International Journal of Engineering, Sciences and Technologies
ISSN : 3079-2045
https://www.ij-est.org/ Volume 3, issue 3, https://doi.org/10.70961/SNIX4926

Optimization of the driver drowsiness detection system for basic vehicles

RAKOTOARISOA Radotiana Michaël[1], RAMANANTSIHOARANA Harisoa Nathalie[2], RASTEFANO Elisée[3]

1- PhD student, Doctoral Research Team in Embedded Systems – Instrumentation – Modeling of Systems and Electronic Devices, Doctoral School of Science and Engineering and Innovation (EDSTII), University of Antananarivo.
2- Assistant Professor, Doctoral Research Team in Embedded Systems – Instrumentation – Modeling of Systems and Electronic Devices, Doctoral School of Science and Engineering and Innovation (EDSTII), University of Antananarivo
3- Professor, Doctoral Research Team in Embedded Systems – Instrumentation – Modeling of Systems and Electronic Devices, Doctoral School of Science and Engineering and Innovation (EDSTII), University of Antananarivo.

Corresponding author : RAKOTOARISOA Radotiana Michaël

E-mail : radotianamichaelr@gmail.com

Phone : +261 34 29 056 94

Abstract :

Intelligent vehicles can significantly improve the driver's experience and safety, as well as that of other road users. However, the vast majority of current intelligent vehicles, often paired with semi-autonomous or fully autonomous driving capabilities, use powerful algorithms that require extensive connectivity and advanced hardware, which can be costly. This study explores one of the options offered by these intelligent vehicles, specifically the drowsiness detection systems, from a more affordable perspective by using and combining visual and auditory methods. The goal is not only to optimize performance but also to explore the possibilities of adapting such systems for low-end vehicles. For visual detection, we use a camera system, and the images are processed using the Hough Transform or H.T. method, primarily a learning model using Random Forests and the Binary Pattern Run Length Matrix. The cost of the entire system in place and a basic type of vehicle is significantly lower than a

mid-range or high-end vehicle offering equivalent features. However, the economic savings can fund other improvements, leading to a system with the same price as mid-range or high-end vehicles but with additional features.

Keyword : driving behavior, drowsiness detection, intelligent vehicle, low-end vehicle.

1. Introduction

As one of the dominant technologies with high potential and major stakes of our time, intelligent vehicles undeniably optimize the transportation sector. However, intelligent vehicles, powered by advanced technologies such as artificial intelligence AI, autonomous driving with high-level hardware, and connectivity, are too costly for the vast majority of developing countries like Madagascar. Not only are modern infrastructures insufficient, or even nonexistent, also the vehicles are outdated, and their lifespan is extended beyond the standards of modern countries. All of this make access to intelligent vehicles difficult in these developing countries, thus depriving road users of the benefits that this technology offers. However, accidents do not differentiate between developed and developing countries, and with this inaccessibility to intelligent vehicles, the number of road accidents is significantly higher in developing countries than in developed ones. This situation led us to design a driver drowsiness detection system that offers the same advantages and safety features as the existing models implemented directly in high-end vehicles, but with optimized hardware and techniques to make it more affordable and accessible, allowing it to be implemented on low-end vehicles. Several independent studies have already laid the foundation for such a system, such as the work on drowsiness detection through voice (P.Martin et al., 2020)(Schuller et al., 2011) and research on robust yet resource-efficient methods for detecting fatigue-related facial expressions (Alioua, 2015).

The system primarily exploits vehicle parameters such as speed, trajectory, and direction, as well as driver-related parameters like reactivity or driver's state, based on physical indicators (pupil state, head position) and the driver's responses to system tests. To develop the system, the use of a learning model is essential for identifying driver parameters, image processing, and sound parameters. This model is also applied to assess vehicle parameters. However, to achieve our goal of creating a low-cost system usable in countries without modern, connected infrastructures, and in low-end vehicles, forgoing connectivity is decided. The system will therefore operate autonomously, without relying on any external vehicle systems. The learning model will rely solely on reference data stored on the vehicle's onboard computer. It is undeniable that this will reduce its performance and the accuracy of its responses compared to a connected system, which can rely on a high-availability server architecture with a significantly larger dataset. To counteract this reduction in performance, accuracy, and responsiveness of the AI component, the system architecture is adjusted to minimize the use of resource-heavy operations for complex calculations.

The system will activate in three stages. First, the parts controlling driving behavior will be activated first and continuously, without using learning algorithms but relying on robust instructions for controlling data from sensors, without the need for reference databases. Then, once the driving behavior checks indicate a possible drowsiness, the system will activate the auditory component to confirm this through a vocal analysis by questioning the driver. And finally, if the result is inconclusive, a final confirmation by analyzing the driver's behavioral signs will minimize false positives and the system's inaccuracies.

2. Methodology

2.1. General description of the system

The system consists of three components that rely on data from sensor networks: the control of driving behavior, the control of the driver's vocal behavior, and finally, the control of the driver's external visual behavioral signs.

i. **Control of Driving Behavior :**

This part monitors vehicle parameters such as speed, trajectory, and direction. It analyzes the driving patterns to detect any irregularities or signs of drowsiness based on the vehicle's motion. This is done through continuous data collection from the vehicle's sensors without requiring complex databases or algorithms.

ii. **Control of Vocal Behavior :**

The second part focuses on assessing the driver's vocal responses. When drowsiness is suspected, the system activates the auditory component, prompting the driver with questions to assess alertness. The system evaluates the clarity and speed of the driver's verbal responses, which helps in confirming or negating the presence of fatigue.

iii. **Control of External Visual Behavioral Signs:**

The third part analyzes visual behavioral signs from the driver, such as eye movement, blinking, head position, and other physical cues that could indicate drowsiness. This phase helps further confirm the driver’s alertness level by observing behavioral indicators beyond what the sensors alone can detect.

