Drowziness prevention system based on artificial intelligence facial features detection

Gonçalo Santos
Department of Informatics, Communications and Fundamental Sciences
Polytechnic Institute of Viseu - School of Technology and Management of Lamego
Lamego, Portugal

Pedro Lopes
Department of Informatics, Communications and Fundamental Sciences
Polytechnic Institute of Viseu - School of Technology and Management of Lamego
Lamego, Portugal

Damiana Guedes
Department of Informatics, Communications and Fundamental Sciences
Polytechnic Institute of Viseu - School of Technology and Management of Lamego
Lamego, Portugal

Daniel Azevedo
CISeD – Research Centre in Digital Services, Instituto Politécnico de Viseu, Portugal
Viseu, Portugal

Francisco Soares
Department of Informatics, Communications and Fundamental Sciences
Polytechnic Institute of Viseu - School of Technology and Management of Lamego
Lamego, Portugal

Abstract - **In Portugal, in 2017, there were 34,416 accidents on the roads. It is estimated that the second major cause of accidents is driver fatigue. In this way, over the years, legislation has been created to mitigate the problem. In parallel with the European Union, the Fédération Internationale de l'Automobile (FIA) has encouraged the automotive industry to develop systems embedded in vehicles to increase their safety and mitigate this and other problems.**
This approach is intended to be an intermediate solution, as to be the in-between the security of a system embedded in a vehicle and the accessibility of a mobile system. In this way, the project aims to be as cheap, fast and applicable as possible.
Using Face API technology provided by Microsoft, it is possible to have access to a set of features based on artificial intelligence, accomplishing tasks previously unthinkable, or very costly.

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

I. INTRODUCTION

The project intended to respond to the need to solve a problem associated with the human condition; the tiredness. In 2017, according to PORDATA, 34,416 road accidents occurred, of which 510 were killed in Portugal (MOR / May 2018). It is estimated that in these accidents, the second leading cause is fatigue (RoSPA, 2019).

This was the motto for devising the solution developed to try to address human errors by warning the individual that he was facing danger. By abstracting and undoing the problem, several goals and objectives were created for the development of the presented solution.

These goals are a data capture system and processing system, in order to decode the image and handle the decoded data, and finally the warning and user feedback system. The main goals of development are to realize a simple, lightweight, and easy to install/configure application in order to maximize almost immediate use after it is placed in the workplace.

Application development focuses on supporting the individual in performing their job. The point is to get to the point where the application is seen as an indispensable tool. This tool is designed to be non-intrusive, but intervene if necessary. Always taking into consideration not preventing the performance of the user function with the necessary interventions.

II. CONCEPTION

The problem arose in the discussion of a problem common to many individuals, drowsiness. During its lifetime, several people may experience drowsiness and/or potentially fall asleep when performing professional duties.
Thus, the idea of thinking about a possible technological solution that could avoid or minimize the risk associated with drowsiness was triggered. The approach to the problem was based on a simple, easy, and light application.
This application needed to be less intrusive and less distracting, in order to allow the normal running of the user's day-to-day, without changing his daily life in case of emergency. However, it was also necessary for the application to interact and intervene with the user if the emergency situation requires it.

A system that might warn the user if he/she enters an emergency situation (eg, falling asleep), the system will use a more aggressive means to remedy the consequences of falling asleep, or even eliminate them once and for all. The scope of the proposed solution is wide, such as long-distance or particularly careful drivers, heavy machinery drivers, and the automotive industry, and can become a standard in new vehicles. However, the application can also be applied in the industrial sector, eg permanent shifts whose function needs special attention, and falling asleep can cause severe problems not only in terms of productivity but also for the individual himself.

A practical example would be a car production line whose poor production capacity due to employee sleep could make one or several car parts defective. This would not only damage the company's image but could also later endanger other people's lives, as well as the individual's own life when handling the production machinery.
The proposed solution is a low-maintenance application at the local level, so it is intended to be easily implemented in the work system without much change to the user's daily life. A solution that can, as far as possible, meet the need of the user to stay awake during his or her job without being distracted or "watched" by it.

