DRIVER’S DROWSINESS DETECTION SYSTEM

SUDHARSAN S : Sudharsan.ec22@bitsathy.ac.in , RITHIGA V: rithiga.ec22@bitsathy.ac.in

Bannari Amman institute of technology, Anna University, Erode, 638401

Abstract—

Driver Drowsiness Detection is an automotive safety technology that helps save a driver's life by preventing accidents when the driver is drowsy. The main goal is to create a system that can identify driver fatigue through continuous retinal monitoring. The device works even if the driver is wearing glasses and in different light situations, to trigger an alarm or a buzzer that alerts the driver when drowsiness is detected. Vehicle speed can be reduce

Introduction :

Falling asleep while driving can easily lead to car crashes and accidents. Many people die each year in car crashes due to alcohol, such as insomnia, poisoning, drug and alcohol abuse, hypertension, or drowsy driving. Automobile manufacturers such as Tesla and Mercedes-Benz offer various driver assistance systems such as lane departure warning, emergency braking, W,0 help drivers avoid accidents. Samsung controls the driver's vision level by reading faces and patterns. At the same time, most of these technologies are proprietary and limited to high-end vehicles. This cognitive process of sleep^{00}qcan be classified in some respects according to the car's history, personality and body .Various methods have been developed in the past to define fatigue. According to the car, the sleep recognition system is used to monitor lane changes, steering turns, acceleration, and pressing the acceleration pedal. These methods include measuring the physical strength of the driver, measuring and recording the performance of the vehicle behavior. Among these technologies, the bio signal measurement method shows the highest potential for detecting drowsy drivers: unlike other methods, it depends only on the driver's state. There should be a camera that can identify fatigue based on features such as eyes closed, yawning and head movement. Another step in the process of recognizing physical fatigue is monitoring fatigue to correlate fatigue with physical signs such as EOG (electrooculogram) and ECG (electrocardiogram). A limitation of seizure detection using physiological methods is that the diver must have electrodes attached to his body. Information about car seizure has limitations such that it is affected by the force affecting the driver, the car, and the road. There are various methods described in different documents, each with some limitations and some benefits.

A drowsiness detection system using YOLO and PyTorch could revolutionize road safety by monitoring driver conditions in real time. This system works by using a camera to continuously photograph the driver's face and assess the driver's level of alertness. Using YOLO's object recognition capabilities, the system can accurately identify the driver's eyes and facial features and accurately analyze key metrics such as eyelid closure, head position, and blink frequency. YOLO is an object detection algorithm that divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself. YOLO is one of the most famous object detection algorithms due to its speed and accuracy. It achieves high accuracy while also being able to run in real-time. Its open source YOLO model can also handle detecting smaller objects and far away objects.

Driver fatigue is a serious and dangerous problem affecting road safety worldwide. Fatigue-related accidents have a huge impact; It costs countless lives and causes significant economic losses every year. The prevalence of long and monotonous journeys and the pressure of work and daily life increase the risk of sleepless driving. Technological solutions are very important to reduce this risk and increase road safety. This article presents an advanced method for detecting drowsy drivers using computer vision technology - the You Only See One (YOLO) object detection model used from PyTorch and integrated with OpenCV.

In recent years, the combination of artificial intelligence, deep learning and computer vision has made it possible to create powerful systems for a variety of applications, including object detection, facial recognition and behavioral analysis. These technologies have also taken their place in the field of vehicle safety, where fatigue detection is important. The current way it detects driver fatigue often depends on sensors such as steering wheel sensors, eye trackers or wearable devices; these have limitations such as invasiveness, cost, and reliability. In contrast, computer vision-based systems offer non-intrusive, cost-effective and excellent solutions for instant monitoring of driver behavior. The core of our planning process is the integration of YOLO, PyTorch and OpenCV. YOLO is known for its performance in detecting real objects and its ability to identify objects in a single pass over the network. By using PyTorch to implement YOLO, we leverage the power of various deep learning methods to obtain powerful and advanced training models. Integration with OpenCV continues to simplify real-time animation, allowing us to further analyze driving behavior.

LITERATURE REVIEW

Driver drowsiness is a critical issue that affects road safety and has been the subject of extensive research. In this section, we review existing literature related to driver drowsiness detection, encompassing both traditional and computer vision-based methods.

Traditional Driver Drowsiness Detection Methods

Traditional approaches to driver drowsiness detection have historically relied on non-computer vision techniques and sensors, such as steering wheel sensors, eyelid closure monitoring, or head-position sensors. These methods have made valuable contributions to understanding drowsiness but have notable limitations:

Intrusiveness: Many traditional methods require the driver to wear uncomfortable and intrusive devices, which can affect their driving experience and acceptance of the technology.

Cost: The cost of specialized hardware and sensors can be a barrier to widespread adoption.