These three parts work together to detect drowsiness and ensure that the system operates efficiently with minimal false positives in result, all while being cost-effective for low-end vehicles.

The sensor networks use standard sensors, thus meeting the desired cost criteria. These sensors include:

(i) **Speed Sensors :**

The data from the speed sensors already present in all vehicles can be utilized. The goal is to detect any inconsistent changes in the vehicle's speed, which might indicate drowsiness or poor control of the vehicle.

(ii) **Steering Wheel Rotation or Movement Sensors:**

Irregular or abrupt steering wheel movements can indicate driver behavior anomalies; such as fatigue or reflex impairments due to other factors.

(iii)**Lane or Road Surface Sensors for Trajectory Monitoring:**

These sensors focus on detecting lane crossing frequencies, encroachment into the left lane without vehicle overtaking or obstacle avoidance, and any unintended drifting off the road toward the shoulder. Infrared sensors and ultrasonic sensors will be used for this purpose.

(iv)**Image and Audio Sensors for Capturing Driver Behavioral Signs:**

These sensors help monitor the driver's physical signs, such as head position, eye movement, and vocal responses. Image sensors (like cameras) will capture visual cues, while microphones will pick up vocal cues to assess the driver's level of alertness and behavior.

All these sensors work together to provide a comprehensive system for detecting drowsiness while maintaining a low-cost approach suitable for low-end vehicles.

2.2. Driving Behavior Control System

This system will be activated continuously and will first control the input data from the sensors. Its output will determine the need to activate the other levels of control.

To limit memory and processor resource consumption, this part will not rely on the use of machine learning algorithms but rather on classic algorithms with precise instructions and reference values directly implemented in the program structure. This is also necessary because infrastructures in countries like Madagascar would make the use of learning algorithms

ineffective without the elements, signs, and indications needed on the roads to feed the learning model. Otherwise, learning would only be effective with powerful algorithms and large datasets to eliminate inaccuracies.

This driving behavior control system will not rely on image processing programs but will use standard values from the sensor outputs. Each part will be controlled by a precise algorithm, detailed below.

Algorithm 1: Trajectory Control

Input :

Height H between the vehicle and the road surface indicated by the sensors

Position Pos between the vehicle and the centerline of the road

Object Obj indicating the presence of an obstacle in the vehicle's lane

Parameters **:**

P1, an integer: reference value for the normal height between the vehicle and the road surface.

P2, an interval of values: reference value for the vehicle's position relative to the centerline.

Output : Boolean value indicating 'true' for a normal trajectory and 'false' for an abnormal trajectory.

If H == P1 AND P in P2 Then :

Trajectory = True

Else :

If H == P1 AND (P not in P2 AND Obj == True) Then :

Trajectory = True

If H == P1 AND (P not in P2 AND Obj = False) Then :

Trajectory = False

If H ! = P1 AND (P in P2 AND Obj == True) Then :

Trajectory = True

If H ! = P1 AND (P in P2 AND Obj == False) Then :

Trajectory = False

Algorithm 1 allows for controlling the trajectory using only reference parameters, without the need to implement a learning model. Since all roads in Madagascar are single-lane in one direction (except for the multi-lane expressways in Antananarivo city) and driving is on the right hand side, we can check if a lane change is justified and if the car is still on the roadway.

Algorithm 2 : Speed and steering movement consistency control

Input :

Vehicle speed V

Trajectory T

Steering movement M

Output : Alert, a boolean value indicating true if an anomaly is detected

If V is increasing (acceleration) AND T == true AND M == false then:

Alert = False

Else :

If V is increasing AND T == false AND M == false then:

Alert = True

If V is decreasing AND T == true AND M == false then:

Alert = True (if no indication of stopping or intent to change trajectory, such as turn signals on) **Else Alert = False**

If V is increasing AND T == true AND M == true then:

Alert = True

Algorithm 2 monitors possible inattention or inconsistency in the driver's decisions through speed and steering movement relative to the trajectory and the driver's displayed indicators (turn signals).

These low-complexity algorithms allow for rigorous and precise control of parameters indicating the beginning of a driving anomaly, helping to save on resource consumption. They only activate the more complex driver behavioral sign detection systems when strictly necessary. This aligns with our goal of having a less complex system adaptable to low-end vehicles.

2.3. Driver Behavioral Signs Control System

For the driver's behavioral signs, the system is also optimized by breaking down the activation into two phases. First, an auditory system will detect drowsiness through sound and will question the driver in case of any anomaly detected by the vehicle behavior control system. Based on the driver's responses, the system will decide whether or not to activate the image control. The latter is the most complex in terms of processing and requires more resources.

2.3.1. Auditory control

For this part, techniques that have already been used and proven to detect drowsiness in the voice are used. A prior recording of the driver's vocal markers in a normal state is necessary during this phase, in which the driver should respond to the system if a question is asked following the detection of an anomaly in the driving parameters. The method chosen here is the detection of drowsiness in the voice through vocal markers (P.Martin et al., 2020).

In the vocal markers, parameters related to the fundamental frequency and the intensity of the voice are used, as well as the extraction of reading errors such as omissions, additions, stumbles, and paralexias.

Moreover, the classification model used is a Support Vector Machine (SVM) for detecting the state of drowsiness, with cross-validation tests on the datasets. Figure 1 shows the proposed system, which calculates performance on two corpus, TILE (Traitement de l'Intensité de la Somnolence et de l'Eveil) for treatment of sleepiness and wakefulness intensity and SLC for Sleepy Language Corpus.