The idealization was foreseen with the voluntary gesture of a given individual in order to install the necessary software and hardware in order to enjoy the solution. This solution is also intended to be universal, and can be installed in any situation requiring minimal preparation.
Thus, with the foregoing idea, the disparity between the proposed solution and the existing solutions serves to link the safety and capacity of an in-vehicle system as well as the ability to transmit the vehicle-to-vehicle system easily without Many complications.
The system also uses a heuristic style in which it associates values with the number of incidents performed and assumes with this data that the user may be sleeping.

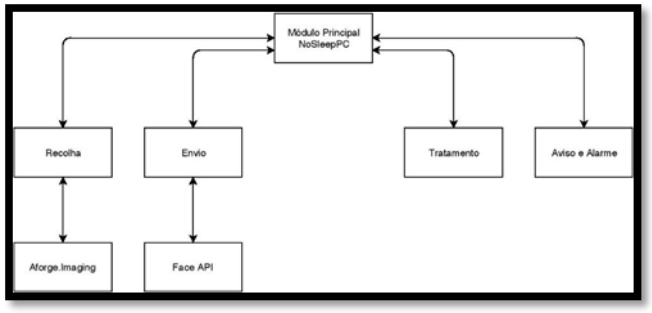

Figure 1. Application Modules Diagram

In Figure 1, it is visible how each component is integrated into the model and how they are interconnected with each other, as the intention to obtain low maintenance application at the local level, is advocated in this work.
It is noticeable when analyzing the scheme of the proposed model, that it consists of a main module, a collection module, a send and treat module, and a warning and alarm. The main module is responsible for the beginning and base in which the modules relate, transacting data and results between them.
When you start your task, you first use the collection module, whose main function is to use the Aforge. Imaging library to access the camcorder. Then “collapse” an image that will return to the main module.
The collection module is responsible for using the system's available camcorders and eventual image capture, the data of which will be sent to the sending module. Only then will you take advantage of the Shipping block by contacting the Face API, which will return the values of the milestones to treat, which are later sent to the treatment block.
This module represents the REST contact with the API, and transposes the received into a string variable. When data is received in the sending module, it will be processed and compared in the treatment module. Understanding if the user may contain signs of drowsiness, creates a value of “confidence”.
This confidence value will be the one on which the warning and alarm module will be based, a warning will be issued, or if an alarm needs to be issued, where the alarm window will be called.

III. Implementation and Results

The application was developed using the C # programming language, with Visual Studio (Microsoft, Visual Studio, 2019), and using the Microsoft Cognitive Services (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 sending and thanks to the structure of artificial intelligence already trained and applied, 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 milestones.

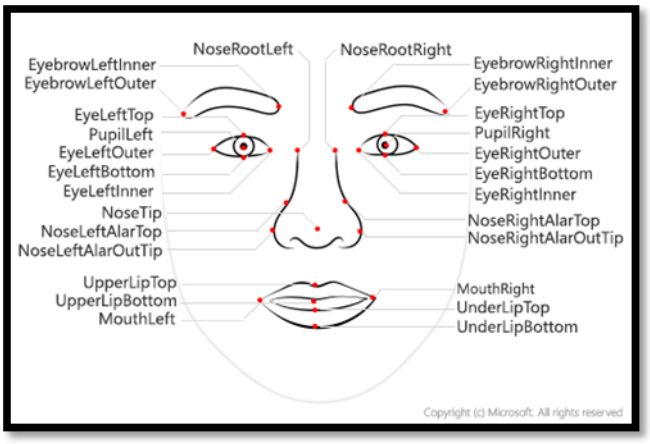

Figure 2. Face Mark

These face landmarks, figure 2, are points outlined and sent by the Face API system, where they mark important points on the face, such as the midpoint of the lip, or the pupil, among others, thus indicating them by coordinates. (*Microsoft*, Face API - Facial Recognition Software | Microsoft Azure, 2019).

