Journal of Theoretical and Applied Information Technology
30th September 2024. Vol.102. No. 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REMOTE MONITORING OF THE DRIVER'S HEART RATE, BREATHING RATE AND FATIGUE SIGNS USING A HYBRID APPROACH ON DIFFERENT EMBEDDED ARCHITECTURES

HODA EL BOUSSAKI[1] , RACHID LATIF[1] , AMINE SADDIK[1,2]

[1]Laboratory of Systems Engineering and Information Technology LISTI, ENSA, Ibn Zohr University, Agadir, Morocco

[2]Faculty of Applied Sciences, Ibn Zohr University, Ait Melloul, Morocco

E-mail: hodaelboussaki@gmail.com, r.latif@uiz.ac.ma
amine.saddik@edu.uiz.ac.ma

ID 55309 Submission	Editorial Screening	Conditional Acceptance	Final Revision Acceptance
10-08-24	11-08-2024	30-08-2024	24-09-2024

ABSTRACT

In this work, a hybrid approach to remote monitoring of the heart rate, the breathing rate and the driver's fatigue signs is presented. The outcomes have been applied across various architectures, including a Desktop, the Odroid XU4, the Jetson Nano and the Odroid H2. The proposed algorithm uses the Dlib library for face detection, facial landmarks and the extraction of the region of interest. It also utilizes remote plethysmography, which is a video-based method allowing non-contact monitoring of blood volume fluctuations by identifying changes in pixel intensity on the skin, for the estimation of the heart rate and the respiratory rate. The efficiency of Dlib's face detection capabilities is assessed in comparison to other algorithms, including those optimized for GPU implementation.

Keywords: *Heart rate, Breathing rate, Fatigue, Driver, Non-contact*

1. INTRODUCTION

Certain prominent indicators of drowsy driving encompass yawning and experiencing difficulty in maintaining wakefulness with open eyes. Although it is more prevalent among drivers covering extensive distances, drowsiness or feeling sleepy is a potential occurrence for anyone. Even highly experienced drivers can be vulnerable to this condition and may overlook the indications that it is no longer safe for them to operate a motor vehicle. The American Automobile Association (AAA) Foundation released a report on traffic safety in 2015, highlighting that fatigue while driving impairs drivers' attention and decision-making capabilities, leading to accidents. The report urges the public to be aware of the hazards and severity associated with driving while fatigued [1]. Their research also revealed that drowsiness played a role as a contributing factor in as many as 9.5 percent of all reported accidents [2]. In the year 2020, there were a total of 633 traffic fatalities resulting from accidents related to drowsy driving according to the National Highway Traffic Safety Administration (NHTSA). Every year, approximately 100,000 police-reported crashes and over 1,500 deaths occur as a result of drowsy driving [3]. And, A staggering 40% or more of drivers have confessed to having dozed off while driving. In addition to accidents resulting from driving while fatigued, there are accidents attributed to medical conditions. Variations in heart rate and respiratory rate while in a seated position can serve as indicators of numerous medical conditions, posing a potential risk of fatal accidents to both the driver and those in proximity. Syncope or passing out as a result of a medical condition exhibited the highest odds ratio for being responsible in crashes. As syncope is a symptom rather than a standalone condition, it could potentially be linked to various underlying conditions like diabetes, high blood pressure, and so on [4]. According to the NHTSA, heart attacks represent 2.4% of road accidents [5]. Pulmonary ventilation disorders that affect the respiratory system can cause accidents when driving [6].

Several methods exist to monitor the heart rate and the respiratory rate with or without contact between the person and the measuring system. There are methods that have been incorporated into various components of vehicles, such as the seat-belt, the seat, or the steering wheel. Additionally, there are methods for measuring vital signs through a radar

system and through camera-based photoplethysmography (PPG). Photoplethysmography (PPG) is an uncomplicated optical method employed to observe alterations in the volume of blood within peripheral circulation. This cost-effective and non-invasive technique conducts measurements on the skin's surface. Given that blood absorbs light more substantially than the surrounding tissues, PPG sensors can discern fluctuations in blood flow by monitoring changes in light intensity [7]. Sandberg and Wahde 2008 examined time-series data, including parameters like vehicle speed and steering wheel angle, for the purpose of detecting fatigue while driving [8]. To monitor both heart rate and respiratory rate, the Harken project was created by integrating the system into the seat-belt [9]. As another system integrated into the seat-belt, S.T. Hamdani and A. Fernando 2015 employed piezo-resistive sensors that were integrated into the seat-belt for the purpose of measuring the driver's heart rate and respiratory rate [10]. Y. Jiang et al. in 2019 introduced an alternative approach, utilizing a pressure sensor that was integrated into the backrest of the seat [11]. K. J. Lee et al. 2016 used continuous-wave Doppler Radar to estimate a driver's heart rate. The radar is installed in the seat. The emitted signal gets reflected and contains information about the heart activity [12]. As for the systems integrated into the steering wheel, S. Heuer et al. 2010 used capacitively coupled ECG electrodes (CCE) integrated in the steering wheel [13]. The second category of monitoring systems are non-contact. S. Kuda et al. 2018 used RGB and near-infrared (NIR) face videos [14]. L. Maurya et al. 2022 used RGB and thermal images to measure the breathing rate [15]. F. Yang et al. 2022 also used the temperature below the nostrils that rises during exhalation and drops during inhalation to estimate the respiratory rate [16]. The last category are systems based on radar technology. Z. Yang et al. 2018 used an impulse ultra-wide band (UWB) radar to monitor the driver's breathing rate. The system is able to detect chest movement [17]. N. Maleˇsevic et al. 2020 utilized a continuous-wave Doppler radar operating at a carrier frequency of 24 GHz, coupled with an artificial neural network (ANN), to gauge heart rate [18].

In our research, we present an algorithm designed to estimate the breathing rate, the heart rate and fatigue signs using an RGB camera. Fatigue indicators encompass yawning and the struggle to maintain open eyes. We employed the dlib mmod, the dlib Hog, opencv haar cascade and opencv DNN for face detection and the dlib library for facial landmarks detection. Once the facial landmarks are identified, we can assess whether the eyes are open or closed and whether the person is yawning. Additionally, facial landmark detection enables us to extract the region of interest (ROI), which is the forehead, where a majority of blood vessels are concentrated. Subsequently, the heart rate and breathing rate are calculated using equations that utilize the information extracted from the ROI. The algorithm was then implemented on different architectures including a Desktop, the Odroid XU4, the Jetson Nano and the Odroid H2. We were able to obtain a frame rate of 12 frames/s on the Desktop, 3 frames/s on the Odroid XU4, 28 frames/s on the Jetson Nano and 8 frames/s on the Odroid H2. The summary of our contribution is is as follows:

- The proposition of a hybrid approach that allows the estimation of both the heart rate, the breathing rate and the detection of fatigue signs using the Dlib library.
- The implementation of the algorithm on different embedded architectures to choose the most performant one.
- The comparison of four different face detection algorithms to determine the least consuming one time-wise and the acceleration using CUDA of Dlib mmod and OpenCV DNN.

The structure of this paper is as follows: Section II offers an overview of recent developments in contactless monitoring of breathing rate and heart rate. Section III delineates our methodology. Following that, Section IV provides the results obtained. Lastly, the conclusion consolidates the study's findings and provides insights into future prospects.

2. BACKGROUND AND RELATED WORK

2.1 Background of rPPG

Numerous studies have explored the estimation of heart rate using an RGB camera. It has been possible to monitor heart rate and respiratory rate by observing the fluctuations in RGB colors within an image caused by changes in blood flow in the capillaries. The signal extracted from these variations is referred to as a photoplethysmography (PPG) signal. The process involves illuminating the human skin with a light source, while a camera captures the changing color patterns [19]. Figure 1 represents the mechanism of specular reflection and diffuse reflection.

