A PROJECT REPORT

on

Real-Time Drowsiness Detection Using Hybrid Computer Vision and Dlib with Deep Learning Classification

Final Review

Submitted for partial fulfilment of the requirements for the award

of

***Bachelor of Technology* by**

BATCH - 12

P.Yashwanth [21HR1A3040]

M.Hemagiri [21HR1A3028]

C.Narendra [21HR1A3011]

S.Vasantha Vani [21HR1A3047]

S.Muneera [21HR1A3051

Under the guidance of

Mr. K .Lokesh M.tech Ph.d

HOD of AI&DS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MOTHER THERESA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Melumoi (Post), Palamaner-517408.

(Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapuramu-515002) (NAAC Accredited & An ISO 9001:2015 Certified Institute) (Email: mtieat@gmail.com Website: www.mtieat.org)

Real-Time Drowsiness Detection Using Hybrid Computer Vision and Dlib with Deep Learning Classification

ABSTRACT

Nowadays, accidents occur during drowsy road trips and increase day by day; It is a known fact that many accidents occur due to driver fatigue and sometimes inattention, this research is primarily devoted to maximizing efforts to identify drowsiness. State of the driver under real driving conditions. The aim of driver drowsiness detection systems is to try to reduce these traffic accidents. The secondary data collected focuses on previous research on systems for detecting drowsiness and several methods have been used to detect drowsiness or inattentive driving.Our goal is to provide an interface where the program can automatically detect the driver's drowsiness and detect it in the event of an accident by using the image of a person captured by the webcam and examining how this information can be used to improve driving safety can be used. . a vehicle safety project that helps prevent accidents caused by the driver's sleep. Basically, you're collecting a human image from the webcam and exploring how that information could be used to improve driving safety. Collect images from the live webcam stream and apply machine learning algorithm to the image and recognize the drowsy driver or not.When the driver is sleepy, it plays the buzzer alarm and increases the buzzer sound. If the driver doesn't wake up, they'll send a text message and email to their family members about their situation. Hence, this utility goes beyond the problem of detecting drowsiness while driving. Eye extraction, face extraction with dlib.

Keywords:

Driver drowsiness detection, OpenCV, Haar Cascades,Dlib, Eye Aspect Ratio (EAR), Blink Rate, Real-time Object Detection.

EXISTING SYSTEM

Limitations of Traditional Machine Learning Models:

Machine learning algorithms like SVM and Random Forests classify features such as eye closure and yawning, but they rely on predefined features, limiting their effectiveness in diverse real-world conditions.

Challenges with Deep Learning Approaches:

Deep learning methods like CNNs improve feature extraction for facial landmarks, but they struggle with real-time performance on low-power devices. LSTMs excel at analyzing sequences, but require large amounts of training data to generalize effectively.

Issues with Hybrid ML-DL Systems:

Hybrid approaches combining ML and DL improve performance but struggle to balance accuracy and efficiency. They are also prone to errors due to dynamic changes like fluctuating lighting or different camera angles, reducing reliability.

Camera-based limitations:

- **Lighting conditions:** Poor lighting, glare, or darkness can affect accuracy.

False positives/negatives:

- Misidentification of normal behavior (e.g., yawning or blinking) as drowsiness can trigger unnecessary alerts.
- Drowsiness might go undetected if it does not show clear physical signs.

Data processing limitations:

Real-time processing demands high-performance hardware and optimized algorithms, which can be challenging to implement in low-end vehicles.

PROPOSED SYSTEM

Methodology Overview:

Input to Output Flow: The system captures live video of the driver's face using an webcam. Haar cascade classifiers and dlib's facial landmark detection identify facial regions. Key metrics, such as the Eye Aspect Ratio (EAR) and lip distance, are computed to classify the driver as active, drowsy, or yawning. Alerts are triggered if drowsiness or yawning is detected.

Modular Design:

Face Detection with Haar Cascade Classifiers: Rapidly identifies the driver's face in real-time, establishing a reliable Region of Interest (ROI) for further analysis.

Facial Landmark Detection with Dlib: Utilizes pre-trained models to detect 68 facial landmarks, enabling accurate tracking of eyes and mouth.

Metric Computation (EAR and Lip Distance): Computes EAR for eye closure detection and lip distance for yawning analysis. These serve as key indicators for drowsiness classification.

