Driver Drowsiness Detection using HAAR and CNN

Gaurav Singh[1], Vikas[2], Durgesh Gaur[3], Nancy Agarwal[4]

[1,2,3,4]School of Computer Application and Technology, Galgotias University Greater Noida(UP), INDIA

Abstract
A major risk factor for traffic accidents is tiredness among drivers . It is a serious issue that emphasizes the necessity of real- time, efficient detection and preventive techniques. In order to effectively identify driver drowsiness, this research presents a hybrid technique that combines Convolunal Neural Network(CNN) and Haar Cascade. Because of its effectiveness in real-time application, the Haar Cascade method is employed for the detection of facial and eye regions. A CNN is then used to examine the identified eye regions and determine if the driver is awake orsleepy. High accuracy in identifying patterns of drowsiness is ensured by training the machine on annotated datasets of eye pictures.The suggested method strikes a balance between accuracy and speed, which qualifies it for real-time vehicle time vehicle deployment. The durability and dependability of this approach are demonstrated by experimental fundings, offering a workable way to improve road safety.

Keywords: Drowsiness, Image Processing, Accuracy, Feature Learning, And Real-Time Processing.

I. INTRODUCTION

One of the main cause of traffic accidents globally is driver fatigue[1-3], which frequently has serious repercussions for both drivers and other road users. Early detection of drowsiness is essential to prevent such accidents and save lives. Traditional methods, such as monitoring vehicle movement or relying on driver feedback, have limitations in accuracy and reliability.

In order to identify drowsiness in real time, this research presents a hybrid techniques that combines the Haar Cascade algorithm [13] with Convolutional Neural Network(CNN)[14]. The CNN evaluates the identified eye regions to as certain if the driver is awake or sleepy, while the Haar Cascade method swiftly locates the face and eye areas in video frames. The suggested solution makes use of both approaches advantages:

CNN's deep learning capabilities for precise classification and Haar Cascades' speed and effectiveness for object detection. In conclusion, our study's CNN algorithm is in charge of precisely determining whether the driver's eyes are open or closed. By giving the Haar cascade the accuracy to required to assess tiredness, it enhances the system's speed and dependability. This method guarantees a quick, dependable, and affordable fix that may be incorporated into car systems to increase traffic safety. Experiments and analysis show that the system can accurately identify tiredness under a variety of circumstances, marking a promising advancement in driver aid technology. The creation of an integrated sleepiness detection system with the following characteristics is the primary objective of this paper.

A. Detect Drowsiness Accurately: Use image processing and machine learning to spot sleepiness indicators like head position changes and eye closing.

B. Real-Time Detection: Ensure the system can process video feeds from camera in real time .
C. Instant feedback: In drowsy driver detection, it provides real-time alerts to the driver when signs of drowsiness are detected, helping to prevent accidents. By using HAAR for face detection and CNN for analysis, the system can quickly identify fatigue and notify the driver immediately.
D. Enhance Road Safety: By reducing the likelihood of accidents brought on by sleepy driving, early identification and alerts can eventually lead to safer driving conditions. By achieving these objectives, the project aims to promote safer driving practices and enhance awareness of the dangers associated with drowsiness on the road.

II. LITERATURE SURVEY

Over the years, many methods have been developed to detect driver drowsiness, aiming to improve road safety. Each method has its advantages and limitations. Below is an overview of the most common approaches:
A. Vehicle Behaviour Monitoring: Some systems track how the vehicle is being driven, such as sudden lane changes, steering patterns, or abrupt braking. While these methods are easy to implement, they can be unreliable because they depend on road conditions and the driver's style, which vary greatly [2].
B. Physiological Monitoring: Advanced methods use sensors to track a driver's heart rate, brain activity (EEG), or other body signals. Although these techniques are very accurate, they require expensive equipment and are often uncomfortable for the driver, making them less practical [3].
C. Camera-Based Methods: With cameras, systems can analyze a driver's face and eyes to detect drowsiness. The Haar Cascade algorithm is a popular choice for detecting faces and eyes quickly. However, this approach struggles in poor lighting or when the background is complex [5].
D. Deep Learning Models: Convolutional Neural Network(CNN) , a type of deep learning techniques , are particularly good at identifying tiredness by examining ocular patterns like blinking rates or closed eyes [5]. Despite their great accuracy, CNNs can be costly because they require a lot of data for training and sophisticated hardware.
E. Combined Approaches: Recently, researchers have started combining traditional methods with modern deep learning techniques. For example, Haar Cascade is used for fast face and eye detection, while CNNs handle the detailed analysis. This combination makes it possible to achieve both speed and accuracy, making the system more suitable for real-time use[14].
In our study, we focus on this hybrid approach. By using Haar Cascade for detecting facial features and CNNs for classifying drowsy or alert states, we aim to create a reliable, real-time system that works effectively in different driving conditions.

