Benchmarking of Computer Vision Algorithms for Driver Monitoring on Automotive-grade Devices

Sebastiano Battiato
Dept. of Mathematics and Computer Science
University of Catania
Catania, Italy
battiato@dmi.unict.it

Sabrina Conoci
University of Messina
Messina, Italy
sconoci@unime.it

Roberto Leotta
Dept. of Mathematics and Computer Science
University of Catania
Catania, Italy
leotta.rob@gmail.com

Alessandro Ortis
Dept. of Mathematics and Computer Science
University of Catania
Catania, Italy
ortis@dmi.unict.it

Francesco Rundo
ADG Central R&D
STMicroelectronics
Catania, Italy
francesco.rundo@st.com

Francesca Trenta
Dept. of Mathematics and Computer Science
University of Catania
Catania, Italy
francesca.trenta@unict.it

***Abstract*—The continuing evolution of technologies in the automotive industry has led to the development of the so-called Advanced Driver Assistance Systems (ADAS). ADAS is the term used to describe vehicle-based intelligent safety systems designed to support the driver, with the aim to significantly improve his safety, and the driving safety in general. In terms of development, current ADAS technologies are based on control functions about the vehicle movements with respect to the objects and entities detected in the same environment (e.g., other vehicles, pedestrian, roads, etc.). However, there is an ever growing interest on the use of internal cameras to infer additional information regarding the driver status (e.g., weakness, level of attention). The purpose of such technologies is to provide accurate details about the environment in order to increase safety and smart driving. In the last few years, Computer Vision technology has achieved impressive results on several tasks related to recognition and detection of customized objects/entities on images and videos. However, automotive-grade devices' hardware resources are limited, with respect to the once usually required for the implementation of modern Computer Vision algorithms. In this work, we present a benchmarking evaluation of a standard Computer Vision algorithm for the driver behaviour monitoring through face detection and analysis, comparing the performances obtained on a common laptop with the same experiments on an existing commercial automotive-grade device based on the Accordo5 processor by STMicroelectronics.**

***Index Terms*—Advanced Driver Assistance Systems (ADAS), Computer Vision, STM Accordo5**

I. Introduction

In recent years, the automotive manufacturers focused on developing advanced assisting technologies to improve driving comfort and safety for the driver. This increasing interest has led to the development of the Advanced Driver Assistance Systems (ADAS). The term ADAS refers to automated control functions and technologies designed to detect and evaluate the environment, in order to alert the driver and take control of the vehicle. Specifically, one of the main ADAS functions consist of processing information about the environment outside of the vehicle, (e.g., road infrastructures, road traffic, etc.), in order to provide effective support while driving by using a wide range of sensors [1]. With the significant progress of ADAS technologies, there is a continuing demand to develop effective warning systems not only to detect the presence of obstacles from the exterior environment, but also to deliver early detection of a car driver's inadequate physiological condition. The main objective is to minimize the risk of road accidents. Indeed, recent studies established that a considerable percentage of crashes are related to the bad driver status [2]. As a consequence, the most recent solutions involve the use of advanced Computer Vision algorithms for the analysis of head pose estimation, eye blink detection alongside the analysis of physiological signals such as PhotoPletysmoGraphy (PPG) and Electrocardiography (ECG) [3]–[8]. In this context, Computer Vision technologies have a crucial role for the vehicle industry because they allow algorithms to automatically analyse driver's face, gaze, and emotions to detect inattentiveness, drowsiness, road rage, and other potential dangerous situations [9], [10]. Computer Vision has achieved high levels of precision and accuracy on interpreting important visual information contained in images and video data. However, the high accuracy comes at the price of large computational cost. As a result, dedicated hardware devices, from the application of specific processors, are needed to optimize complex workloads of the Computer Vision methods. In automotive industry, the current capabilities of automotive-grade hardware devices are limited. For this reason, the scientific community is working intensely to overcome the constraints in intelligent in-vehicle technologies. Our study describes the development, implementation and testing of a drowsiness detection system that tailors to the car environment. Specifically, the main objective is to compare the performances of a common laptop and an existing commercial automotive-grade device in order to provide a benchmarking evaluation of a Computer Vision algorithm for driver monitoring through eye blink detection. The hardware used to run the experiments