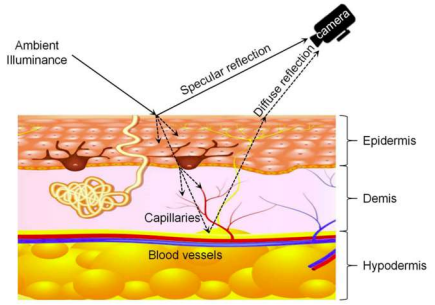

Figure 1: Reflection model of rPPG [20]

The reflection of a skin pixel is defined the following equation 1:

$$C_K(t) = I(t).(v_s(t)+v_d(t)+v_n(t)) \quad (1)$$

Where k is the kth pixel, I(t) denotes the illumination intensity level, $v_s(t)$ denotes the specular reflection that occurs on the surface, $v_d(t)$ denotes the diffuse reflection on the blood vessels and $v_n(t)$ denotes the noise measurement.

The function $v_s(t)$ represents a mirror-like light reflection from the skin surface, devoid of pulsatile information. It varies over time due to motion, which alters the distance (or angle) between the light source, the skin surface, and the camera. The specular reflection is a mirror-like reflection and does not contain information of the pulse. It can be expressed in equation 2 [20].

$$u_s(t) = u_s.(s_0+s(t)) \quad (2)$$

Where u_s denotes the unit color vector of the light spectrum and s_0 and s(t) denote the stationary and changing parts of the specular reflection.

The diffuse reflection is related to the absorption of the light. v_d is linked to the absorption and scattering of light within the skin tissues. It is defined in equation 3 [20].

$$v_d(t) = u_d.d_0+u_p.p(t) \quad (3)$$

Where u_d denotes the unit color vector of the skin, d_0 denotes the stationary reflection,
up denotes the relative strength of the pulse in the channels and p(t) is the signal.

2.2 Heart rate

Heart rate (HR) is defined as the frequency at which cardiac beats occur within a specified time interval, typically measured in beats per minute [21]. Typical resting heart rate values for adults generally range from 60 to 90 beats per minute (bpm). The American Heart Association, on the other hand, defines a normal sinus heart rate as falling between 60 and 100 bpm [22]. The most basic test to evaluate the heart's rhythm and measure its electrical activity is the electrocardiogram (ECG)[23]. In a typical ECG reading, the standard upward deflection of a QRS complex is observed at the apex of the R wave [24]. The signal is normally denoise to obtain the heart rate [25]. In their research, Poh et al. 2011 introduced an uncomplicated and cost-effective approach for quantifying physiological parameters such as heart rate (HR), respiratory rate (RR), and heart rate variability (HRV) employing a standard webcam. They extracted the pulse signal through an independent component analysis (ICA) based blind source separation (BSS) technique applied to the three RGB color channels of facial video recordings, yielding three distinct independent components. In their study, the facial region of interest (ROI) was defined as a rectangular bounding box, automatically identified using the Viola-Jones face detector [26]. W. Verkruysse et al. 2008 suggested a manual selection of the forehead region of interest (ROI). Subsequently, raw signals were computed as the average pixel values within this forehead ROI. These signals were then subjected to bandpass filtering using a fourth-order Butterworth filter [27]. Heart rate (HR) was extracted from the signal's frequency content using Fast Fourier Transform (FFT) for each 10-second interval. The researchers noted variations in the strength of PPG signals across different channels of the RGB camera, with the green channel exhibiting the most pronounced pulsatile signal [28]. This finding aligns with the fact that hemoglobin's light absorption is most sensitive to changes in oxygenation when exposed to green light [29]. S. Sahin et al. in 2021 also employed the Viola-Jones algorithm to detect specific regions of interest, namely the forehead and the cheeks. To mitigate noise, they utilized both Independent Component Analysis (ICA), particularly FastICA, and Principal Component Analysis (PCA). They then subjected the resulting signal to a 3rd order Butterworth bandpass filter to eliminate values falling outside the range of 40 bpm to 180 bpm. The signal was further processed using Discrete Fourier Transform (DFT) to convert it into a power spectrum, where the peak corresponds to the heart rate. This algorithm underwent testing with 10 different participants, achieving an accuracy of 85% [30]. When FastICA was utilized, the accuracy improved to 92% [30]. Z. El khadiri et al. 2023 employed signal decomposition, dimensionality reduction via Principal Component Analysis, signal denoising using the moving average algorithm [31]. They also used conversion of the PPG signal from the time domain to the frequency domain using Fourier Transform in order to estimate the heart rate [32].

2.3 Breathing rate

The respiratory rate, which signifies the count of breaths taken per minute, is meticulously controlled to ensure that cells generate the optimal

amount of energy as required at any given moment [33]. C. B. Pereira et al. in 2015 devised their method on the premise that the temperature around the nostrils undergoes fluctuations throughout the respiratory cycle. They implemented a process to detect and track the region of interest, specifically the nostrils. They conducted their experiments by capturing three distinct thermal recordings from a group of 11 healthy volunteers [34]. Y. Takahashi et al. in 2021 introduced a system for estimating respiratory rate (RR) utilizing thermal imaging cameras. Their approach involves measuring temperature variations during exhalation and inhalation to monitor the subject's respiration in real-time, whether they are in a supine or seated position, and all without any physical contact. The facial region is detected using YOLOv3, an object detection model grounded in deep learning techniques [35]. J. Kempfle et al. 2018 employed an RGB-D camera for their study. They were able to accurately estimate the respiratory rate by analyzing specific portions of the depth data collected from a Kinect v2 camera. This depth data provides insights into the movements of the chest [36].

2.4 Fatigue signs

C. Sravan at al. 2018 employed the Viola-Jones algorithm to first detect the face and subsequently identify the eyes in a given picture [37]. An alternative approach involves utilizing funnel-like features for multi-view face detection. These features exhibit a wider top and taper down to a narrower bottom, resembling an inverted pyramid shape. This method encompasses a three-stage classifier: a rapid LAB cascade, a coarse Multilayer Perceptron (MLP), and a fine MLP cascade classifier [38]. The fast LAB cascade quickly eliminates portions of the image that do not contain faces. The coarse Multilayer Perceptron employs Speeded-Up Robust Feature (SURF) to further refine regions that contain faces. SURF features are preferred as they offer greater expressiveness compared to LAB features. MLP, a neural network consisting of three layers (input, output, and hidden layers), is then applied for more refined classification. Zhu et al. 2022 employed a tasks-constrained deep convolutional network (TCDCN) algorithm to identify facial features. They utilized an AdaBoost classifier to assess whether the eyes are open or closed based on the percentage of eye closure (PERCLOS). If the PERCLOS value surpasses 0.8, it indicates that the driver is deemed drowsy [39]. S. Mehta at el. 2019 used the dlib library for the detection of the facial landmarks and the EAR is measured [40].

Table 1 summarizes the works cited above.

Table 1: Summary of the existing work on heart rate, breathing rate and fatigue signs.

