Real-time Driver Drowsiness Detection

Jahnavi Bonagiri, Evangeli Silva, Andrew Okoro, Tobin Cherian
Instructor: Professor Ming Ching Chang

CSI 531: Data Mining
Spring 2023

Abstract

Driver drowsiness is a significant problem on the roads that can cause accidents, injuries, and fatalities. This issue has been present for decades, and it is important to develop reliable and accurate methods for detecting driver drowsiness in real-time. Research has shown that facial expressions, such as head movements, eye blinking, and body movements, are strongly correlated with drowsiness. However, existing driver drowsiness systems face challenges such as low accuracy, high false negatives and false positives, and limited robustness to different facial landmarks. This project aims to develop a reliable and accurate driver drowsiness system by using images as the dataset and applying feature extraction to create a Region of Interest (ROI) on the training set. The ROI will then be fed to a classifier to detect if the driver's eyes are closed or open. Multiple approaches, including Eye Aspect Ratio (EAR), developing a Convolutional Neural Network (CNN) model from scratch, and transfer learning with a pre-trained CNN model, were employed to detect the state of the eye. The best-performing model achieved a Precision-Recall Area under the Curve score (PR AUC) of 0.983 and was then used for real-time drowsiness detection. The system will send alerts to the driver when drowsy movements are detected. Although the system has some limitations, such as sensitivity to lighting conditions and camera distance from the person, data augmentation techniques can be applied to improve its accuracy and reliability. In conclusion, the project aims to develop a reliable and accurate driver drowsiness system that utilizes non-intrusive methods to increase driver alertness and safety. The project's outcome can contribute to preventing accidents, injuries, and fatalities caused by driver drowsiness on the roads.

1. Introduction

Driver drowsiness is a major problem on the roads, leading to numerous accidents, injuries, and even fatalities. In 2017 alone, there were 91,000 police-reported crashes that involved drowsy drivers, resulting in an estimated 50,000 injuries and 800 deaths (NHTSA, 2019) [1]. This issue has been present for several decades, and it is important to come up with reliable and accurate methods for detecting driver drowsiness in real-time to prevent accidents from occurring.

Research has shown that facial expressions, such as head movements, eye blinking, and body movements, are strongly correlated with drowsiness. Therefore, monitoring these facial expressions can help identify if a driver is drowsy. However, existing driver drowsiness systems face challenges such as low accuracy, high false negatives and false positives, and limited robustness to different facial landmarks. A reliable driver drowsiness system should have high accuracy, low false negatives, and operate in real-time while being robust to different facial landmarks.

Our project aims to address these challenges by using images as our dataset and applying feature extraction to create a Region of Interest (ROI) on our training set. We will then feed this ROI to a classifier to detect if the driver's eyes are closed or open. Finally, we will evaluate the accuracy of our model using our training dataset. To accomplish this, we will utilize TensorFlow and Keras for our CNN, and OpenCV to create the ROI on our dataset.

In order to detect the state of the eye we employed multiple approaches which included a mathematical approach utilizing Eye Aspect Ratio (EAR), developing a Convolutional Neural Network (CNN) model from scratch for image classification and transfer learning with a pre-trained CNN model. To identify the best performing model out of the three approaches, we evaluated the performance of each model using various metrics. Our primary metric for evaluation was the Precision-Recall Area under the Curve score (PR AUC). By achieving a PR AUC score of 0.983, the model seems to be the most effective method for open/closed eye detection. The transfer learning model then used to perform real-time drowsiness detection. The system will send alerts to the driver when drowsy movements are detected.

The primary objective of our system is to utilize non-intrusive methods to accurately determine the driver's state in real-time and generate effective and acceptable warnings to increase driver alertness and safety. However, our system does have some limitations that need to be considered. For instance, the system may not work properly when the camera is too far from the person, and it is sensitive to the lighting conditions in the environment. To improve our system's accuracy and reliability, we could apply data augmentation techniques, such as applying transformations like flips, crops, and rotations to our dataset. This will increase the size of the dataset and make the model more robust to variations in the input.

In conclusion, driver drowsiness is a serious problem that needs to be addressed to prevent accidents, injuries, and fatalities. Our project aims to develop a reliable and accurate driver drowsiness system that utilizes non-intrusive methods to accurately determine driver state in real-time and generate effective and acceptable warnings to increase driver alertness and safety. Although there are limitations to our system, we can improve its accuracy and reliability by applying data augmentation techniques and addressing the limitations mentioned earlier.

2. Literature Review

Drowsy driving is a significant problem and is a leading cause of accidents on the road. Many studies have been conducted to detect drowsiness in drivers to prevent accidents. In this report, we will discuss the existing practices for detecting drowsiness in drivers, including visual monitoring, lane deviation monitoring, head nod monitoring, activity monitoring, and mental reaction time.

Visual Monitoring of the Driver

Visual monitoring of the driver is a widely used approach to detect drowsiness, and it can be achieved by analyzing the driver's facial expressions and physical movements. Several studies have been conducted to detect drowsiness in drivers based on visual monitoring. For example, Rahman et al. (2022) [2] proposed a system that uses a camera mounted on the dashboard to capture the driver's facial expressions and detect drowsiness. One way to analyze the driver's facial expressions is by using facial landmarks such as the eyes, mouth, and other features. For instance, in [4], the authors use eye aspect ratio (EAR), mouth aspect ratio (MAR), mouth aspect ratio over eye aspect ratio, and pupil circularity as features for machine learning algorithms such as KNN, CNN, LSTM, and transfer learning to detect drowsiness. Another study by [4] uses a real-time video to analyze the driver's face and extract visual characteristics, and then applies CNN to show the degree of drowsiness of a driver. Haar Cascades for face detection were used in [5], while [6] utilized Eye Detection on Grey Intensity Face to detect drowsiness in drivers.

Overall, visual monitoring techniques that utilize facial landmarks and machine learning algorithms have shown promising results in detecting drowsiness in drivers. However, there is still a need for further research to improve the accuracy and reliability of these methods.

Lane Deviation Monitoring

Lane deviation monitoring is an important technique for detecting driver drowsiness. This method involves monitoring the driver's position in relation to the lane markings on the road. A driver who is drowsy is more likely to drift out of their lane, so this technique can be used to predict drowsiness before it becomes a problem. McDonald et al. (2012) [7] proposed a system for real-time detection of drowsiness-related lane departures using steering wheel angle. In their study, they showed that the steering angle can be used to predict lane departures up to six seconds before they occur, giving the driver enough time to take corrective action. The study also proposed the use of a random forest algorithm, paired with an alert system, to significantly reduce vehicle crashes caused by drowsiness. The results of this study suggest that lane deviation monitoring, combined with machine learning algorithms, can be an effective method for detecting and preventing driver drowsiness.

Head Nod Monitoring

Head nod monitoring is a method of detecting drowsiness in drivers by measuring the movement of the driver's head. Choi et al. (2014) [8] developed a system that uses gaze direction tracking and head pose estimation to detect drowsiness in drivers. This system estimates the driver's head pose by calculating optic flow of facial features using a corner detection algorithm. The optic flow provides information on how the facial features move, allowing the system to detect the three moving components of the driver's head behavior: nodding, shaking, and tilting. The study showed that the system was effective in detecting drowsiness, as it was able to identify head nodding patterns associated with sleepiness. Additionally, the system was able to distinguish between head nodding due to drowsiness and head nodding due to other reasons such as the road conditions.

