**Table of contents**

Supervisor: Peter Passmore

|  |  |
| --- | --- |
| A picture of a winding road and trees  Real-Time Fatigue Detection Using Machine Learning and Computer Vision | Erik Vasquez Chafla  **M00866654** |

1- Declaration ………………………………………………………………………………………….………**2**

2- Abstract ………………………………………………………………………………….……………………**3**

3- Introduction …………………………………………….………………………………………………….**4**

4- Background ……………………………………………………………………………………....…….….**4**

4.1- Problems With Fatigue …………………………………………….….………………....**4**

4.2- Effects of fatigue ……………………………………………………………………….……**5**

4.3- Technology to identify fatigue………………………………………………………….**7**

4.4- Aims ……………………………………………………………………………………………….**7**

5- Literature Review ………………………………………………………………………………….…….**7**

6- Design and Methodology …………………………………………………………………………..**14**

6.1- Equipment: Hardware and Software………………………………………………**14**

6.2- System Requirements ……………………………………………………………………**15**

6.3- System Initialization ………………………………………………………………………**16**

6.3- Database Selection ……………………………………………………………………….**17**

7- Implementation ………………………………………………………………………………………..**18**

7.1- Database pre-processing ………………………………………………………………**18**

7.2- Model selection and training …………………………………………………………**19**

7.3- Main Fatigued Detection implementation ……………………………………..**20**

7.4- Modules and Resources ………………………………………………………………..**21**

7.5- Detecting facial landmarks with computer vision …………………………..**21**

7.6- Integrating Trained CNN model ……………………………………………………..**21**

8- Problems and challenges ……………………………………………………………………………**22**

9- Results ………………………………………………………………………………………………………**23**

10- Ethical and Privacy Concerns ……………………………………………………………………**24**

11- Conclusion ………………………………………………………………………………………………**25**

**Declaration**

I, Erik Ricardo Vasquez Chafla, hereby certify that all of the work presented in this report and all other related materials are my own. The information derived from the literature has been appropriately cited in the text, and a list of references has been supplied. No section of this dissertation has been presented previously for another degree or diploma at this or another institution

**Signature:** Erik Ricardo Vasquez Chafla

**Date:** 02/04/2025

**Abstract**

Accidents caused by fatigue are a serious concern in many industries, but particularly in transportation. Convolutional Neural Networks (CNNs) and facial landmark analysis have been used to create a real-time fatigue monitoring system that tracks indicators of drowsiness in order to address this problem. In addition to a CNN model trained to identify fatigue states from facial images in real time, the system uses computer vision techniques to measure eye aspect ratio (EAR) and mouth aspect ratio (MAR) for blink rate and yawning detection.

To improve model performance, the system incorporates pre-processing methods such as data augmentation, resizing, and rescaling. The system uses TensorFlow for classification and OpenCV and dlib for facial recognition in real-time. All processing is done locally on the device, and no facial data is saved to protect user privacy.

This study demonstrates how AI driven fatigue detection systems can improve safety while upholding morally and acceptable AI practices as accurate as possible.

**Introduction**

Fatigue-related events are a serious issue in the modern world, especially in industries like industrial labour, healthcare, and transportation that demand a high level of focus. The system that I made, trained and tested offers a real-time fatigue detection system that uses computer vision solve this problem. Eye closing and head tilting and yawns are the primary indicators of weariness or tiredness that the system continuously observes in the user's face. To help the user avoid accidents or a loss in performance, an audible warning is activated when fatigue is detected.

The main goal of this project is to create, train, and then deploy a machine learning model that can discriminate between fatigue and non-fatigue states. The algorithm is trained on a dataset of video frames/facial photos to identify important signs of fatigue. After training, the model is included into a real-time system that analyses visual data and predicts the user's level of fatigue. The goal of this research is to develop an effective, automated method for fatigue detection and prevention by combining machine learning with real-time visual analysis.

**Background**

Road safety is a huge concern on a global scale, as millions of traffic accidents occur each year for several reasons, including human error. One of these, driver fatigue, is a serious problem that is often overlooked despite having a significant impact on traffic accidents. Fatigue increases the chances of traffic accidents as fatigue affects a driver's awareness, response time, and decision-making abilities, it raises the risk of traffic accidents. As drowsy drivers might not take caution before contact, research shows that crashes caused by drowsiness are usually more severe. Since tiredness driving is a significant issue on an international scale, there are countless papers and studies on the subject.

**Problems Fatigue has with vehicle safety:**

[1] According to a report by the Royal Society for the Prevention of Accidents (ROSPA), a charity, 1,300 accidents in 2022 are estimated to have resulted in injuries due to driver fatigue alone. Compared to other accident categories, these collisions are extremely dangerous since they have a 50% greater chance of causing fatalities and serious injuries to victims. This is because crashes often happen when a vehicle is moving rapidly and continues to accelerate. When a driver is exhausted or sleeping, they may not brake or swerve, which can result in an impact at a dangerous speed.

Several countries have conducted research to better understand drowsy driving, with the American Automotive Association, aaaFoundation, conducting the first study in December 2016. [2] The purpose of this study was to learn more about the causes of fatigue driving and the backgrounds of drivers involved in fatigue driving incidents. The age of the driver, the time and place of the collision, and their sleep patterns were all examined in the study. 7,234 drivers who were involved in 4,571 crashes made up the study's final sample as the remaining 3,000 were either hospitalised with serious injury, under investigation during the time or data didn’t help with the study.

Medical professionals say that individuals between the ages of 18 and 64 require 7 to 9 hours of sleep, or more if they are young adults, sick, or trying to recover from sleep debt. Notably, drivers who get fewer than two hours of sleep are not fit to operate a motor vehicle. However, according to a “Centers for Disease Control” survey of 444,000 American adults, 35% of US people sleep fewer than seven hours a night, and 12% of Americans sleep for five hours or less. 35% of drivers are not obtaining the recommended amount of sleep to stay awake throughout the day.

According to the AAA report, crashes most frequently happened between 5:30 and 7:59 in the morning (12.4%) and between 9 and 11:59 at night (3%). Ages 35 to 54 were the most affected by fatigued driving, accounting for 30.2% of accidents. Young drivers also played a significant role in the number of collisions; those between the ages of 18 and 20 made up 11.4% and those between the ages of 21 and 24 made up 11.3%. Either sleep deprivation or a change in sleep schedule were the main causes of these crashes. Compared to drivers who slept for seven hours or more, those who slept for less than four hours had an 11.5-fold increased risk of being involved in collisions making them the biggest contributors.

Notably, 3,570 of the 7,234 drivers in the research were contributors—the reason or cause of the collision—while 3,655 were non-contributors. A driver who gets more sleep is less likely to be involved in an accident than one who gets fewer than four hours. The likelihood of a crash is 4.3 times higher for drivers who sleep 4–5 hours, 1.9 times higher for those who sleep 5–6 hours, and 1.3 times lower for those who sleep 6–7 hours.