On the technical side, we use the AForge Framework (*AforgeTeam*, 2019), which aids access to hardware, such as the webcam, and enables its use and upload to the Face API servers.

Aforge Framework is a framework designed for computer vision or artificial intelligence programmers or researchers. It contains several libraries, the one used in the application is Aforge. Imaging, in order to enable image processing, for its submission in the Face API.

SQL is a relational database language that was initially worked on by Dr. E. F. Codd, in his paper “A Relational Model of Data for Large Shared Data Banks” (Oracle, 2019).

Later, his work was accepted as the definitive model in relational databases, so the IBM group made the first language following the Codd model.

This first language was called SEQUEL, however, thanks to an existing patent of an aircraft company (*Oppel*, 2004), had to be changed to SQL.

Today, SQL is the standard language for relational databases. Microsoft SQL Server is a commercial solution from Microsoft, which allows you to configure an SQL server to store data (*Microsoft*, 2019).

It also allows applications to be connected to the server in whatever language the consumer prefers, on any platform, be it Windows, Linux, etc.

C #, read as C Sharp, is a high-level programming language that was based on the C language in 2002 by Anders Hejlsberg (Microsoft, The History of C #, 2019), such as C ++. Like C ++, C # was intended to be an "increment" of an "increment" because the suffix "++" in C ++ in the programming language is a numeric increment. Therefore, the “#” symbol in C # was intended to be four “+”, a step beyond what C ++ would be. Thus, C # intended to be (C # DESIGN GOALS, 2011):

- Simple, modern, and general, being object-oriented;
- Include type checking, vector boundary checking, uninitialized variable usage detection, source code portability, and garbage disposal.
- Must aim to develop components that can take advantage of distributed environments.
- Must have programmatic portability, especially for individuals who already know C and C ++.
- Provide international support.
- Enable application programming in embedded and hosted systems (websites, e.g.).

Because the language belongs to Microsoft, this language is intact with Visual Studio. Visual Studio is an IDE produced by Microsoft. This is a powerful tool that allows the clustered use of various other tools, such as a debugger, a compiler, and an interpreter.

It also utilizes IntelliSense technology, which allows easy programming by completing code and facilitating if there are poorly written syntax methods or other common errors. (*Microsoft*, Intelligence, 2019).

It also allows the use of various other platforms such as Microsoft API, Windows Forms, WPF, etc. (Microsoft, Visual Studio, 2019). However, Visual Studio not only supports C #, but also supports Visual Basic, F #, C ++, Javascript, Typescript, and more.
Also with this, it is important to refer to its biggest feature, the "Visual" part of "Visual Studio", which allows controls to be added to the application with the mere use of the mouse, rather than having to program these interfaces controls from the application user of nothingness.
The project development was based on the implementation of the application flow described in Figure 3.