Work	Feature extracted	Method	Reference
Poh et al. 2011	HR/RR	Viola-Jones face detector and ICA	[26]
W. Verkruysse et al. 2008	HR	The green channel method	[29]
S. Sahin et al. in 2021	HR	FastICA and PCA	[30]
Z. El khadiri et al. 2023	HR	PCA	[31]
C. B. Pereira et al. in 2015	RR	The thermal recordings of the nostrils	[34]
Y. Takahashi et al. in 2021	RR	YOLOv3 for face detection and the thermal recordings of the nostrils	[35]
J. Kempfle et al. 2018	RR	The depth data collected providing insights into the movements of the chest	[36]
C. Sravan at al. 2018	Fatigue	Funnel-like features for multi-view face detection	[37]
Zhu et al. 2022	Fatigue	Tasks-constrained deep convolutional network (TCDCN) and Adaboost classifier	[40]
S. Mehta at el. 2019	Fatigue	Dlib library	[41]

3. Methodology and algorithm study

We can summarize our methodology as follows: We begin by detecting the face and facial landmarks using the Dlib library. Following this, we designate the region of interest as the area immediately above the facial landmarks of the eyebrows. Once these initial steps are completed, we can extract the heart rate and breathing rate by analyzing changes in light intensity captured by the camera. Figure 2 represents the flow chart of the

global algorithm. In figure 2, the face and facial landmarks are detected. Subsequently, if the signal lacks sufficient amplitude, new values are introduced, corresponding to the pixel values of the red, green, and blue bands. Following this, only the green band is retained due to it containing the most pertinent information regarding blood flow. The extracted raw signal undergoes normalization, and a combination of a moving average filter and discrete Fourier transform is applied to it. Band masking is then employed for estimating heart rate and respiratory rate, utilizing the maximum of the power spectrum for calculation. Following the estimation of vital signs, fatigue detection is initiated. The eye aspect ratio is computed, and if it falls below a threshold value of 0.23, a counter is incremented, representing the duration the driver keeps their eyes closed. When this counter reaches a value of 10, an alarm is supposed to be triggered. Yawning detection involves calculating the distance between the upper lip and lower lip, and if it exceeds 60, the driver is identified as yawning.

3.1 Face detection and ROI extraction

Dlib is a modern C++ toolkit equipped with machine learning algorithms designed for addressing real-world challenges by creating intricate software solutions in C++ [41]. Dlib is frequently employed in tasks like facial recognition, object detection, image processing, and numerous other applications. Dlib is implemented using OpenCV and C++. The Dlib library is used to identify and precisely locate facial landmarks. More specifically, Dlib's pre-trained facial landmark detector, which is powered by a shape predictor model trained on the i-Bug 300-W dataset. This model is proficient at identifying and localizing 68 facial landmarks within a facial image.

Dlib is employed with the specific aim of identifying landmarks associated with the eyes and mouth to detect signs of fatigue. Additionally, it is used to pinpoint landmarks on the forehead, which serves as the region of interest (ROI). This ROI is crucial for calculating both heart rate and breathing rate. The Dlib library employs the ERT (ensemble of regression trees) cascade regression algorithm, specifically relying on the gradient learning-based regression tree approach. This algorithm utilizes a cascade of regression factors and begins by training on a set of carefully calibrated face images, ultimately generating a model for its operations [42]. Fig. 3 shows the location of the 68 facial landmarks.

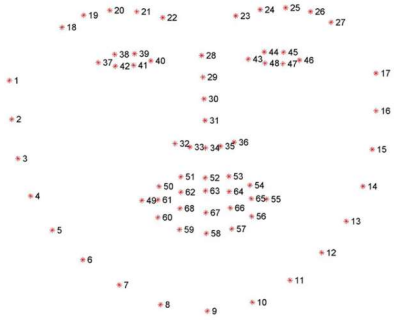

Figure 3: The 68 facial landmarks [43]

The right eye is associated with landmarks numbered 37 to 42, the left eye corresponds to landmarks numbered 43 to 48, and the mouth encompasses landmarks ranging from 49 to 68. To extract the forehead region, we utilize landmarks 20 and 25, applying a translation to encompass the forehead area

3.2 Heart rate and respiratory rate monitoring

To estimate the heart rate and breathing rate, we begin by extracting a signal from the region of interest. At this state, the signal comprises variations in pixel intensity. Subsequently, we filter this signal and apply band masks to retain only the frequencies that contain relevant information pertaining to both the heart rate and breathing rate.

3.2.1 Raw signal extraction

The raw signal is derived from the image using a function that calculates the average pixel values for each channel (red, green, and blue). These channel averages are then merged to create a composite signal. Within the region of interest (ROI), the RGB components are spatially averaged across all pixels, generating an RGB component for each frame. These averaged RGB components collectively constitute the raw signals. As additional frames are processed, their values are incorporated into the signal. Consequently, the signal portrays the fluctuations in pixel values between consecutive frames. Figure 4 represents the raw signal extraction algorithm.

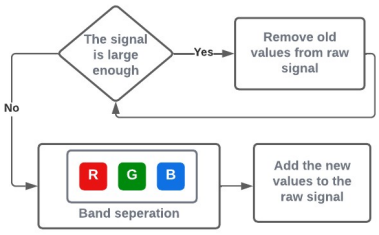

Figure 4: Raw signal extraction algorithm [19]

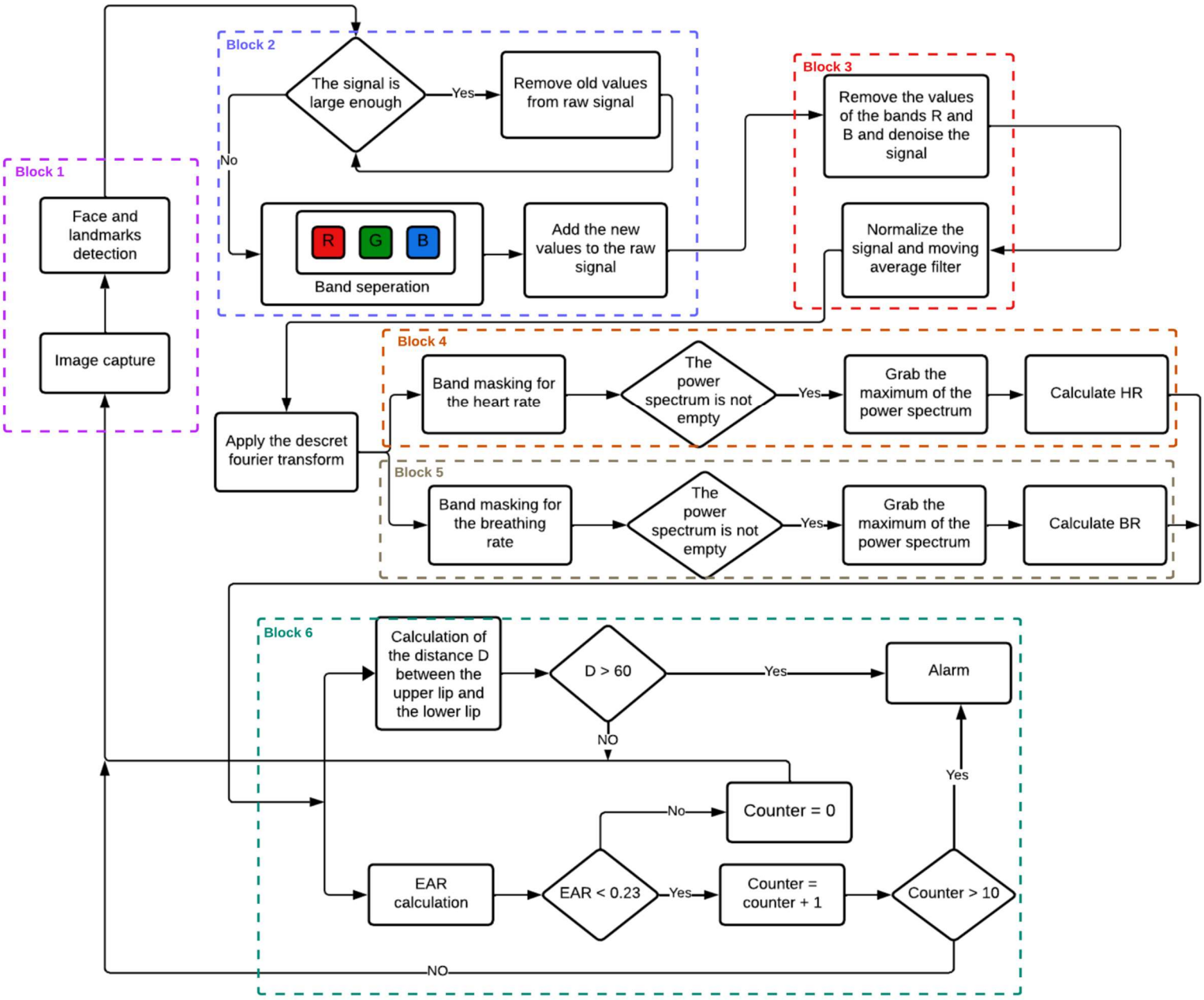

Figure 2: Global algorithm

The image is separated into three bands, red, green and blue. Their pixels' values are added to the signal. If the signal is large enough, the old values are removed. Otherwise, new values are added for each frame.