978-8-8872-3749-8 ©2020 AEIT

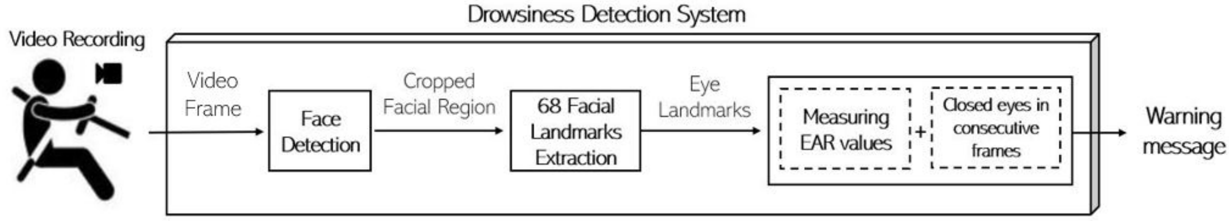

Fig. 1. Pipeline to detect car driver's drowsiness.

consists of a ST board based on Accordo 5 processor, developed by STMicroelectronics [11]. Although the results are not encouraging, since the board takes a long time to process the algorithms compared to a common laptop, we aim to open several paths for further studies to speeding up Computer Vision methods and implementing more advanced solutions on this automotive-grade technology. This paper is organized as follows. Previous related works exposing Computer Vision techniques that have been employed for automatic drowsiness detection are presented in Section II. The description of Computer Vision algorithms used for face and eye blink detection are presented in Section III. Section IV describes our experimental setup, providing details about the dataset and the hardware devices used to carry out our study. Section V shows the quantitative results of our benchmarking evaluation. The final considerations and a brief discussion on possible future research are presented in Section VI.

II. Related Works

Several studies have investigated the application of Computer Vision algorithms with the purpose of estimating driver's status. In [12], the authors propose a system that measures eye blinking rate and eye closure duration. The main contribution of this work is the skin-segmentation procedure. Specifically, the authors implemented a robust and adaptive skin segmentation method to detect facial regions (e.g., eyes, lips, nose, eyebrows, etc.) by extracting Speeded Up Robust Features (SURF) facial features [13]. After performing segmentation, eye blinking frequency and eye closure are estimated by monitoring the positions of irises through time. However, the calculation of Euclidean distance between SURF descriptors and the skin segmentation take a long time to process. Indeed, the work set the computation time as the critical point, which is not feasible for real-time applications. In this context, developing a robust, reliable, and efficient embedded system has become crucial in the automotive industry. The technological advances and expansion of vehicle technology has led to the development of more sophisticated embedded systems for the vehicle control. Embedded systems are commonly used in several applications including infotainment and telematics, safety and powertrain control. The recent electronic innovation has contributed to introduce the development of new features for different functionalities. Hence, the modern days embedded systems include microprocessors capable of efficiently running Computer Vision algorithms for several applications. However, there are many concerns related to automotive-grade hardware resources which present some limitations in terms of size, memory, power, cost, etc. Based on this, we listed some interesting solutions which balance all requirements and limitations in order to execute advanced algorithm for several ADAS applications. In [14], a novel approach for the real time detection of car driver drownsiness and alcoholic intoxication is proposed. The authors used an embedded system board Raspberry-pi with Raspbian-OS, including Python-IDLE and OpenCV library. The overall system is composed of an alcohol gas sensor application and an open source embedded system board Arduino Uno with I2C protocol for performing the alarm notification. To capture the car driver's face and eyes, they used an open source 5-megapixel digital camera to record a video while driving. Another example is [15], it uses a 5-megapixel Raspbian camera to capture driver's face and eyes. To detect the blink eye duration of the driver, the authors calculate Eye Aspect Ration (EAR) by using Haar Cascade Classifiers. The Raspberry Pi 3 with Raspbian Operating Systems is used to perform the entire algorithm for drowsiness detection. When the driver closing his eyes for more than a fixed threshold, a warning message is generated. This alert message is sent to the car owner along the car plate number through the use of a Ubidots cloud service and Twilio API.