The TILE corpus is a dataset containing vocal data used to study sleepiness and wakefulness through the analysis of vocal parameters. It is often employed in research related to voice activation, sleepiness or wakefulness detection, as well as for developing automatic sleepiness detection systems based on voice. The SLC corpus is a vocal database primarily used for detecting drowsiness through speech analysis. Collected in 2011, it has become a reference in this field. The SLC includes recordings of 99 German-speaking participants who took part in six partial sleep deprivation studies. The recorded vocal tasks vary, including sustained vowels, text reading, and spontaneous speech. The participants' level of sleepiness was assessed using the Karolinska Sleepiness Scale (KSS), a subjective scale ranging from 1 to 10, where 1 represents extreme alertness and 10 indicates intense drowsiness. For the SLC, an average of three KSS evaluations (one self-assessment and two assessments by external observers) was used to determine the level of sleepiness (Schuller et al., 2011).

Firstly, the vocal parameter centering by speaker involves subtracting the average vocal markers of a speaker from all their markers in order to eliminate individual factors (such as sex, age, or the physiology of the respiratory system) and retain only the instantaneous variations of the vocal parameters, allowing for a more accurate assessment of short-term subjective sleepiness.

Secondly, the Spearman correlation between each vocal marker and the sleepiness measure (KSS) is calculated, thus allowing the markers to be ranked based on their correlation. Unlike traditional reduction techniques, statistical methods preserve the meaning of the vocal markers, thereby facilitating the association between sleepiness and physiological manifestations. This calculation is performed on the entire training and development datasets.

Thirdly, the selection of the number of markers and the optimal parameters of the classifier is carried out. To do this, the system's performance is calculated (on the training set vs. the development set) for the previously ranked vocal markers, and the number of vocal markers and classifier parameters that provide the best performance are retained.

Finally, the parameters C and γ obtained in the previous step are used to train the SVM on the training and development sub-corpus, thereby providing the estimated sleepiness classes for each sample in the test sub-corpus.

Through the SVM, the system will recognize whether the driver is in an abnormal state, either through their responses or vocal markers, or in a normal state. The activation of the head pose detection systems and the detection of the driver's eye closure will depend on this state.

2.3.2. Facial features analysis system

The techniques for detecting drowsiness based on the driver's facial features have the advantage of being non-intrusive, and we can use standard cameras for image acquisition. Since they rely on cameras to capture physical signals, the equipment used is less expensive than that of other categories. However, the limitations of using cameras must be considered, such as sensitivity to lighting changes and the need for infrared lighting to enable acquisition in dark environments (Alioua, 2015). The method used is detecting the eyes to identify potential micro-sleeps and the mouth for yawning (wide mouth opening for at least 2 seconds).

The procedure used is to locate the regions of interest for the eyes and mouth operates on a grayscale face area, extracted using a library based on SVM (Alioua, 2015).

First, the eye search area is restricted by defining a lower and upper boundary on the face. This operation is performed using proposed procedure, which is divided into the following sub-steps.

(i) The face gradients in the horizontal and vertical directions, denoted as G_x and G_y are calculated. A significant spacing between the points in each direction is considered to obtain a fine contour. Subsequently, the face gradient is refered as :

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

The gradient image obtained is illustrated in Fig 2.a.

(ii) The horizontal projection of the gradient image in Fig 2.b is calculated. An element of the horizontal projection is the sum of the pixels in the given line *i*, as given by the following equation:

$$proj_h(i) = \sum_j \big(grad(i,j)\big) \quad (2)$$

where grad symbolizes the gradient image and *j* represents the index of a column.

(iii) The number of peaks in the horizontal projection is reduced by smoothing. This involves replacing every *k* elements of the projection with their average:

$$proj_h(i) = \frac{\sum_{l=i}^{i+k} proj_h(l)}{k} \quad (3)$$

Thus, the result is a reduction of the projection vector size by a factor of *k*.

(iv) Since the eye level is neither at the upper nor lower extremity of the face, we proposed zeroing out the elements of the projection vector corresponding to these parts of the face. Thus, the elements located before the maximum projection of the first third of the face and the elements located after the maximum projection of the last third are nullified (Alioua, 2015).

For the mouth, its location is determined on the lower half of the face gradient. These steps are illustrated in Fig 2.

Once the eye location is determined, one proceeds with detecting drowsiness in this area using the Hough Transform (HT) theory (Schuller, et al., 2011).

One typically group under the name HT transformations that allow the detection of the presence of parametric curves belonging to a known family in images, based on a set of selected points called characteristic points. HT primarily uses the spatial information of the characteristic points (their position in the image), but sometimes it also takes into account the information contained in the image signal itself, which corresponds to the luminance value at a given point. This signal is considered to be a scalar function : grayscale image. But nothing prevents it from being vectorial : color or multispectral image.

One will denote by *n* the dimension of the image space dimension. Let $\mathbb{R}$ be the image space, and E a set of N points selected by preprocessing, E = $\{M_i, i = 1 \dots N\} \in \mathbb{R}$. A point M in $\mathbb{R}$ is identified by its coordinates x. Let $\Omega \subset \mathbb{R}^p$ be a parameter space, and F a family of curves in $\mathbb{R}^n$ parameterized by a.

$$F = \{\{x : f(x,a) = 0, x \in \mathbb{R}^n\}, : a \in \Omega \quad (4)$$

Particularly in the case considered here, one focuses more on the theory of the Circular Hough Transform (CHT). It is used to detect circular contours in an image. Knowing that the equation of a circle is given by : $r^2 = (x-1)^2 + (y-1)^2$, the parametric representation of the circle can be written, assuming a constant radius, as follows :

$$x = a + r\cos\theta \ \ et \ y = b + r\sin\theta \quad (5)$$

In order to detect circles in an image, it is estimated that the necessary preprocessing consists of edge detection, which must be performed using an efficient edge detector.

For each point on the contour, a circle is drawn while maintaining a predefined radius. The coordinates located on the perimeter of this circle are then incremented and recorded in an accumulator. Once the circles are drawn for all points on the contour with the desired radius, the coordinates corresponding to these circles are incremented again in the accumulator. In this way, the accumulator records the number of circles passing through each coordinate. The coordinates that appear most frequently indicate the center of the circle present in the image. Figure 3 provides an illustration of the detection of the center of a circle by the CHT.