3.2.2 Signal filtering

Various techniques are available for deriving the filtered signal. Following the computation of channel averages, only the green channel remains. According to a study by W. Verkruysse et al. in 2008, it was elucidated that the green channel harbors the most relevant information pertaining to a PPG signal [29]. This predominance is primarily attributed to hemoglobin's superior absorption of green light compared to red and blue light. Figure 5 represents the signal filtering algorithm.

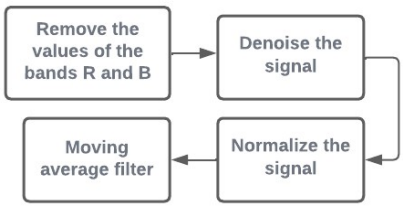

Figure 5: Signal filtering algorithm

Only the green channel remains by removing the R and B bands. After that the signal is denoised, normalized and a moving average filter is applied.

3.2.3 Heart rate and breathing rate estimation

The heart rate (HR) and the breathing rate are assessed through a frequency analysis, and this particular block has remained unchanged from the original algorithms. It is executed solely when the signal possesses sufficient magnitude. In such instances, the discrete Fourier transform (DFT) is employed to calculate the heart rate and the breathing rate. Figure 6 represents the heart rate and breathing rate extraction algorithm.

After applying the discrete Fourier transform to move to frequency domain, the Band masks are applied to the signal in order to isolate frequencies within specific ranges: 0.7 Hz to 3 Hz for heart rate and 0.1 Hz to 0.5 Hz for breathing rate [29]. Then if the power spectrum is not empty, the maximum is extracted to calculate both the heart rate and the breathing rate.

The heart rate is calculated with equation 4 [19].

$$BPM = (MaxFr*fps*60)/Size \quad (4)$$

Where:

- MaxFr is the maximum frequency
- fps is the number of frames per second
- Size is the size of the signal

The breathing rate is calculated with equation 5 [44].

$$BPM = Max*60 \quad (5)$$

Where Max is the maximum frequency

3.3 Fatigue monitoring

In order to assess if the eyes are closed or open, the Eye Aspect Ratio (EAR) is calculated. The vertical distance between the upper landmarks and the lower landmarks is employed to compute the Eye Aspect Ratio (EAR), with a set threshold value of 0.23. This threshold value was set after many trials by comparing it with other values.

Equation 6 is utilized to calculate the EAR. If the EAR value surpasses the established threshold, it indicates that the eye is open. Conversely, if the eye remains closed for more than 10 consecutive frames, it is considered closed.

$$EAR = \frac{||P2-P6|| + ||P3-P5||}{2||P1-P4||} \quad (6)$$

Where P1, P2, P3, P4, P5, P6 represent the positions of the eye's landmarks, as depicted in the accompanying figure 7.

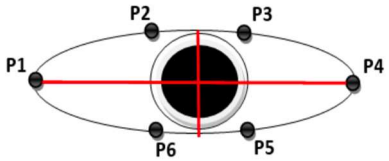

Figure 7: The positions of the eye's landmarks [45]

The average EAR value for both the left and right eyes is computed using equation 7. If the calculated EAR value falls below a certain threshold, it indicates that the eye is considered closed.

$$EAR = \frac{EAR_L + EAR_R}{2} \quad (7)$$

Where:

- EAR_L is the eye aspect ratio of the left eye
- EAR_R is the eye aspect ratio of the right eye

To ascertain whether a person is yawning, indicating their mouth is open, we calculate the distance between the upper lip and the lower lip. If this measurement exceeds 60, it indicates that the person is in a yawning state.

The algorithm iterates over each detected face in the input image or video frame. For each face, it extracts facial landmarks using the Dlib library. These landmarks are used to calculate specific measurements. The Eye Aspect Ratio (EAR) is calculated separately for the left and right eyes using the formula provided, which involves the distances between certain facial landmarks. EAR is then averaged over both eyes. If the calculated EAR falls below a certain threshold (0.23), it indicates that the eyes are closed or partially closed. A counter is incremented to track consecutive frames where eyes remain closed. If the counter reaches a threshold (10 frames), it signifies prolonged eye closure. If the total counter reaches 10, indicating prolonged eye closure, the algorithm sets a flag (Total) to 1, indicating that the eyes are closed. Otherwise, it resets the counter. The distance (D) between specific facial landmarks associated with the mouth is calculated. If this distance exceeds a threshold (60), it suggests yawning. Depending on the calculated

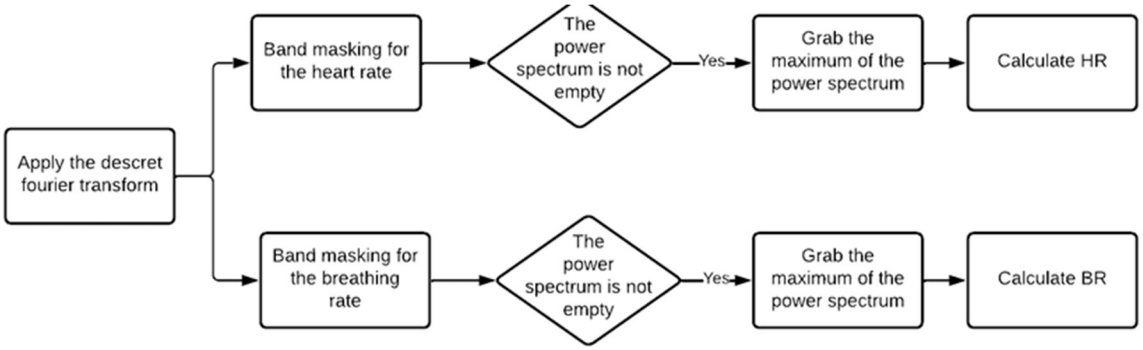

Figure 6: Heart rate and breathing rate estimation

values and thresholds, the algorithm classifies whether the eyes are open or closed and whether yawning is detected. This information can be used for further actions, such as alerting the driver in a drowsiness detection system.

Figure 8 illustrates the flow chart for the algorithm designed to detect the state of the eyes and yawning. The algorithm utilizes an EAR threshold of 0.23 and a threshold for the distance between the upper lip and the lower lip set at 60. These specific values were chosen following a comparison of various alternatives.

4. Results and discussion

4.1 Experimental results

Different architectures were used in this work. It was carried out on a Desktop system equipped with an Intel i7-1165G7 processor. Additionally, the implementation was extended to three distinct embedded architectures: the Odroid XU4, featuring an Exynos 5422 processor with an ARM A15 CPU running at 2 GHz and an ARM A7 CPU operating at 1.4 GHz and the Odroid H2. Lastly, the NVIDIA Jetson Nano, featuring an ARM A57 CPU clocked at 1.43 GHz and a Maxwell-based GPU, was also employed for this study. Table 2 represents the systems' specifications.