Activity Monitoring

Activity monitoring is a technique that uses wearable sensors to track the driver's motion and detect drowsiness. One type of wearable device used in this technique is wrist-worn devices that integrate Photoplethysmogram (PPG) sensors and motion sensors. The study by Cal et al. (2023) [9] proposes a drowsiness detection system that uses these sensors to detect drowsiness in drivers. The system analyzes heart rate variability (HRV) and motion data to estimate the driver's level of drowsiness. The authors claim that this approach provides accurate drowsiness detection with a low false positive rate, making it a promising technique for real-world application.

Another study by Tavakoli et al. (2021) [10] uses smartwatches for passive sensing to classify elements of driving context. The authors propose a multi-task deep learning framework that classifies various driving contexts, such as driving style, route type, and traffic congestion, by exploiting different types of sensors embedded in smartwatches. The study also investigates the correlation between driving context and driver fatigue to improve the accuracy of drowsiness detection. By leveraging the passive sensing capability of smartwatches, this technique can provide real-time detection of drowsiness without requiring additional hardware.

Mental Reaction Time

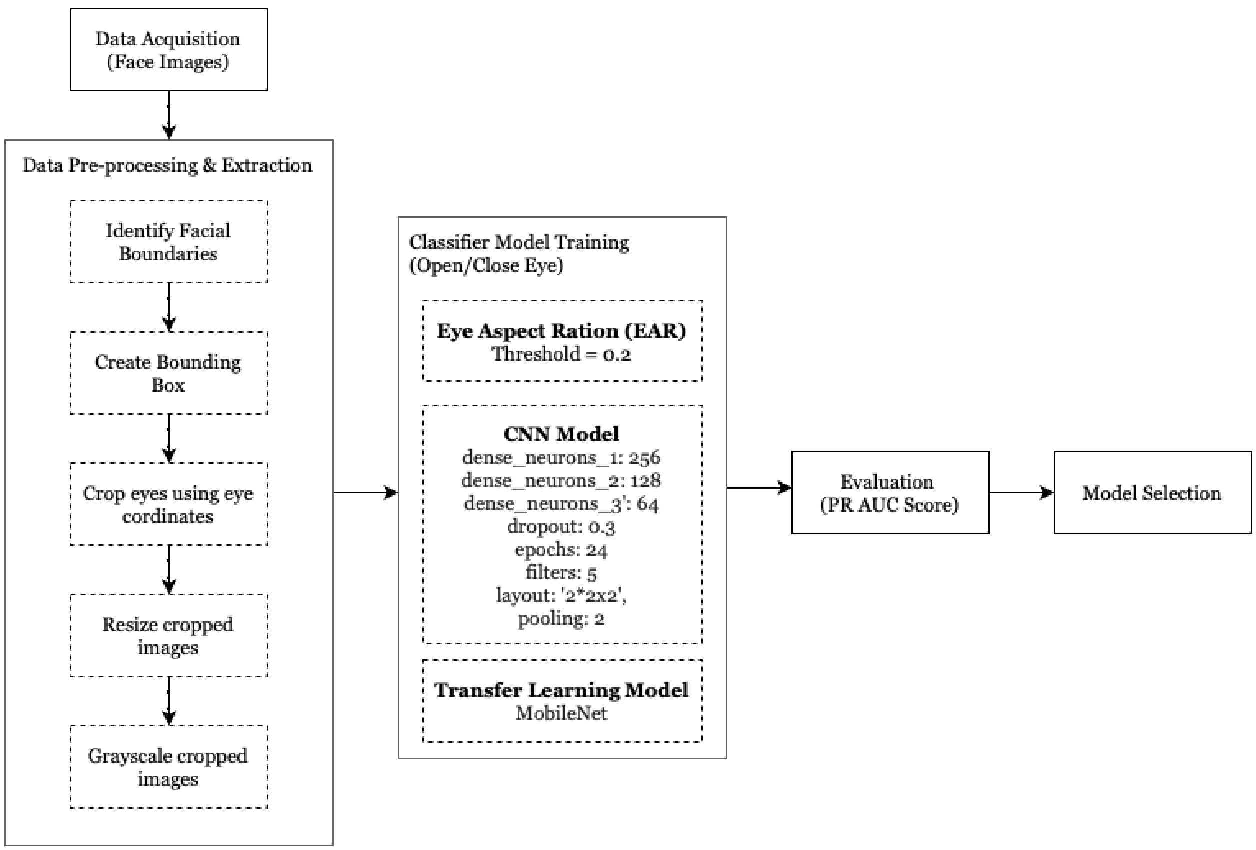
Mental reaction time is a technique that measures the time it takes for a driver to respond to a random warning signal. The study by Rahaman et al. (2019) [11] proposes a novel approach to detect driver sleepiness using a wearable electroencephalogram (EEG). This technique is based on the analysis of brain electrical activity, which is a reliable indicator of sleepiness. The study used the EMOTIV Epoc+ device to collect EEG signals and decompose them into narrowband frequencies such as delta, theta, alpha, and beta using Discrete Wavelet Transform (DWT). The decomposed signals were then fed to a machine learning algorithm to detect sleepiness. The study found that the proposed technique achieved a high accuracy of 90% in detecting drowsiness, indicating its potential as a promising alternative to conventional drowsiness detection systems. The wearable EEG device used in this study is lightweight and easy to use, making it a convenient tool for monitoring driver sleepiness in real-time.

In conclusion, detecting drowsiness in drivers is crucial in preventing accidents on the road. Various techniques have been proposed for detecting drowsiness, including visual monitoring, lane deviation monitoring, head nod monitoring, activity monitoring, and mental reaction time. These techniques utilize different sensors and machine learning algorithms to detect drowsiness accurately. While existing practices have shown promising results, there is still a need for further research to improve the accuracy and reliability of these methods. Overall, the development of effective drowsiness detection systems can significantly reduce accidents caused by drowsy driving.

3. Methodology and Implementation

In order to monitor facial expressions and specifically the eye closure time of a person, we have developed a two-phase approach that involves detecting the open or closed state of the eyes and then using this information to detect drowsiness in real-time.

Phase 1: Open/Close Eye Detection



The first phase of our approach involves data acquisition and pre-processing, followed by the training of a classifier model for eye state detection. We started by acquiring face images, which were then pre-processed to identify the facial boundaries and create a bounding box around the eyes. We then cropped the eyes using the eye coordinates and resized the cropped images to a standard size. Finally, we converted the images to grayscale to simplify the processing.

Next, we trained multiple models for eye state detection, including the Eye Aspect Ratio (EAR) model, a Convolutional Neural Network (CNN) model, and a Transfer Learning model. The EAR model is based on the ratio of the distance between eye landmarks to the eye width, and we set a threshold of 0.2 to determine whether the eyes are open or closed. The CNN and Transfer Learning models were trained on the pre-processed eye images, with the Transfer Learning model using a pre-trained model as the base.

To evaluate the performance of the models, we used the Precision-Recall Area Under Curve (PR AUC) score. We selected the model with the highest PR AUC score for eye state detection.

Once the best model was selected, it was used to detect the driver's eye state in real-time by processing the video feed from a camera. The resulting information is then used to calculate the time for which the eyes are closed to detect whether the driver is drowsy.