[10] Between 2015 and 2019, fatigue-related crashes accounted for about 12% of road fatalities in Queensland. However, the true number may be higher as it can be challenging to determine whether fatigue is a contributing factor.  According to research by Queensland’s government, staying up past 17 hours can affect one's ability to drive in a similar way as having a blood alcohol content above 0.05. Fatigue is particularly dangerous for drivers as it can cause microsleeps, which are short, involuntary bursts of sleep that last anywhere from a fraction of a second to ten seconds.

**Effects of fatigue:**

Exhausted drivers are frequently involved in crashes on long motorways and highways between midnight and six in the morning. They are also common during the day, especially when the driver has eaten lunch or had a single alcoholic beverage, between two and four in the afternoon, if the driver has had little to no sleep before driving, if the driver is intoxicated, they are more likely to fall asleep, if they are taking medication that causes fatigue, and most frequently, because they work long hours, particularly if they are active during the day, work night shifts, and attempt to drive home.

[1] In 2023, a controlled and small-scale investigation was carried out in India. To find out how exhausted drivers would respond to dangerous situations, a research team put 50 participants through a 30- to 35-minute driving simulation in three different settings. The three experimental circumstances included driving seven to eight hours after a week of the usual amount of recommended sleep, driving again after one day of sleep deprivation (between 3.75 and 4.75 hours), and driving again after two days of sleep deprivation. Two hazards would appear at random points throughout each driving simulation: a pedestrian crossing the street and other cars merging into the driver's road. The study will include the driver's entire braking time as well as their reaction time to the hazard.

The driver's reaction time increased by 10% in the pedestrian crossing hazard after one day of sleep deprivation and remained unchanged after two days. After the first day of sleep deprivation, the drivers' total braking time dropped by 25%, and after the second day, it dropped by 28%. After the first day of sleep deprivation, the driver's reaction time and braking time were 44% and 17% slower on the second, when they had to deal with vehicles merging into their road. The fact that delayed reaction times increased significantly after only one day of sleeping less than advised demonstrates how dangerous exhausted driving can be.

According to the ROSPA survey, young male drivers, truck drivers, business car drivers, and shift workers were the most likely to be involved in collisions because they were fatigued from their jobs. Compared to other vehicle types, commercial drivers—particularly those operating heavily loaded vehicles—had about 40% of sleep-related collisions and two-thirds of drivers fell asleep while operating a motor vehicle. Male drivers make about 85% of those involved in sleep fatigue collisions, and more than one-third are under 30. A study of 2,170-night shift workers found that 57% of them had been in an accident or near-miss while driving home after finishing their shift, and 84% of them claimed they were too exhausted to drive home.

[10] In order to prevent accidents, it is critical to recognize the warning indications of fatigue. The following are typical signs of fatigue, according to a Queensland government article:

* Yawning
* Eyes closing for a moment
* Blinking more than usual
* Feeling drowsy, exhausted
* Having trouble keeping you head up
* forgetting the last few minutes spent behind the wheel
* starting to hallucinate
* starting to hear buzzing or humming in ears
* stiffness or experiencing cramps
* aches and pains
* daydreaming
* noticing slower reaction times
* when driving and tired, you may change speed randomly for no reason
* If driving manual car when tired, you may fumble when changing gears
* Drifting the vehicle or going over the lane lines

**Technology to identify fatigue**

In modern times artificial intelligence (AI) and machine learning is at the core of most technology and growing, transforming numerous industries by allowing their systems to learn data and make better decisions. Machine learning algorithms can identify patterns in data and utilize that information to make judgments on their own. The more data they have, the more effective they will be at making decisions. Artificial intelligence (AI), which is currently being utilized by millions of people and expanding quickly, will follow from this. This includes techniques such as [3] “natural language processing and computer vision - the ability for computers to use human language and interpret images - to automate tasks, accelerate decision making”.

Artificial Intelligence (AI) and Machine Learning (ML) have become effective technologies for real-time driver monitoring and tiredness detection in response to this problem. With Advanced Driver assistance Systems (ADAS), which include lane-keeping aid, speed regulation, and danger recognition, AI-powered technologies are being incorporated into contemporary automobiles to improve road safety. However, most current systems rely on indirect vehicle-based measurements such steering wheel movement or lane departure which isn’t accurate when the issue is driver, human error. These techniques frequently fall short in identifying fatigues in the early stages which leads to road accidents due to the driver.

**Aims:**

The goal of this project is to create a real-time system that can identify driver fatigue by analysing head motions and facial expressions using computer vision and machine learning algorithms. The device will use a camera-based method to identify early signs of tiredness, including head tilting, gaze direction, yawning frequency, and blinking patterns. The driver will then receive an alert from the system before their degree of weariness becomes unsafe.

**Literature Review**

AI and ML are now used in numerous products, such as smartphones, refrigerators, watches, homes, and even vehicles. Using AI, vehicles now have advanced driver assistance systems (ADAS) which significantly decrease the number of accidents, this is done by implementing lane keeping assistance, speed regulation and hazard detection. However, even with today's smart vehicles they still lack in addressing human error/issues. Millions of people die in auto accidents worldwide each year, with vulnerable road users such as cyclists and pedestrians contributing to most of these fatalities.

These include [4] drunk driving, which accounted for 6.1% of all car accidents in Great Britain in 2023 causing 4,089 accidents; driving while under the influence of drugs causing 1,853 accidents annually in GB; distracted driving, which is typically caused by mobile devices (even with hands-free phones not making it safer) causing 463 accidents; aggressive driving, which contributed 0.9%; and drowsiness driving being a critical but overlooked factor, which will be the main focus of this project.

Accidents caused by driver fatigue is a major concern that is responsible for thousands of accidents and deaths annually. When driving while being drowsy, the driver’s reaction time, decision making and awareness is significantly impaired. [5] In a research article that was based in China in 2019, “according to the statistics of the National Bureau of Statistics of China, traffic accidents caused by fatigue driving account for >20% of the total number of traffic accidents and account for >40% of serious traffic accidents”. To further research and combat fatigue driving, an automatic detection system was created and categorized into 3 methods: physiological bases which monitors heart rates using sensors, vehicle based which analysis the steering wheel movement and lane control, and vision-based approaches which uses cameras and AI algorithms to monitor the drivers’ facial expressions and head movements.

physiological baseswere implemented by monitoring drivers in real time through the ab embedded sensor on the steering wheel by Jung et al.Through the sensors the drivers state could be categorized as “normal, fatigued and drowsy states” by monitoring the drivers heart rate. Although not widely used, this method has been implemented into some smart cars to monitor a driver’s heart rate and alert the driver if they have been categorized as fatigued driving, although not 100% accurate. There will be times where the smart car will mistake the driver for fatigue and lock the steering wheel (like lane assist). By using the sensors to monitor heart rate, there is an 83.6% accuracy.