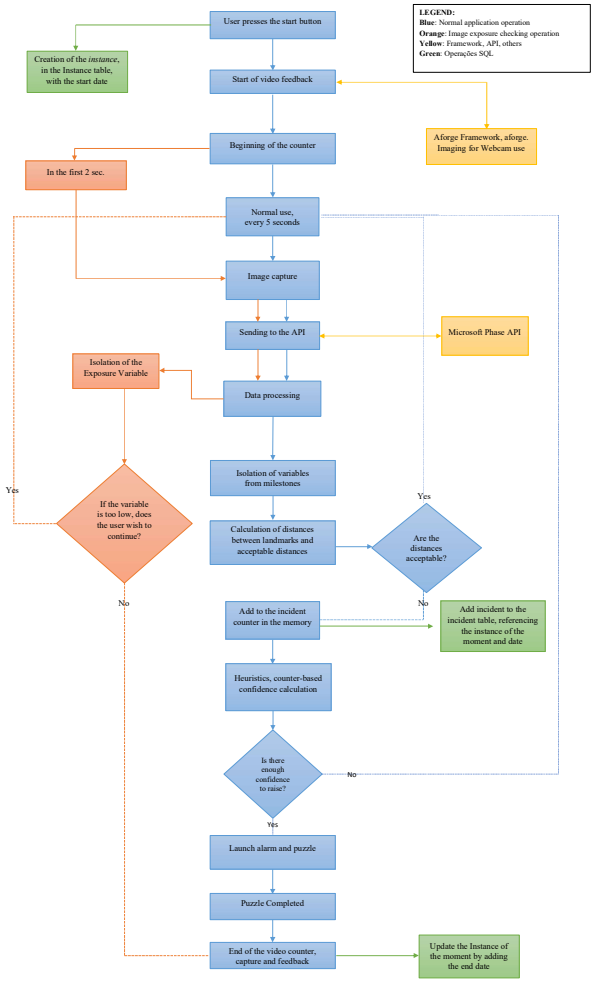

Figure 3. Application Flowchart

The COLLECTION MODULE as described in Figure 3, initializes a counter, which in turn initializes the capture of an image every five seconds and sends it to the upload module, where it will be converted and, as the name implies, sent.
Since images are not recorded for upload but periodically captured for upload, no video is saved. Captured images are immediately processed and deleted in memory, and are not stored in any directory.
Because images are processed on Microsoft servers, it is, therefore, necessary to comply with their privacy policy if they accept it. (Microsoft, Azure Cognitive Services, 2019).
When the image is captured it is treated in memory, so it becomes a bitmap image and later a byte array to take advantage of the API. A REST call is then made, named in Figure iv as "Submit to API", so that data is sent according to the API documentation, returning a string variable to be handled. Therefore, the Face API contact method is called, identified in the previous flowchart, Figure iv, by the state "Upload to API", whereby the captured image is inserted.

This method will then perform a REST connection to the API servers, so it will require return parameters such as face milestones, image exposure, and head position. These values will all be placed in a string to allow their treatment.
These isolated and treated variables are the top and bottom coordinates of the right eye and the left eye, the orientation of the head, and the coordinates of the upper and lower lip. These are the milestones that will be used and treated for fatigue detection and later drowsiness.

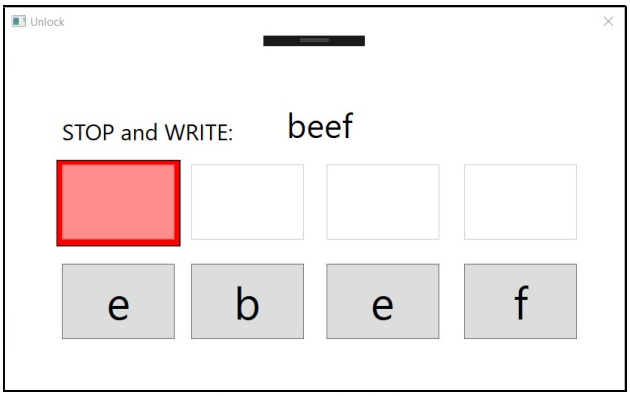

Figure 4. Puzzle window

In Figure 4, the alarm-triggered puzzle is provided, which forces the user to solve it in order to disable the alarm. This window is made up of several text boxes and buttons in order to type the sentence that the window makes explicit for the user to type.
If it is correct, the window closes and restarts the program so as to maximize its use almost immediately. If the individual completes the puzzle incorrectly, the puzzle will remain active with the alarm active and restart the text boxes so that the puzzle can be solved. The word requested, however, will not change.
The user, in normal use of the program, would not be able to access this menu willingly, however, in this version of the program, you can use the controls in the main window to access this window without first triggering the alarm.
Soak Testing is a type of test that is performed by letting the software run for a long time, with its normal operation, with a normal number of transactions.
This period then serves to test whether the program can last long, to predict whether performance deteriorates over prolonged use, such as memory leaks, or bugs not previously experienced. (Guru99, 2019)
The test was then performed initially by keeping the application in its dormant state for fifteen minutes to deliberate if there is a memory problem without anything occurring.

Figure 5. Sleeping Process

As you can see in Figure 5, in this mode the dormant process shows no leak or problem during its course without any action, only having a rapid rise in memory usage in the first few seconds due to the application code being started.
Processor usage has been kept to a minimum, with spikes in use throughout the processor. The next process, already running normally, would have been an hour.