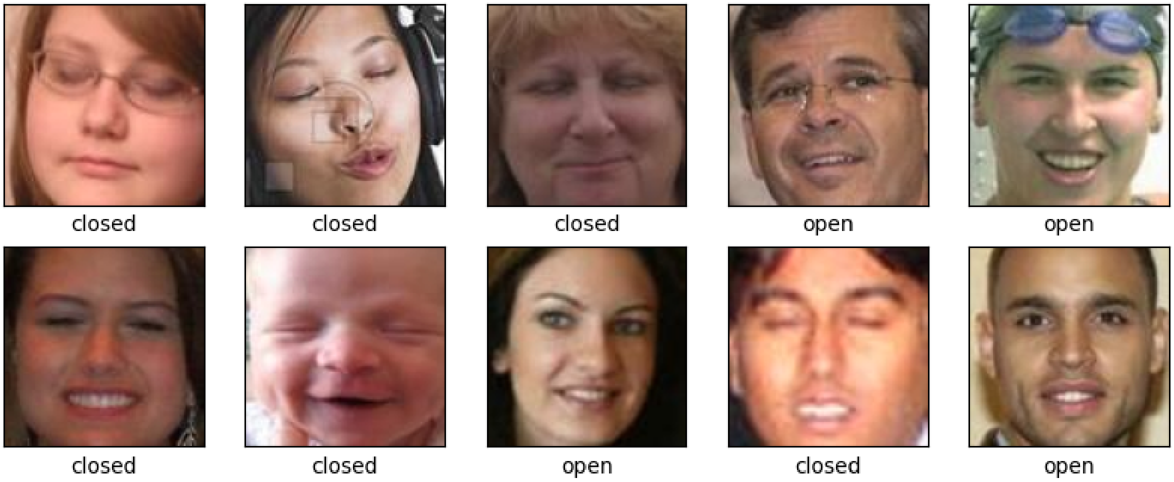
A. Data Acquisition

The dataset consists of three sources, including two facial image datasets and one eye dataset, that contain various attributes such as grayscale, RGB, high/low quality, gender, glasses, and diverse facial structures. The diverse characteristics of the dataset ensure that the classification models are trained to recognize eye state accurately.

1. **Labeled Faces in the Wild (Link)**
   The Labeled Faces in the Wild (LFW) dataset is provided by the University of Massachusetts Amherst. It contains 18,983 images of people's faces with both eyes open. The dataset has images of different resolutions and qualities, captured under different lighting conditions, and from different angles. It includes grayscale and RGB images of male and female faces with or without glasses.

2. **Closed Faces in the Wild (Link)**
   The Closed Faces in the Wild (CFW) dataset is provided by the Nanjing University of Aeronautics and Astronautics. It contains 1192 images of people's faces with both eyes closed. The dataset includes images of different resolutions and qualities, captured under different lighting conditions, and from different angles. It includes grayscale and RGB images of male and female faces with or without glasses.

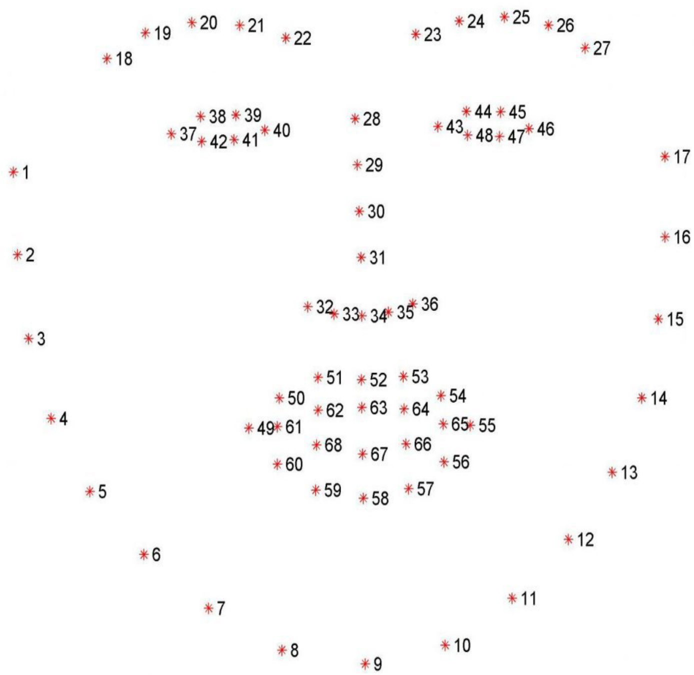
3. **Kaggle Dataset (Link)**
   The Drowsiness Dataset is obtained from Kaggle and provided by Dheeraj Perumandla. It contains a collection of 1452 images, including 726 images of open eyes and 726 images of closed eyes. The dataset includes grayscale and RGB images of both male and female eyes, captured under different lighting conditions, and from different angles. The images are of high quality and resolution.



Sample Data

B. Data Preprocessing

In order to detect the open and closed state of a person's eyes, we extracted the relevant features from the face images. We used the face_recognition python library, which is built using dlib's state-of-the-art face recognition built with deep learning, and claims to have an accuracy of 99.38% on Labeled Faces in the Wild benchmark. The face_recognition library uses a training set of labeled facial landmarks on an image, where each image is manually labeled with 68 (x, y)-coordinates of regions surrounding each facial structure. Given this training data, an ensemble of regression trees are trained to estimate the facial landmark positions directly from the pixel intensities themselves. This allowed us to detect the key facial structures on the face ROI, including the eyes.

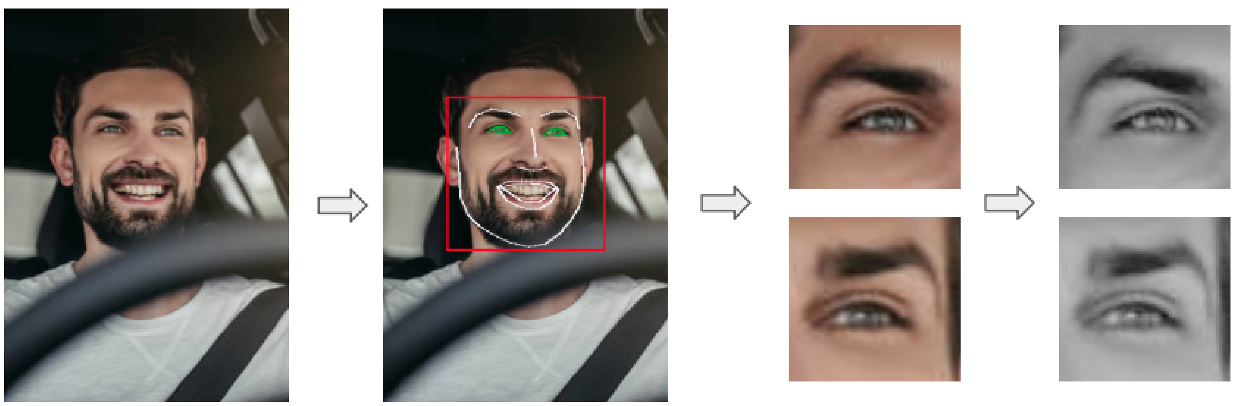


Visualization of the 68 facial landmark coordinates from the iBUG 300-W dataset

Once we have identified the locations of the eyes, we cropped the image to only include the regions around the eyes. In order to ensure that the full eye is captured in the cropped image, we added 50% cushion to the x and y coordinates of the bounding box that encloses the eye landmarks. This ensures that any eye movements or blinks can be captured within the cropped image.

After cropping the images, we resized them to a standard size and converted them to grayscale. This allows us to reduce the dimensionality of the images and standardize the inputs to our models, which is important for accurate and consistent eye state detection.