Table 2: Specification of the systems used

Type	Desktop	Odroid XU4	Jetson Nano	Odroid H2
Processor	11th Gen Intel Core™	Exynos 5422	Tegra SoC	Intel
CPU	Intel i7	ARM Cortex A15/A7	ARM A57	Quad-core processor J4115
GPU	NVIDIA GeForce MX330	Advanced Mali	Nvidia Maxwell	Intel UHD Graphics
Frequency	2.8 GHz	2GHz/1.4GHz	1.43 GHz	2.5 GHz
Weight	1,78kg	60g	136g	285g
Energy	90W	5W	10W	60W

The suggested algorithm underwent a time-based evaluation, and its processing duration was compared across all architectural platforms. The implementation language for this research is C/C++, utilizing the OpenCV library. The temporal assessment involved a sequence of 64 images. The algorithm relies on the diagrams shown before for heart rate estimation, breathing rate estimation, yawning detection, and eye closure detection. The algorithm makes use of the Dlib libraries for both face detection and facial landmarks detection. The outcome of the algorithm is depicted in the figure below Figure 9, which displays the interface when a face is detected. It also presents the values of EAR (Eye Aspect Ratio) and the distance between the upper lip and lower lip. Furthermore, it indicates the state of the eyes and whether the person is yawning. The camera must be positioned at a distance of one meter from the person concerned with a certain amount of lighting. Sudden motion may affect the accuracy of the measurements. Figure 9 and 10 represents the interface of the system proposed.

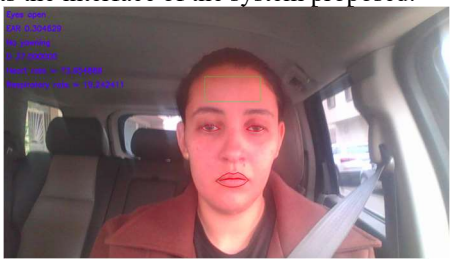

Figure 9: Interface of the system proposed when eyes are open

Journal of Theoretical and Applied Information Technology
30th September 2024. Vol.102. No. 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

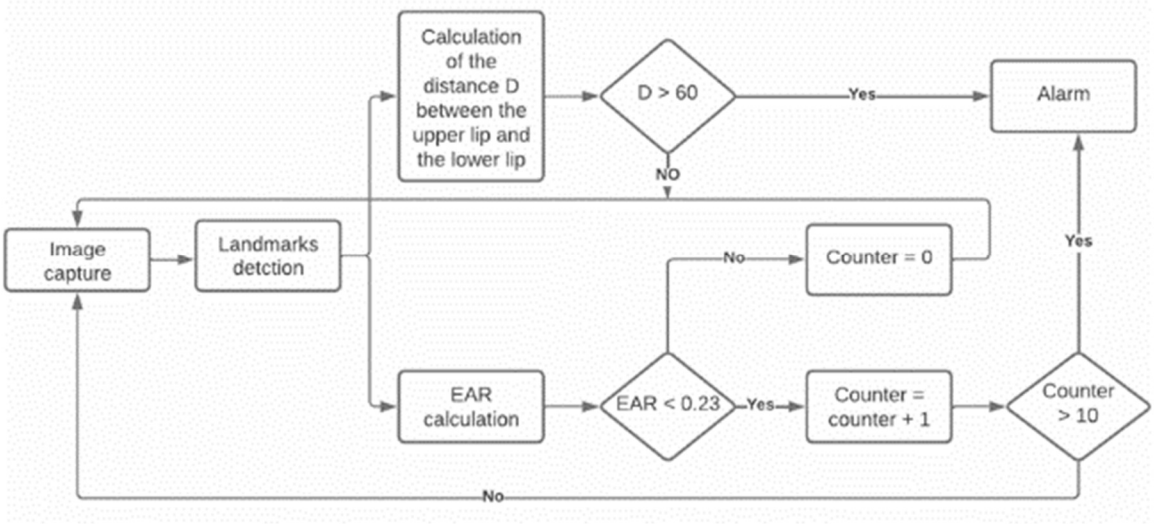

Figure 8: Flow chart of the eyes' state and yawning detection algorithm

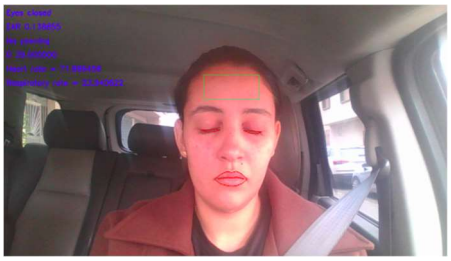

Figure 10: Interface of the system proposed when eyes are open

We can observe an EAR value of 0.3 when the eyes are open and a value of 0.13 when the eyes are closed.

4.2 Processing time

Figures 11, 12, 13 and 14 show the results of the global processing time on the Desktop, the Odroid XU4, the Jetson Nano and the Odroid H2, respectively

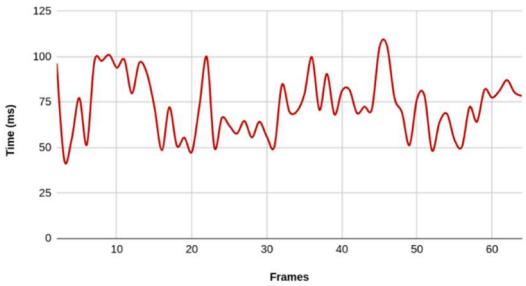

Figure 11: Processing time based on the Desktop

Figure 11 displays the results obtained from evaluating our algorithm on a Desktop system. The evaluation indicated a maximum processing time of 115 ms, a minimum of 45 ms, and an average of 80 ms, suggesting a frame rate of approximately 12 frames/s.

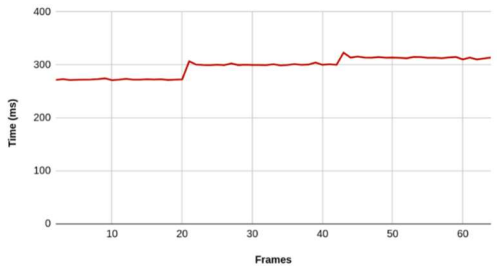

Figure 12: Processing time based on the Odroid XU4

Figure 12 presents the outcomes of assessing our algorithm on the Odroid XU4. The assessment demonstrated a peak processing time of 340 ms, a minimum of 280 ms, and an average of 314 ms, indicating a frame rate of around 3 frames/s.

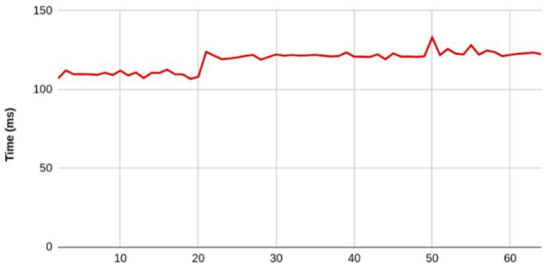

Figure 13: Processing time based on the Jetson Nano

Figure 13 showcases the results derived from the evaluation of our algorithm on the Jetson Nano. The assessment unveiled a peak processing time of 210 ms, a minimum of 201 ms, and an average of 204 ms, implying a frame rate of about 5 frames/s.

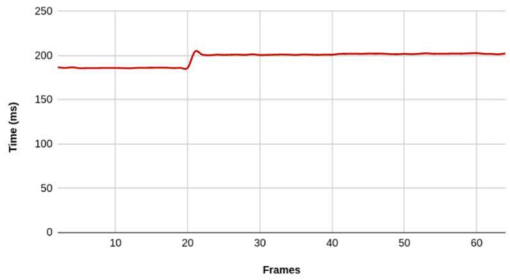

Figure 14: Processing time based on the Odroid H2

Figure 14 displays the outcomes obtained from assessing our algorithm on the Odroid H2. The evaluation revealed a maximum processing time of 133 ms, a minimum of 106 ms, and an average of 119 ms, indicating an approximate frame rate of 8 frames/s.

The variations in processing time observed with desktop systems compared to the consistency seen with embedded platforms such as the Jetson Nano, Odroid H2, and Odroid XU4 can be attributed to differences in hardware architecture, software optimization, operating environment, and resource management strategies. Embedded platforms offer more predictable and consistent performance due to their specialized design for edge computing and real-time processing applications.

