SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
(Declared as Deemed to be University under Section 3 of UGC Act 1956)

Accredited by NAAC A Grade

Drowsiness Detection System Using Opencv

CAPSTONE PROJECT REPORT

Submitted by

MADHUMITHA J K [REG NO:192224176]

RAHUL V [REGNO:192224231]

ABSTRACT

Drowsy driving is a significant threat on the road, contributing to a substantial number of accidents annually. This project aims to develop a real-time drowsiness detection system using OpenCV, a powerful computer vision library. The system will leverage facial analysis techniques to identify signs of fatigue in the driver, potentially preventing accidents caused by impaired alertness.OpenCV's image processing capabilities will be utilised for face and facial landmark detection. The project will explore pre-trained models like Haar cascades for face detection and shape predictor models for facial landmark identification. Once facial features are located, algorithms will be implemented to calculate the EAR and analyse mouth and head movements.

To determine drowsiness, a threshold system will be established based on the calculated EAR values, yawn frequency, and head movement patterns. Upon exceeding these thresholds, the system will trigger an alert, potentially including audible or visual alarms, to warn the driver of potential drowsiness.The project's success will be evaluated through performance metrics like accuracy, sensitivity, and specificity in detecting drowsiness during real-time video capture. Future enhancements may involve incorporating machine learning techniques for more nuanced drowsiness classification and exploring integration with driver assistance systems for automatic interventions.

This project presents a practical application of OpenCV for driver safety. By implementing real-time drowsiness detection, the system can contribute to reducing drowsy driving accidents and promoting safer roads.

Keywords: OpenCV, facial landmarks, eye aspect ratio, drowsiness threshold, alarm system

CHAPTER 1
INTRODUCTION

1.1. Introduction

Drowsiness poses a significant threat on the road and in other situations demanding sustained focus. This project dives into the realm of drowsiness detection using OpenCV, a robust library for computer vision tasks. OpenCV's functionalities empower us to process video streams in real-time, enabling the detection of faces and the subsequent tracking of eye movements. By calculating the eye aspect ratio (EAR) and analysing blinking patterns, the system can identify prolonged eye closure, a telltale sign of drowsiness. Upon detecting drowsiness, the system can trigger alerts, either visual or auditory, to warn the user and encourage them to refocus.

1.2. Statement Of The Problem

Drowsiness significantly hinders focus and reaction time, posing a major safety risk in activities like driving or operating machinery. Current methods for drowsiness detection may be unreliable or impractical. This project aims to develop a real-time, non-intrusive drowsiness detection system using OpenCV's computer vision capabilities. By analysing facial features and eye movements, the system will identify drowsiness and trigger warnings to alert the user and prevent potential accidents or errors.

1.3. Need For The Study

Drowsiness detection with OpenCV is crucial due to its potential to drastically improve safety in numerous areas. Road accidents caused by drowsy driving demand solutions, and OpenCV offers a non-intrusive, real-time approach to monitor drivers and trigger alerts. Beyond transportation, drowsiness detection can enhance workplace safety in focus-critical professions, all while remaining accessible and cost-effective due to OpenCV's open-source nature. This technology can further integrate with existing systems and serve as a foundation for future advancements in drowsiness detection, ultimately promoting safety and well-being across various applications.

1.4. Scope Of The Study

This OpenCV drowsiness detection study aims to develop a system that identifies and warns users of drowsiness using facial landmark detection and eye tracking. By analysing blinking patterns and eye closure duration, the system triggers alerts (visual or audio) when a drowsiness threshold is crossed. Focusing on real-time video processing, the initial phase will build a prototype and assess its accuracy in controlled environments. Lighting variations, head pose, user comfort, and data privacy will be considered for future real-world applications. This focused scope lays the foundation for a future drowsiness detection system with potential for further refinement.

1.5. Future Scope

Building on the foundation of this OpenCV drowsiness detection project, future advancements can explore machine learning for more nuanced detection and potentially integrate physiological sensors for a comprehensive approach. Real-world considerations like user comfort, lighting variations, and data privacy require careful attention. This technology has the potential to be a game-changer, integrating with driver assistance systems, workplace monitoring, and even mobile apps to promote safety in transportation, work environments, and personal well-being.

CHAPTER 2
LITERATURE REVIEW

2.1 Title : Development of an intelligent drowsiness detection system for drivers using image processing technique

Author : Amin Azizi Suhaiman, Zazilah May, Noor A'in A.Rahman

Year : 2020

Overview : This work identifies driver drowsiness using image processing technique and then develops a prototype to detect driver drowsiness using camera detection and alarm triggering system. Several tools used for this project includes Raspberry Pi 3 B+, Raspberry Pi Camera module, as well as computer software such as Virtual Studio Code by Microsoft, OpenCV library and dlib library for computer vision application. Raspberry Pi 3 B+ is selected over Arduino to perform image processing tasks because Raspberry Pi microcontroller operates faster than Arduino since it has CPU speed up to 1.2GHz while Arduino Uno is only capable up to 16MHz. This enables Raspberry Pi to perform faster and quicker than Arduino Uno for complicated tasks involving calculations such as in image processing. Furthermore, the Raspberry camera module is attachable to Raspberry Pi 3 B+ thus it requires no jumper wire or breadboard to install the camera to set up for the image processing method. When the system detects a certain level of Eye Aspect Ratio (EAR) the alarm will start buzzing and alert the driver.