Overall, this data preprocessing step is critical for obtaining accurate eye state detection results. By using a high-quality face recognition library and careful cropping and resizing of the eye images, we can extract the relevant features needed for training our eye state detection models.



A sample image after pre-processing

C. Classifier Model Training (Closed/Open Detection)

In order to detect the state of the eye we employed multiple approaches which included a mathematical approach utilizing Eye Aspect Ratio (EAR), developing a Convolutional Neural Network (CNN) model from scratch for image classification and transfer learning with a pre-trained CNN model.

Approach 1: Eye Aspect Ratio

One of the approaches we explored for open/closed eye detection is the mathematical approach, which uses the eye aspect ratio (EAR). EAR is a measure of the relationship between the vertical and horizontal landmarks of the eye and is calculated as the ratio of the distance between two vertical landmarks to the distance between two horizontal landmarks. Specifically, EAR is computed based on the (x,y)-coordinates of the eye, where P2, P3, P5, and P6 key points represent the value of the longitudinal coordinate, and P1 and P4 key points represent the value of the horizontal coordinate.

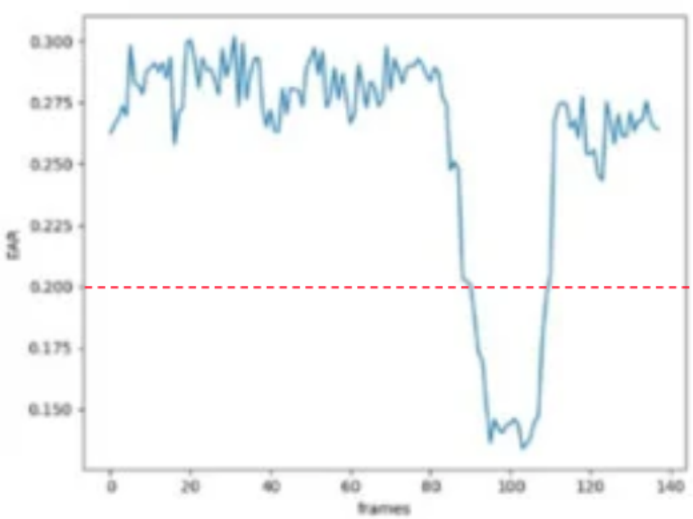
Associated Landmarks

The distance between the vertical landmarks is calculated as the Euclidean distance between P2 and P6, and the distance between the horizontal landmarks is calculated as the Euclidean distance between P1 and P5. The EAR is then calculated as the ratio of these two distances.

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

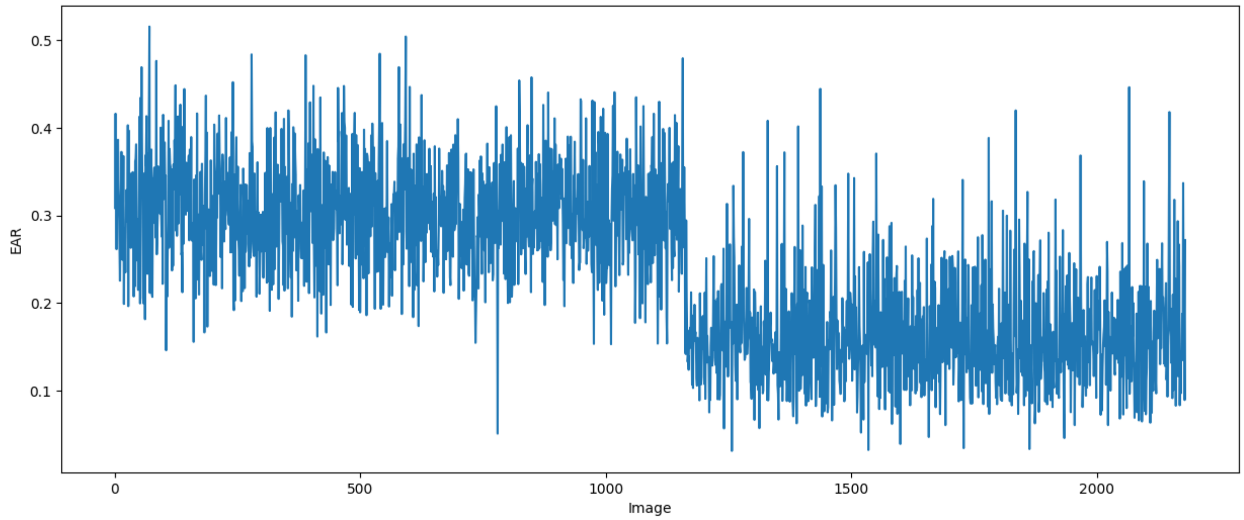
When the eyes are open, the distance between the vertical landmarks is longer than when the eyes are closed, resulting in a higher EAR value. Conversely, when the eyes are closed, the distance between the vertical landmarks is shorter, resulting in a lower EAR value.

By setting a threshold EAR value, we can classify an eye as either open or closed. According to T Zhu et al (2022) [12], when the value of EAR is greater than 0.2, it is identified as an open eye, and when it is less than or equal to 0.2, it is identified as a closed eye. This threshold value of 0.2 was chosen based on the analysis of a large dataset of eye images, where it was found to provide good accuracy in detecting eye state.



The values of EAR at open and closed states

The graph below illustrates the EAR values obtained from the preprocessed images.



We observed that EAR fluctuates greatly when given an image of an eye with glasses, and this may lead to incorrect classification of the eye state. To mitigate this issue, we explored other approaches such as building a CNN model from scratch for image classification and using transfer learning with a pretrained CNN model.

In summary, while the mathematical approach using EAR is a straightforward and efficient method for open/closed eye detection, it has limitations in certain scenarios. Hence, we explored other methods to achieve more accurate and robust results.

Approach 2: Convolutional Neural Network

The second approach we explored was building a CNN model from scratch for image classification. We used a simple CNN architecture consisting of multiple convolutional layers followed by fully connected layers. We trained the model on the cropped eye images using a binary cross-entropy loss function and Adam optimizer. This approach performed better than the EAR approach and was able to accurately detect the open and closed state of the eyes.

As the initial step under training a convolutional neural network we performed hyper-parameter tuning. Hyper-parameter tuning is an essential step in building a robust and accurate machine learning model. We conducted hyper-parameter tuning to explore different combinations of model parameters and identify the best configuration that produces the highest accuracy. For the exhaustive search over specified parameter values we used `sklearn.model_selection.GridSearchCV`.

We trained and evaluated approximately 100 models, varying the parameters in the following categories: convolutional layers, filters, pooling, dense layers, and dropout. For each parameter, we tested several options to find the best combination.