The physiological approach has drawbacks despite its high level of accuracy. The fact that there are several variables that may not be connected to fatigue is a major disadvantage and the primary cause of physiological approaches' lack of accuracy. For example, anxiety, caffeine consumption, under the influence of drugs or alcohol or underlying medical issues that the driver has might affect physiological signals, which can lead to uneven accuracy and false positives for exhaustion, leading to an accuracy of 83.6% instead of 100%.

An additional disadvantage of the physiological based approach is the expensive and complicated cost of the necessary technology. Although a small number of smart cars have begun to employ them, the concept of physiological monitoring is not adopted in most smart cars due to the complicated and expensive technology required, which raises the product's price. Due to the necessary costs, owners of smart cars that have embraced this concept now find them expensive, which makes drivers reluctant to spend a lot of money which will not help reduce fatigue related driving accidents.

Furthermore, this approach has given rise to privacy problems. Drivers worry that their personal information is being utilized and retained without their consent. Some drivers are hesitant to have sensors in their cars because they fear that their biometric information will be exploited without their permission.

The second method was vehicle-based detection, studied by Ma et al, which relies on the way the vehicle is being operated by the driver. The data collected for this method was gathered from steering wheel movement, vehicle speed and lateral distance. By using a neural network to analyse the vehicles behaviour, they could detect a pattern for detecting drowsy driving. The average accuracy of this method was 78.01%.

Monitoring the steering wheel movement, lane departure, acceleration, and braking patterns was how the vehicle-based approach operated. This strategy is simply indirect as external factors that are not frequently related to driver fatigue. For instance, weather conditions, road conditions, or personal driving preferences can affect how a car is being operated. Strong gusts of wind and rain, for example, may result in frequent steering adjustments, which could result in false positives when detecting fatigued driving.

The system's ability to identify driver fatigue late was another major flaw in the vehicle-based approach. The system's effectiveness to avoid fatigue related accidents is reduced since changes in vehicle behaviour are usually only noticeable when the driver is already extremely exhausted. Furthermore, the driver is the only one who shows early signs of exhaustion, which are entirely overlooked. This involves factors such as subtle head movements or slower blink rates. The inability of the vehicle to collect this data reduces the system's efficiency to detect exhaustion. The driver may have already been involved in an accident by the time the system detects their weariness. Because of these factors, the research team Ma et al. was only able to obtain an accuracy of 78.01.

The third method that was used to identify fatigued drivers is vision-based This is done by using cameras and machine learning algorithms for capturing and analysing face visuals while driving. The data that would be gathered from the camera include the number of blinks in a short amount of time, yawning, head movement such as leaning and gaze direction. This method was studied and implemented by Mandal et al to monitor bus drivers for fatigue driving. The system had an accuracy of 98% when combined with the algorithm that used the support Vector Machine algorithm that was proposed by Alioua et al.

Despite the method being non-intrusive and closely related to human behaviour, this approach has a variety of drawbacks in its present implementations. One major issue is environmental conditions, which can lower their dependability because they are greatly impacted by lighting conditions, including glare, shadows, and driving at night. Furthermore, these systems have trouble detecting face features accurately due to obstructions and distractions like masks, eyeglasses, and facial hair. Due to biases in the training data, models trained on demographic groups could not perform well on a varied population, which is another major drawback. The study group Mandal et al achieved 98% accuracy using vision-based detection, but performance decreased significantly under varying lighting conditions and occlusions.

These three techniques are currently the most widely used in technology to prevent fatigue driving, and they are being applied in a variety of ways by several researchers. For example, PERCLOS (percentage of eye closure) was created using the vision-based approach. The research team determined that a driver's blinking frequency and duration were the most significant signs of early fatigue, which is why they only implemented eye closure frequency. They avoided wasting time, money, and energy creating and training algorithms to detect additional factors by concentrating on only the one.

[6] This was developed and tested by Zhang, Jianping, et al. to assist air traffic control, who might be fatigued while instructing pilots where to steer their aircraft. Air traffic control officers are essential to the operation of an aircraft, despite not being the drivers, yet they typically suffer from a significant degree of mental exhaustion. Because of its excellent validity, reliability, and noncontact use, the research team decided to utilize only the percentage of eye closure to assess weariness. The research team Ma et al.'s vision base differs from this groups in that they do not seek to determine the quantity of blinks. In contrast to Ma et al., who examined the quantity of blinks, this team examined the quantity and percentage of blinks, allowing them to measure slow eyelid closures instead of actual blinks.

To collect the officers' visual data, the team utilized a laptop equipped with a camera, a multi-task convolutional neural network (MTCNN-CNN) model was then employed by the group. In computer vision, the MTCNN-CNN model is frequently used for precise face alignment and detection. P-Net (Proposal Network), R-Net (Refine Network), and O-Net (Output Network) are the three cascading networks that make up the Multi-Task Cascading Convolutional Neural Network. In order to perform accurate face detection, these networks first create candidate face regions, then refine the bounding boxes that are found. High accuracy and speed are achieved with the effective removal of non-face regions made possible by focusing on just the eyes.

[7] Following face detection, the generated face images are then sent to a CNN (Convolutional Neural Network) for classification or facial recognition tasks. MTCNN is perfect for applications like facial recognition systems, face verification, and emotion detection because of its multi-task learning capabilities, which allow it to concurrently recognize faces and landmarks. It is a common option in contemporary computer vision systems due to its great efficiency and precision.

[6] The model was proven to identify PERCLOS in the field by combining a convolutional neural network for PERCLOS calculations with a multi-task convolutional neural network for eye detection. To improve the system a parallel computing mechanism, the method's robustness was enhanced, and the group employed the P70 index as a metric in this work to adapt it to field research. P70, which is used as an exhaustion metric when evaluating PERCLOS data, is defined as the amount of time spent with the eyelids closed for at least 70%.

To get the data needed to train the system and make it as precise as possible, the researchers collected image data of the ATCOs' faces and employed a computer program that extracted the PERCLOS values for each image using the MTCNN-CNN model. After the collected image was loaded into the computer program, the participants' faces and eyes were recognized using the MTCNN model. The image was reduced to 26 × 34 to remove extraneous elements and cropped to just include the eye as the region of focus for fast, accurate computation.

Every picture involving just eyes was turned to black and white to make processing the visual data easier. The CNN model was then used to identify the eye status, assigning an open or closed PERCLOS status to the image and computing the PERCLOS based on the P70 value. The team implemented a parallel computing system for real-time performance to improve results. This made it possible for one thread to record video while another processed pictures at the same time, improving the system's ability to identify exhaustion by 28%. From August 26 to August 30, 2019, the team gathered 69 healthy ATCOs from one of the busiest control centres in China to test the system. They will all be evaluated at four different times: morning (7:30 - 8:30), afternoon (11:30 - 12:30), evening (17:30 - 18:30), and midnight (23:30 - 00:30). All of them had Class A medical certificates. To collect data before and after shifts, participants faced a camera for five minutes each. The video data was recorded at 24 frames per second.