Classification Mechanism: Classifies driver states (active, drowsy, yawning) based on thresholds for EAR and lip distance, ensuring accurate classification under real-time conditions.

Real-Time Alerts and Notifications: When thresholds are crossed for EAR or lip distance, the system emits audible warnings. Persistent alerts trigger SMS notifications via Twilio to designated contacts.

Adaptability and Scalability: Ensures robust detection under varying conditions (e.g., lighting, head orientation) and provides modular flexibility for adding features like head pose estimation.

EYE-TRACKING

The eye condition is open or close; it most crucial aspect that defines drivers weariness. The eyelid muscles tend in a state of drowsiness to speed the way to sleep. Viola-Jones monitors the location of the drivers eyes. Both eyes are then separated using the identification of the edge, and the eyes focus is resolved following the symmetrical properties of the eye. Finally, the analysis is well known. In the event of the opening of the eyes, the usual state of vigilance is handled at this stage. If the eye is turned off, so the failure of the eye is treated as a precaution.

YAWNING DETECTION

Another distinctive sign of fatigue in the driving area is the gauntlet, which appears due to the bodys re exes when a person is exhausted and nods. Suppose Viola-Jones is found in the mouth districts. Only the middle region is separated by K and accompanied by the coordination of relationship coef cients. K denotes that the items in K no of essentially unrelated classes are sections, so protests in each bunch are closest and farthest from artifacts in separate groups. Each of the K groups center or focal points is identi ed. K-capacity implies that the K-means are clustered through an iterative calculation that distinguishes objects within each bunch.

HARDWARE REQUIRMENTS

1.**USB/Webcam Compatibility:** USB 2.0/3.0 or compatible with processing unit (laptop/PC)

2.**Processing Unit:**

Type: Desktop or Laptop

RAM: 8GB or higher (for smooth processing, especially for deep learning models)

CPU: Intel Core i5 or equivalent, preferably i7 or Ryzen 7 for better performance

3.**Storage:**

Minimum: HDD 250GB / SSD 128GB

Recommended: SSD 512GB+ for faster data processing

SOFTWARE REQUIRMENTS

1. **Libraries/Frameworks:**

   Open CV: For real-time computer vision, face detection, and landmark extraction.

   Dlib: For detecting facial landmarks and eye status.

   TensorFlow/PyTorch/Keras: Deep learning techniques for drowsiness classification.

   NumPy: For data manipulation and analysis.

   Pyaudio : For generating audio alerts.