Thus, based on this process, an algorithm for iris detection can be implemented, schematized in Fig. 4. The proposed iris contour detector is based on the morphology of the eye. When observing an open eye, one notices that it consists of a dark disk in the center, corresponding to the pupil, surrounded by a larger disk whose intensity varies depending on the color of the eyes. This disk is the iris, which is itself surrounded by the sclera, a lighter area than the iris. Finally, the entire eye is framed by the upper and lower eyelids, made of skin. This specific configuration of the eye allowed us to extract the iris contour by using the intensity differences between the iris and the sclera.

Using the same principle and theory, the detection of closed eyes can be extended to the detection of yawning using the algorithm illustrated in Fig. 5.

The main steps in the algorithm described in Fig. 5 are:

I: Face detection

II: Localization of the eyes and mouth

III: Checking the status of the left eye

IV: If closed, checking the status of the right eye. If it is also closed, counting the number of consecutive frames with both eyes closed. Once this counter exceeds 2 seconds of closure, a drowsiness alert is triggered.

V: If one of the eyes is open, checking the status of the mouth

VI: If wide open, counting the number of consecutive instances of the mouth in a yawning state. If this counter corresponds to a duration greater than 2 seconds, the number of yawns is incremented, and a fatigue alert is triggered when this number becomes significant (Alioua, 2015).

2.3.3. Head pose control system

For head pose detection, which refers to the position of the head relative to the camera, in complementing the previous systems, it will help to minimize false positives. A random forest classifier and a binary pattern run-length matrix are used. This method allows us to have a robust system against lighting variations.

The proposed approach can be summarized by the steps described below. Firstly, Random Forests are used as classifiers. With this classifier, the system can operate in real-time and handle a large training dataset. Secondly, for binary testing, the Binary Pattern Run Length matrix is proposed. This method combines a binary pattern and a run-length matrix. The binary pattern is calculated using a randomly selected operator, such as the Local Binary Pattern (LBP), Centralized Binary Pattern (CBP), and Local Directional Pattern (LDP). Statistical texture features, such as Short Run Emphasis and Long Run Emphasis, are used. This strategy makes the system robust to lighting variations, and classification performance is improved. Thirdly, the key parameters for binary tests at each node are optimized using information gain. The resulting optimal binary test improves the discriminative power of individual trees in the forest. Finally, to achieve more efficient data splitting, the number of iterations for generating parameters is increased. With this strategy, patches are roughly divided at initial depth levels and split more finely at higher depth levels (Kim, et al., 2014).

For Random Forests, the choice is justified by the fact that it is one of the most popular methods while being suitable for low-cost and low-power hardware such as microcontrollers or Raspberry Pi.

In the context of real-time pose estimation, multi-class random forests have been proposed for the real-time determination of head pose from 2D video data. A person-independent head pose estimation method is proposed by the research of Li Y, Wang. They use half-face classifiers and tree-structured classifiers with the AdaBoost cascade algorithm to detect faces with different head poses. After localization, a regression using random forests is trained and applied to estimate the head orientation (Kim, et al., 2014).

For the binary pattern, the original LBP operator labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the central value, then considering the result as a binary number, whose corresponding decimal number is used for labeling. Formally, for a pixel located at (xc, yc), the resulting LBP can be derived by the following equation.

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n \quad (6)$$

where *n* iterates over the 8 neighbors of the central pixel, i_c and in are the grayscale values of the central pixel and the neighboring pixels, respectively, and the sign function *s(x)* is defined as follows :

$$s(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases} \quad (7)$$

According to the definition above, the LBP operator is invariant to monotonic transformations of the grayscale that preserve the order of pixel intensities in local neighborhoods. The histogram of the LBP labels computed over a region can be used as a texture descriptor (Kim, et al., 2014).

For the gray-level run length matrix part, the Gray Level Run Length (GLRL) method is a way to extract higher-order statistical texture features (Fanelli, et al., 2013). A group of adjacent pixels with the same grayscale level, arranged in a specific direction, forms a gray-level run. The length of this run corresponds to the number of pixels that compose it, while the run length value represents the number of occurrences of this sequence in the image.

The proposed algorithm for head pose estimation is shown by Fig.6.

(i) For the random forest framework

A tree T in a forest $F=\{T_i\}$ is constructed from a set of annotated patches $P = \{P_i = (I_i, c_i)\}$ randomly extracted from training images. Here, I_i and c_i represent the intensity of the patches and the class labels associated with head orientation, respectively. Starting from the root, each tree is built recursively by assigning a binary test $\emptyset\ (I) \rightarrow \{0, 1\}$ to each non-leaf node. This test directs each patch either to the left child or to the right child, thereby dividing the set of training patches P arriving at the node into two distinct sets, $PL(\emptyset)\ and\ PR(\emptyset)$.

The best test $\emptyset^*$ is chosen from a pool of randomly generated ones ($\{\emptyset\}$). All patches arriving at the node are evaluated by all tests in the pool and a predefined information gain of the split $IG(\emptyset)$ is maximized by the following equation.

$$\emptyset^* = Argmax_{\emptyset} IG(\emptyset) \tag{8}$$

The process continues with the left and the right child using the corresponding training sets $PL(\emptyset^*)$ and $PR(\emptyset^*)$ until a leaf is created when either the maximum tree depth is reached, or less than a minimum number of training samples are left (Gross, et al., 2010).

(ii) Training

All the trees are trained on different training sets. These sets are generated from the original training set using the bootstrap procedure. For each training set, N data points are randomly selected from the original set. The data are chosen with replacement. That is, some data points will appear multiple times and others will be absent. Then, M patches of fixed size are randomly extracted.