The Critical Peak Power (CPP) serves as a metric for evaluating the performance of implementations across various processing units. To compute this index, one must utilize the frequency (F) in hertz, the execution time (T), and the image resolution (R) [46]. The computed CPP yielded results of 729 for the Desktop, 2044 for the Odroid XU4, 950 for the Jetson Nano and 968 for the Odroid H2. It was calculated with equation 8.

$$CPP = \frac{F*T}{R} \quad (8)$$

Table 3 summarizes the results of the computed CPP.

Table 3: Results of the CPP

Type	CPP
Desktop	729
Odroid XU4	2044
Jetson Nano	950
Odroid H2	968

The algorithm was segmented into six distinct blocks. The initial block handles face detection through dlib, the second block deals with raw signal extraction, the third block is responsible for signal filtering, the fourth block is dedicated to estimating heart rate, the fifth block is utilized for respiratory rate estimation and the sixth block focuses on fatigue detection. Figure 15 represents the processing time of block 1, since it is the most consuming block with a processing time of 67.36 ms for the Desktop, 272.22 ms for the Odroid XU4, 186,24 ms for the Jetson Nano and 109,58 for the Odroid H2.

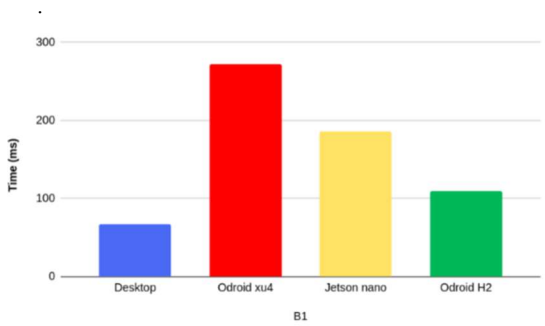

Figure 15: Comparison of the processing time of block 1

Figure 15 shows the processing time of block 1. The Odroid XU4 consumes the most compared with the Desktop, the Jetson nano and the Odroid H2, the Desktop consuming the least. Figure 16 represents the processing time of the rest of the blocks on the Desktop, the Odroid XU4, the Jeston

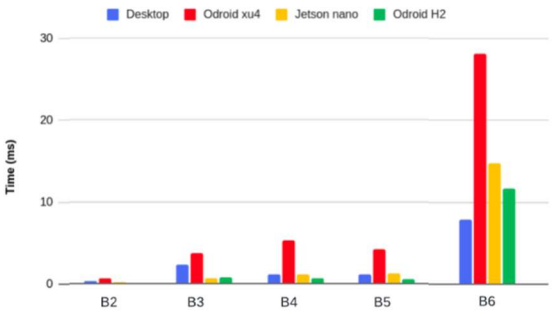

Figure 16: comparison of the processing time of block 2 to block 6

Nano and the Odroid H2. Figure 15 represents the processing time of block 2 to block 6.

In the last block involving fatigue detection, the Odroid XU4 requires more time than the other three architectures. However, the subsequent four blocks each take less than 5 ms for all four architectures.

Table 4 summarizes the time evaluation. For each block, the average time was calculated.

Table 4: Processing time of each block

Type	Desktop	Odroid XU4	Jetson Nano	Odroid H2
B1	67.3 ms	272.22 ms	186.2 ms	109.58 ms
B2	0.3 ms	0.7 ms	0.18 ms	0.085 ms
B3	2.35 ms	3.7 ms	0.62 ms	0.77 ms
B4	1.14 ms	5.32 ms	1.08 ms	0.62 ms
B5	1.1 ms	4.19 ms	1.2 ms	0.54 ms
B6	7.9 ms	28.12 ms	14.76 ms	11.64 ms
Total	80.15 ms	314.25 ms	204.09 ms	123.24 ms
CPP	729	2044	950	968

The Desktop system has a shorter processing time. Nevertheless, it is important to note that the significant issue lies in its high power consumption when contrasted with the cost-effective and low-power Odroid XU4 embedded architecture.

Figure 17 represents the comparison between the average value, the maximum and the minimum of the processing time of both the desktop and the Odroid XU4.

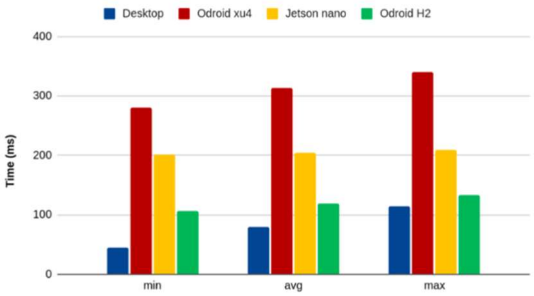

Figure 17: Comparison of the processing time of the Jetson Nano, the Odroid XU4, the Odroid H2 and the Desktop

In figure 17, the Desktop consumes the least with an average of 80 ms, a maximum of 115 ms and a minimum of 45 ms. The Odroid H2 consumes an average of 119 ms, a maximum of 133 ms and a minimum of 106 ms. On the other hand, the Jetson Nano consumes an average of 204 ms, a maximum of 210 ms and a minimum of 201 ms. Finally, the Odroid XU4 consumes the most with an average of 314 ms, a maximum of 340 ms and a minimum of 280 ms

4.3 Comparison of four different face detection algorithm

We decided to implement different face detection algorithms in order to compare their processing time as it is the block that consumes the most. We used previously the dlib Histogram of gradients (Hog) that works on the CPU. In this section, we introduce the dlib Maximum-Margin Object Detector (MMOD) with CNN-based features method, the opencv haar cascade and the opencv DNN. Figure 18 gives a comparison between the processing time of each of these face detection methods on the CPU

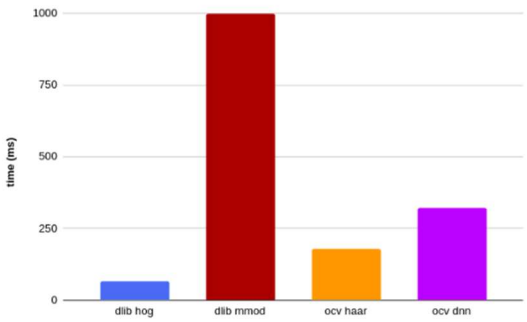

Figure 18: Comparison of the processing time of different face detection algorithms on the desktop

Dlib hog that we used previously is the fastest on the CPU, followed by the haar cascade, the opencv DNN and the dlib mmod. However, opencv DNN and dlib mmod that are slower when implemented on the CPU and can be accelerated thanks to the use of CUDA and the GPU. They both are supported by CUDA. Hence, we are going to implement them on the Jetson Nano to take advantage of its GPU. Figure 19 represents the comparison of the processing time of dlib mmod on the CPU and the GPU on the Jetson Nano.

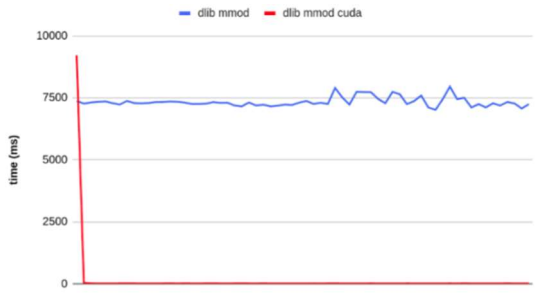

Figure 19: Comparison of the processing time of dlib mmod on the CPU and the GPU on the Jetson Nano

Dlib mmod on the CPU takes approximately 7500 ms. On the other hand, its take 9s for the first frame and 17 ms on average for the rest of the frames on the GPU. At the onset, the GPU-accelerated MMOD model undergoes initialization steps. These tasks encompass loading the model into GPU memory, configuring CUDA kernels, and executing other GPU-specific initialization operations. Given that these actions are typically executed only once at the outset of the application, they induce a delay for processing the initial frame.