III. Drowsiness Detection System

This section presents the pipeline developed to evaluate car driver's drowsiness. Our approach is based on assessing driver's fatigue level through facial analysis by extracting face component features (e.g., eyes, nose, mouth, etc.) from human face image. Feature extraction is a key component of most computer vision applications, from face recognition to age estimation [16], since providing a vast amount of information. In automotive field, facial analysis systems have been widely used to measure the eye closure duration in order to find hints of drowsiness. Indeed, previous research activities have demonstrated the existing correlation between vigilance and eye blink duration [17]. Inspired by the work of Soukupová and Čech [18], we computed the Eye Aspect Ratio (EAR) values, estimating the eye blinks, with the aim of assessing the car driver's drowsiness. Fig. 1 shows the overall pipeline. At the first stage, we collected video sequences of car driver's face emulating a drowsy scenario. To conduct our experiments, we used a common laptop and a development board based on Accordo5 processor in order to compare their performances in terms of computing time. After recording the sequences, we reduced the video resolution from 1280×720 to standard VGA size of 640×480 pixels. The amount of memory allocated, related to the board, did not allow us to process videos with higher resolution. For this reason, we converted videos to a lower resolution. In order to detect facial landmarks, we cropped facial region from video frames. Therefore, we load the OpenCV's Haar Cascade related to frontal face landmarks detection. Haar Cascade is a machine learning object detection algorithm based on the concept of features proposed by Viola and Jones [19]. Despite being an effective procedure to perform face detection in images and videos, Haar Cascade presents some drawbacks related to the demanding training time and the poor performances under difficult light conditions. We detected and extracted facial landmarks from frames of video sequence from dataset by using *Facemark*, the OpenCV's facial landmark API. Specifically, we used *FacemarkLBF* which represents implementation of the algorithm proposed by Ren et al. [20]. The approach described in [20] consists in a set of local binary features, and a locality principle for learning those features. The main novelty is related to learning discriminative local binary features to perform regression for the final output and elaborating facial landmarks independently. The main advantage of the proposed approach is the lower cost in terms of computation time since regressing local binary features is a very cheap operation. At this stage, we created an instance of *Facemark* class which is wrapped inside OpenCV pointer, optimizing the memory management. We load model related to landmarks detector which has been trained on a considerable number of training images and the corresponding annotations. We repeat this procedure for each video frames of recorded sequences. We also estimate computing time in order to measure the performance of both devices. Although the dlib library [1] is considered the "gold standard" to estimate facial landmarks in Computer Vision applications, it requires a high computational cost. For this reason, we adopted the *Facemark* API. Once detecting the 68-facial landmarks, we developed an effective drowsiness detection system by taking into account the eye landmarks. According to [18], Eye Aspect Ratio (EAR) value represents the distance used to determine if a person is blinking. Each eye is represented by 6 (x, y)-coordinates. Also, we choose the number of consecutive frames in which a driver closes his eyes. This is a threshold to not exceed. When the driver close his eyes and the EAR value is below a given threshold for a number of consecutive frames, a warning message appears to alert that the fatigue of car driver is high and could be dangerous. The EAR value is computed using the following equation:

$$EAR = \frac{\| p2 - p6 \| + \| p3 - p5 \|}{2 \| p1 - p4 \|} \tag{1}$$

where p1,...,p6 represent the 2D eye landmarks locations. The numerator indicates the Euclidean distance between vertical eye landmarks. The denominator represents the Euclidean distance between horizontal eye landmarks. Fig. 2 illustrates the eye landmarks and their distances.