2.4. Hardware Requirements

To ensure a cost-effective and practical implementation of the drowsiness detection system suitable for low-end vehicles, the hardware components were carefully selected based on availability, affordability, and compatibility with embedded systems such as Arduino and Raspberry Pi platforms. The system comprises the following key components :

- **Microcontroller/processing unit :** An Arduino Uno board is used for basic sensor data acquisition and processing, while a Raspberry Pi 4 is considered for more demanding tasks such as vocal and image analysis.
- **Driving behavior sensors :** Speed detection is achieved via Hall effect sensors mounted on the wheel, while steering wheel movements are captured using rotary encoders. Trajectory monitoring leverages infrared (IR) sensors for lane detection and ultrasonic sensors to detect nearby obstacles.
- **Vocal behavior sensors :** A MEMS microphone captures the driver's vocal responses for auditory analysis.
- **Visual behavior sensors :** A combination of a standard RGB camera and an infrared (IR) camera module enables facial feature detection and analysis even in low-light conditions, complemented by IR illumination when necessary.

- **Additional modules :** Data storage is facilitated through SD card modules ; audio alerts are delivered via a buzzer ; and optional display modules (LCD or LEDs) provide visual feedback to the driver.

This selection balances performance and low cost, ensuring the system remains accessible for use in developing countries with limited vehicle and infrastructure resources.

A detailed summary table of the components in Table I, including pricing information, clearly demonstrates that the overall cost remains well within reach of an average budget. Indeed, the total cost of the components, when added to the standard price of a low-end vehicle, remains significantly lower than the standard price of a high-end car offering the same features, as illustrated in Table II.

3. Results

The methods described in the second phase and the third and final phase are both proven methods in various studies (P.Martin, et al., 2020)(Alioua, 2015) and the results are all conclusive in the literature. The originality, simple yet aligned with objective to be atteined here, lies in the combination of these three methods : the control of driving behavior, the control of the driver's vocal behavior, and finally, the control of the driver's external visual behavioral signs. A first step consists of a simple system reduced to a network of sensors and processing of sensor outputs using a precise algorithm but without a learning model. The second phase calls the sound control modules, followed by the visual control modules if the results of the first phase require additional control by these modules.

The practical results, on infrastructures where the roads contain only two lanes in opposite directions with a speed limit of 80 km/h on average (as is the case for most roads in Madagascar, except for a few highways in the capital), show results that meet expectations and demonstrate fairly good responsiveness. Indeed, the detectors provide timely alerts and allow for a reaction

before the incident occurs. To obtain results in a real-world setting, tests were conducted using a miniature car model equipped with the mentioned sensors and an Arduino board where implemented algorithms as illustrated in Fig.7.

Tests conducted on a motorized miniature car on a paved track showed a detection rate of 92.5%, with a false positive rate of 5% across a total of 40 test runs. However, such tests primarily assess the reliability of Phase 1, which focuses on behavioral monitoring. The vocal and visual monitoring phases were simulated, as it was not possible to place a real driver inside a miniature vehicle. Therefore, actual real-world performance is expected to be slightly below these figures. Camera quality and processor speed will be key factors affecting performance, but the deviation from the simulated results is not expected to be significant. Simulations were carried out under a variety of test scenarios.

4. Discussion

The approach is quite efficient and significantly more cost-effective compared to the complex systems found in high-end and modern vehicles from automotive manufacturers. However, with a relatively simple algorithm that does not require a more complex system and that utilizes a learning model (in this case, Random Forest, which is the most suitable for low- or medium-power hardware like Arduino), the system will struggle to operate in real-time if the vehicle is moving at high speeds, such as on highways and expressways. Not only the processing time is relatively slow, but the number of false positives will be high, and several parameters may be inaccurately calculated by the system at the right time. Thus, the system is better suited for low to medium speeds (on average, 80 km/h), regardless of road conditions, but beyond this speed, it may not be entirely relevant.. Nevertheless, in the context of Madagascar's infrastructure, this average speed is rarely reached, not only due to the absence of highways but also because of the poor condition of the roads, forcing drivers to

travel at reduced speeds that rarely reach normal averages. However, driving at low speed on a deteriorated road does not mean reduced fatigue or drowsiness; on the contrary, this situation is extremely tiring. This system is more than adequate for this context.

5. Conclusion

The system designed here was primarily created to adapt existing technology to a broader range of users, particularly in terms of methodology for implementing embedded systems that make vehicles 'smart' in the technical sense of the term, so that these systems can adapt to the context of the Malagasy transport sector. The prototype is adapted to the country's infrastructure, both in terms of road conditions and the aging vehicle fleet, which significantly limits the speed of drivers. A speed limitation that allows the system to react in time (responsiveness is not good at high speeds due to the use of inexpensive components and a low-resource computing technique) while optimizing both economic cost and technical complexity. This optimization offers both economic and technical flexibility to add additional options such as detection, or even predictive analysis, of vehicle components requiring maintenance. Since breakdowns are frequent on our roads, this option will be welcome.

6. References

Vincent P. Martin, Jean-Luc Rouas, Pierre Philip, (2/2020), Tal V61, « Détection de la somnolence dans la voix : nouveaux marqueurs et nouvelles stratégies », pp. 67-90

Björn Schuller, Stefan Steidl, Anton Batliner, Florian Schiel, Jarek Krajewski, (2011) « The interspeech 2011 Speaker State Challenge », 10.21437/ Interspeech.2011-801

Nawal Alioua, (2015), « Extraction et analyse des caractéristiques faciales : application à l'hypovigilance chez le conducteur », Informatique [cs]. INSA Rouen, University Mohammed V-Agdal (Rabat, Maroc). NNT : 2015ISAM0002.