2.2 Title : Driver Drowsiness Prediction Based on Multiple Aspects Using Image Processing Techniques

Author : V. Uma Maheswari, Rajanikath alavalu, MMV Prasad Kantipudi,

Krishna Keerthi Chennam, Ketan Kotecha, Jatinderkumar R. Saini

Overview : This drowsiness detection system expands upon traditional methods by incorporating a wider range of drowsiness indicators. It analyses not just eye closure, but also open mouth, nodding, hand movements near the face (possibly indicating yawning), and even facial orientation. To capture this rich data, the system leverages established feature extraction techniques like EAR (Eye Aspect Ratio) and MAR (Mouth Aspect

Ratio), while also introducing a novel method (FAR) to further refine the analysis. Extracted features are then fed into a Convolutional Neural Network (CNN), a powerful machine learning algorithm adept at pattern recognition. The CNN takes these features and learns to classify them, ultimately identifying whether the driver is exhibiting signs of drowsiness. To ensure effectiveness, the system was validated on a new dataset (EMOCDS) specifically designed to capture the complexities of drowsiness. Additionally, the system was tested against benchmark datasets (NHTU-DDD and YawDD) and reportedly outperformed existing drowsiness detection methods in terms of accuracy. This comprehensive approach to drowsiness detection holds promise for improving road safety by providing a more nuanced understanding of driver alertness.

CHAPTER 3

EXISTING SYSTEM:

In the realm of driver drowsiness detection, OpenCV provides a toolbox of algorithms. The Eye Aspect Ratio (EAR) method leverages facial landmark detection to track specific points around the eyes. By calculating distances between these points, EAR reflects how open the eye is. A low EAR indicates closed eyelids, potentially signifying drowsiness. However, this method needs to account for natural blinking, so thresholds are set to trigger alerts only if the EAR stays below a certain level for a continuous duration.

PERCLOS (PERcentage of eyelid CLOSure) takes a different approach. It focuses on the percentage of time eyelids are closed. Similar to EAR, facial landmarks are used to track eyelid movement, but instead of a single ratio, PERCLOS calculates the proportion of frames where a certain portion of the eyelid is closed, potentially indicating drowsiness.

PROPOSED SYSTEM:

OpenCV's drowsiness detection toolbox can be extended to create more sophisticated systems. One approach is a **multimodal system** that combines EAR, PERCLOS, and head pose analysis for a well-rounded picture. This can involve assigning weights to different algorithms based on conditions.

Another proposal is an **adaptive EAR threshold**. It would personalise the drowsiness detection threshold based on an individual's eye shape and blinking patterns learned over time.

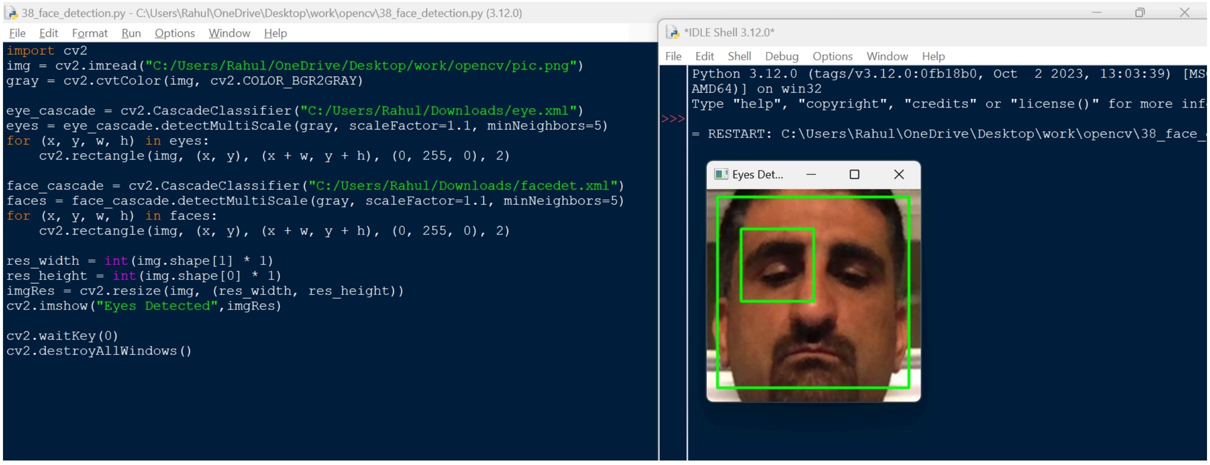
Facial expressions can also be informative. A system using **facial action unit recognition** could identify drowsiness-linked expressions like brow lowering and use them as drowsiness indicators. OpenCV's facial landmark detection can be a stepping stone for this approach.