According to the study's findings, ATCOs who worked 8:00–12:00 (shift 1) and 12:00–18:00 (shift 2) had significantly higher levels of weariness than those who worked night shifts (shifts 3 and 4) from 18:00–00:00 and 00:00–8:00. From shift 1 to shift 2, fatigue stayed constant; it then rapidly climbed in shift 3 before declining in shift 4. This indicates that pre-shift weariness rose steadily from morning to night and peaked during the third shift.

[8] Another team of researchers, under the direction of Dr. Christos Padadelis, a professor, decided to use brain activity monitoring to assess how alert or sleepy a driver was. The research team employed electroencephalography (EEG), a physiologically based method. EEG and peripheral physiological data were collected from sleep-deprived patients during nighttime driving in an experimental vehicle as part of the study. The team focused on microscopic event analysis by concentrating on physiological changes of the driver during driving incidents or errors and macroscopic analysis by tracking physiological measurements over time while the driver was driving, which were used in the data analysis.

Healthy drivers who were sleep deprived had their EEG and peripheral physiological data recorded while they were operating a real car on the road at night. With an average age of 33 ± 10 years, 21 seasoned drivers, 20 men and 1 woman, participated in the experiment and were monitored while driving. According to the report, all participants had an average of 12.28 ± 8.66 years of driving experience, normal or corrected vision, no history of neuropsychiatric problems, and no medication use. To obtain the most accurate data, this was the criterion for each participant.

Prior to the experiment, each participant was sleep deprived for a minimum of 24 hours. These physiological measurements, together with information from two on-board systems, the Lane Detection System (LDS) and the Eye Leads Sensors (ELS), were used to continually monitor the individuals' levels of fatigue and drowsiness in the experimental vehicle. Wearing sensors like electroencephalograms to track brain activity, electrooculogram to detect eye movements and blinks, and electromyograms and electrocardiograms to track heart and muscle activity was required of the drivers in order to observe their physiological behaviours.

The experiment was conducted at the “Center for Research and Technology”, Hellenic Institute of Transport in Thessaloniki, Greece. The participants had to drive a 100 km route from Thessaloniki to Veria and back while following traffic rules. Once completed two independent neurophysiologists visually interpreted the data to make sure the test was fair and accurate. The result of this method was that in the driver’s brain wave frequencies, significant variations were observed in the delta, alpha, beta, and gamma frequency bands. there was an increase in alpha and delta activities and decreases in beta and gamma activities as driving time increased.

When alpha activity rises, it can represent a sign of drowsiness, even when a person is still awake, their brain goes into a relaxed/restful state. The slowest brainwaves, known as delta waves, are usually linked to deep sleeping. A significant drop in alertness and a shift to sleep-like brain activity can be detected by an increase in delta activity; this is an obvious indicator of growing fatigue. Performance, alertness, and focus are linked to beta and gamma waves. A decline in these activities signifies that the brain is losing concentration and engagement which means that the driver might be having trouble concentrating driving.

Even though this technique can collect real-time data on a driver's level of weariness, its drawbacks prevent its use in real life. The cost of purchasing and maintaining the gear and sensors is the biggest problem. At the moment, this approach is not cost effective and causes discomfort for the driver. The system required the drivers to wear a helmet-like device that stored all the sensors and a sensor that ran across their chest. Wearing this will be uncomfortable, especially on long journeys, and putting on and taking off the sensors requires a lot of time, making it impracticable for quick trips.

[11] After looking into many camera-based fatigue detection systems, I've seen a tendency toward the use of dlib. Dlib is a C++ software library that is open-source and cross-platform, offering a variety of computer vision and machine learning techniques. Davis King developed Dlib, which was intended to be extremely modular, fast to run, and easy to use through a modern, clear C++ API. Although Dlib is used in many other fields, its primary function is object identification for computer vision and machine learning applications, which is essential for camera-based detection. It offers effective implementations of deep learning, object tracking, face detection, and image processing methods. In fatigue detection systems, dlib is being used for face detection and recognition by using a powerful Histogram of Oriented Gradients (HOG) based face detector and a deep learning CNN face detector.

[12] OpenCV is another module that frequently works with dlib in fatigue detection systems. For real-time image processing and analysis, OpenCV is an open-source computer vision and machine learning package. OpenCV can be applied to robotics, object detection, facial recognition, and other systems. OpenCV's primary features for fatigue detection systems include image processing, object and face detection, feature detection, integration with machine learning models like KNN and SVM, and support for integration of deep learning models.

For more precise face detection and tracking of facial landmarks such the eyes, nose, mouth, eyebrows, and jawline, OpenCV is frequently used in conjunction with Dlib. Users may now track their face in an image or camera, as well as their head posture, blink detection, yawns, and facial alignment, by combining Dlib and OpenCV. The user must first identify a face in an image, which may be accomplished with ease using OpenCV's pre-trained SVM object detector or the deep learning face identification feature included in dlib. While OpenCV is less accurate than dlib's face detection and dlib is better at determining region of interest (facial landmarks), it is still preferable to combine the two techniques for a more accurate face detection.

After recognizing a user's face, the computer must identify facial landmarks by concentrating on the following areas of interest: the nose, eyes, eyebrows, and jaw. The software will then be able to recognize these particular spots. The dlib library has an open source, pre-trained facial landmark detector called shape\_predictor\_68\_face\_landmarks that can identify 68 points or coordinates that map facial features.

A screenshot of a puzzle

AI-generated content may be incorrect.

Figure 1 – demonstration of facial structure being mapped with shape predictor 68 face landmarks.

As shown in figure 1, the pre-trained facial landmark detector is distributing the 68 points to just the points of interest that I mentioned before:

* Jawline – Points 1–17
* Eyebrows – Points 18–27
* Nose – Points 28–36
* Eyes – Points 37–48
* Mouth/Lips – Points 49–68

[9] After some more research, I can see that datasets plays a key part when creating systems that can detect fatigue. Datasets are crucial as they contain visual data which users can use to train a model to get a better accuracy when detecting fatigue. The most popular dataset being the NTHU Driver Drowsiness Detection Dataset (NTHU-DDD) due to its wide range of visual data, providing high accuracy to machine learning models and it is a publicly available, free dataset. This extensive dataset was created to aid in the development of systems that can identify driver fatigue. Video recordings and images taken in actual driving situations make up this type of data. Drivers are seen in these recordings displaying symptoms of exhaustion, including head motions, yawning, and eye closes. Using this dataset, users have been able to train algorithm’s that will allow machines to identify fatigue related head movements, blinking patterns, yawns, and facial expressions.