Hyunduk Kim, Sang-Heon Lee, Myoung-Kyu Sohn, Dong-Ju Kim, (2014), « Illumination invariant head pose estimation using random forests classifier and binary pattern run length matrix », Human-centric Computing and Information Sciences, 4-9

Fanelli G, Danotone M, Gall J, Fossati A, Van Fool L, (2013), « Random forests for real time 3D face analysis », International J of Computer Vision 437-458

Gross R, Matthews I, Cohn JF, Kanade T, Baker S, (2010), « Multi-PIE », Image Vis Comput 807-813

7. Tables

Table I : Component Summary Table

Component	Model / Type	Technical Reference / Specifications	System Usage	Price
Microcontroller	Arduino Uno	ATmega328P, 16 MHz, 14 GPIO pins	Basic sensor data acquisition and processing	60.000 MGA
Microprocessor	Raspberry Pi 4	Quad-core 1.5 GHz, 4 GB RAM	Advanced voice and image processing	1.000.000 MGA
Speed Sensor	Hall Effect Sensor	Digital output, wheel rotation detection	Vehicle speed measurement	15.000 MGA
Steering Position Sensor	Rotary Encoder	20 pulses/rev resolution, digital output	Steering angle/movement detection	20.000 MGA
Trajectory Sensors	IR Reflectance Sensor	White line detection, range 2 – 10 cm	Lane position tracking	5.000 MGA
Obstacle Sensor	HC-SR04 Ultrasonic Sensor	Range 2 – 400 cm, 15° detection angle	Obstacle and boundary detection	8.000 MGA
Microphone	MEMS Microphone SPW2430	Sensitivity – 38 dBV, frequency 100 Hz – 10 kHz	Driver voice capture	30.000 MGA
RGB Camera	Standard USB Webcam	Resolution 640X480 or 1280X720, USB 2.0	Facial, eye, and mouth detection	40.000 MGA
Infrared Camera	Raspberry Pi NoIR Camera	IR sensitivity, 8 MP resolution	Night vision, facial detection	75.000 MGA

IR Lighting	850 nm IR LED	1 W power, range 1 – 2 m	Face illumination under low-light conditions	1.400 MGA
Storage Module	Micro SD Card	Capacity 32 GB	Storage of audio and image data	50.000 MGA
Audio Module	Piezo Buzzer	3 – 5 V, variable sound frequency	Audible alerts for the driver	3.000 MGA
Display	16 X 2 LCD Screen	I2C interface, backlit	System status and visual alerts display	20.000 MGA
Connection cable	Dupont wires, connectors	45 Dupont wires and 10 connectors	Wires and connectors, Power supply	44.000 MGA

Table II : Table of Cost Comparison

Product / Service	**Average Cost**
Low-end vehicle	15.000.000 MGA
Set of hardware components	1.375.400 MGA
Software analysis and development work	1.600.000 MGA
High-end vehicle	30.000.000 MGA

8. Figures

International Journal of Engineering, Sciences and Technologies
ISSN : 3079-2045
https://www.ij-est.org/ Volume 3, issue 3, https://doi.org/10.70961/SNIX4926

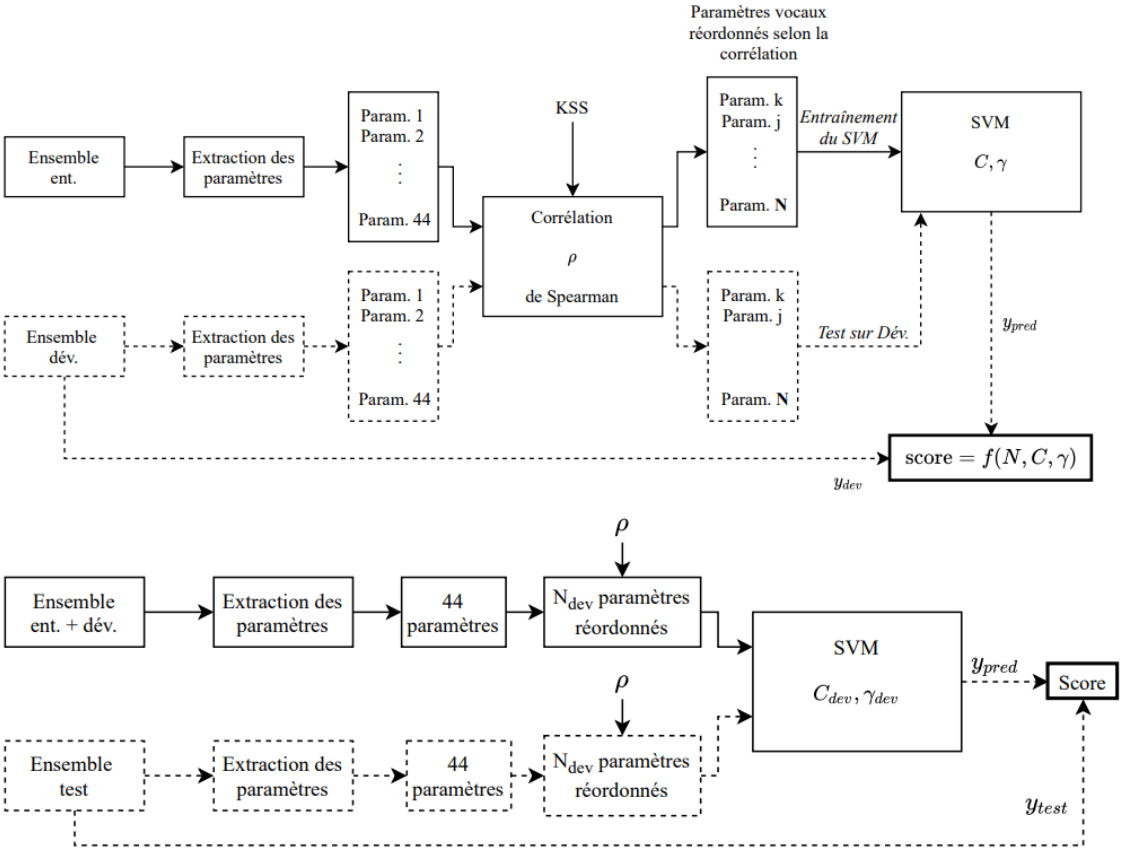

Figure 1 : Diagram of the proposed system with training and development [1]