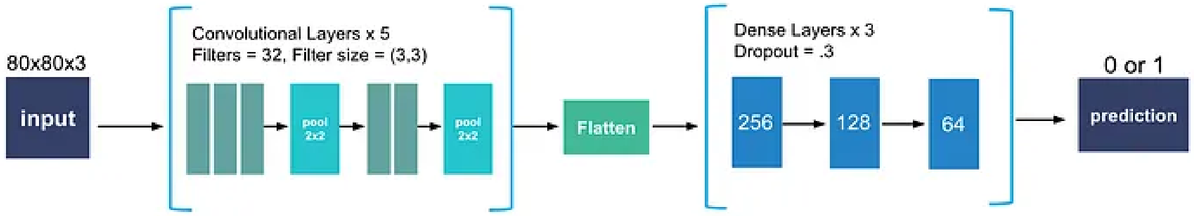
1. Convolutional Layers (Number of layers: 1, 2, 3, 4, 5)
2. Filters (Number of filters: 8, 16, 32 | Size of filters: 3x3, 5x5)
3. Pooling (Pooling of features 2x2: Yes, No)
4. Dense Layers (Number of Layers: 1, 2, 3 | Neurons: 64, 128, 256, 512, 1024)
5. Dropout: (Percentage: 10%, 20%, 30%, 40%, 50%)

We used grid searching with k-fold cross-validation (k = 3) to evaluate the performance of each model. Grid searching involves testing all possible combinations of the given parameter values, while k-fold cross-validation involves dividing the dataset into k equal parts and training the model k times, using a different part as the validation set each time. The following are the top 5 scores obtained from the grid search.

Dense Layer 1 Neurons	Dense Layer 2 Neurons	Dense Layer 3 Neurons	Dropout Rate	Number of Iterations	Input Filters	Shape of Filters	Pooling Layer	Average Test Score
256	128	64	0.3	24	32	3x3	Yes	0.9631
256	512	1028	0.4	30	32	3x3	Yes	0.9614
256	512	1028	0.3	30	64	3x3	Yes	0.9610
128	128	128	0.1	10	32	5x5	No	0.9123
128	128	128	0.1	10	16	3x3	No	0.9028

We analyzed the performance of each model based on accuracy and chose the best configuration of parameters that produced the highest accuracy. This approach allowed us to optimize the model's performance and improve its ability to accurately classify open and closed eyes.

After extensive grid searching, the optimal CNN architecture model which produced an AUC score of 0.963 is as follows.



The Final Network Architecture

Convolutional Layers are a fundamental building block of Convolutional Neural Networks (CNNs). They are responsible for creating subsets of pixels (also called filters or kernels) from an image and convolving them with the image to extract local features. By using subsets of pixels instead of the full image, convolutional layers allow for faster models and can learn more complex relationships using fewer resources.

The convolutional layers in the model are using the Rectified Linear Unit (ReLU) activation function, which helps prevent the exponential growth in computation required to operate the neural network. The output layer is using the Sigmoid activation function, which maps input values to a value between 0 and 1. The model is compiled using the Adam optimizer, which has faster computation time and requires fewer parameters for tuning. The model was trained for 24 epochs, as there were diminishing returns after 20 epochs.

In terms of the convolutional layers, the model uses two 3x3 layers pooled together followed by three 3x3 layers also pooled together, with 32 filters set. The flatten function is used to ensure that the image array can enter the dense layers. For the dense layers, three layers are used with ReLU activation at a decreasing rate of neurons (256, 128, 64), with a 30% dropout after each layer.

For the output layer, the Sigmoid activation function is used as this is a binary classification problem. When compiling the model, the metric is set to PR AUC or recall, and the loss is set to binary_crossentropy. The batch size is set as high as possible, without overloading the memory, and 32 batches are used in this specific CNN.

The aforementioned CNN model was developed using Tensorflow’s Keras API and the model summary is as follows.

Layer (type)	**Output Shape**	**Param#**
conv2d_76 (Conv2D)	(None, 78, 78, 32)	896
conv2d_77 (Conv2D)	(None, 76, 76, 32)	9248
conv2d_78 (Conv2D)	(None, 74, 74, 32)	9248
max_pooling2d_70 (MaxPooling2D)	(None, 37, 37, 32)	0
conv2d_79 (Conv2D)	(None, 35, 35, 32)	9248
conv2d_80 (Conv2D)	(None, 33, 33, 32)	9248
max_pooling2d_71 (MaxPooling2D)	(None, 16, 16, 32)	0
flatten_35 (Flatten)	(None, 8192)	0
dense_140 (Dense)	(None, 256)	2097408
droput_105 (Dropout)	(None, 256)	0
dense_141 (Dense)	(None, 128)	32896
droput_106 (Dropout)	(None, 128)	0
dense_142 (Dense)	(None, 64)	8256
droput_107 (Dropout)	(None, 64)	0
dense_143 (Dense)	(None, 1)	65

Approach 3: Transfer Learning

The third approach we explored was transfer learning with a pretrained CNN model. We used the pre-trained MobileNet model and fine-tuned it for the open/closed eye detection task. We replaced the last few layers of the model with our own fully connected layers and trained the model on the cropped eye images. This approach performed the best among the three methods with an ROC AUC of 0.986, as the pre-trained model already had learned features that were relevant to the task, and we were able to fine-tune it to our specific task.

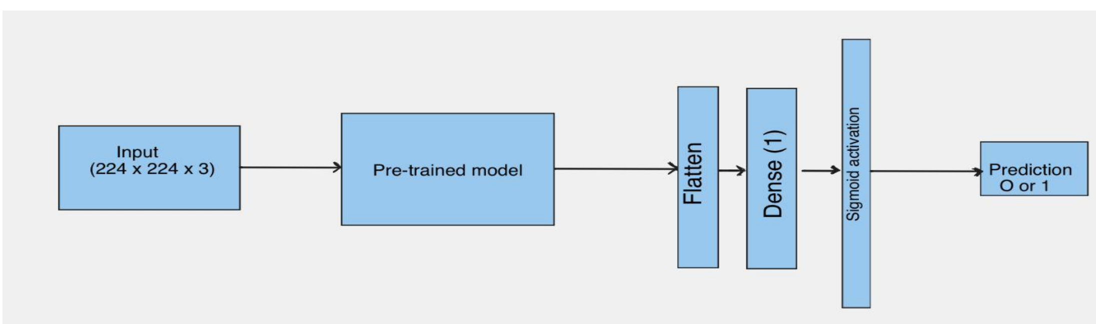
In image classification tasks, transfer learning is a widely used technique that involves using a pre-trained deep learning model as a starting point for the new task. This is because pre-trained models have already learned relevant features from large datasets, making them effective in solving new related problems with less data.

MobileNet is one such pre-trained model that was trained on the ImageNet dataset. The model is specifically designed for mobile and embedded vision applications and uses depth-wise separable convolution to reduce the number of parameters and make it more efficient for deployment on mobile devices.

To use MobileNet for our image classification task, we removed the last few layers of the pre-trained model, which were trained for ImageNet classification and not specific to our task. We added a few new layers to the model, namely a flatten layer and a dense layer for binary classification. The flatten layer is used to flatten the output of the pre-trained model, and the dense layer has a sigmoid activation function for binary classification.

For training the model, we used binary cross-entropy loss function and the Adam optimizer. The binary cross-entropy loss function is commonly used for binary classification tasks and calculates the difference between the predicted and actual class probabilities. The Adam optimizer is an efficient optimizer that adjusts the learning rate adaptively during training.

To monitor the training progress and prevent overfitting, we used early stopping and callbacks. Early stopping allows us to stop the training process if the validation loss does not improve after a certain number of epochs. Callbacks can be used to monitor the training process and take action when specific conditions are met, such as saving the best model weights or reducing the learning rate if the validation loss is not improving.