Figure 20 represents the comparison of the processing time of opencv DNN on the CPU and the GPU on the Jetson Nano

Figure 20: Comparison of the processing time of opencv DNN on the CPU and the GPU on the Jetson Nano

Opencv DNN on the CPU takes approximately 600 ms. On the other hand, it takes 5s for the first frame and 30 ms on average for the rest of the frames on the GPU due the setup procedures of CUDA.

Dlib mmod is the fastest on the GPU when ignoring the time consumed on the first frame.

4.4 Discussion

Continuous monitoring of vital signs and fatigue indicators is essential to guarantee the safety of both the driver and other individuals. The algorithm incorporates facial detection and drowsiness extraction through the dlib library, as well as the monitoring of fatigue signs, heart rate, and respiratory rate estimation. This implementation extends across various platforms, including the Desktop, Odroid XU4, and Jetson Nano. The computed CPP (Computations Per Pixel) values reveal notable differences across the tested architectures: 729 for the Desktop, 2044 for the Odroid XU4, 87 for the Jetson Nano, and 968 for the Odroid H2. These metrics signify the computational efficiency of each platform in processing image data, with higher values indicating a greater capacity for parallel computation.

Furthermore, the findings demonstrate the achievable frame rates, which provide insights into the real-time processing capabilities of each architecture: 12 frames/s on the Desktop, 3 frames/s on the Odroid XU4, 8 frames/s on the Odroid H2, and 28 frames/s on the Jetson Nano. These frame rates reflect the speed at which each platform can process incoming frames, influencing the responsiveness and smoothness of applications utilizing real-time image processing.

Upon analysis, the Jetson Nano emerges as the optimal choice in terms of performance when compared to the other architectures. Despite its compact form factor, weighing 136g and consuming only 10W of power, the Jetson Nano delivers significantly higher frame rates and competitive CPP values. This highlights its efficiency and suitability for applications requiring rapid image processing in resource-constrained environments.

In summary, the comprehensive evaluation of CPP values and frame rates underscores the Jetson Nano's prowess as a high-performance yet energy-efficient platform for image processing tasks, positioning it as a compelling choice.

Table 5 outlines the time assessment, presenting the average time calculated for each block in comparison when employing dlib mmod accelerated with CUDA on the Jetson Nano.

Table 5: Processing time of each block

Type	Desktop	Odroid XU4	Jetson Nano after acceleration	Odroid H2
B1	67.3 ms	272.22 ms	17.71 ms	109.58 ms
B2	0.3 ms	0.7 ms	0.18 ms	0.085 ms
B3	2.35 ms	3.7 ms	0.62 ms	0.77 ms
B4	1.14 ms	5.32 ms	1.08 ms	0.62 ms

B5	1.1 ms	4.19 ms	1.2 ms	0.54 ms
B6	7.9 ms	28.12 ms	14.76 ms	11.64 ms
Total	80.15 ms	314.25 ms	35.55 ms	123.24 ms
CPP	729	2044	87	968

The Jetson Nano is the best option as it allows the use of CUDA to accelerate the face detection algorithm with a global processing time of 35.55 ms which allows us a speed of 28 frames/s.

The research design in this study was selected based on the nature of the research objectives, namely the contactless estimation of physiological parameters such as heart rate, respiratory rate, and signs of fatigue, and their implementation on various hardware platforms. An experimental design was chosen as it allows for testing, validating, and comparing the performance of the algorithms developed for signal estimation across multiple devices (desktop, Jetson Nano, Odroid H2). This design choice is particularly suitable as it enables the evaluation of not only the accuracy of the obtained results but also the efficiency of the algorithms based on the technical specifications of each platform.

The design addresses the research questions by empirically testing the formulated hypotheses: the effectiveness of the contactless method for estimating physiological and behavioural characteristics and the capability of different platforms to handle these algorithms in terms of processing speed, accuracy, and energy consumption. By comparing experimental results across various hardware configurations, it becomes possible to determine the optimal solution based on specific constraints.

Thus, this design is appropriate for meeting the research objectives, as it combines a rigorous experimental approach with cross-platform comparisons, ensuring a comprehensive evaluation of the system's performance.

5. CONCLUSION

This paper presents a non-contact algorithm for monitoring heart rate, breathing rate, and signs of fatigue. It is designed to measure the driver's heart rate and breathing rate, as well as to detect the state of their eyes (open or closed) and whether the driver is yawning. The approach consists of detecting the facial landmarks of the face using the different face detection methods and detecting the region of interest that is the forehead. The system is able to detect if the eyes are open or closed for a period of time and if the driver is yawning. It is also able to extract a signal from the region of interest that provides information used to calculate the heart rate and the breathing rate. Our algorithm has undergone evaluation on various embedded architectures, including the Desktop, Odroid XU4, and Jetson Nano. The outcomes demonstrate the capability to attain a frame rate of 12 frames/s on the Desktop, 3 frames/s on the Odroid XU4, 28 frames/s on the Jetson Nano and 8 frames/s on the Odroid H2. Future works also involve the implementation of the algorithm into a prototype and conducting testing in a real driver's environment. Future perspectives include developing more sophisticated algorithms capable of handling variations in lighting during daylight or nighttime, skin tones, and individual physiological differences.

ACKNOWLEDGEMENTS:

We owe a debt of gratitude to the Ministry of National Education, Vocational Training, Higher Education and Scientific Research (MENF-PERSRS) and the National Center for Scientific and Technical Research of Morocco (CNRST) for their financial support (grant number: 27UIZ2022) and for the project Cov/2020/109.

REFERENCES:

[1] H. He et al., "A Real-time Driver Fatigue Detection Method Based on Two-Stage Convolutional Neural Network," IFAC-PapersOnLine, Jan. 01, 2020

[2] E. Rivelli, "Drowsy driving statistics and facts 2022," Bankrate, Dec. 2022,[Online]. Available: https://www.bankrate.com/insurance/car/drowsy-driving-statistics

[3] "Drowsy Driving — NHTSA," NHTSA. https://www.nhtsa.gov/risky-driving/drowsy-driving

[4] P. C. Dischinger, S. M. Ho, and J. A. Kufera, "Medical conditions and car crashes.," PubMed, vol. 44, pp. 335–46, Jan. 2000, [Online]. Available:https://pubmed.ncbi.nlm.nih.gov/11558092

[5] D. Hs, "National Motor Vehicle Crash Causation Survey," NHTSA, Jan. 2008,[Online]. Available:https://thenewspaper.com/rlc/docs/2008/us-crashcause.pdf

[6] Pulmonary ventilation disorders and driving," Fundaci´on MAPFRE, Mar. 10, 2021. https://www.fundacionmapfre.org/en/education-outreach/road-safety/mobility-safe-health/clinical-topics-and-safe-driving/respiratory-disease-driving/pulmonary-ventilation-disorders/

[7] News-Medical.net, "Photoplethysmography (PPG)," News-Medical.net, Feb. 27, 2019. https://www.news-medical.net/health/Photoplethysmography-(PPG).aspx

[8] Sandberg, D. and Wahde, M. (2008). Particle swarm optimization of feedforward neural networks for the detection of drowsy driving. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 788–793. IEEE

[9] M. Subramaniyam et al., "Recent developments on driver's health monitoring and comfort enhancement through IoT," IOP Conference Series, vol. 402, p. 012064, Sep. 2018, doi: 10.1088/1757-899x/402/1/012064.

[10] Hamdani, S. U., Fernando, A. (2015). The Application of a Piezo-Resistive Cardiorespiratory Sensor System in an Automobile Safety Belt. Sensors, 15(4), 7742–7753. https://doi.org/10.3390/s150407742

[11] Jiang, Y., Sanpeng, D., Sun, H., Qi, Y. (2019). Unconstrained Monitoring Method for Heartbeat Signals Measurement using Pressure Sensors Array. Sensors, 19(2), 368.

[12] K. J. Lee, C. Park, and B. Lee, "Tracking driver's heart rate by continuous-wave Doppler radar," Annu Int Conf IEEE Eng Med Biol Soc. 2016, Aug. 2016.