2. **Operating System:**

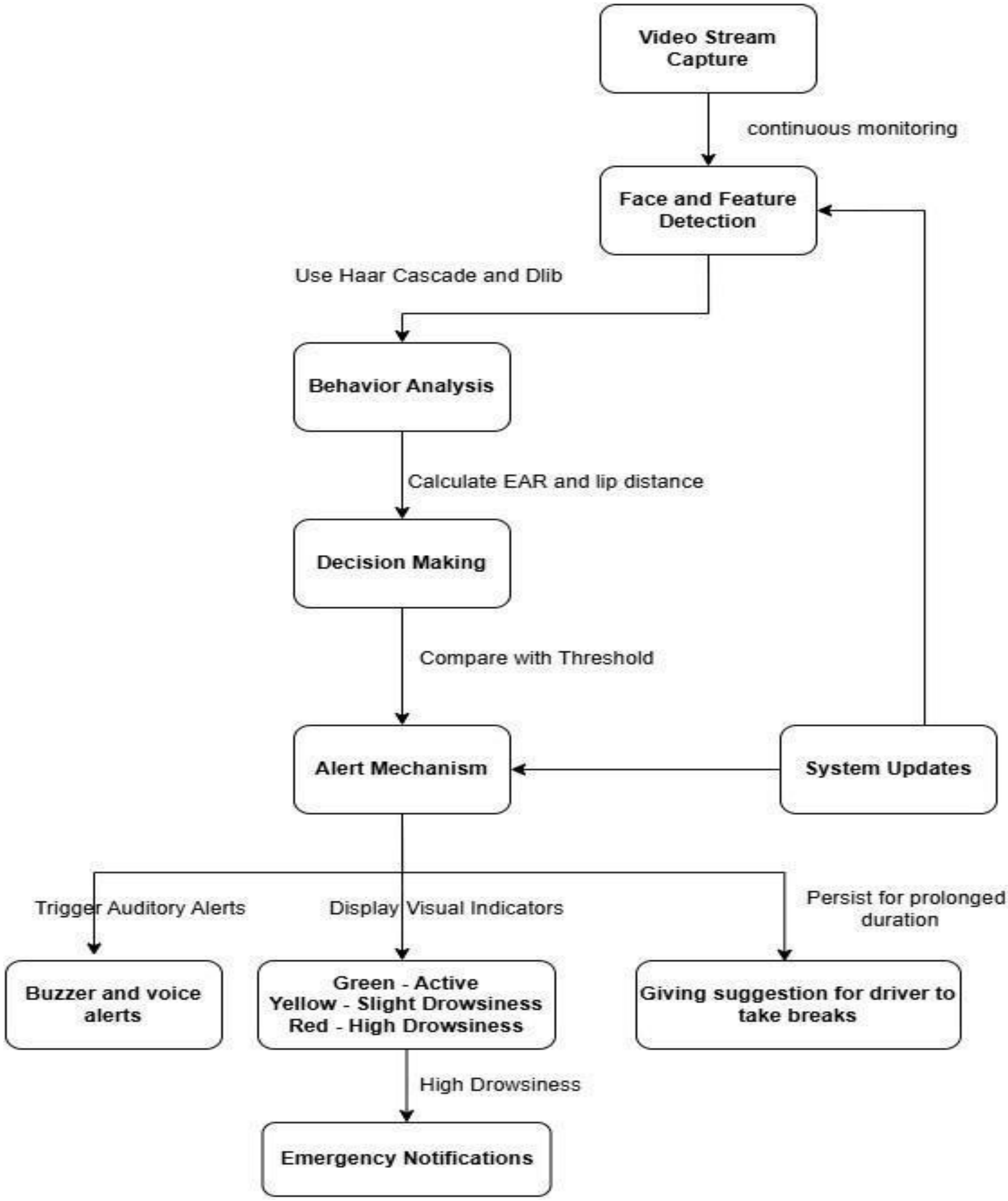
   Windows/Linux/Mac OS

3. **IDE/Development Environment:**

   VS Code / PyCharm: IDEs for Python development.

   Jupyter Notebooks: For prototyping and experimenting with machine learning models.

ARCHITECTURE



Fig(1): Architecture of Real-Time Drowsiness Detection Using Hybrid Computer Vision and ML Techniques

MODULE

1. **Data Collection Module:**

   Gather real-time video feed from a camera focusing on the driver's face.

   Capture facial landmarks required for drowsiness detection (eyes, mouth, head movement).

   Store collected frames in a structured dataset for preprocessing and analysis.

2. **Preprocessing Module:**

   - Face Detection: Identify and isolate the driver's face from the video feed.
   - Image Normalization: Resize and standardize images for consistent input.
   - Noise Reduction: Apply filters to remove unwanted variations in lighting and background.
   - Convert images into a usable format for feature extraction.

3. **Processing & Computation Module:** Face Tracking: Continuously track facial features in realtime.

   Feature Extraction:

   Eye Blink Rate: Measure blink duration and frequency.

   Yawning Detection: Identify yawning through mouth landmarks.

   Drowsiness Classification Algorithm:

   Analyze features to classify driver state as alert, mildly drowsy, or highly drowsy.

   Set threshold values for drowsiness levels.

4. **Alert Mechanism Module:**

   Audio Alerts: A loud buzzer or voice warning.

   Visual Alert: A warning message displayed on the screen.

   SMS Alert: An automated SMS notification sent to peers with a message.