III. HAAR ALGORITHM

The Haar Cascade algorithm is a computer vision technique used to quickly and efficiently detect objects like faces or eyes in an image. It works by scanning an image in smaller sections, called regions, to find patterns that match predefined features, such as the shape of a face or eyes [6]. The initial step in this study is to identify the driver's face and eyes from a video stream using the Haar algorithm.
This step is crucial because it allows us to focus only on the relevant areas (the eyes) for further analysis. The key advantage of Haar is its speed, as it quickly discards regions that don't match the pattern, making it suitable for real-time applications [6, 19].
Here's how it works in simple terms:

- Predefined EyePatterns**:** The algorithm uses pre-trained data to recognize common patterns in eye regions. This data is stored in what is called a "cascade classifier."
- Image Scanning**:** The algorithm looks at the video frame in multiple sections and scales (sizes). It starts with smaller areas and moves across the entire frame, checking if the current section matches the predefined eye pattern.
- Feature Matching: Haar uses simple rectangular patterns to analyze contrast differences, such as between the dark pupil of an eye and the lighter surrounding skin. If these patterns match enough, the algorithm decides that eyes are present in that region.
- Efficiency**:** One key advantage of Haar is its speed. By quickly discarding regions that don't match the pattern, it focuses only on promising areas, making it suitable for real-time applications.

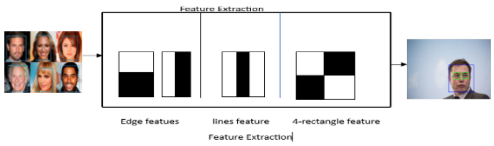

Fig.1. Haar feature extraction for object detection

In order to ascertain whether the driver is awake or sleepy, the eye regions of the driver's face are effectively recognized using Haar Cascade in Figure 1. These regions are subsequently sent to a Convolutional Neural Network (CNN) for in-depth analysis. Accuracy and speed are guaranteed by this combination, which is crucial for real time car sleepiness detection.

IV. CNN ALGORITHM

One kind of machine learning technique called a convolutional neural network (CNN) is made to evaluate images by finding features and patterns. It is widely used for tasks like recognizing objects or classifying images [20] because it mimics how the human brain processes visual information. In order to identify whether a motorist is awake or tired, our driver drowsiness detection system uses the CNN algorithm to examine the ocular areas of the driver, which were previously identified using the Haar Cascade technique.

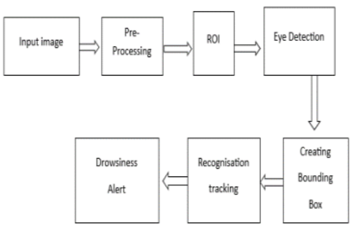

Fig. 2 Block diagram for drowsiness detection

Fig. 2 illustrates the process of drowsiness detection using a Convolutional Neural Network (CNN). Here's a short explanation of each step:

1. Input Image: The system takes an image (or video frame) as input.
2. Pre-processing: The input is processed to improve the quality and prepare it for further analysis.
3. ROI (Region of Interest): The region containing relevant features, such as the eyes, is extracted. Haar algorithm is used for this purpose.
4. Eye Detection: The system detects the eyes in the extracted region using CNN-based methods.

5. Creating Bounding Box: A bounding box is created around the detected eyes for further analysis.
6. Recognition Tracking: Tracks the movement or state (open/closed) of the eyes over time.
7. Drowsiness Alert: If the system detects drowsiness (e.g. prolonged eye closure), it generates an alert to notify the user.

This flow is designed to analyze eye behavior and issue alerts to prevent accidents caused by drowsiness.

V. PROPOSED FRAMEWORK

We train the suggested framework using 1456 photos from a Kaggle dataset. Three convolution layers make up the Convolutional Neural Network (CNN) model that is used to process these photos in order to extract information. Following features extraction, the most significant features are chosen using a pooling layer. After that, a fully connected layer. receives these features for additional processing. The training phase and the deployment phase, shown in Fig. 2, are the two stags of the suggested framework.

A. Training Phase:

The mode is trained using a CNN technique, which consists of three main layers: one or more hidden layers, an input layer, and an output layer.

The CNN version architecture includes the following layers (Fig 1):

- Convolutional layer: three kernel sizes, 32 nodes.
- Convolutional layer: kernel size 3, 64 nodes.
- Convolutional layer: kernel size three, 128 nodes.
- 256 nodes make up the fully connected layer.