Limited Accuracy: Traditional methods often lack precision and may generate false alarms or miss subtle signs of drowsiness.

Non-Real-Time Analysis: Some traditional approaches involve post-analysis rather than real-time detection, which can delay intervention.

Computer Vision-Based Approaches

The advent of computer vision and deep learning has brought about a paradigm shift in drowsiness detection. Computer vision-based approaches are non-intrusive, cost-effective, and capable of real-time analysis. This section focuses on the relevance of computer vision and the emergence of YOLO, PyTorch, and OpenCV in the context of driver drowsiness detection.

Facial Feature Analysis: Computer vision methods have harnessed the power of facial feature analysis, including eye tracking, to detect signs of drowsiness. These methods can monitor parameters like blink rate, eye closure, and head pose, which are reliable indicators of driver alertness.

Deep Learning: Deep learning models, particularly convolutional neural networks (CNNs), have shown exceptional performance in object detection and facial feature recognition. YOLO, in particular, stands out for its ability to identify objects in real-time, making it well-suited for detecting facial landmarks and monitoring driver behavior.

PyTorch and OpenCV Integration: PyTorch, as a versatile deep learning framework, has gained popularity for its ease of use and robust capabilities. When integrated with OpenCV, a powerful computer vision library, it offers an efficient platform for real-time video processing, making it suitable for drowsiness detection in a vehicle.

Recent Advancements: Recent research in computer vision-based driver drowsiness detection has demonstrated the promise of such systems. These solutions show high accuracy in detecting drowsiness indicators such as closed eyes, head nodding, or prolonged blinking. The integration of YOLO, PyTorch, and OpenCV represents a significant advancement in the field, enabling real-time, non-intrusive, and highly accurate driver drowsiness detection.

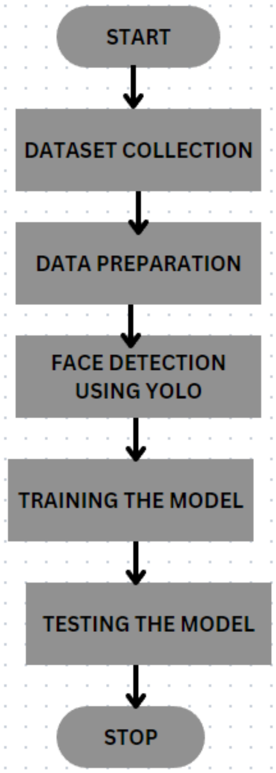
Significance of Real-Time Detection

Real-time detection is paramount in driver drowsiness detection, as it allows for immediate intervention to prevent accidents. The ability to raise alarms and initiate corrective measures when sleep is detected can improve road safety. In conclusion, while traditional methods help us understand driver sleep, computer vision-based ideas, especially those running on YOLO, PyTorch and OpenCV, provide a more promising and practical way to find a way to understand momentary sleepiness. . This article builds on the foundation provided by these advances and highlights the effectiveness of these systems in improving road safety.

METHODOLOGY:

1. Setting up the environment:

It is important to set up the development environment before you start developing the sleep quest. This includes developing the necessary libraries and frameworks. PyTorch is a deep learning program that will form the basis of our system. OpenCV is an open source computer vision program essential for image and video processing. In addition, before you can display the product, you must receive training on the YOLO model and weight, especially on YOLOv4 or YOLOv8



2. Data collection: All machine learning needs data, and the search for sleep is no exception. Collecting graphs is necessary for training and evaluating the system. This information should include both sleep and non-sleep conditions. You'll also collect data by tagging each photo with a tag indicating whether the person in the photo is asleep or not.

3. Preliminary data: Raw data that is not preprocessed is rarely used for machine learning. In this step you will resize and normalize the image collection. Resizing ensures that all images have the same size, making them compatible with neural network architectures.

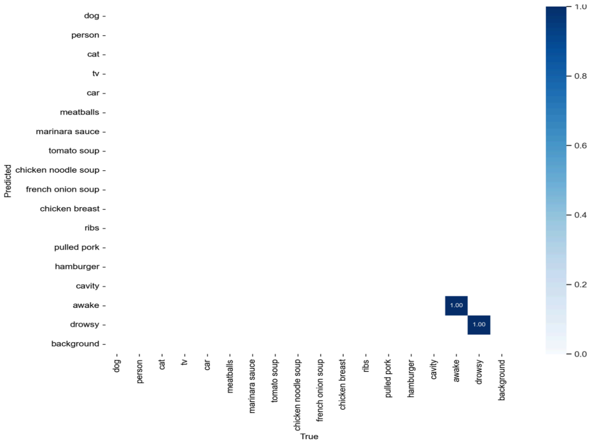
Normalization normalizes pixel values, which can improve the training process. The dataset is divided into training set, validation set and testing set to train the model, improve the model and evaluate its performance.