The Transfer Learning Model

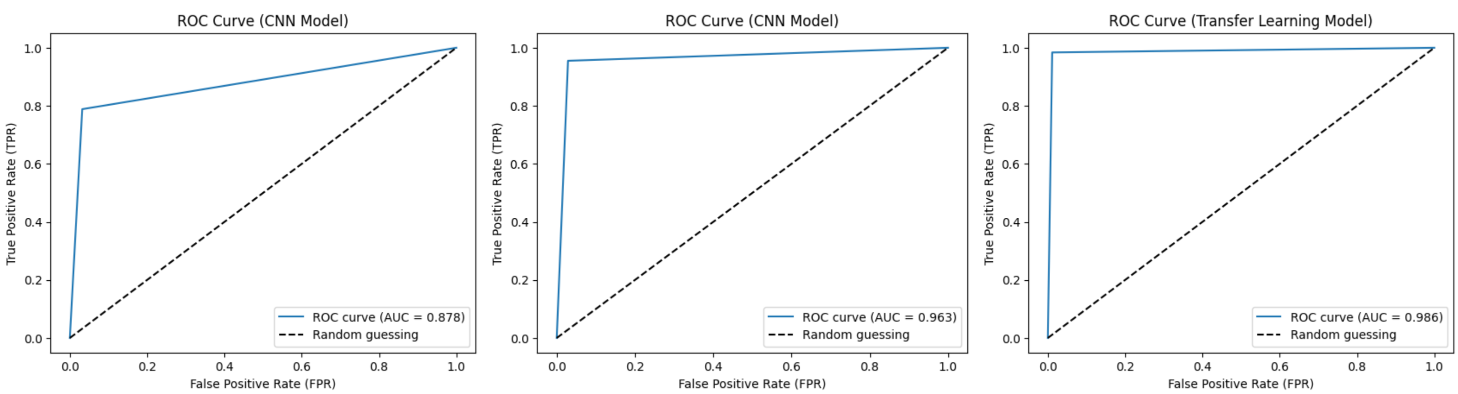
In summary, fine-tuning the pre-trained MobileNet model for image classification involves removing the last few layers, adding new layers, using binary cross-entropy loss function, Adam optimizer, and early stopping and callbacks for training monitoring. This approach is effective in achieving good performance in image classification tasks with limited data.

D. Model Evaluation

To identify the best performing model out of the three approaches, we evaluated the performance of each model using various metrics. Our primary metric for evaluation was the Precision-Recall Area under the Curve score (PR AUC). The PR AUC score is a commonly used evaluation metric for binary classification problems, especially when the positive class is rare or imbalanced. It measures the trade-off between precision and recall for different classification thresholds. Precision is the fraction of true positives out of all predicted positives, while recall is the fraction of true positives out of all actual positives. The PR AUC score takes into account both precision and recall over all possible classification thresholds and gives a single number that summarizes the overall performance of the model. A score of 0.5 is considered as the baseline, which represents a random guess. A perfect classifier would have a PR AUC score of 1.0. In practice, a higher PR AUC score indicates a better performance of the model.

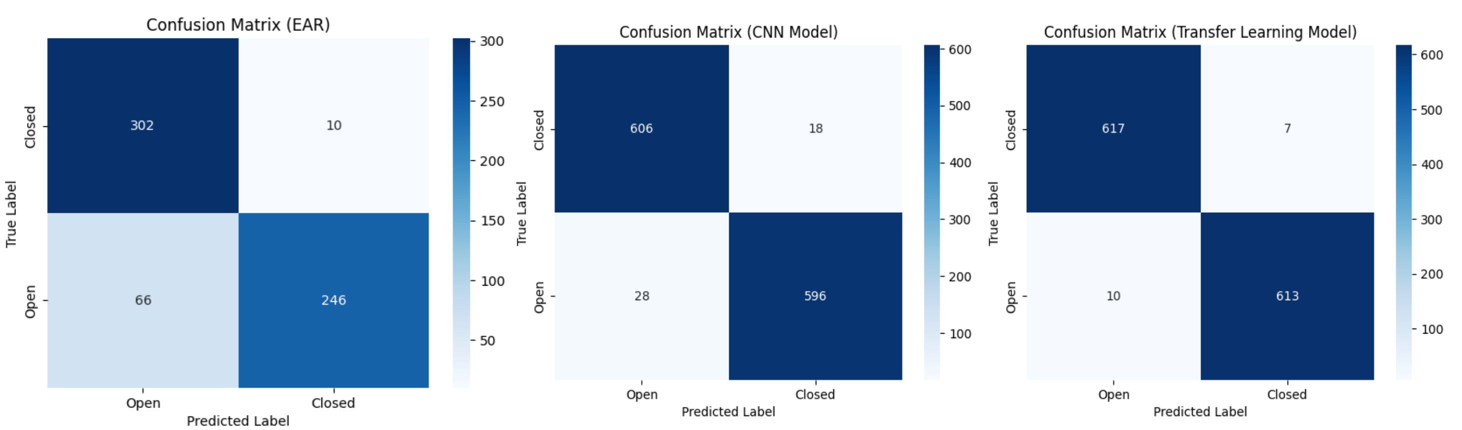
In the context of driver drowsiness detection, it is more important to correctly predict when the driver is asleep (true positives) than to correctly predict when the driver is awake (true negatives). This is because the consequences of a false negative (predicting that the driver is awake when they are actually asleep) can be more severe than a false positive (predicting that the driver is asleep when they are actually awake).

Therefore, the PR AUC score is a suitable metric for evaluating the performance of a driver drowsiness detection model. By achieving a PR AUC score of 0.983, the model seems to be the most effective method for open/closed eye detection.



ROC-AUC scores of the Eye State Classifier Models

The confusion matrices of the eye state classifier models are as follows.