The final layer consists of two nodes and is also fully connected.The output layer employs a SoftMax activation function, but the network as a whole uses ReLU activation.The steps used in this training phase are listed below.

- The images from the training data is passed to the Haar algorithm.
- The Haar Cascade technique is used to recognize faces.
- The eyes are then tracked using CNN (Convolutional Neural Network) layers to determine if they are open or closed.

Finally, CNN layers along with activation layers, i.e. ReLu and softmax classify whether the person is awake or drowsy based on the eye-tracking data.

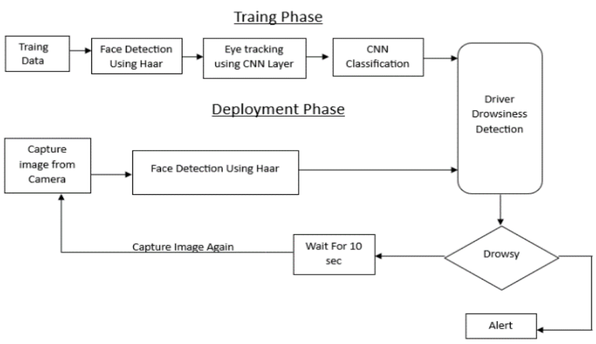

Fig. 3. Proposed System Architecture

B. Deployment Phase:

We can track the driver's level of awareness by using the trained model to detect faces and eyes on the webcam.The mechanism is passive unless drowsiness is detected, in which case an alert is generated. Below is a description of the step-by-step procedure.

- A camera captures a live image of the driver.
- The system detects the face using the Haar Cascade method.
- If the driver is drowsy, an alert is triggered to wake him up. If drowsiness is not detected, the system waits for 10 seconds and captures another image.

Once the model is skilled, device will begin taking pictures snap shots of driving force at the same time as riding. If for a given time percentage of program language period, the share of detected drowsy photographs whilst using is greater than a threshold percent of drowsiness then the driving force is observed drowsy and alarm rings.

VI. MODEL IMPLEMENTATION

A. Dataset:

The dataset typically consists of images and video frames of drivers' faces, specifically focusing on the eyes. The experimental dataset used in this study is from kaggle dataset consisting 726 pictures of closed eyes and 726 pictures of open eyes. The dataset might contain grayscale images to reduce complexity and focus on key features like eye shape and position. The images shown in Fig. 4 will be given to the detection system where CNN will classify them into one-of-a kind lessons- drowsy and not drowsy. This dataset will include both drowsy in addition to non-drowsy photos.

Fig. 4. Dataset Images for Drowsiness Detection

B. Data Preprocessing

a) Data Augmentation: Use strategies like flipping, scaling, and rotation to expands datasets and make them more resistant to overfitting.
b) Normalization**:** Standardize the images to a uniform size and pixel value range to facilitate CNN training.

C. Face Detection using Haar

Use Haar Cascade classifiers to detect faces in images before feeding them into the CNN. This will help focus the model on relevant features.

D. Training:

Convolutional neural network(CNN) model constructed with Tensorflow/keras Sequencial API.

We have used some layers like as:

- Conv2D Layers: Extract eye region (e.g., shapes indicating open or closed states) from the faces in images detected by Haar.
- Pooling Layers: Simplify the feature maps, reducing unnecessary details while retaining key patterns.
- Flatten & Dense Layers: classify if the driver is awake or sleepy by combining all learnt data.

E. Evaluation:

a) Epochs: To obtain a satisfactory accuracy of 98% for training accuracy and 96% for validation, the

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

model has been trained 50 times over 50 datasets.

b) Loss: Indicates how closely the model's predictions correspond to the actual labels. Better performance is indicated by a lower loss.
c) Accuracy: Indicates the proportion of accurate predictions the model made.
d) Validation Loss: This metric gauges performance on the validation set (unseen), much as training loss. information while training).
e) Validation Accuracy: The model’s ability to generalize to new data is indicated by it’s accuracy on the validation set .

F. Accuracy Graph:

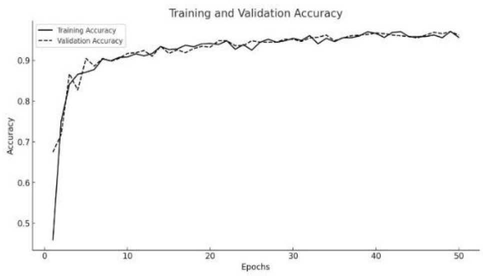

Fig.6. Graph of Training and Validation Accuracy

Training Accuracy: Solid line in Fig. 6 indicates how well the model is learning to classify drowsiness during training. The curve shows consistent improvement, stabilizing close to 100%
Validation Accuracy: The model’s performance on The model's performance on unknown data is shown by the dot line in Figure 6. Although it exhibits some oscillations, perhaps as a result of overfitting or differences in the validation data, it nearly tracks the training accuracy.