**Methodology: Fatigue detection system with machine learning model**

After researching all the different methods and ways that other groups have used to detect fatigue, I have chosen to use a video-based method as this method seems to frequently be the most accurate, is the most cost effective and can capture early signs of fatigue quicker compared to the other methods. Using the information research in the literature review I can plan to create and train a system that can detect fatigue.

**Hardware Equipment**

The equipment needed to create a fatigue detection system is:

* Camera - A webcam or the built-in front camera of a laptop will be crucial for gathering the visual data (images and videos) of the driver’s face. The camera will gather the data needed to detect fatigue.
* Speaker/Buzzer - An audio output device will be needed to warn the driver with an audible alert if the system detects fatigue. A speaker, buzzer or a built-in speaker can be used for this.

**Software equipment**

* OpenCV – will be applied to face detection and feature extraction in real-time image and video processing.
* Dlib - Dlib has a strong face detection and facial landmark detection model to improve the model’s accuracy
* Tensorflow - Will be used to develop and implement deep learning models that use facial features to categorize different levels of fatigue.
* shape\_predictor\_68\_face\_landmarks.dat - Contains 68 important facial landmarks—such as the eyes, nose, mouth, jawline, etc.—that are necessary for feature extraction are detected by this pre-trained model file for Dlib.

The system's design focuses on real-time performance and high-accuracy detection. It is divided into five primary modules: video capture, face and landmark detection, feature extraction, state classification, and an alert system.

1. **Video Capture Module**: This module will use the camera to record live video frames of the driver.
2. **Face and Landmark Detection Module**: This module will detect facial landmarks, including the eyes, mouth, and head orientation, which are crucial for identifying signs of fatigue.
3. **Feature Extraction Module**: This module will compute essential measures such as the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR), which are key indicators for detecting yawning and drowsiness.
4. **State Classification Module**: Using machine learning models, this module will classify the driver's state (fatigued or alert) based on the extracted features.
5. **Alert System**: The system will include an adaptive alert system that provides context-aware notifications, ensuring the driver receives a timely warning without unnecessary distractions.

**System Requirements:**

Establishing the system's main needs is crucial before delving into the fatigue detection system's construction and modules. These specifications list the technical and functional requirements that the system must fulfil in order to function properly. The essential elements that follow should be present in the system:

* Load and process a dataset: The system must be able to load in a dataset and process the visual data/image frames into a format that can be used in an algorithm.
* Face detection: The system must be able to detect the user’s face from a camera or web camera and accurately locate the position of the user’s face and draw a boundary line around it.
* Facial landmark detection: The system must be capable of detecting key facial landmarks, including the eyes, mouth, and nose. This will help in tracking eye movement, blinking patterns, and yawning, which are critical for assessing fatigue.
* Work in real time: The system must be able to operate in real time with little delay. As the purpose of this program is to detect fatigue in a driving simulation, it must be able to process the video data quickly and predict whether the user is fatigued or not in real time.
* Feature Extraction: The system must be to use the landmarks that were detected, the system must calculate facial features such the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) in order to measure indicators of exhaustion or fatigue (such as yawning or eyelid closure).

**System Initialization**

A diagram of a driver

AI-generated content may be incorrect.

Using vision-based techniques, the driver fatigue detection system is intended to continuously observe the driver's face and detect signs of drowsiness. To evaluate the driver's level of awareness, the system captures video data of the driver's face using a camera, processes the data to identify facial landmarks, and examines the movements of the mouth and eyes. The technology warns the driver by sounding an alert if it detects fatigue.

The web camera inside the car or similar simulation must be positioned and calibrated correctly for the system to work. To get a clear picture of the driver's complete head and face, the camera has to be mounted on the dashboard or next to the rearview mirror or in a similar environment. The camera placement is essential as it minimizes driving distractions while providing the best visibility. To allow for constant and more precise data processing for driver monitoring, the frames must be captured and analysed at a set pace.

I'll use the OpenCV computer vision library to create the program in Python. I have decided to utilize OpenCV since it would enable real-time image and video frame processing in the system. The system's real-time video frame reading will be done by OpenCV, which first converts all the visual frames to grayscale (black and white). The frames must be converted to simplify image processing, which will make it simpler for the system to handle the data and increase accuracy since it will only need to concentrate on one colour instead of the original three (RGB).

OpenCV can identify the driver's face specifically in webcam image frames by using deep learning-based DNN models. The system will begin searching for the region of interest as soon as it recognizes the driver's face. I intend to focus on the driver's entire head, while paying particular attention to the eyes and lips to identify PERCLOS, eye blinking frequency, and yawns, in comparison with earlier study groups that employed vision-based techniques that only focused on the drivers’ eyes.

Using Dlib, a library that my supervisor suggested, I will be able to access its trained facial landmark detection, expression, and eye blink analysis. Dlib offers machine learning algorithms as well as tools for computer vision and image processing. My primary focus will be on the eye, nose, mouth, and jawline when OpenCV is included. This will help me follow the movement of the mouth, jawline, and eye which I use to calculate the eye aspect ratio and mouth aspect ratio, which will be utilized as a metric system.

With the use of these metrics such as the eye aspect ratio, I can use the landmarks surrounding the eye to develop a blink detection. The system will indicate that the driver is feeling tired if it determines a low eye aspect ratio across a set of frames. Using the landmarks surrounding the mouth and the mouth aspect ratio, I will be able to determine when the driver's mouth is open. This means that if the user leaves their mouth open for several seconds on numerous occasions, the algorithm will be able to determine when they are yawning. Both metrics can also be used to calculate the driver's head movement and position by tracking landmarks on the nose, eyes, and mouth. If the user starts to nod their head, the system will be able to detect it.

If the system detects fatigue, then an alert will be triggered. An audible alarm will be played through a speaker or buzzer to warn the driver and wake them up if they are asleep, all in real time.

**Database Selection**

In order to develop an algorithm model and train it to detect and predict a user's condition (fatigue or alert), as well as to estimate the accuracy of the user's fatigue, I will require a dataset for my fatigue detection system. I plan to get the visual data from a free, publicly accessible dataset in order to train the algorithm for fatigue. I had originally intended to use the NTHU Driver Drowsiness Detection Dataset (NTHU-DDD), which is publicly available and free, the most widely used and advised dataset for fatigue detection, has a lot of visual data, and includes multiple factors that can determine fatigue, all of which can improve the accuracy of the model. However, because of a few changes, the dataset could only be accessible by requesting permission from the dataset developer, which could take too long and not fit into the timeline for developing the fatigue detection system.

[13] After doing some investigation, I discovered a publicly available dataset that is free of charge and includes visual data of people that are both alert and fatigued. The Drowsy Detection Dataset "Stay Awake, Stay Safe: A Dataset for Drowsiness Detection" was located on Kaggle. The dataset was more helpful to me even though it is much smaller than the NTHU-DDD since smaller datasets are easier to use and can be loaded and processed into the fatigue detection system more quickly.

In addition, the visual data is already divided into data that will be used for model testing and training, which makes this dataset more useful for my fatigue detection than other datasets. The dataset will be used to assess the accuracy of fatigue categorization, validate feature extraction techniques, and train machine learning models. Through the use of a dataset, I can guarantee that the model is trained on a variety of realistic, accurate, and diverse facial images, allowing it to recognize trends in face features associated with fatigue. I can also guarantee that the visual input has variety by using the Kaggle dataset, which will allow the model to identify various people in spite of variations in appearance, lighting, and camera angles.

**Implementation**

**Dataset pre-processing**

As the data is now unsuitable for the model due to its image nature, which renders it difficult to read by the system, pre-processing the dataset is essential. Rescaling the dataset was the first method utilized at this stage. The pixels in each image ranged from 0 to 255, which is why I had to rescale the dataset. The model will suffer because of the algorithm's wide range since it would have to modify its weights to account for the huge inputs, which would make the model sluggish or unstable. Therefore, the data must be scaled down to a limited range (0 to1) in order to perform better. This will increase the neuron network's speed and stability and avoid frequent huge weight updates.

Data augmentation is used in the pre-processing step to improve the model's generalization and robustness. Deep learning models need a lot of different training data to function properly, so data augmentation helps to grow the dataset by making modified versions of pre-existing images. This keeps the model from overfitting, which is when it memorizes the training data rather than learning patterns that are to generalize which would negatively affect the model

Rotation, zooming, flipping horizontally, and rescaling are the augmentation techniques that have been used. The rotation range is set to 20 which helps the model identify faces at various angles by randomly rotating images by up to 20 degrees. Zoom range is set to 0.2 to simulate how close a driver's face might look in real-world situations by allowing for a small amount of zooming in and out. In order to learn symptoms of fatigue of the driver's regardless of their orientation, horizontal flip is set to true to do this by replicating the image. Finally, for quicker and more reliable training, I rescaled the pixel to 1/255 which will guarantee that pixel values stay inside the 0,1 range. Through the use of these augmentation strategies, the dataset is made more representative and diversified, which aids the model in learning to identify fatigue in various lighting conditions and orientations, ultimately increasing its accuracy in real-world situations.

To properly assess the model's performance, the dataset is divided into training and testing sets once the images have been pre-processed. The neural network learns patterns and correlations between the input features and the target labels (fatigued and alert) by using the training set. The purpose of the testing set is to evaluate the trained model's performance on new data, making sure it does not only memorize the training images but also generalizes well to new data.

20% percent of the dataset is utilized for validation, while 80% percent is used for training. This validation will monitor the model's performance during training and avoid overfitting. After the model is fully trained, the testing set, which is separate from the training process, is used to assess the model's accuracy and robustness.

The dataset's images may have different aspect ratios and resolutions, so resizing all the images to a fixed size (128x128 pixels) is an essential preprocessing step because models need a constant input size in order to process data effectively. The model would have trouble handling various image sizes if the images weren't scaled, which could result in error and inconsistencies during training. resizing all images in the dataset guarantees that every input has an equal number of features by standardizing all images to 128x128 pixels, which improves computing efficiency and avoids form mismatch problems.

**Model selection and training**

Support Vector Machines (SVM) was the model I initially chose to employ for the fatigue detection system. SVM is a powerful machine learning algorithm that is used for classification tasks and performs particularly well with small-scale datasets, which is why I wanted to employ it. Another reason I initially decided to use SVM was because, according to my research in the literature review, SVM was the algorithm most employed in various fatigue detection methods to identify fatigue and provided positive results.

[14] But after more investigation and analysis, I choose to employ a Convolutional Neural Network (CNN) rather than SVM. CNN is also better suited for my fatigue detection since it can automatically learn hierarchical features from images, which enables it to pick up on complex patterns like drowsiness-related micro-expressions, facial muscle activity, and expressions. Second, after selecting my dataset, I realized that SVM was unable to process large datasets, so I would need to pick an algorithm that could. Since CNN employs several layers to extract patterns, it meets these requirements.

In order to interpret images of faces and categorize them as either alert or exhausted, the Convolutional Neural Network (CNN) model for fatigue detection was created. In order to extract and learn characteristics from the input images, the model is made up of several layers, each of which carries out different responsibilities. The model starts with the input layer that accepts 128 x 128 RGB (red, green, and blue) images as input.  The input shape, which is defined as (128, 128, 3), guarantees that all images are consistent before the network processes them, enabling reliable feature extraction and effective training.

In order to extract and learn characteristics from the input images, the model is made up of three convolution layers and followed by a max-pooling layer. These layers reduce the spatial dimensions of the input images while extracting significant characteristics. 32 3x3 filters are applied in the first layer to identify fundamental patterns like corners and edges. The model can learn detailed correlations by introducing non-linearity through the application of the ReLU activation function. After that, a 2x2 max-pooling layer reduces the feature maps' spatial size while keeping crucial data.

The second convolution layer detects complex facial structures by expanding the number of filters to 64 and the feature size is further reduced by an additional max-pooling layer. Variables such as head posture, yawning, and eye opening are extracted by the third convolution layer, which has 128 filters. The CNN can automatically learn and identify important facial traits associated with fatigue thanks to these layers.

The model uses a flattening layer to transform the multi-dimensional feature maps into a one-dimensional vector following feature extraction. The data is then ready for categorization in the fully connected layers thanks to this modification. With 512 neurons, the first layer uses the ReLU activation function to identify patterns in the data. By randomly deactivating half of the neurons during training, a 50% probability Dropout layer ensures that the model generalizes effectively to unseen data and prevents overfitting.

The output layer, the last stage of the model, is made up of a single neuron with a sigmoid activation function. A probability score between 0 and 1 is produced by this function, where values nearer 0 will predict an alert driver, and values nearer to 1 predicts a fatigued driver. The model can produce precise and trustworthy predictions thanks to this binary classification method.

A batch size of 32 is used to train the model over ten epochs, and performance is tracked in real time so that learning parameters can be modified as necessary. Once training is complete, the model undergoes evaluation on a separate test dataset to measure its accuracy on unseen images. This ensures that the model generalizes well beyond the training data. The trained model is then saved as 'fatigue\_detection\_model.h5', which enables it to be integrated into the file that employs computer vision and a camera to track a driver in real time.

**Main Fatigued Detection implementation**

The main.py file is in charge of real-time fatigue detection by combining the trained CNN model, computer vision, and facial landmark detection. The script analyses eye and mouth movements, extracts face features, examines webcam video frames, and predicts a user's level of fatigue.

**Modules and Resources:**

Important libraries like OpenCV, dlib, NumPy, TensorFlow, and time are first imported to the file. These libraries are used for deep learning inference, image processing, face landmark detection, and time calculations. To determine whether a user is alert or exhausted, TensorFlow loads the trained CNN model (fatigue\_detection\_model.h5). Additionally, the dlib shape predictor (shape\_predictor\_68\_face\_landmarks.dat) is loaded to identify and extract important facial landmarks like the mouth and eyes.

**Detecting facial landmarks with computer vision:**

I used my laptop's built-in webcam to apply computer vision to my fatigued detection system. Real-time user video frame capturing is carried out with the function cv2.VideoCapture(0). Using dlib's frontal face detector, the system must first identify the user's face before it can identify facial landmarks. Now that the user's face has been detected, the algorithm can extract 68 facial landmarks, which are essential for analysing eye and mouth movements, using the shape predictor file.

This allows me to compute the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR), which are used to assess user fatigue. To determine whether the user is blinking or keeping their eyes closed, the Eye Aspect Ratio is computed. Closed eyes are indicated by a lower EAR value, which may indicate weariness. Another indicator of tiredness is yawning, which is detected by computing the mouth aspect ratio.

The system employs a number of factors, including blink detection, to determine the user's level of fatigue. The system employs EAR to detect blinks. A blink is recorded if the EAR drops below the 025 threshold. Fatigue is suspected if the user blinks excessively (more than 20 times per minute). Prolonged Eye Closure is an additional characteristic. The system detects that the user is sleeping if their eyes are closed for longer than ten seconds and shows a warning. Additionally, the system will employ MAR to identify the user's yawns. If the MAR is greater than 0.50, the system will recognize yawning and issue a fatigue warning. Using cv2.putText(), each detection is shown on the screen, warning the user if they are yawning, blinking excessively, or are fatigued.

**Integrating Trained CNN model:**

The script classifies the user's state of fatigue in real time using the pre-trained CNN model in addition to the facial landmark analysis. To match the input size required by the CNN model, the face region is taken from the detected face and resized to 128×128 pixels. To improve model performance, the image is normalized (divided by 255.0) to guarantee that pixel values fall between 0 and 1. The CNN model receives the processed face image and uses it to predict the user's level of fatigue. If the prediction probability is less than 50%, the system classifies the user as fatigued; if it is greater than 50%, the user is considered alert. The final classification result is then shown on the screen.

**Problems and Challenges**

Despite showing promising outcomes, the fatigue detection system has a number of drawbacks and difficulties that limit its functionality and practicality. The fatigue detection system was developed in a short amount of time, which is the main problem. This has resulted in issues like managing various illumination conditions. Because it is more difficult to recognize eye features and face landmarks in low light or overexposed situations, the model's accuracy declines.

Another disadvantage was that I did not have enough time to create all of the features I had originally intended to include. The trained model, EAR for (PERCLOS) blink frequency, is currently used by the fatigued detection system to time eye closure, and MAR for yawn detection. Despite the fact that these features have produced great results, I was unable to incorporate eye gaze tracking and head movement tracking due to time constraints. The detecting system may have achieved a higher and better accuracy score if these fatigued features had been included.

Additionally, it appears that the user's head posture and camera angle are causing some issues with facial recognition. The system may have trouble if the user's head is tilted or partially turned but it works best when the user's face is directly in front of the camera or slightly angled. The efficiency of eye aspect ratio (EAR) and mouth aspect ratio (MAR) computations gets reduced by side views or extreme angles. Due to head positions and camera angle, the system might not even be able to determine the user's level of exhaustion as the face detection feature might not be able to identify the user's face. If the face is not identified, facial landmarks cannot be monitored, rendering MAR and EAR null.

Receiving false positives and false negatives was the final difficulty I encountered when developing the fatigued detection model. The system occasionally generates false negatives (failing to detect fatigue when the user is drowsy) and false positives (detecting fatigue when the user is alert) even after adjusting the EAR and MAR thresholds. Individual variations in eye shape, blinking patterns, and yawning patterns are mostly to blame for this. In order to counteract this, I had to modify the EAR and MAR thresholds and time EAR and MAR using the time library to obtain the most accurate results. Although rarely, I still get some false results.

There is also a problem with MAR making it not perform as well as it was intended to. The system will consider the user to be yawning every time the user opened their mouth. This means that if the user was talking or opened their mouth just for a second then the system will consider the user as yawning. The system also results in false positives when the user’s mouth was closed due to camera angle and the differences in different user’s lips. To combat this in the future I could have used the time module to set a timer for when the user’s mouth was opened. If the user’s mouth was opened for around 6 seconds (the average length of a yawn) or more then the system would consider that as the user yawning.

**Results**

I ran the software to test the system's performance once the tired detector was established. If a saved version was not saved, the system will load the database and do preprocessing to run the data in the CNN model to train. The CNN model would take about five minutes to complete, due to the amount of data in the database being inputted into it, and it would run through ten epochs if no saved version was made. On the other hand, if the CNN model was trained and saved, the system will load it into the main computer vision file in roughly two minutes.

The system will launch the machine's default camera (my laptop's built-in webcam) after loading the saved CNN model into the main file. After identifying the user's face, the application will display a green rectangle to show that it has done so. The algorithm then recognizes the user's eyes, nose, and mouth as facial landmarks after detecting the face. The points of these traits are shown on the screen once the system has identified these facial landmarks.

As soon as the system recognizes and shows the user's face and facial landmarks, it instantaneously and in real time identifies the user's level of fatigue and state (either alert or fatigued). This is shown on the screen of the webcam. I receive the status alert with an average percentage of 92.53% to 96.37% when I stare directly into the webcam, keeping my head straight and level with the camera, and keep my lips shut and eyes wide open. This is a positive outcome, and the system has given me a status and level that accurately reflects my lack of clear signs of fatigue.

Additionally, the system can record how frequently I blink and show the total number of blinks in the upper left corner of the camera screen. The system has so far produced no false positives in the position I started with, and it appears to accurately track all of my blinks in real time (depending on the machine signals, the speed of blink detection may be influenced). The blink counter resets to 0 after a minute. The system does not appear to mind if the user blinks a couple of time times but if the user blinks 20 times in less than a minute, it will interpret that as fatigue and display the message "User is fatigued." However, the state stays alert, and the fatigue level drops to 86.42%.

The system uses MAR to detect when the user opens their mouth to detect for yawns. When I opened my mouth, the system displayed the messages “user is yawning” which appeared to be correct. However, there was an issue with MAR, the system considers the user to be yawning whenever their mouth was opened. This was a problem as it meant that if the user slightly opened their mouth or was talking then the system would consider that as yawning. Another problem was that the system would result in false positives even when the user’s mouth was closed, this was due to the user’s lips. Due to camera angle or the characteristic of the user’s lips, the system would consider the users mouth to be open even when closed.

The system can still determine the user's state and level of fatigue even if they tilt their head slightly in either the horizontal or vertical directions meaning it can still track the user's face and facial landmarks. However, even though the user's face is still fully in the webcam, the system will lose sight of it if the user tilts their head close to 90 degrees, which would prevent the system from functioning.