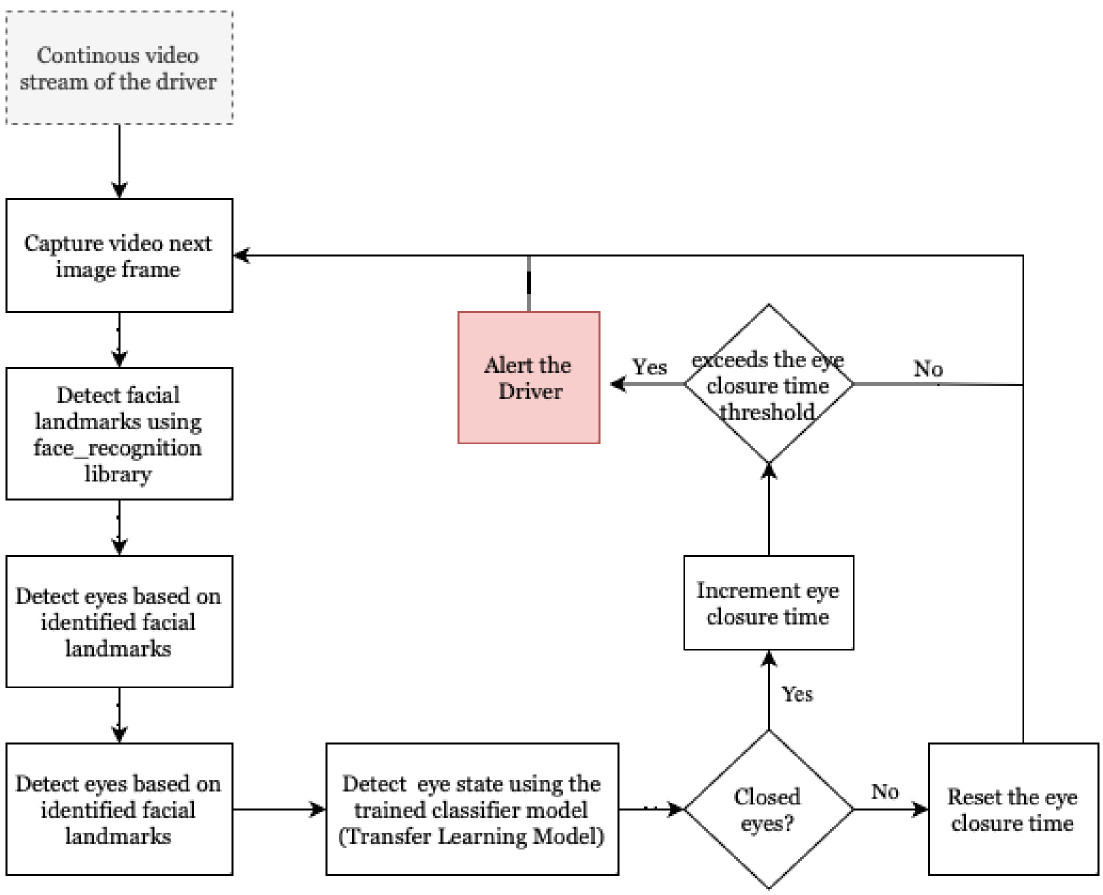


Confusion Matrices of the Eye State Classifier Models

Based on the evaluation matrix, we can conclude that the transfer learning approach performed the highest with an accuracy of 98.6% followed by the CNN model of accuracy 96.31%. We can conclude that Transfer learning has several benefits with the main advantages being saving training time and computational resources as the pre-trained model has already learned useful features and patterns.

Metric	EAR Based Classification	CNN Model	Transfer Learning Model
Accuracy	87.8%	96.31%	98.6%
Precision	96.1%	97.07%	98.87%
Recall	78.84%	96.51%	98.39%
F-measure	86.62%	96.28%	98.63%

Phase 2: Real-time Drowsiness Detection



The second phase utilizes the trained transfer learning model to perform real-time drowsiness detection by implementing the following steps:

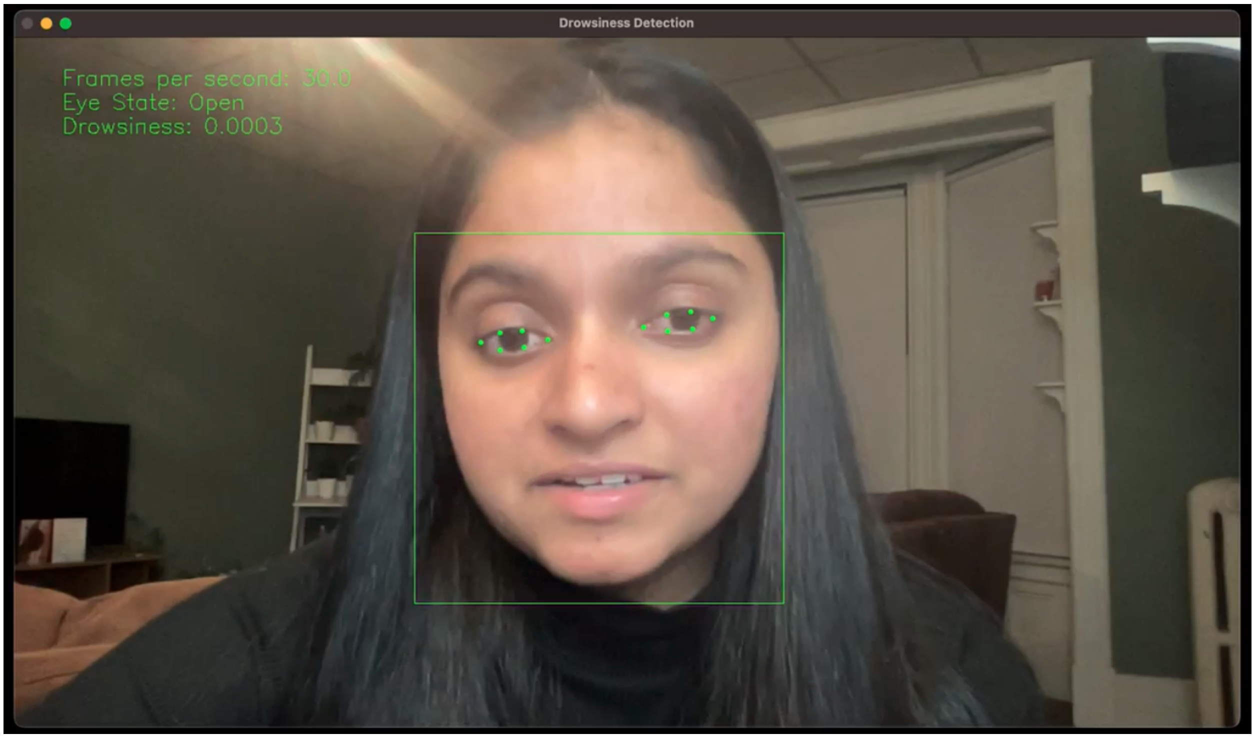
1. **Continuous video stream of the driver:** A video stream of the driver's face is captured continuously in real-time.
2. **Capture next image frame:** The next image frame from the video stream is captured.
3. **Detect facial landmarks:** The face_recognition library is used to detect the facial landmarks in the captured image. These landmarks include the position of the eyes, nose, mouth, etc.
4. **Detect eyes:** Using the identified facial landmarks, the positions of the eyes are determined.
5. **Detect eye state:** The trained transfer learning classifier model is then used to determine the state of the eyes - whether they are open or closed.
6. **If the eye is open, reset the eye closure time and analyze the next frame:** If the eyes are open, the eye closure time is reset to zero, and the process starts again from step 2 for the next image frame.
7. **If the eye is closed, check whether eye closure time exceeds the threshold:** If the eyes are closed, the eye closure time is then checked to see if it exceeds the threshold of 2 seconds. If the eye closure time exceeds the threshold of 2 seconds, an alert is generated to notify the driver to wake up. The process then starts again from step 2 for the next image frame.

The eye closure time is a critical parameter in the real-time drowsiness detection system as it is used to determine whether a driver is becoming drowsy or not. The duration of eyelid closure is an indicator of drowsiness and fatigue, and it is crucial to set a threshold that can accurately detect drowsiness while minimizing the false alarms.