G. Loss Graph:

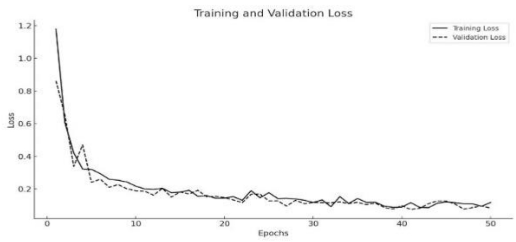

Fig.7.Graph of Traning and Validation Loss

Training Loss: Solid line in Fig. 7 demonstrates how well the model is minimizing the error during training. A steady decrease indicates that the model is learning effectively.
Validation Loss: Dotted line in Fig. 7 Shows the error on validation data. Although it decreases initially, spikes in the curve suggest moments where the model struggles to generalize.
The accuracy and loss performance metrics for a machine learning model during the training and validation stages are depicted in the two graphs in Figure 6 and Figure 7.
Training Accuracy:
Starting Accuracy (Epoch 1)**:** ~48.92%
Final Accuracy (Epoch 50)**:** ~97.48%
Validation Accuracy:
Starting Validation Accuracy (Epoch 1)**:** ~74.05%
Final Validation Accuracy (Epoch 50)**:** ~96.54%

Validation Loss:
Starting Loss (Epoch 1): ~1.12
Final Loss (Epoch 50): ~0.072
The model seems to be effectively learning, with a notable improvement in both training and validation accuracy, reaching an impressive final performance with ~97% accuracy on the training data and ~96.5% on the validation set.

VII. DISCUSSION

The proposed system offers the following features:
Drowsiness Detection: When the model detects that both eyes are closed for a sustained period (indicating potential drowsiness), the system triggers the alarm, which can be measured by how quickly the system reacts to prolonged eye closure. The score mechanism helps track the driver's alertness level over time, and when it exceeds the threshold (score > 15), the alarm goes off.

A. Limitations:

The following section discuss the limitations of the proposed system.

- Variations in lighting and angles affecting the accuracy of eye detection.
- Handling partial occlusions (e.g., if the eyes are partially covered by the eyelids or other objects).

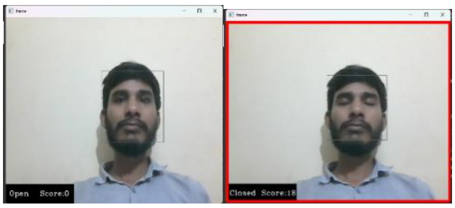

Fig.9.Captured-image with eyes open&close

Here Fig.9 Above image are capturing from the camera which detectig person are drowsy or not . If Score are are greater than 15 frames the model detects drowsy, if frames count is less than 15 model will ring the alarm.

VIII. CONCLUSION

In this study, we developed a system for detecting sleepiness by combining computer vision and machine learning techniques, specifically utilizing the
CNN algorithm and the Haar cascade classifier. We gathered a diverse dataset consisting of images and videos that showcase individuals exhibiting various levels of sleepiness. The Haar algorithm specializes in recognizing facial features, particularly the eyes and mouth. We implemented a series of Haar classifiers that are trained to detect distinct patterns within the visual data. This process allows us to extract relevant characteristics and categorize them into two groups: drowsy and not drowsy, resulting in a labeled dataset. For the model training phase, we employed TensorFlow and Keras to construct a deep neural network using the prepared dataset, which underwent validation on a separate validation set. We assessed the performance of the trained model using various metrics, including recall, accuracy, and precision. The final model is capable of classifying new images in real-time, such as those encountered while driving, providing audio alerts to the driver if signs of fatigue are detected.