[13] Heuer, S., Chamadiya, B., Gharbi, A., Kunze, C., Wagner, M. (2010). Unobtrusive in-vehicle biosignal instrumentation for advanced driver assistance and active safety. IEEE-EMBS Conference on Biomedical Engineering and Sciences.

[14] Kado, S., Monno, Y., Moriwaki, K., Yoshizaki, K., Tanaka, M., Okutomi, M (2018). Remote Heart Rate Measurement from RGB-NIR Video Based on Spatial and Spectral Face Patch Selection. International Conference of the IEEE Engineering in Medicine and Biology Society.

[15] Maurya, L., Zwiggelaar, R., Chawla, D., Mahapatra, P. (2022). Non-contact respiratory rate monitoring using thermal and visible imaging: a pilot study on neonates. Journal of Clinical Monitoring and Computing.

[16] Yang, F., He, S., Sadanand, S., Yusuf, A., Bolic, M. (2022). Contactless Measurement of Vital Signs Using Thermal and RGB Cameras: A Study of COVID 19-Related Health Monitoring. Sensors, 22(2), 627.

[17] Z. Yang, M. Bocca, V. Jain, and P. Mohapatra, "Contactless Breathing Rate Monitoring in Vehicle Using UWB Radar," Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks, Nov. 2018.

[18] N. Maleˇsevi´c, V. L. Petrovi´c, M. Beli´c, C. Antfolk, V. R. Mihajlovi´c, and M. M. Jankovi´c, "Contactless Real-Time heartbeat detection via 24 GHz Continuous-Wave doppler radar using artificial neural networks," Sensors, vol. 20, no. 8, p. 2351, Apr. 2020.

[19] H. El Boussaki, R. Latif, and A. Saddik, "Video-based Heart Rate Estimation using Embedded Architectures," International Journal of Advanced Computer Science and Applications, 14(5), 2023.

[20] X. Chen, J. Cheng, R. Song, Y. Liu, R. K. Ward, and Z. J. Wang, "Video-Based Heart Rate Measurement: recent advances and future prospects," IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 10, pp. 3600–3615, Oct. 2019.

[21] J. Kranjec, S. Beguˇs, G. Gerˇsak, and J. Drnovˇsek, "Non-contact heart rate and heart rate variability measurements: A review," Biomedical Signal Processing and Control, vol. 13, pp. 102–112, Sep. 2014

[22] R. Avram et al., "Real-world heart rate norms in the Health eHeart study," Npj Digital Medicine, vol. 2, no. 1, Jun. 2019,

[23] A Pon Bharathi, Srinivasan, P., Sarika, A. S., Vinodha, D. V., Parthiban, K. (2023). Chronological golden search optimization-based deep learning for classification of heartbeat using ECG signal. Computer Methods in Biomechanics and Biomedical Engineering. Imaging and Visualization.

[24] Jenkal, W., Latif, R., Toumanari, A., Dliou, A., B'charri, O. E., Maoulainine, F. M. R. (2016). An efficient algorithm of ECG signal denoising using the adaptive dual threshold filter and the discrete wavelet transform. Biocybernetics and Biomedical Engineering, 36(3), 499–508.

[25] El B'charri, O., Latif, R., Elmansouri, K., Abenaou, A., Jenkal, W. (2017). ECG signal performance de-noising assessment based on threshold tuning of dual-tree wavelet transform. Biomedical Engineering Online.

[26] M.-Z. Poh, D. J. McDuff, and R. W. Picard, "Advancements in noncontact, multiparameter physiological measurements using a webcam," IEEE Trans. Biomed. Eng., vol. 58, no. 1, pp. 7–11, Jan. 2011.

[27] El Boussaki, H., Latif, R., Saddik, A. (2023b). A review on Video-Based Heart Rate, Respiratory

Rate and Blood Pressure Estimation. In Lecture notes in networks and systems (pp. 129–140).

[28] Bella, A., Latif, R., Saddik, A., Jamad, L. (2020). Review and Evaluation of Heart Rate Monitoring Based Vital Signs, A case Study: Covid-19 Pandemic. 2020 6th IEEE Congress on Information Science and Technology (CiSt).

[29] W. Verkruysse, L. O. Svaasand, J. S. Nelson, ”Remote plethysmographic imaging using ambient light,” Optics Express, 16(26), 21434, 2008.

[30] Sahin, S. M., Deng, Q., Castelo, J. L. V., Lee, D. (2021). Non-Contact Heart Rate Monitoring from Face Video Utilizing Color Intensity. Journal of Multimedia Information System, 8(1), 1–10

[31] El Khadiri, Z., Latif, R., Saddik, A. (2023). Remote heart rate measurement using plethysmographic wave analysis. In Lecture notes in networks and systems (pp.254–267)

[32] El Khadiri, Z., Latif, R., Saddik, A. (2023b). An efficient hybrid algorithm for non-contact physiological sign monitoring using plethysmography wave analysis. Computer Methods in Biomechanics and Biomedical Engineering. Imaging and Visualization, 1–17.

[33] C. Chourpiliadis, “Physiology, Respiratory Rate,” StatPearls - NCBI Bookshelf, Sep. 2022, [Online].Available:https://www.ncbi.nlm.nih.gov/books/

[34] C. B. Pereira, X. Yu, M. Czaplik, R. Rossaint, V. Bla˘zek, and S. Leonhardt,“Remote monitoring of breathing dynamics using infrared thermography,” Biomedical Optics Express, vol. 6, no. 11, p. 4378, Oct. 2015

[35] Y. Takahashi, Y. Gu, T. Nakada, R. Abe, and T. Nakaguchi, “Estimation of Respiratory Rate from Thermography Using Respiratory Likelihood Index,” Sensors, vol. 21, no. 13, p. 4406, Jun. 2021

[36] J. Kempfle and K. Van Laerhoven, “Respiration Rate Estimation with Depth Cameras,” Conference Paper, Sep. 2018

[37] Sravan, C., Onesim, K., Bhavana, V. S. S., Arthi, R., Srinadh, G.: Eye Fatigue Detection System. 2018 International Conference on System Modeling Advancement in Research Trends (SMART), pp. 245-247.

[38] El Boussaki, H., Latif, R., Saddik, A. (2023). A review on the driver’s fatigue detection methods. In Lecture notes in networks and systems (pp. 464–473).

[39] Zhu, T., Zhang, C., Wu, T., Ouyang, Z., Li, H., Na, X., Liang, J., Li, W. (2022). Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences. Applied Sciences, 12(4), 2224.

[40] Mehta, S., Dadhich, S., Gumber, S., Bhatt, A. J. (2019). Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio. Social Science Research Network.

[41] H. Nigam, “Review of Facial Recognition Techniques,” International Journal for Research in Applied Science and Engineering Technology, vol. 10, no. 1, pp. 1740–1743, Jan. 2022

[42] X.-H. Han and C. Li, “Face Recognition and Application of Film and Television Actors Based on Dlib,” 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, Oct. 2019.

[43] Anzalone, L. (2021, August 1). Training alternative DlibShape Predictor models using Python.Medium.https://medium.datadriveninvestor.com/training-alternative-dlib-shape-predictor-models-using-python-d1d8f8bd9f5c

[44] H. El Boussaki, R. Latif, and A. Saddik, ”Non-contact respiratory rate monitoring based on the principal component analysis,” International Journal of Advanced Computer Science and Applications, 14(9), 2023.

[45] Dewi, C., Chen, R., Chang, C., Wu, S., Jiang, X., Yu, H. (2022). Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety. Electronics, 11(19), 3183.

[46] Saddik, A., Latif, R., Elhoseny, M., Elouardi, A. (n.d.). Real-time evaluation of different indexes in precision agriculture using a heterogeneous embedded system. Sustainable Computing: Informatics and Systems, 30, 100506.