IV. Future Improvements

Future improvements and potential advancements of the proposed solution can be discussed. For example, the current system could be enhanced by incorporating additional sensors, such as a heart rate monitor, to increase its accuracy and effectiveness. Furthermore, the system could be integrated with other smart technologies, such as the Internet of Things (IoT), to allow for greater connectivity and data sharing.

Additionally, ongoing research and development could lead to new advancements in the field of artificial intelligence that could be incorporated into the system to further improve its performance. Ultimately, it is important to continually explore new possibilities and improvements to ensure that the system remains effective and relevant in addressing the issue of driver fatigue and increasing road safety.

V. Conclusion

Driver fatigue is estimated to be one of the major causes of road accidents. In Portugal, fatigue comes in second place, which highlights the need for a driver drowsiness effective solution. This project consists of a system designed to mitigate this problem by creating an application intended to detect drowsiness and warn the driver.
The technologies used are artificial intelligence-based technology, specifically the Microsoft Face API. With this, we can detect the driver's facial features in order to determine if the driver is drowsy. The proposed solution is designed to be lightweight, easy to configure, and mainly non-intrusive while being able to warn and intervene if necessary to ensure the driver's safety.
It is a low-maintenance application at the local level and can be easily implemented in the work system with minimal changes to the user's daily life. It is a low-maintenance application at the local level and can be easily implemented in the work system with minimal changes to the user's daily life.
Furthermore, the solution can be applied in various sectors, including the automotive and industrial sectors. Overall, the proposed solution provides an intermediate solution that bridges the gap between the security of a system embedded in a vehicle and the accessibility of a mobile system, thus increasing road safety and saving lives.

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 Ref. UIDB/05583/2020. Furthermore, we would like to thank the Research Centre in Digital Services (CISeD) and the Instituto Politécnico de Viseu for their support

REFERENCES

[1] AforgeTeam. (2019). *AForge.NET: Computer Vision, Artificial Intelligence, Robotics*. Retrieved April 19, 2023, from http://www.aforgenet.com/framework/

[2] ANSR/MAI, P. (2018). *Acidentes de viação com vítimas, feridos e mortos - Continente*. PorData. Retrieved April 19, 2023, from https://www.pordata.pt/Portugal/Acidentes+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*. Retrieved from https://azati.ai/how-much-does-it-cost-to-utilize-machine-learning-artificial-intelligence/

[5] Bizdirect. (2019). *Quem somos*. Retrieved April 19, 2023, from https://www.bizdirect.pt/pt/quem-somos

[6] *C# DESIGN GOALS*. (2011, January 23). Retrieved April 19, 2023, from 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] Guru99. (2019). *What is Soak Testing? Definition, Meaning, Examples*. Available at: https://www.guru99.com/soak-testing.html [Accessed: April 19, 2023].

[9] Hall-Geisler, K. (2019). *How Anti-sleep Alarms Work*. Available at: https://electronics.howstuffworks.com/gadgets/automotive/anti-sleep-alarm.htm [Accessed: April 19, 2023].

[10] IDEOU. (2019, August 4). *Design Thinking*. IDEO U. https://www.ideou.com/pages/design-thinking

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

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

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

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

[19] Microsoft. (2019, August 13). *IntelliSense*. Microsoft. https://code.visualstudio.com/docs/editor/intellisense

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

[21] Microsoft. (2019, August 13). *The History of C#.* Microsoft. https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history

[22] Microsoft. (2019, August 13). *Visual Studio*. Microsoft. https://visualstudio.microsoft.com/vs/

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

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

[25] Oppel, A. (2004). *Databases Demystified.* McGraw-Hill Osborne Media.

[26] Oracle. (2019, August 13). *History of SQL*. Oracle. https://docs.oracle.com/cd/B12037_01/server.101/b10759/intro001.htm

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

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

[29] 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

[30] 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.

[31] 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.

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

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

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