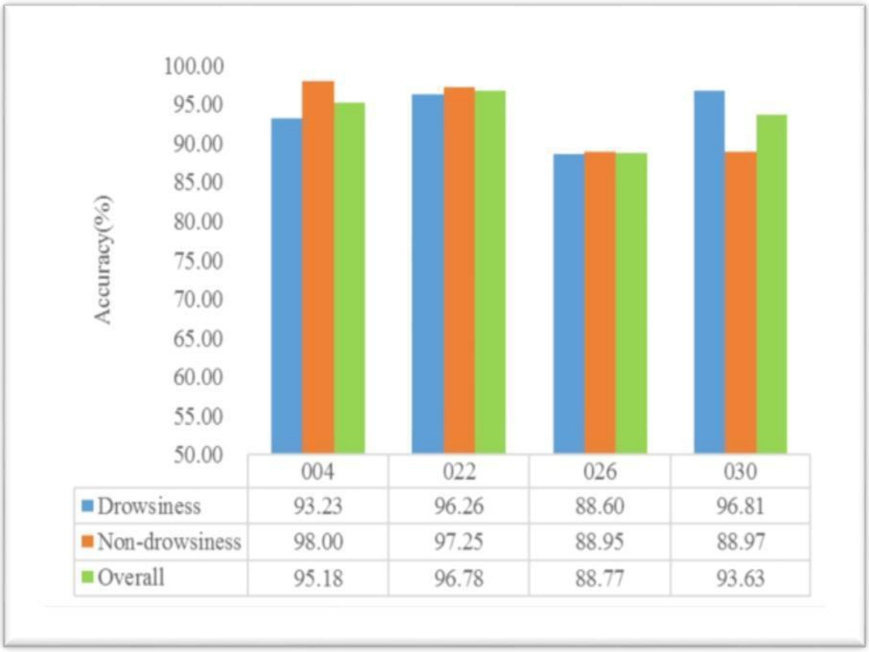
5. **Testing & Evaluation Module:**

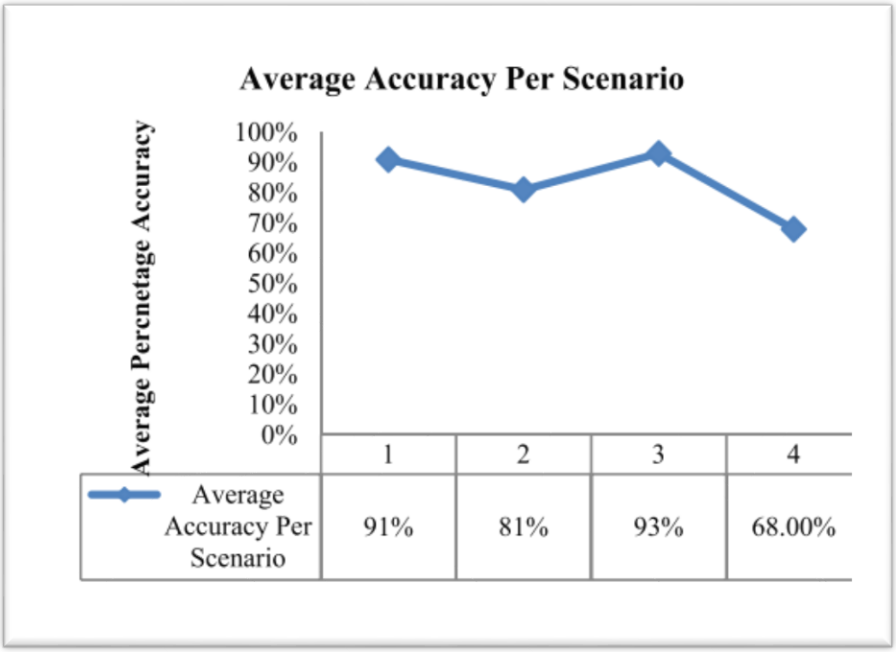
   - Evaluate accuracy of face detection and tracking algorithms.
   - Test drowsiness detection under different lighting and facial angles.
   - Debug false positives and false negatives in the alert system.

6. **Deployment Module:**

   - Deploy the model on a local system or integrate with an embedded system in vehicles.