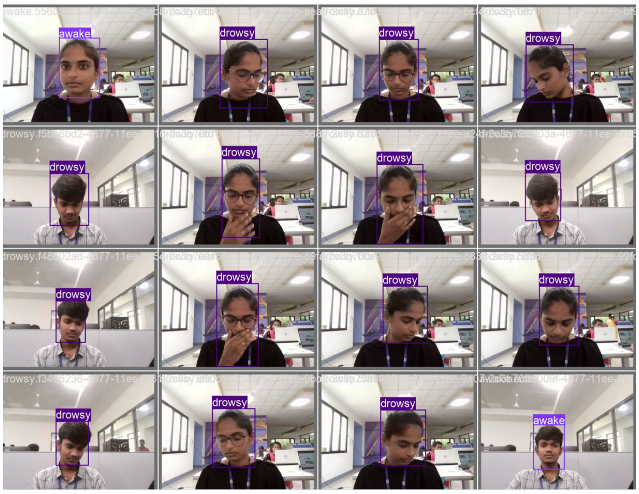
4. Face detection: One of the main tools of sleep detection is the ability to find faces in photos or videos. The YOLO algorithm, known for its ability to instantly detect objects, can be used for this purpose. YOLO can detect and recognize faces in high-quality images. The output will contain a bounding box around the visible face, which is important for the next step.

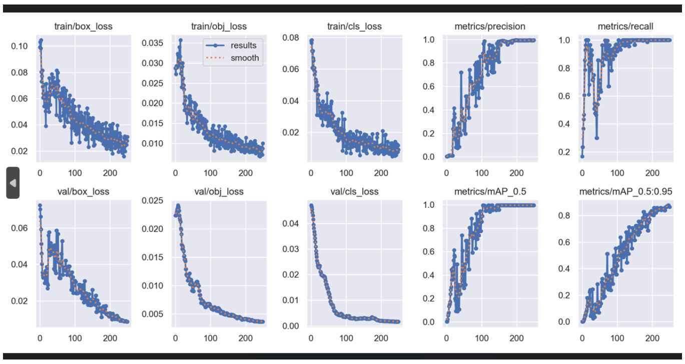
5. Eye detection: Once the face is detected, the next step is to detect the eyes in the visible area with the bounding box. Eyes are important in measuring sleep. Similar to face detection, you can use YOLO or similar to do eye detection. Extract bounding boxes to detect eyes in each region of the face.

6. Real-time application: The application of fatigue detection techniques is real-time. In this step, video frames are usually captured in time on the webcam or other camera sources. Each frame is made using YOLO-based face and eye capture. The EAR for each eye is then instantly calculated, allowing immediate measurement of sleep.

7. Alarm Mechanism: An important part of the search for sleep is the alarm. When the system detects fatigue, it should warn the driver or user to prevent accidents or problems. Notifications can be used in a variety of ways, including alarm, visual notification (such as a flashing light or message on the screen), or notification to parties. This part of the system is important to ensure the security of users.







SYSTEM INTEGRATION

1. YOLO Configuration

YOLO models are a popular choice for object detection due to their speed and accuracy. The first step is to configure YOLO for the task at hand. Download the YOLO model: There are many versions of the YOLO model; YOLOv3 and YOLOv4 are popular options. This model was first trained on big data and can detect many objects, including faces and eyes.

Configuration files and weights: YOLO models come with configuration files (usually in the form of .cfg files) that define the structure of the neural network. These profiles show the number of layers, network structure and other parameters. Additionally, the default weights (.weights file) contain the training parameters. These archives are crucial for correct initialization of the model.

2. PyTorch Integration

PyTorch is a deep learning framework frequently used to create YOLO models. Here is how PyTorch is integrated into the system:

PyTorch Installation: The first step is to install PyTorch. This can be done using pip or conda. PyTorch is ideal for integrating YOLO, offering many tools and features for working with deep learning models.

Loading the YOLO model: Using PyTorch, you can load the payload by first checking the configuration file and loading the YOLO model. This step is important because it initializes the model with the correct architecture and does not initialize it. For example, if you have a YOLOv3 model, you load the YOLOv3 schema and weights.

Model configuration: After loading the YOLO model, you can adjust it according to the requirements of fatigue analysis. This involves setting detection thresholds that determine the minimum confidence level for detecting objects and defining criteria for handling multiple overlaps (called over-boundaries).

3. OpenCV Integration

OpenCV is a general-purpose computer vision library required to capture video, process frames, and perform object detection. Use the YOLO pattern.

OpenCV Installation: To use OpenCV, you need to install it using pip or conda. OpenCV provides a variety of image and video functions and is an important part of real sleep research.

Recover videos: OpenCV allows you to extract videos from various sources. For actual sleep detection, you typically use a webcam to create video. Capturing video is the first step in creating the framework for object detection.