The system gave me a fatigue level of 36.80% when I showed obvious symptoms of fatigue, such as lowering my head (with my face still in the camera) and closing my eyes. This indicated that I was now fatigued because my level was below 50%. After about 20 seconds of being in this state with my eyes closed, I received a level of 34%, and because my eyes were closed for 20 seconds, the system displayed the text "user is sleeping." During this time, I was receiving levels ranging from 36% to 48%, which represented that I was fatigued. This is a good result because I showed obvious signs of exhaustion and was given the right level of fatigue and correct state.

Numerous more individuals, including friends and family, volunteered to test the fatigued detection system and got similar results. With some small differences, the system produced findings that were positive and comparable to what I got. But when I tested with other individuals, I discovered a new problem. Due to differences in appearance, such as users with beards, glasses, hats, or hair that falls over their faces. Although it will have difficulty, the system can still identify the user's face and facial landmarks. The main problem appears to be that the system will have trouble detecting the user's face, track it, or identify it at all if the user's forehead is covered by hair or a hat.

**Ethical and Privacy Concerns**

The user's facial information is never recorded or stored by the fatigue detection model. I made the decision not to store, transmit, or save any face data, even though doing so can increase accuracy for users who record and store their data. Every facial analysis is done in real time, and every image frame is instantly discarded after processing. This reduces the possibility of illegal access, data leaks, or misuse of facial data by guaranteeing that no personally identifying information is gathered. Users' privacy will be protected, and their confidence in the detection model will grow as a result.

Additionally, there is no requirement for cloud-based processing as the system runs locally on a device (such as a laptop), which lowers the possibility of data breaches by third parties. Since no user data is exchanged or stored, this method complies with privacy requirements like the General Data Protection Regulation (GDPR) and others.

**Conclusion**

In conclusion, this study demonstrates how AI-driven fatigue detection can enhance safety and lower hazards associated with fatigue. By advancing intelligent safety monitoring systems, my study can help create surroundings that are safer and more secure.

The creation and deployment of a real-time fatigue detection system utilizing deep learning and computer vision techniques has been described in detail in this report. An automatic and effective monitoring system is beneficial when combatting fatigue-related accidents. Based on eye closure, blink frequency, and yawning recognition, the system uses facial landmark detection and a Convolutional Neural Network (CNN) to detect fatigue.

When fatigue symptoms are identified, the system efficiently sends out text alerts thanks to its excellent accuracy and real-time performance. There are still issues, though, such as the effects of obstructions, lighting, and individual facial differences. To overcome these constraints, more optimization, a larger dataset, and even integration with physiological based methods such as sensors to improve robustness could be beneficial.

**Appendix**

A questionnaire with text and a checklist

AI-generated content may be incorrect.

A close-up of a form

AI-generated content may be incorrect.

**References**

1. Tefft, B., & Velaga, N. R. (2024). Road safety fact sheet: Driver fatigue and road accidents. In *Road Safety Fact Sheet: Driver Fatigue and Road Accidents* [Report]. Available at:

<https://www.rospa.com/getmedia/25999a67-6e67-4a8c-b32d-13b5c1090010/Driver-Fatigue-Factsheet-2022-updated.pdf> (Original work published 2011)

1. Tefft, B. C. & AAA Foundation for Traffic Safety. (2016). *Acute sleep deprivation and risk of motor vehicle crash involvement*.

Available at:

<https://aaafoundation.org/wp-content/uploads/2017/12/AcuteSleepDeprivationCrashRisk.pdf>

1. *Artificial Intelligence (AI) vs. Machine Learning | Columbia AI*. (2023, October 3). CU-CAI.

Available at:

<https://ai.engineering.columbia.edu/ai-vs-machine-learning/>

1. Yurday, E. (2025) *Leading causes of car accidents UK 2025*, *NimbleFins*.

Available at:

<https://www.nimblefins.co.uk/cheap-car-insurance/top-causes-car-accidents-uk>

1. Xiao, Zhitao, et al. “Fatigue Driving Recognition Network: Fatigue Driving Recognition via Convolutional Neural Network and Long Short‐Term Memory Units.” *IET Intelligent Transport Systems*, vol. 13, no. 9, 6 June 2019, pp. 1410–1416,

Available at:

<https://doi.org/10.1049/iet-its.2018.5392>

1. Zhang, Jianping, et al. “A Field Study of Work Type Influence on Air Traffic Controllers’ Fatigue Based on Data-Driven PERCLOS Detection.”, vol. 18, no. 22, 13 Nov. 2021, pp. 11937–11937, [www.ncbi.nlm.nih.gov/pmc/articles/PMC8622576/](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC8622576/),

Available at:

<https://doi.org/10.3390/ijerph182211937>

1. Du, Juan. “High-Precision Portrait Classification Based on MTCNN and Its Application on Similarity Judgement.” *Journal of Physics: Conference Series*, vol. 1518, Apr. 2020, p. 012066,

Available at:

<https://doi.org/10.1088/1742-6596/1518/1/012066>

1. Papadelis, Christos, et al. “Monitoring Sleepiness with On-Board Electrophysiological Recordings for Preventing Sleep-Deprived Traffic Accidents.” *Clinical Neurophysiology*, vol. 118, no. 9, Sept. 2007, pp. 1906–1922,

Available at:

<https://www-sciencedirect-com.ezproxy.mdx.ac.uk/science/article/pii/S1388245707002945>

1. Albadawi, Yaman, et al. “A Review of Recent Developments in Driver Drowsiness Detection Systems.” *Sensors*, vol. 22, no. 5, 7 Mar. 2022, p. 2069,

Available at:

<https://doi.org/10.3390/s22052069>

1. “Driving Tired.” *Qld.gov.au*, 2015,

Available at:

[www.qld.gov.au/transport/safety/road-safety/driving-safely/driving-tired?](http://www.qld.gov.au/transport/safety/road-safety/driving-safely/driving-tired?)

1. King, Davis. “Certificate Error.” *Dlib.net*, 2024, dlib.net/.

Available at:

<https://dlib.net/>

1. Rosebrock, Adrian. “Facial Landmarks with Dlib, OpenCV, and Python.” *PyImageSearch*, 3 Apr. 2017, pyimagesearch.com/2017/04/03/facial-landmarks-dlib opencv-python/

Available at:

<https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

1. Yashar Jebraeily. “Drowsy Detection Dataset.” *Kaggle.com*, 2024,

Available at:

[www.kaggle.com/datasets/yasharjebraeily/drowsy-detection-dataset](http://www.kaggle.com/datasets/yasharjebraeily/drowsy-detection-dataset)

1. GeeksforGeeks. “Introduction to Convolution Neural Network.” *GeeksforGeeks*, 14 Mar. 2024,

Available at:

[www.geeksforgeeks.org/introduction-convolution-neural-network/](http://www.geeksforgeeks.org/introduction-convolution-neural-network/)