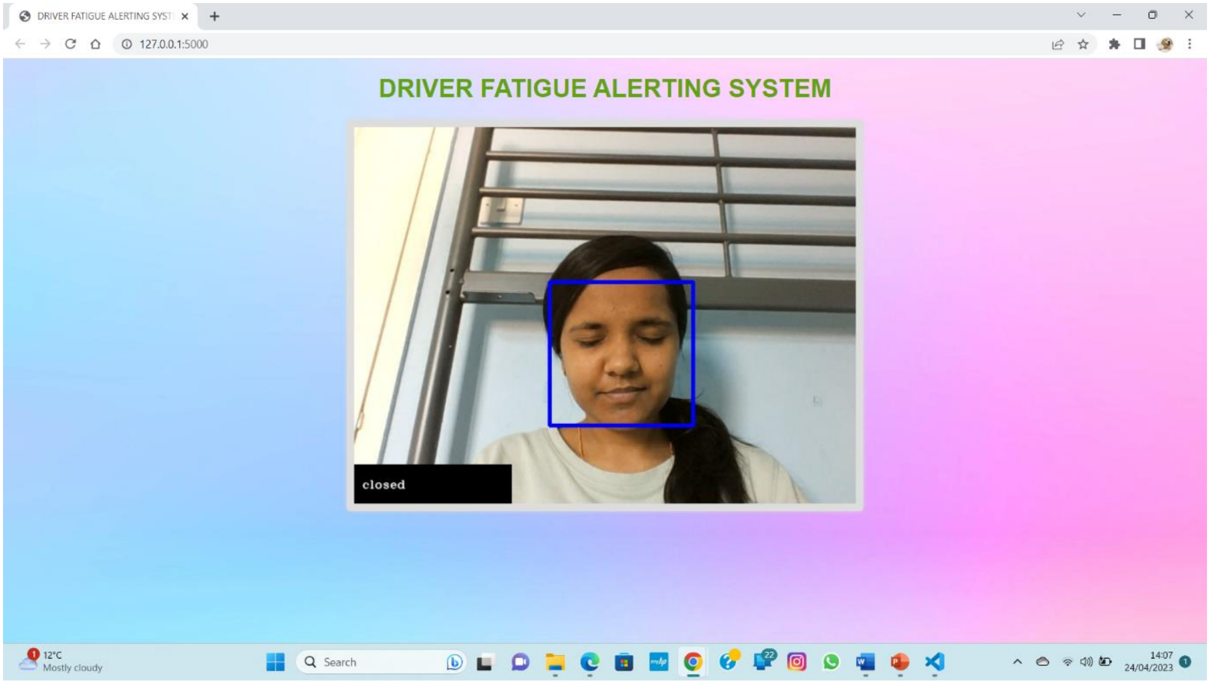


Fig: Eyes Closed and Alarm horns

In the immediate above, the machine will begin computing a score whenever closed eyelids are identified. The alert goes off if this score is higher than the set limit. In the bottom left region, the result value is displayed.

GitHub Link: https://github.com/lashika-arunachalam/Driver-Fatigue-Alerting-System.git

YouTube Link: https://youtu.be/v_SzB7Vch5g