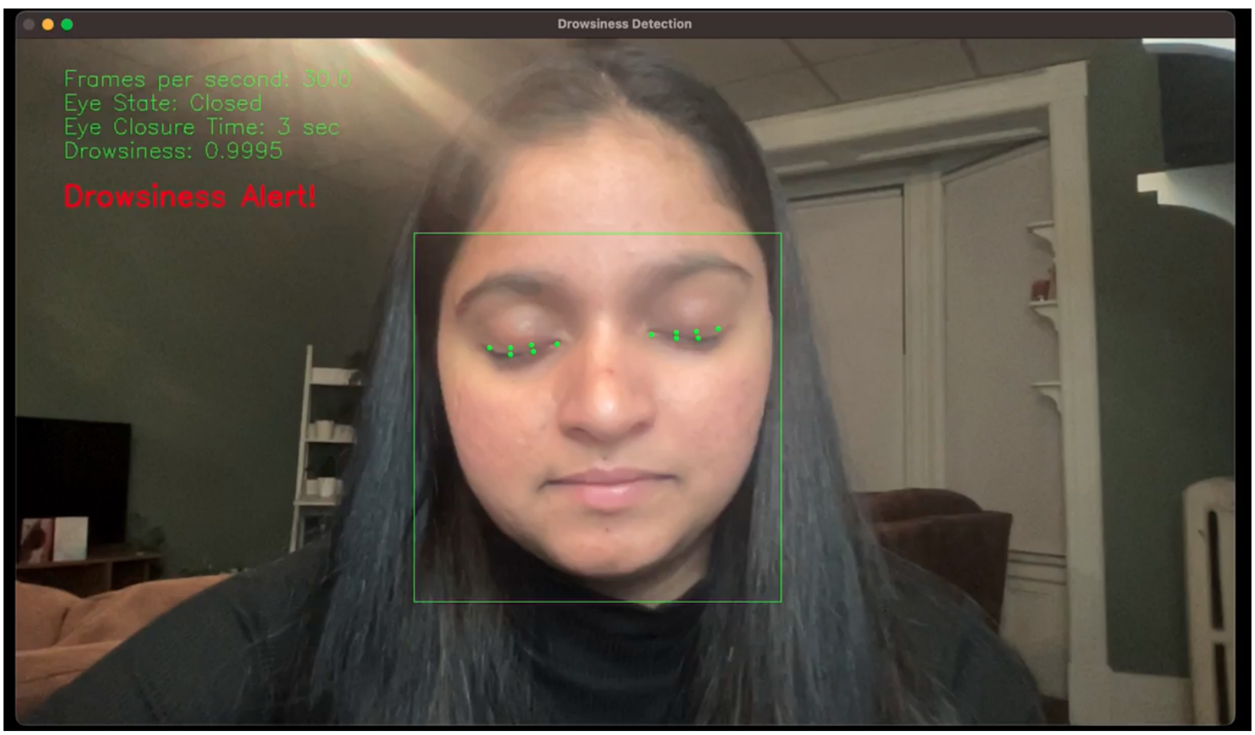
Studies such as the one published in the Journal of Safety Research in 2009 have shown that the mean duration of eyelid closure prior to a simulated crash was 3.79 seconds. This duration is significant, and it shows that a driver can be completely unaware of their drowsiness until it is too late. The same study also found that eyelid closure durations of 2 seconds or longer were highly indicative of drowsiness, suggesting that this is a critical threshold to monitor.

Another study published in the Journal of Sleep Research in 2018 found that a duration of eyelid closure of 2 seconds or more was associated with increased levels of drowsiness and impaired driving performance. This further confirms the importance of setting the closure time threshold to 2 seconds in our drowsiness detection system.

Therefore, by setting the closure time threshold to 2 seconds, we can detect drowsiness accurately and alert the driver in a timely manner, potentially preventing accidents and promoting safe driving practices.



Real-time Driver Drowsiness System when the Driver is not Drowsy (Eyes Open)



Real-time Driver Drowsiness System when the Driver is Drowsy (Eyes Closed)

4. Limitations and Improvements

The limitations and improvements of the real-time drowsiness detection are as follows:

1. **Proximity to Camera**

   The system may not work properly when the camera is far from the person. The face detection and facial landmark detection algorithms used in the system are dependent on the size of the face in the image. If the face is too small, the algorithm may not be able to detect it, resulting in incorrect or unreliable output. One solution to this limitation is to use a camera with a higher resolution or move the camera closer to the person's face.

2. **Sensitivity to the Lightening Condition in the Environment**

   Another limitation is that the system is sensitive to the lighting conditions in the environment. This is because the facial landmark detection algorithm is sensitive to variations in lighting and contrast. To address this limitation, we can use techniques such as histogram equalization or adaptive thresholding to improve the contrast and brightness of the image.

3. **Data Augmentation**

   A potential improvement to our drowsiness detection system is to apply data augmentation by applying transformations like flip, crop, rotations, to the data set. This will increase the size of the dataset and make the model more robust to variations in the input. This will help to address the issue of overfitting, which occurs when the model is too specific to the training data and cannot generalize well to new data.

5. Conclusion

In conclusion, driver drowsiness is a serious issue that can lead to devastating consequences. The development of a reliable and accurate driver drowsiness system that utilizes non-intrusive methods to detect driver state in real-time and generate effective and acceptable warnings can greatly increase driver alertness and safety. This project addressed the challenges faced by existing driver drowsiness systems, such as low accuracy, high false negatives and false positives, and limited robustness to different facial landmarks. The approach of utilizing images as the dataset and applying feature extraction to create a Region of Interest (ROI) on the training set, followed by a classifier to detect the state of the eyes, was found to be highly effective. The project's outcome, achieving a PR AUC score of 0.983, indicates that the system can accurately and reliably detect drowsiness in real-time. While there are limitations to the system, the application of data augmentation techniques and addressing the limitations can enhance its accuracy and reliability. Overall, the project's outcome can contribute to preventing accidents, injuries, and fatalities caused by driver drowsiness on the roads.

References

1. Centers for Disease Control and Prevention. (2021, March 25). Drowsy Driving: Asleep at the Wheel. https://www.cdc.gov/sleep/features/drowsy-driving.html
2. Rahman, A., Hriday, M. B. H., & Khan, R. (2022). Computer vision-based approach to detect fatigue driving and face mask for edge computing device. Heliyon, 8(10), e11204. https://doi.org/10.1016/j.heliyon.2022.e11204
3. Singh, H. (2019, August 9). Drowsiness Detection with Machine Learning. Towards Data Science.https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a
4. Q. Ji and X. Yang, “Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance,” Real-Time Imaging, vol. 8, no. 4, pp. 357–377, 2002.
5. GeeksforGeeks. (2021, January 26). Face Detection using OpenCV-Python. GeeksforGeeks. https://www.geeksforgeeks.org/face-detection-using-opencv-python/.
6. A Robust Algorithm for Eye Detection on Grey Intensity Face without Spectacles - JCS&T Vol. 5 No. 3
7. McDonald, A. D., Schwarz, C., Lee, J. D., & Brown, T. L. (2012). Real-time detection of drowsiness related lane departures using steering wheel angle. Proceedings of the Human Factors and Ergonomics Society 56th Annual Meeting.
8. In-Ho Choi and Yong-Guk Kim, "Head pose and gaze direction tracking for detecting a drowsy driver," 2014 International Conference on Big Data and Smart Computing (BIGCOMP), Bangkok, 2014, pp. 241-244, doi: 10.1109/BIGCOMP.2014.6741444.
9. Yetong Cao, Fan Li, Xiaochen Liu, Song Yang, and Yu Wang. 2023. Towards Reliable Driver Drowsiness Detection Leveraging Wearables. ACM Trans. Sen. Netw. 19, 2, Article 39 (May 2023), 23 pages. https://doi.org/10.1145/3560821
10. Tavakoli, A., Kumar, S., Boukhechba, M., & Heydarian, A. (2021). Driver State and Behavior Detection Through Smart Wearables. arXiv preprint arXiv:2104.13889.
11. Rahma, O. N., & Rahmatillah, A. (2019). Drowsiness Analysis Using Common Spatial Pattern and Extreme Learning Machine Based on Electroencephalogram Signal. Journal of medical signals and sensors, 9(2), 130–136. https://doi.org/10.4103/jmss.JMSS_54_18
12. Zhu, T.; Zhang, C.; Wu, T.; Ouyang, Z.; Li, H.; Na, X.; Liang, J.; Li, W. Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences. Appl. Sci. 2022, 12, 2224. https://doi.org/10.3390/app12042224