REFERENCES

1. Amodio, A., Ermidoro, M., Maggi, D., Formentin, S., Savaresi, S.M. (2018). Automatic detection of driver impairment based on pupillary light reflex. IEEE Transactions on Intelligent Transportation Systems, pp. 1-11. https://doi.org/10.1109/tits.2018.2871262
2. Yang, J.H., Mao, Z.H., Tijerina, L., Pilutti, T., Coughlin, J.F., Feron, E. (2009). Detection of driver fatigue caused by sleep deprivation. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 39(4): 694-705. https://doi.org/10.1109/tsmca.2009.2018634
3. Hu, S., Zheng, G. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. Expert Systems with Applications, 36(4): 7651-7658. https://doi.org/10.1016/j.eswa.2008.09.030
4. Mardi, Z., Ashtiani, S.N., Mikaili, M. (2011). CNN-based drowsiness detection for safe driving using chaotic features and statistical tests. Journal of Medical Signals And Sensors, 1(2): 130-137. https://doi.org/10.4103/2228-7477.95297
5. Noori, S.M., Mikaeili, M. (2016). Driving drowsiness detection using fusion of electroencephalography, electrooculography, and driving quality signals. J Med Signals Sens, 6: 39-46. https://doi.org/10.4103/22287477.175868
6. Dang, K., Sharma, S. (2017). Review and comparison of face detection algorithms. 7th International Conference on Cloud Computing, Data Science & Engineering, pp. 629-633. https://doi.org/10.1109/confluence.2017.7943228
7. VenkataRamiReddy, C., Kishore, K.K., Bhattacharyya, D., Kim, T.H. (2014). Multi-feature fusion based facial expression classification using DLBP and DCT. Int J Softw Eng Appl, 8(9): 55-68. https://doi.org/10.14257/ijseia.2014.8.9.05
8. Krajewski, J., Sommer, D., Trutschel, U., Edwards, D., Golz, M. (2009). Steering wheel behavior based estimation of fatigue. The Fifth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 118-124. https://doi.org/10.17077/drivingassessment.1311
9. Danisman, T., Bilasco, I.M., Djeraba, C., Ihaddadene,
10. N. (2014). Drowsy driver detection system using eye blink patterns. 2010 International Conference on Machine and Web Intelligence, Algiers, Algeria, pp. 230-233. https://doi.org/10.1109/icmwi.2010.5648121
11. Abtahi, S., Hariri, B., Shirmohammadi, S. (2011). Driver drowsiness monitoring based on yawning detection. 2011 IEEE International Instrumentation and Measurement Technology Conference, Binjiang, pp. 1-4. https://doi.org/10.1109/imtc.2011.5944101
12. Dwivedi, K., Biswaranjan, K., Sethi, A. (2014). Drowsy driver detection using representation learning. Advance Computing Conference (IACC), IEEE, pp. 995-999. https://doi.org/10.1109/iadcc.2014.6779459
13. Alshaqaqi, B., Baquhaizel, A.S., Amine Ouis, M.E., Boumehed, M., Ouamri, A., Keche, M. (2013). Driver drowsiness detection system. 8 th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA). https://doi.org/10.1109/wosspa.2013.6602353
14. Park, S., Pan, F., Kang, S., Yoo, C.D. (2017). Driver drowsiness detection system based on feature representation learning using various deep networks. In: Chen CS., Lu J., Ma KK. (eds) Computer Vision – ACCV 2016 Workshops, Lecture Notes in Computer Science, pp. 154-164. https://doi.org/10.1007/978-3- 319-54526-4_12

15. Tadesse, E., Sheng, W., Liu, M. (2014). Driver drowsiness detection through HMM based dynamic modeling. 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, pp. 4003-4008. https://doi.org/10.1109/ICRA.2014.6907440
16. Said, S., AlKork, S., Beyrouthy, T., Hassan, M., Abdellatif, O., Abdraboo, M.F. (2018). Real time eye tracking and detection- a driving assistance system. Advances in Science, Technology and Engineering Systems Journal, 3(6): 446-454. https://doi.org/10.25046/aj030653
17. Picot, A., Charbonnier, S., Caplier, A. (2012). On-Line Detection of drowsiness using brain and visual information. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 42(3): 764- 775. https://doi.org/10.1109/tsmca.2011.2164242
18. Mandal, B., Li, L., Wang, G.S., Lin, J. (2017). Towards detection of bus driver fatigue based on robust visual analysis of eye state. IEEE Transactions on Intelligent Transportation Systems, 18(3): 545-557. https://doi.org/10.1109/tits.2016.2582900
19. Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., Jiang, S. (2018). Real-time driver drowsiness detection for android application using deep neural networks techniques. Procedia Computer Science, 130: 400-407. https://doi.org/10.1016/j.procs.2018.04.060
20. Viola, P., Jones, M.J. (2004). Robust real-time face detection. International Journal of Computer Vision, 57(2): 137-154. https://doi.org/10.1023/b:visi.0000013087.49260.fb
21. Jensen, O.H. (2008). Implementing the Viola-Jones face detection algorithm (Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark).