DROWSINESS DETECTION AND PREVENTION USING ARTIFICIAL FEATURE RECOGNITION

DETEÇÃO E PREVENÇÃO DA SONOLÊNCIA ATRAVÉS DO RECONHECIMENTO ARTIFICIAL DE CARACTERÍSTICAS

10.29073/e3.v9i2.873
Receção: 10/07/2023. Aprovação: 20/09/2023. Publicação: 28/12/2023

Miguel Mota [a]; Daniel Azevedo [b]; Damiana Guedes [c]; Gonçalo Santos [d]; Francisco Soares [e]; Pedro Lopes [f];

[a] CERNAS-IPV Research Centre for Natural Resources, Environment and Society, Polytechnic Institute of Viseu. Campus Politécnico, Portugal; mmota@estgl.ipv.pt; [b] CISeD - Research Centre in Digital Services, Polytechnic Institute of Viseu. Campus Politécnico, Portugal; dazevedo@estgl.ipv.pt; [c] Polytechnic Institute of Viseu - School of Technology and Management of Lamego, Portugal; dguedes@estgl.ipv.pt; [d] Polytechnic Institute of Viseu - School of Technology and Management of Lamego, Portugal; estgl3267@estgl.ipv.pt; [e] Polytechnic Institute of Viseu - School of Technology and Management of Lamego, Portugal; fsoares@estgl.ipv.pt; [f] Polytechnic Institute of Viseu - School of Technology and Management of Lamego, Portugal; plopes@estgl.ipv.pt;

ABSTRACT

In Portugal, in the year 2022, there were 32.788 reported road accidents. Among the identified causes, driver fatigue emerged as the second major contributor to these incidents. Efforts to address this issue have resulted in the development of legislation over the years. Concurrently, aligning with initiatives in the European Union, the Fédération Internationale de l'Automobile (FIA) has actively promoted collaboration with the automotive industry to integrate safety-enhancing systems in vehicles, aiming to alleviate challenges such as driver fatigue.The proposed strategy represents an intermediary solution, bridging the gap between the securi-ty provided by in-vehicle systems and the accessibility offered by mobile sys-tems. The project's overarching goal is to ensure cost-effectiveness, efficiency, and widespread applicability. Leveraging Microsoft's Face API technology, the project seeks to capitalize on artificial intelligence to unlock a range of features, thereby achieving tasks that were previously deemed impractical or financially burdensome.

Keywords: Artificial Intelligence, Alarm, Security, Control, Car Industry, API, IoT.

RESUMO

Em Portugal, no ano de 2022, registaram-se 32.788 acidentes rodoviários. Entre as causas identificadas, a fadiga do condutor surgiu como o segundo maior contribuinte para estes incidentes. Os esforços para abordar esta questão resultaram no desenvolvimento de legislação ao longo dos anos. Simultaneamente, alinhando-se com as iniciativas da União Europeia, a Fédération Internationale de l'Automobile (FIA) tem promovido ativamente a colaboração com a indústria automóvel para integrar sistemas de reforço da segurança nos veículos, com o objetivo de atenuar desafios como a fadiga do condutor. A estratégia proposta representa uma solução intermediária, colmatando a lacuna entre a segurança proporcionada pelos sistemas de bordo e a acessibilidade oferecida pelos sistemas móveis. O objetivo global do projeto é assegurar a relação custo-eficácia, a eficiência e a aplicabilidade generalizada. Tirando partido da tecnologia Face API da Microsoft, o projeto procura capitalizar a inteligência artificial para desbloquear uma série de funcionalidades, realizando assim tarefas que anteriormente eram consideradas impraticáveis ou financeiramente onerosas.

Palavras-chave: Inteligência Artificial, Alarme, Segurança, Controlo, Indústria automóvel, API, IoT.

1 Introduction

The first work about this theme was published this year (Santos et al, 2023). This present work arises as a new line of research related to the implementation of algorithms to better adapt to the user, potentially enabling the application to 'get to know' the user to enhance its fatigue detection (access to data from digital watches such as heart rate measurements, accelerometers, etc).

The project's goal is to address the problem of tiredness, which is a significant issue associated with human behavior. In 2022, there were 32.788 road accidents in Portugal, resulting in 462 fatalities, and fatigue was identified as the second leading cause of these accidents (PORDATA, 2022). The project's solution primarily focuses on warning individuals about potential dangers to reduce human errors.

Key objectives of the project include the development of data capture and pro-cessing systems, image decoding, user feedback, and alerts. The main development goals are to create a simple, lightweight, and easily deployable application for immediate workplace use. The application's purpose is to support individuals in their tasks and become an essential tool that intervenes only when necessary, without interfering with user performance.

2 Conception

The genesis of this project stems from the recognition of a common issue afflicting many individuals - drowsiness. Throughout their lives, numerous people encounter instances of drowsiness or even nod off while engaged in professional responsibilities. Consequently, the concept of devising a technological solution to prevent or mitigate the risks associated with drowsiness was conceived. The approach to addressing this issue centered on creating a simple, user-friendly, and lightweight application.

This application was designed to be minimally intrusive and non-distracting, allowing users to proceed with their daily routines seamlessly while ensuring the preservation of normalcy in case of emergencies. However, the application was also structured to interact and intervene if the situation demanded emergency measures.

The envisioned system aims to alert users in the event of an emergency, such as the onset of drowsiness, employing more assertive measures to mitigate or eliminate the consequences of falling asleep. The proposed solution has broad applicability, targeting long-distance or particularly vigilant drivers, operators of heavy machinery, and the automotive industry, potentially becoming a standard feature in new vehicles. Additionally, it holds relevance in the industrial sector, especially for roles with permanent shifts that demand heightened attention, where falling asleep could have severe repercussions for both productivity and individual well-being.

A tangible example could be a car production line, where compromised production capacity due to employee fatigue might lead to the production of defective car parts, tarnishing the company's reputation and posing risks to lives, including the individual operating the machinery. The proposed solution, being a low-maintenance application at the local level, is intended to seamlessly integrate into work systems with minimal disruption to users' daily lives. The goal is to meet users' needs to stay alert during their tasks without causing distraction or making them feel constantly monitored. Implementation of the solution envisions voluntary participation, with individuals opting to install the necessary software and hardware for its functionality. This universality allows for its installation in diverse situations with minimal preparatory requirements. The project distinguishes itself from existing solutions by bridging the gap between the safety features of in-vehicle systems and the ease of transmission found in vehicle-to-vehicle systems, minimizing complexities. Employing a heuristic approach, the system associate’s values with the frequency of incidents, utilizing this data to infer the likelihood of the user being in a state of sleep.

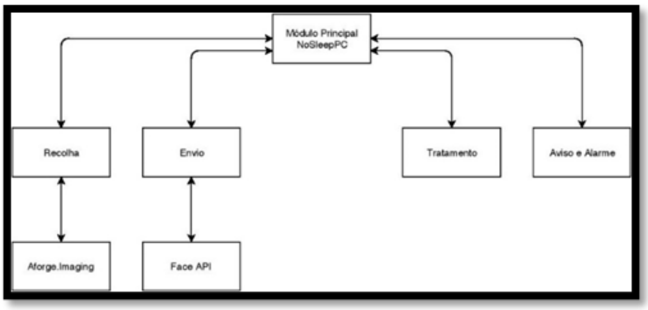

Fig. 1. Application Modules Diagram

In Figure 1, the integration of each component into the model and their interconnections are visually represented, aligning to achieve a low-maintenance application at the local level as advocated in this study.

Upon scrutinizing the schematic representation of the proposed model, it be-comes evident that it comprises four main modules: the main module, collection module, send and treat module, and warning and alarm module. The main module serves as the foundation through which the other modules interact, facilitating the exchange of data and results among them. Initiating a task involves the utilization of the collection module, which employs the Aforge. Imaging library to access the camera, capturing an image that is subsequently relayed to the main module.

The collection module manages the utilization of available system cameras for image capture, and the captured data is transmitted to the sending module. The Shipping block is then employed to communicate with the Face API, retrieving milestone values for processing, which are subsequently forwarded to the treat-ment block. This module establishes REST communication with the API, converting the received data into a string variable. Upon reception in the sending mod-ule, the data is processed and compared in the treatment module to ascertain if the user exhibits signs of drowsiness, generating a "confidence" value.

The confidence value becomes the basis for the warning and alarm module, triggering a warning or, if necessary, an alarm where the alarm window is activated. This comprehensive process ensures that the system can effectively detect and respond to potential instances of user drowsiness.

Implementation and results

The application was crafted using the C# programming language within the Visual Studio environment (Microsoft, Visual Studio, 2019). Microsoft Cognitive Services, specifically the Face API (Microsoft, Azure Cognitive Services, 2019), was employed to facilitate the submission and processing of facial landmarks, thereby enabling the computation of project metrics. The application is built on the Windows platform, utilizing the Windows Presentation Foundation, which adheres to the application syntax.

This foundation serves as a cohesive programming model for constructing Windows applications (Microsoft, Windows Presentation Foundation, 2019). Operationally, the application stores specific data, such as total recording hours, start and end times of each recording session, and details of each detected "incident" (such as eye closure), in an SQL database. This database serves as a repository for all incidents and recordings, contributing to comprehensive incident tracking.

Additionally, the application leverages the AForge. Imaging library to interface with and optimize the usage of webcams installed on the system (AforgeTeam, 2019). The technological framework employed in this project revolves around Microsoft's Face API, a specialized cognitive service. This API enables data transmission and, thanks to its pre-trained artificial intelligence structure, returns a detailed list of information. This information encompasses various details, such as the person's hair color, estimated age, and, most importantly, facial landmarks. The integration of the Face API enhances the application's capabilities by providing valuable insights derived from facial analysis. We also utilise and implement Dlib (Dlib.net. s.d.).

The Dlib library is a popular choice for facial recognition and offers a set of ready-to-use tools and templates. Dlib is an open-source C++ library widely used for computer vision tasks, including face detection, object tracking, facial recognition and much more. Here are some examples of projects that can be implemented with Dlib:

Face Detection and Landmarks:

Implement a face detection system that localises faces in images or videos and then identifies the landmarks on the faces, such as eyes, nose and mouth.

Facial Recognition:

Create a facial recognition system that compares faces detected in an image with faces in a database to identify people.

Eye or Facial Expression Tracking:

Implement an eye-tracking system to track eye movement in a video. This can be useful in visual attention research. Develop a facial expression recognition system that identifies emotions such as happiness, sadness, anger, etc., based on changes in facial features.

Object detection:

Use Dlib to detect and track objects in videos, such as vehicles on a road or people in a crowd.

Keypoint Tracking in Medical Images:

Apply Dlib to track key points in medical images, such as magnetic resonance imaging (MRI) or computerised tomography (CT) scans.

Gesture Detection:

Create a system that detects hand gestures or other body gestures and interprets them to control an application or device.

3 Implementation and Results

The application was developed using the C # programming language, with Visual Studio (Microsoft, Visual Studio, 2019), and using the Microsoft Cognitive Ser-vices (Microsoft, Azure Cognitive Services, 2019) Face API, to enable submission and treatment of face milestones, thereby enabling project calculations. The application is based on Windows, using the Windows Presentation Foundation, which is based on the application syntax. This foundation represents a unified programming model for building Windows applications (Microsoft, Windows Presentation Foundation, 2019). The application itself saves certain data, such as total hours, each time it starts recording and ends, as well as each “incident” it detects (such as closing its eyes) in an SQL database, to instantiate all incidents as well as all recordings made. It also uses the AForge. Imaging library to describe the webcams installed on the system and to enhance their use for capture (AforgeTeam, 2019). The technology used to accomplish this project is based on Microsoft's more specific cognitive service, the Face API. This API allows send-ing and thanks to the structure of artificial intelligence already trained and ap-plied, returns a list of data, which can specify the hair color of the person, the estimated age, etc. However, the most important is that it returns the face mile-stone

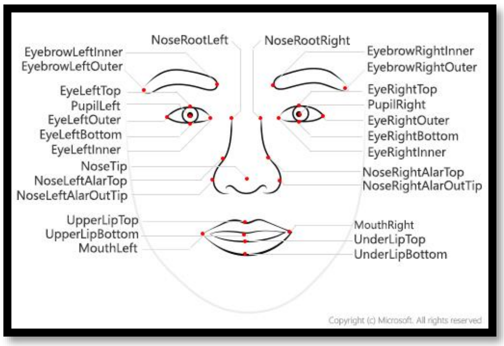

Fig. 2. Face Mark

These facial landmarks, as illustrated in Figure 2, represent distinctive points outlined and transmitted by the Face API system. These points correspond to significant facial features, such as the midpoint of the lip or the pupil, pinpointed by coordinates (Microsoft, Face API - Facial Recognition Software | Microsoft Azure, 2019). On the technical front, the AForge Framework (AforgeTeam, 2019) is utilized to interface with hardware, specifically the webcam. This framework facilitates the utilization and transmission of webcam data to the Face API servers. AForge Framework is tailored for computer vision and artificial intelligence programmers, containing various libraries. The application specifically employs the AForge. Imaging library to enable image processing for submission to the Face API. SQL, a relational database language, originated from the work of Dr E. F. Codd, as detailed in his paper "A Relational Model of Data for Large Shared Data Banks" (Oracle, 2019). Initially known as SEQUEL, it later became SQL due to patent constraints. Today, SQL serves as the standard language for relational databases, and Microsoft SQL Server, a commercial solution, allows the configuration of an SQL server to store data (Microsoft, 2019). It facilitates connections from applications in any preferred language on various platforms, including Windows and Linux. C#, pronounced as C Sharp, is a high-level programming language developed by Anders Hejlsberg in 2002, based on the C language and intended as an "increment" beyond C++. The language is designed to be simple, modern, and object-oriented, with features such as type checking, vector boundary checking, and garbage disposal. Visual Studio, an IDE produced by Microsoft, is integral to C# development, providing a robust environment with debugging, compiling, and interpreting capabilities. IntelliSense technology assists in error detection and code completion, making programming more efficient (Microsoft, Intelligence, 2019). Visual Studio also supports other languages, including Visual Basic, F#, C++, Javascript, and Typescript. The development of the project was centered on implementing the application flow depicted in Figure 3, leveraging the capabilities of Visual Studio and the selected programming languages and frameworks.

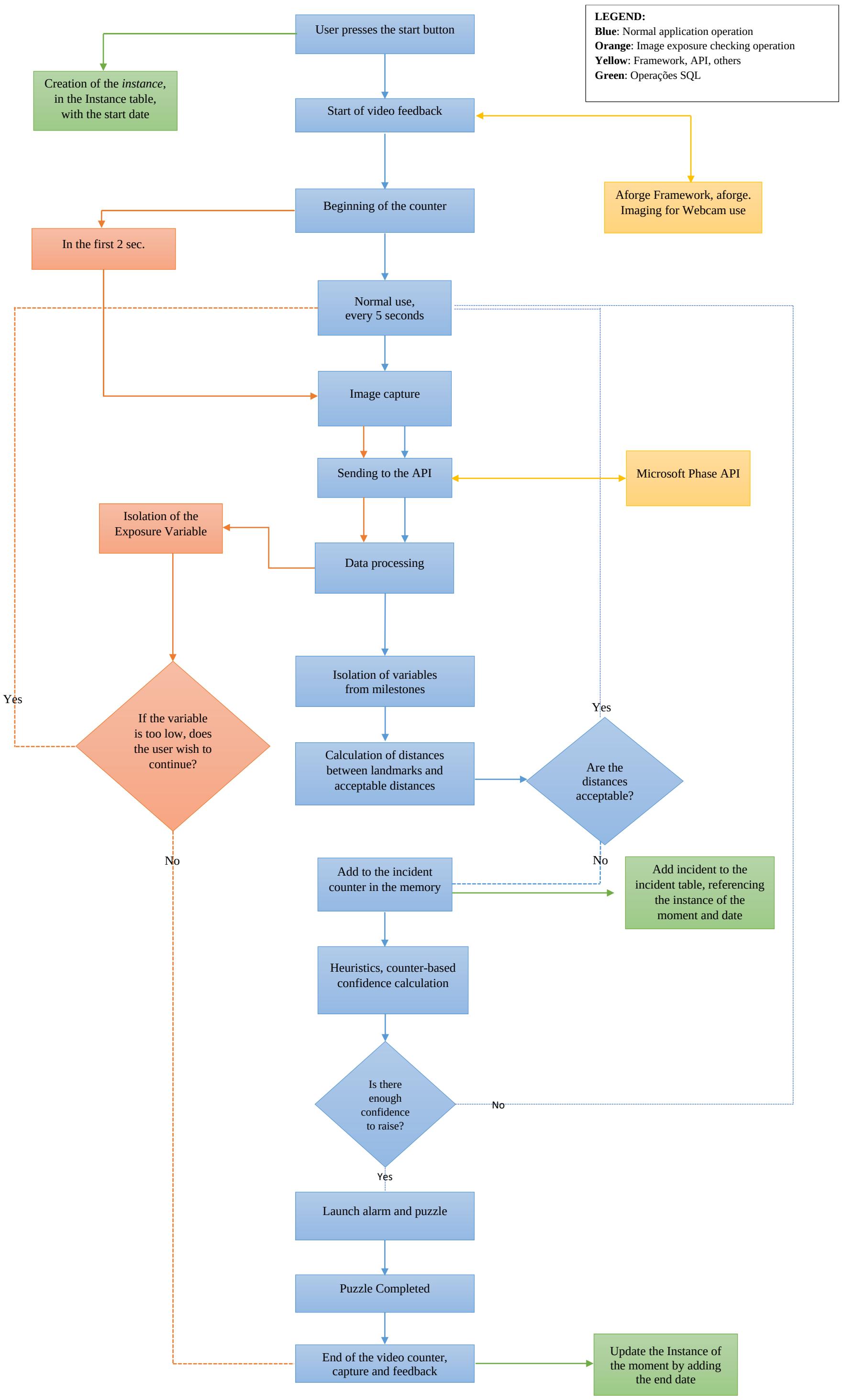

Fig. 3. Application Flowchart

The COLLECTION MODULE, as delineated in Figure 3, initiates a counter to facilitate the capture of an image at five-second intervals. These captured images are then transmitted to the upload module, where they undergo conversion and subsequent transmission, as implied by its name.

Given that images are not recorded for storage but are intermittently captured for upload, no video data is retained. Captured images undergo immediate processing and are deleted from memory without being stored in any directory. Since image processing occurs on Microsoft servers, adherence to their privacy policy is imperative for users who opt to accept it (Microsoft, Azure Cognitive Services, 2019). Upon image capture, in-memory processing takes place, converting it into a bitmap image and subsequently into a byte array for API utilization.

A REST call, designated as "Submit to API" in Figure iv, is executed to transmit data in accordance with the API documentation, resulting in the retrieval of a string variable for further handling. The contact with the Face API, as indicated in the preceding flowchart (Figure iv) with the label "Upload to API," involves the insertion of the captured image into the API servers. This method initiates a REST connection to the API servers, necessitating the retrieval of parameters such as face landmarks, image exposure, and head position upon its completion. These retrieved values are consolidated into a string variable to facilitate further pro-cessing. The isolated and processed variables encompass the top and bottom co-ordinates of both the right and left eyes, the orientation of the head, and the co-ordinates of the upper and lower lip. These landmarks constitute the crucial data points utilized for fatigue detection and subsequent identification of drowsiness.

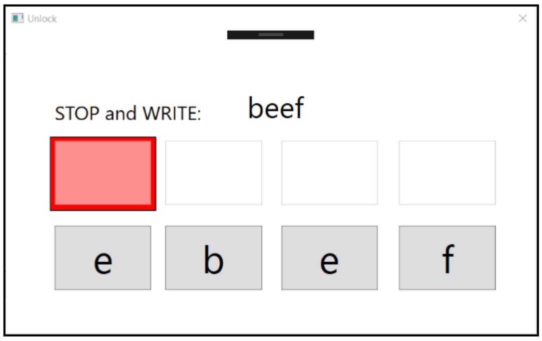

Fig. 4. Puzzle window

In Figure 4, the alarm-triggered puzzle interface is presented, compelling the user to solve it to deactivate the alarm. This window incorporates various text boxes and buttons designed for entering the specific sentence provided by the window.

If the user correctly inputs the sentence, the window closes, and the program re-starts to facilitate nearly immediate resumption of normal use. In the event of an incorrect completion, the puzzle persists with the alarm active, resetting the text boxes for subsequent attempts. Notably, the requested word for the puzzle re-mains unchanged throughout this process. Under typical usage conditions, users cannot voluntarily access this menu. However, in this program version, controls in the main window allow users to access this interface without triggering the alarm initially. Soak Testing, a form of test, involves prolonged software operation under normal conditions and transaction volumes. This extended duration serves to assess the program's endurance, identifying potential issues like memory leaks or previously unnoticed bugs that might arise with prolonged use (Guru99, 2019). The initial phase of this test involved maintaining the application in a dormant state for fifteen minutes to evaluate any potential memory-related problems without active user interactions.

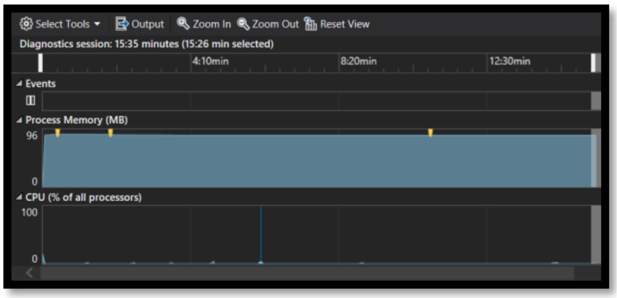

Fig. 5. Sleeping Process

As depicted in Figure 5, the dormant process in this mode exhibits no evidence of leaks or issues throughout its duration without any user action. There is only a swift initial surge in memory usage within the first few seconds, attributed to the initiation of the application code. Processor usage has been consistently maintained at a minimal level, with occasional spikes observed throughout its operation. The subsequent process, which has already resumed normal functioning, would have been active for an hour.

4 Future Improvements

Potential enhancements and future developments for the proposed solution can be considered. One avenue for improvement involves the integration of addition-al sensors, such as a heart rate monitor, to augment the system's accuracy and efficacy. Expanding the system's capabilities could also entail integration with other smart technologies, such as the Internet of Things (IoT), fostering increased connectivity and data exchange. Furthermore, ongoing research and development efforts may yield novel advancements in artificial intelligence.

Incorporating these innovations into the system could elevate its overall performance. Continuous exploration of new possibilities and avenues for improvement is crucial to ensure the system's sustained effectiveness and relevance in addressing driver fatigue and enhancing road safety.

5 Conclusion

Driver fatigue stands out as a significant contributor to road accidents, ranking second in Portugal's accident statistics. This underscores the imperative for an effective solution to address drowsiness among drivers. The presented project introduces a system designed to mitigate this issue through the development of an application geared towards detecting and alerting drivers about potential drowsi-ness. Employing artificial intelligence-based technology, specifically the Microsoft Face API, the system analyzes the driver's facial features to assess drowsiness. The proposed solution is intentionally lightweight, user-friendly, and primarily non-intrusive, yet equipped to issue warnings and intervene when necessary to uphold driver safety. This low-maintenance application, seamlessly inte-grated into local systems, ensures minimal disruption to users' daily routines. Its adaptability extends beyond the individual user, making it suitable for implementation across various sectors, including automotive and industrial domains. In essence, the proposed solution serves as an intermediary approach, bridging the gap between the security offered by in-vehicle systems and the accessibility of mobile systems. By enhancing road safety, this solution holds the potential to save lives and contribute to accident prevention.

Acknowledgement and Funding:

This work is funded by National Funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project Refª UIDB/00681/2020. Furthermore, we would like to thank the CERNAS Research Centre and the Poly-technic Institute of Viseu for their support.

REFERENCES

1. AforgeTeam. (2019). AForge.NET: Computer Vision, Artificial Intelligence, Robotics. http://www.aforgenet.com/framework/

2. ANSR/MAI, P. (2018). Acidentes de viação com vítimas, feridos e mortos - Continente. PorData. https://www.pordata.pt/Portugal/Acidenes+de+via%c3%a7%c3%a3o+com+v%c3%adtimas++feridos+e+mortos+++Continente-326-3587

3. Ataei, M., Esmaeilpour, A., & Yousefi, M. R. (2019). Driver drowsiness detection using facial landmark estimation and eye aspect ratio. Measurement, 139, 373-381.

4. Azati (2019) reports on the cost of artificial intelligence (AI) in 2019. https://azati.ai/how-much-does-it-cost-to-utilize-machine-learning-artificial-intelligence/

5. Bizdirect. (2019). Quem somos. https://www.bizdirect.pt/pt/quem-somos

6. C# DESIGN GOALS. (2011, January 23). https://feeldotneteasy.blogspot.com/2011/01/c-design-goals.html

7. Choi, Y., Jeong, J., & Lee, J. (2021). Driver drowsiness detection using facial landmarks, head movement, and bio-signal features with deep learning. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2021.3119049

8. Dlib.net. (s.d.) DLIB C++ Library. http://dlib.net/

9. Guru99. (2019). What is Soak Testing? Definition, Meaning, Examples. https://www.guru99.com/soak-testing.html

10. Hall-Geisler, K. (2019). How Anti-sleep Alarms Work. Available at: https://electronics.howstuffworks.com/gadgets/automotive/anti-sleep-alarm.htm

11. IDEOU. (2019). Design Thinking. IDEO U. https://www.ideou.com/pages/design-thinking

12. Jafari, S. A., Zanjanizadeh Ezazi, M. P., Torkamani, M. J., & Rahimpour, A. (2021). An overview of deep learning in medical imaging: Focus on MRI. Journal of Medical Signals and Sensors, 11(3), 137-151.

13. Kim, H., Yoo, J., & Han, J. (2021). A Comparative Study on Driver Drowsiness Detection Systems based on Facial Landmark Estimation and Machine Learning Techniques. Sensors, 21(11), 3875.

14. Kim, M., Lee, W., & Choi, S. (2020). Real-Time Driver Drowsiness Detection System based on Facial Features using Convolutional Neural Network. Journal of Advanced Transportation, 2020. doi: 10.1155/2020/8827386.

15. Lee, H., Lee, S., & Lee, S. (2022). Driver Drowsiness Detection System Based on Facial Landmarks and Artificial Neural Network. Sensors, 22(3), 957.

16. Liu, Y., Wang, Z., Lv, Z., & Sun, J. (2017). Driver Drowsiness Detection using Dynamic Facial Analysis. IEEE Transactions on Intelligent Transportation Systems, 18(4), 945-954.

17. Mercedes-Benz. (2019, August 4). Safety. Mercedes-Benz USA. https://www.mbusa.com/mercedes/benz/safety#module-3

18. Microsoft. (2019, August 11). Azure Cognitive Services. Microsoft Azure. https://azure.microsoft.com/en-us/support/legal/cognitive-services-compliance-and-privacy/

19. Microsoft. (2019, August 11). Windows Presentation Foundation. Microsoft. https://docs.microsoft.com/en-us/dotnet/framework/wpf/

20. Microsoft. (2019, August 13). IntelliSense. Microsoft. https://code.visualstudio.com/docs/editor/intellisense

21. Microsoft. (2019, August 13). SQL Server. Microsoft. https://www.microsoft.com/en-us/sql-server/sql-server-2017

22. Microsoft. (2019, August 13). The History of C#. Microsoft.

https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history

23. Microsoft. (2019, August 13). Visual Studio. Microsoft. https://visualstudio.microsoft.com/vs/

24. Microsoft. (2019, August 4). Face Detection and Attributes. Microsoft Azure. https://docs.microsoft.com/en-us/azure/cognitive-services/face/concepts/face-detection

25. Microsoft. (2019, June 25). Face API - Facial Recognition Software | Microsoft Azure. Microsoft Azure. https://azure.microsoft.com/en-us/services/cognitive-services/face/

26. Oppel, A. (2004). Databases Demystified. McGraw-Hill Osborne Media.

27. Oracle. (2019). History of SQL. Oracle. https://docs.oracle.com/cd/B12037_01/server.101/b10759/intro001.htm

28. Ozkan, M. & Turk, M. (2017). Driver drowsiness detection based on shape and texture analysis of facial images. Image and Vision Computing, 57, 38-46.

29. Plattner, H., Meinel, C., & Leifer, L. J. (2011). Design thinking: Understand, improve, apply. Berlin: Heidelberg.

30. PORDATA. (2022). Acidentes de viação com vítimas - Total e por tipo de acidente. PORDATA - Base de Dados de Portugal Contemporâneo. https://www.pordata.pt/

31. RoSPA. (2019, August 11). Driver Fatigue and Road Accidents - a Literature Review and Position Paper. Road Safety Knowledge Centre. http://www.roadsafetyknowledgecentre.org.uk/knowledge/545.html

32. Santos, G., Lopes, P., Guedes, D., Azevedo, D., & Soares, F. (2023). Drowziness prevention system based on artificial intelligence facial features detection. Special Issue - ICE2MAS, 92-99.

33. Singh, P., Bajaj, P., & Gupta, S. (2017). A Vision-Based System for Driver Drowsiness Detection using Pupil Detection and Mouth Detection. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 935-940). IEEE.

34. Sun, H. F., Park, S. S., & Cho, S. H. (2018). Facial landmarks-based driver drowsiness detection using deep neural networks. Neural Computing and Applications, 30(10), 3177-3186.

35. TECH, T. (2019, August 5). Nap Zapper. TBO TECH. https://www.tbotech.com/napzapper.htm

36. Volvo. (2019, August 4). Driver Alert Control. YouTube. https://www.youtube.com/watch?v=sVDTnFeutOs

37. Zaidi, B., Ilyas, M. U., & Imran, A. (2020). Driver Drowsiness Detection Based on Fa-cial Features using Machine Learning Techniques. International Journal of Advanced Computer Science and Applications, 11(7), 87-93.

PROCEDIMENTOS ÉTICOS

Conflito de interesses: Nada a declarar. **Financiamento**: Nada a declarar. **Revisão por pares**: Dupla revisão anónima por pares.