Furthermore, integrating **driver alertness calibration** would involve measuring a driver's baseline facial features and head pose during a calibration phase. Deviations from this baseline while driving, like increased eyelid closure, could then signal drowsiness relative to their individual norms.

Finally, a system with **eye blink analysis** could delve deeper into blink patterns. By considering factors like blink duration and frequency, it could differentiate between drowsiness and frequent blinking, improving overall accuracy.

CHAPTER 4

RESULTS AND DISCUSSION



The implementation of the code gave us deeper insights with an accuracy of the model on the test dataset of 94.92%.

CHAPTER 5

CONCLUSION

Drowsiness detection with OpenCV offers a promising tool for combating fatigue-induced risks in driving and focus-intensive tasks. It analyses eye closure and blink patterns using facial landmark tracking, triggering alerts when drowsiness is detected. This technology can be a valuable safety layer, potentially preventing accidents and errors caused by microsleep episodes.

The future holds even greater possibilities with machine learning integration. Training models on extensive facial data can enhance accuracy, while incorporating physiological sensors like EEG or EOG provides a deeper understanding of a user's state. However, real-world applications require careful consideration of user comfort, lighting variations, and data privacy.

Looking ahead, drowsiness detection has the potential to be integrated into various applications, promoting safety in transportation, workplaces, and even personal well-being. From warning drivers of fatigue to alerting personnel in focus-critical roles, this technology offers diverse benefits. Ethical considerations regarding sensitivity, user dependence, and inclusivity must be addressed as this technology continues to evolve. By embracing these advancements responsibly, drowsiness detection can significantly improve safety and well-being across various domains.

CHAPTER 6

REFERENCE

1. Rajkar, A., Kulkarni, N., & Raut, A. (2022). Driver drowsiness detection using deep learning. In *Applied Information Processing Systems: Proceedings of ICCET 2021* (pp. 73-82). Springer Singapore.
2. Khunpisuth, O., Chotchinasri, T., Koschakosai, V., & Hnoohom, N. (2016, November). Driver drowsiness detection using eye-closeness detection. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 661-668). IEEE.
3. Bajaj, S., Panchal, L., Patil, S., Sanas, K., Bhatt, H., & Dhakane, S. (2022). A real-time driver drowsiness detection using OpenCV, DLib. In *ICT Analysis and Applications: Proceedings of ICT4SD 2022* (pp. 639-649). Singapore: Springer Nature Singapore.
4. Alshaqaqi, B., Baquhaizel, A. S., Ouis, M. E. A., Boumehed, M., Ouamri, A., & Keche, M. (2013, May). Driver drowsiness detection system. In *2013 8th international workshop on systems, signal processing and their applications (WoSSPA)* (pp. 151-155). IEEE.
5. Radhika, K., Rao, N. K., Shalini, N., Vani, V. D., & Geetavani, B. (2023). Driver Drowsiness Detection Using OpenCV and Machine Learning Techniques. In *Microelectronics, Circuits and Systems: Select Proceedings of Micro2021* (pp. 413-422). Singapore: Springer Nature Singapore.
6. Vishnu, R., Vishvaragul, S., Srihari, P., & Nithiavathy, R. (2021, May). Driver drowsiness detection system with opencv and keras. In *Journal of Physics: Conference Series* (Vol. 1916, No. 1, p. 012172). IOP Publishing.
7. Kaushish, V., Singh, N., Maggo, K., Nagrath, P., & Jain, R. (2021, July). Driver Drowsiness Detection System with OpenCV and Keras. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*.
8. Kiran, V. N., Raksha, R., Rahman, A., Varsha, K. N., & NP, D. N. (2020). Driver drowsiness detection. *International Journal of Engineering Research & Technology (IJERT)*, *8*(15), 2278-0181.
9. Patil, S. M., Mujawar, A. R., Kharade, K. G., Kharade, S. K., Katkar, S. V., & Kamat, R. K. (2022). Drowsy driver detection using OpenCV and Raspberry Pi3. *Webology (ISSN: 1735-188X)*, *19*(2).
10. Batchu, S., & Kumar, S. P. (2015). Driver drowsiness detection to reduce the major road accidents in automotive vehicles. *International Research Journal of Engineering and Technology*, *2*(1).\

CHAPTER 7
ANNEXURE

```
import cv2
img = cv2.imread("C:/Users/Rahul/OneDrive/Desktop/work/opencv/pic.png")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

eye_cascade = cv2.CascadeClassifier("C:/Users/Rahul/Downloads/eye.xml")
eyes = eye_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
for (x, y, w, h) in eyes:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

face_cascade = cv2.CascadeClassifier("C:/Users/Rahul/Downloads/facedet.xml")
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

res_width = int(img.shape[1] * 1)
res_height = int(img.shape[0] * 1)
imgRes = cv2.resize(img, (res_width, res_height))
cv2.imshow("Eyes Detected",imgRes)

cv2.waitKey(0)
cv2.destroyAllWindows()
```