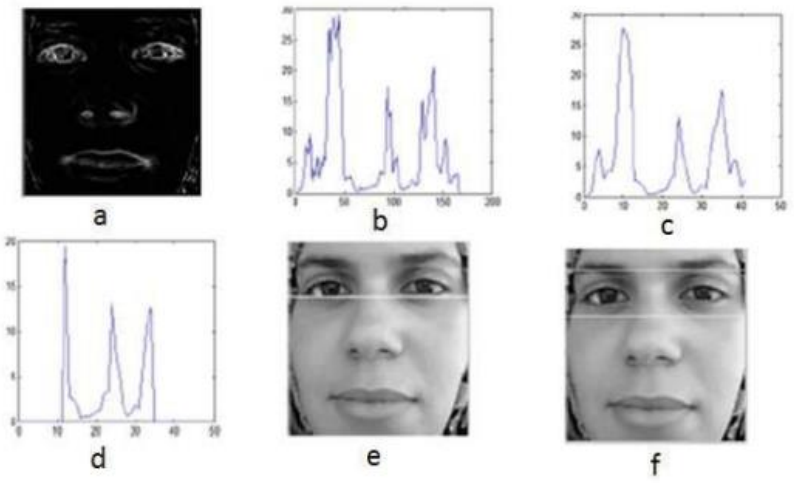

Figure 2 : Steps for locating the lower and upper boundaries of the eyes. (a) Image Gradient; (b) Horizontal Projection; (c) Raised Horizontal Projection; (d) Processed Horizontal Projection; (e) Eye Levels; (f) Eye Area. [3]

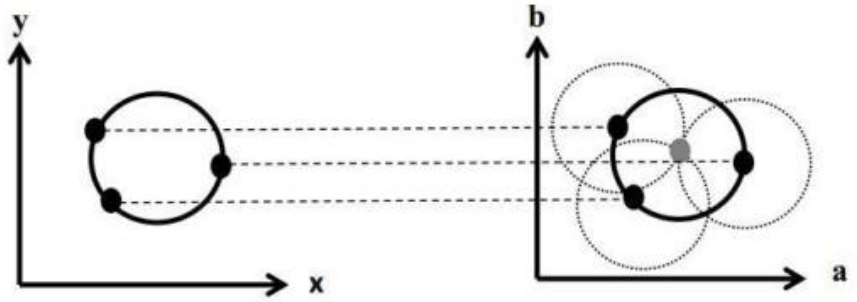

Figure 3 : Illustration of circle center detection using the Circular Hough Transform (CHT) *[3].*

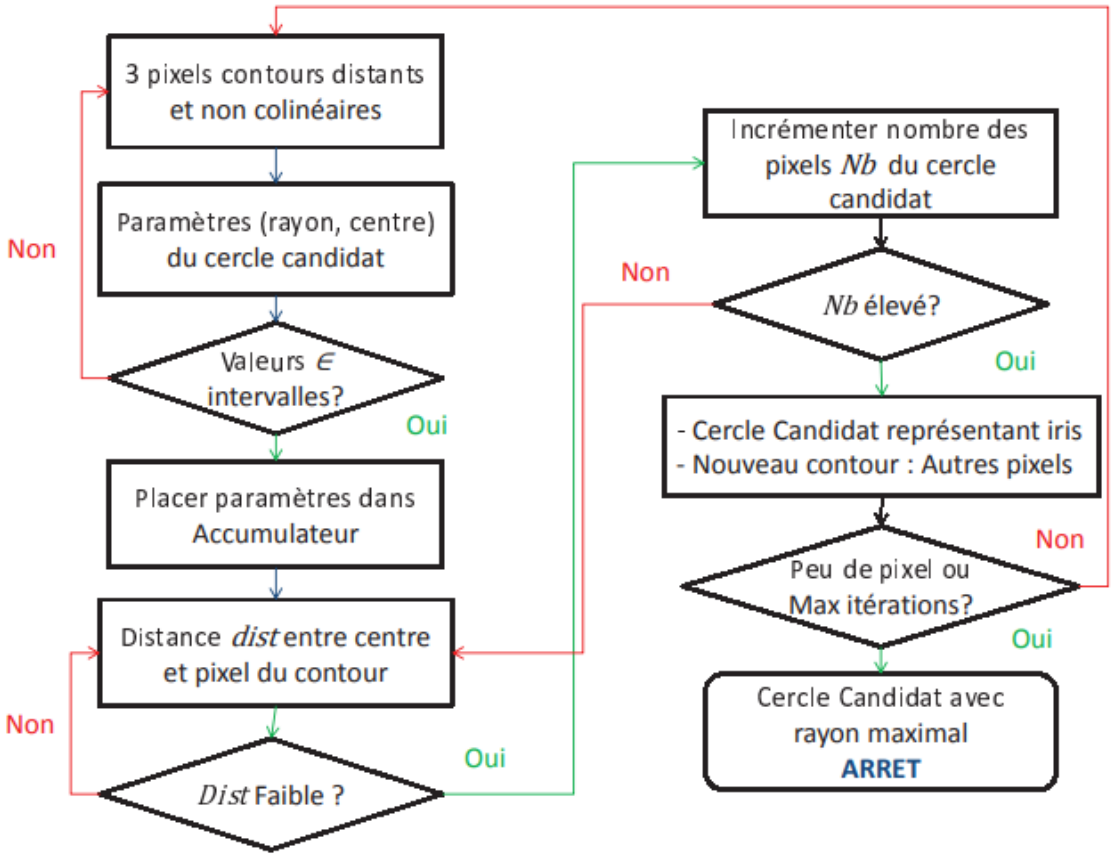

Figure 4 : Iris detection algorithm using CHT [3]

International Journal of Engineering, Sciences and Technologies
ISSN : 3079-2045
https://www.ij-est.org/ Volume 3, issue 3, https://doi.org/10.70961/SNIX4926