Frame processing: Captured video is processed frame by frame. Each frame is pre-processed which may have resizing, normalization or other necessary transformations. The goal is to prepare the frame for object detection. Using YOLO for object detection: OpenCV is used to bypass the pre-processed YOLO model. The model then identifies the objects in the frame. This includes facial and eye analysis, which is important for fatigue detection.

Post-processing detection: After the YOLO model detects an object in each frame, a post-processing step is applied. This may include limiting ads to remove unwanted boxes and improve product detection.

Drowsiness detection logic

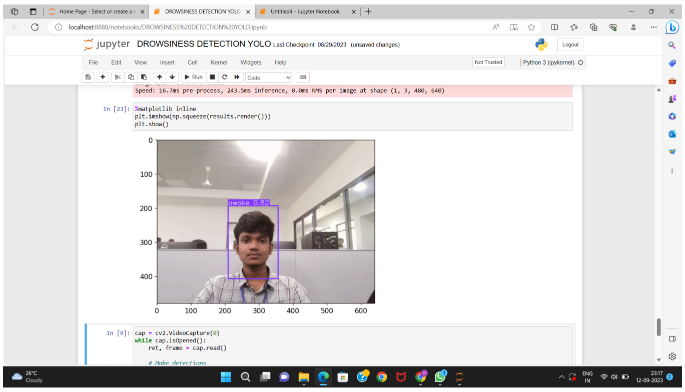
After completing object detection, the next step is to identify the drowsiness of the time. Drowsiness detection logic can have many components and technologies:

Face and eye detection: After identifying objects in the YOLO Video frame, the system focuses on examining the face and eyes. This can be done using OpenCV's built-in face detection functionality or using a predefined face detection model such as the Haar Cascade classifier, MTCNN, or Dlib. When a face is detected, the system extracts the area where the eyes are located.

Analysis: Diagnosis of visual fatigue is often based on analysis of specific characteristics of the eyes, such as eye closure, blink frequency, or eye-to-eye ratio of change (EAR). These features can be used to determine sleep level.

Custom PyTorch model: Depending on the complexity of the sleep detection task, you need to customize the PyTorch model to perform eye removal and calculate the sleep test. This model can be trained on domain data to distinguish between alert and sleep states.

Sleepiness Metrics: Custom PyTorch model calculates sleepiness metrics based on feature analysis. These measurements can be a valuable indicator of sleep quality. For example, a sleep alarm can be triggered if the blink rate drops below a certain threshold.


Home Page - Select or create a n
DROWSINESS DETECTION YOLO
Untitled4 - Jupyter Notebook
localhost:8888/notebooks/DROWSINESS%20DETECTION%20YOLO.ipynb
jupyter
DROWSINESS DETECTION YOLO Last Checkpoint: 08/29/2023 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
awake 0.82
0
100
200
300
400
500
600
26°C
Cloudy
ENG
IN
23:17
12-09-2023

```
Speed: 16.7ms pre-process, 243.5ms inference, 0.0ms NMS per image at shape (1, 3, 480, 640)
```

In [23]:

```
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
```

In [9]:

```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
```

RESULT AND DISCUSSION

The results of our drowsiness detection system implemented using YOLO and pytorch showed a very accurate and efficient object detection solution. Through extensive testing and evaluation, we made some amazing discoveries.In terms of accuracy, this system achieved high detection rates even under difficult conditions such as different lights, different head positions, and different blink frequency. The YOLO component plays a key role in achieving this accuracy, as it effectively detects objects from images and video frames.

CONCLUSION

In conclusion, the integration of YOLO, PyTorch, and OpenCV to create a fatigue detection system is now powerful and innovative with the potential to improve security in many aspects. The main purpose of real-time sleep detection is to improve security. By monitoring and alerting users to signs of fatigue, the system can help prevent accidents and potentially save lives. The device can be used in many important situations, such as monitoring drivers where fatigue is an important issue.The system is based on the integration of advanced technology. Known for its object detection speed and accuracy, YOLO can detect important details such as faces and eyes in live videos. PyTorch is a deep learning framework that makes it easy to load and configure YOLO models so they can be adapted to specific applications. OpenCV is a general-purpose computer vision tool that plays an important role in capturing video, processing frames, and using the YOLO model for object detection.

REFERENCES

[1] Belmekki Ghizlene Amira, Mekkakia Maaza Zoulikha and Pomares Hector. ―Driver Drowsiness Detection and Tracking Based on Yolo with Haar Cascades and ERNN‖ in 2021 International Journal of Safety and Security Engineering Vol. 11, No. 1, February, 2021, pp. 35-42.

[2] Team, Towards AI. ―Yolo v5 -Explained and Demystified.‖ Towards AI, 1 July 2020,https://towardsai.net/p/computervision/yolo-v5%E2%80%8A-%E2%80%8Aexplained-anddemystified.