[1]Dlib library: http://dlib.net/

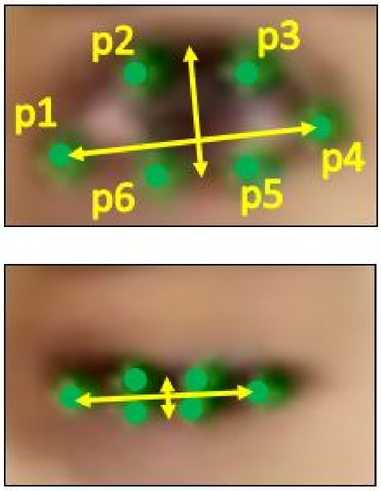

Fig. 2. Example of blink eye. The Eye Aspect Ration (EAR) is computed by using the distances of eye landmarks.

IV. Resources and Materials

A. Dataset

The dataset is composed of 12 videos. Specifically, this set contains video sequences of a face of people while driving. We acquired video sequences from drivers of different genders and ages. For the sake of completeness, we set up experiments also including drivers while wearing eyeglasses, in order to evaluate the effectiveness of the drowsiness detection algorithm considering different scenarios. The facial camera were at a distance of approximately 30 cm from the subject. In particular, it was placed in front of the driver in order to record the subject's frontal face while driving. Each video is 100 seconds in length. The resolution is set to $720p$ and frame rate is 30 fps. We emulate a drowsy scenario by performing eye blink closures while recording videos under a high light condition.

B. Hardware and devices

In this section, we briefly provide an overview of the used hardware, listing the main characteristics of each device. In order to evaluate the benchmarks, we used OpenCV (v. 3.4.3) installed on both systems, running C++ (std11) implementation of drowsiness detection algorithm to improve overall computational efficiency.

LAPTOP. Experiments were carried out on a laptop with an Intel Core i7 4710HQ CPU with 4 cores, 16GB of RAM and a N550JK motherboard, running Ubuntu 18.04 LTS.

ST BOARD. The other device involved in experiments is the development board STA1295 based on Accordo 5 processor produced by STMicroelectronics (software environment with embedded Linux) [11]. Specifically, the Accordo5 Evaluation Board (A5EVB) is a highly integrated "Car Radio," mounting

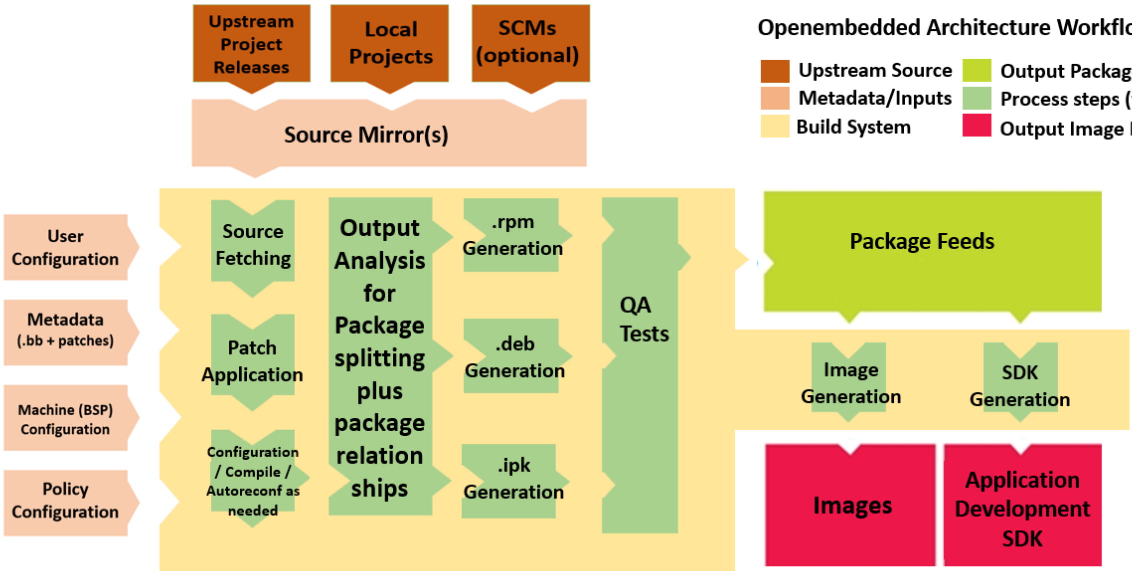

Fig. 3. YOCTO workflow.

the STA1295 version of the Accordo5 device, which is packaged in a 19x19mm LFBGA Package with 529 balls, pitch 0.8mm. The board is also composed of a 720p 10-inch Display Panel.

ACCORDO 5. The Accordo 5 multi-processor is a new line of digital-infotainment chips developed by STMicroelectronics [11]. The main advantage of the Accordo 5 chips is their integration of a dedicated, isolated ARM® Cortex®M3 core that secures the interface between the head unit and the main vehicle network. The microcontroller features built-in boot-code authentication, secure interconnect, and high-performance data encryption to manage secure CAN (Control Area Network) connectivity in real time. Furthermore, Accordo 5 devices provide several attractive features, such as smartphone mirroring (e.g., music and navigation services, etc.). In addition, the Accordo 5 family includes many state-of-art features, such as digital instrument clusters, consisting of complex and elegant displays which replace conventional dials and indicator lamps. Moreover, these chips integrate graphics, video and audio functionality alongside Audio/Video/Navigation (AVN) head units and dual screen capability in order to allow simultaneous user-interface plus rear-view camera with navigation and video previews, making driver safer and more comfortable.

YOCTO. The Operating System on STA1295 Accordo5 embedded automotive platform is ad-hoc YOCTO Linux-based distribution (v. 1.8) [21]. Specifically, YOCTO is a collection of tools and meta-data that allows a developer to build their own custom distribution of Linux for their embedded platform. The main benefit of Yocto consists in building custom Linux OS for about any kind of computing device, allowing to create Linux images for all the major hardware architectures. In addition, it makes the creation of custom Linux distribution faster, easier and cheaper, which is a key aspect when developing for embedded products. The main parts of the Yocto Project consist of the build system, the package meta-data, and the developer tools. Specifically, the build system uses a tool called *bitbake* to process the meta-data and produce a complete Linux distribution. It also builds the kernel, libraries, and programs that comprise a Linux distribution. In order to perform the deployment to the target device, it prepares the resulting software by placing it into appropriate bundles (including packages, images, or both). Moreover, Yocto provides application development and debugging to support developers. Fig. 3 overviews the Yocto workflow.

V. Results and Discussion

In this section, we present and analyze the obtained results. In order to perform the proposed benchmark evaluation, we executed the above described drowsiness detection algorithm 3 times, estimating the average time taken by both used devices to run it. Table I shows the computational time for each video of the dataset. The average time is highlighted in bold. As evident, the results revealed that the performance of the common PC is quite efficient. More specifically, we observed that the maximum running time required to compute drowsiness detection algorithms is $31,515$ seconds in relation to the video in which we performed the highest number of eye blink closures (Video 6), while the minimum running time is about 24 seconds. On the other hand, our experiments reported poor performances concerning the embedded STA1295 board. As expected, our tests show that the automotive designed board yields longer computational time than the common laptop. In general, to run a Computer Vision algorithm on our embedded system faced with several limitations both in terms of memory and power consumption, since it was not developed to execute demanding algorithms. In the process of analyzing performances of the used board, it turned out that the most

TABLE I
THE PROCESSING TIME OF A COMMON LAPTOP.

Run	Video elaboration (PC)											
	Video 1	**Video 2**	**Video 3**	**Video 4**	**Video 5**	**Video 6**	**Video 7**	**Video 8**	**Video 9**	**Video 10**	**Video 11**	**Video 12**
1	24.606 s	25.469 s	26.689 s	24.162 s	25.941 s	31.831 s	27.336 s	28.641 s	27.091 s	27.042 s	27.127 s	28.132 s
2	24.380 s	24.786 s	25.501 s	23.376 s	25.961 s	31.118 s	27.812 s	28.429 s	27.269 s	27.605 s	27.306 s	28.218 s
3	25.138 s	25.797 s	25.532 s	23.263 s	26.127 s	31.597 s	27.671 s	28.571 s	27.056 s	26.911 s	27.467 s	28.167 s
Avg Time	**24.708 s**	**25.351 s**	**25.907 s**	**23.600 s**	**26.010 s**	**31.515 s**	**27.606 s**	**28.547 s**	**27.139 s**	**27.186 s**	**27.300 s**	**28.172 s**

TABLE II
THE PROCESSING TIME OF STA1295 ACCORDO5 EMBEDDED AUTOMOTIVE PLATFORM.