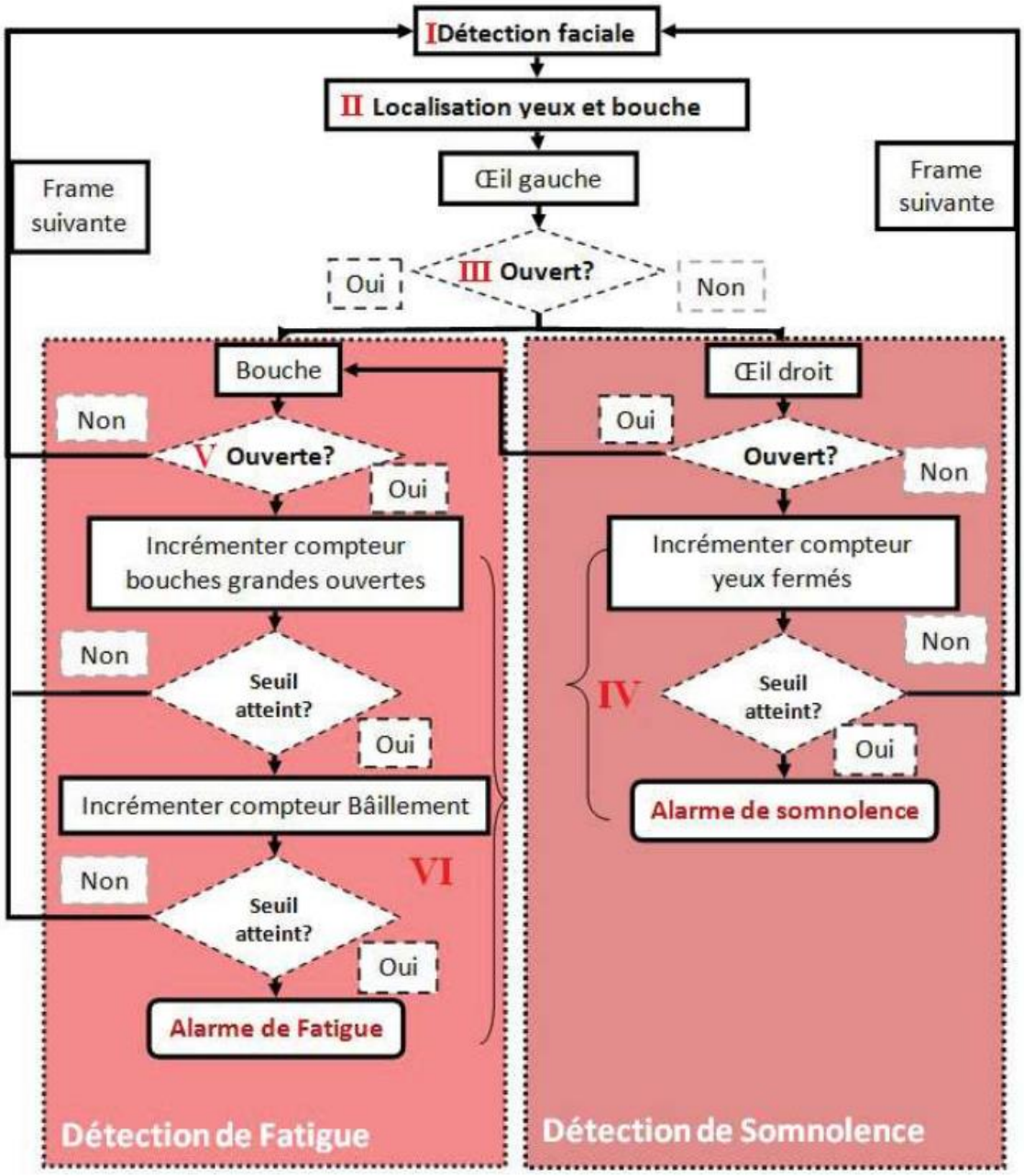

Figure 5 : General Diagram of fatigue and drowsiness detection in the driver [3]

Algorithm 1 Training the random forest

Input: T_{max}: the maximum number of trees to grow
D_{max}: the maximum depth of trees to extend
B_{max}: the maximum number of binary tests to split
S: training sets
Initialize: $i = 0, j = 0$

Loop: $T_i < T_{max}$
$i = i + 1$
Select n new boostrap samples from training set S.
Extract m sample patches from each sample.

Loop: $D_j < D_{max}$
$j = j + 1$
1. Grow an unpruned tree using the nm sample patches.
2. Each internal node, randomly generate $B_{max}D_j/D_{max}$ binary tests and determines the best binary test. The binary test $\phi(I)$ iteratively splits the training data into left node P_L and right node P_R using equation (10).

$$P_L = \{I_i \in P \mid \phi(I_i) < \tau\}, \quad P_R = P \setminus P_L \tag{10}$$

The threshold τ is randomly chosen by the binary test $\phi(I)$ in the range $\tau \in (\min \phi(I), \max \phi(I))$
Loop end

Add the i-th decision tree to the random forests.
Loop end

Output: random rorest F

Figure 6 : Training the random forest algorithm [4]

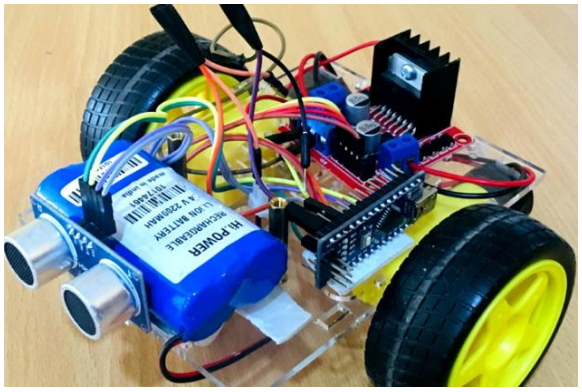

Figure 7 : Experimental mini-vehicle with sensors and Arduino board