Experimentation Results

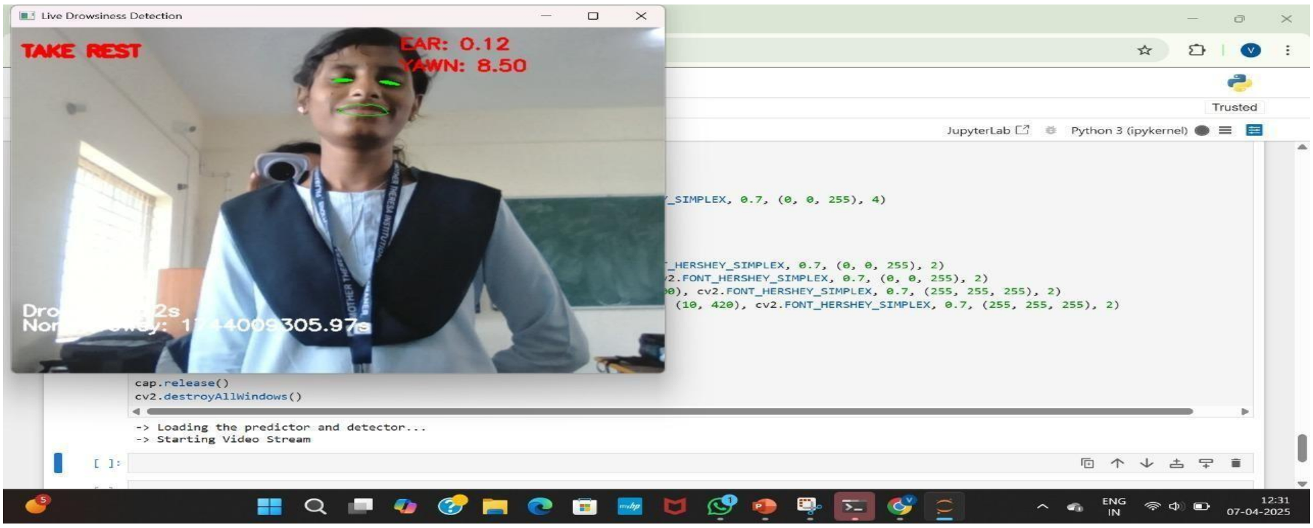




Sample coding

```
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2

EYE_AR_THRESH = 0.2
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 30
alarm_status = False
alarm_status2 = False
alarm_status3=False
danger_alarm_status = False
COUNTER = 0
yellow_border_duration = 3  # Duration to show yellow border (in seconds)
yellow_border_time = None  # Time to start showing yellow border
drowsy_start_time = None  # Time when drowsiness started
drowsy_time = 0  # Total drowsy time in seconds
non_drowsy_time = 0  # Total non-drowsy time in seconds

print("-> Loading the predictor and detector...")
detector = cv2.CascadeClassifier(r"C:\Users\Project\Desktop\PPREOJECTS\DROWSINESS_AND_YAWN_DETECTION\haarcascade_frontalface_default.xml")
predictor = dlib.shape_predictor(r"C:\Users\Project\Desktop\PPREOJECTS\DROWSINESS_AND_YAWN_DETECTION\shape_predictor_68_face_landmarks.dat")
print("-> Starting Video Stream")
cap = cv2.VideoCapture(0)

def play_alarm():
    alarm_path = r"C:\Users\Project\Desktop\PPREOJECTS\DROWSINESS_AND_YAWN_DETECTION\alarm.wav"
    winsound.PlaySound(alarm_path, winsound.SND_FILENAME | winsound.SND_ASYNC)

def play_danger_alarm():
    alarm_path = r"C:\Users\Project\Desktop\PPREOJECTS\DROWSINESS_AND_YAWN_DETECTION\Danger.wav"
    winsound.PlaySound(alarm_path, winsound.SND_FILENAME | winsound.SND_ASYNC)

def play_yawn_alaram():
    alarm_path = r"C:\Users\Project\Desktop\PPREOJECTS\DROWSINESS_AND_YAWN_DETECTION\yawn_sound.wav"
    winsound.PlaySound(alarm_path, winsound.SND_FILENAME | winsound.SND_ASYNC)
```

Sample Output



References

1. Turki, O. Kahouli, S. Albadran, M. Ksantini, A. Aloui, and M. B. Amara, “A sophisticated drowsiness detection system via deep transfer learning for real time scenarios,” AIMS Math., vol. 9, no. 2, pp. 3211–3234, 2024.

2. L. Yang, H. Yang, H. Wei, Z. Hu, and C. Lv, “Video-based driver drowsiness detection with optimised utilization of key facial features,” IEEE Trans. Intell. Transp. Syst., early access, Jan. 5, 2024, doi: 10.1109/TITS.2023.3346054.

3. J. Singh, R. Kanojia, R. Singh, R. Bansal, and S. Bansal, “Driver drowsiness detection system: An approach by machine learning application,” 2023, arXiv:2303.06310.

4. B. Bhowmick and C. Kumar, "Detection and classification of Eye State in IR Camera for Driver Drowsiness Identification", in Proceedings of the IEEE International Conference on Signal and Image Processing Applications, 2009.

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

Signature of Guide