Run	Video elaboration (STA1295 Board)											
	Video 1	**Video 2**	**Video 3**	**Video 4**	**Video 5**	**Video 6**	**Video 7**	**Video 8**	**Video 9**	**Video 10**	**Video 11**	**Video 12**
1	551.671 s	682.129 s	704.965 s	682.965 s	543.469 s	613.612 s	536.201 s	650.899 s	651.623 s	633.570 s	683.854 s	663.421 s
2	525.358 s	648.010 s	671.192 s	685.960 s	715.312 s	593.900 s	534.827 s	633.245 s	504.850 s	616.359 s	690.002 s	729.140 s
3	540.734 s	653.270 s	676.755 s	679.879 s	688.844 s	597.820 s	535.847 s	540.152 s	609.027 s	671.887 s	517.740 s	635.540 s
Avg Time	**539.254 s**	**661.136 s**	**684.304 s**	**682.935 s**	**649.208 s**	**601.777 s**	**535.625 s**	**608.099 s**	**588.500 s**	**640.605 s**	**630.532 s**	**676.034 s**

demanding part was related to I/O operations. To minimize the computational costs, we optimized I/O operations to increase the processing efficiency. In this respect, we performed the following steps:

- video decoding, computing CV algorithms, frame pushing on a buffer;
- closing reading stream
- frame popping from buffer and writing on file system

Despite attempts to increase the overall performance, the automotive-grade board takes long time to compute the implemented drowsiness detection algorithm. As seen in Table II, the results reported an average running time over 530 seconds for each video which is not acceptable for a real-time application. Indeed, the resulting performance is a consequence of multiple factors, spanning over OpenCV implementation and video properties (resolution, fps, etc.). Although it is certainly possible to cross-compile the OpenCV source code on embedded devices, memory constraints and other architectural considerations may pose a problem, complicating the task of optimizing CV functions on a new target device. Furthermore, video properties, such as resolution, fps, etc., will directly affect the efficiency of the overall system. As previously reported, we reduced the resolution of videos (Standard VGA resolution) since our automotive-grade device presents limited resources. To sum up, we used a ready-to-go development board taking advantage of its integrated features to execute a Computer Vision algorithm for evaluating drowsiness of a car driver. Despite its limitations, and consequently the poor results reported in Table II, our findings suggest that this device could be employed for automotive applications based on running CV algorithms. In fact, by developing a fierce trade-off with respect to memory usage, I/O bandwidth and algorithm complexity, we will provide a significant improvement to maximize the potential of Computer Vision functions on our embedded platform.

ACKNOWLEDGMENT

This research was funded by the National Funded Program 2014-2020 under grant agreement n. 1733, (ADAS + Project).

VI. CONCLUSIONS

In this paper, we proposed a benchmarking evaluation of Computer Vision algorithms, which allowed us to perform direct comparisons between the performances of an automotive designed computational board (STA1295 Accordo5 MCUs) with a common laptop, with the aim to assess the possibility to run Computer Vision algorithms on already existing hardware designed for automotive purposes. Although the results are not satisfactory, we observed that it is possible to run Computer Vision algorithms on our embedded development board. In this instance, we remain confident that, with adequate improvements, it is possible to implement Computer Vision algorithms on our automotive-grade device in order to define an efficient attention monitoring system. Finally, there are a few directions for further studies. Specifically, we aim to implement more advanced Computer Vision algorithms to provide a complete study regarding driver status analysis. Moreover, considering the remarkable success of Deep Learning architecture over the last years [22], our future works will focus on optimizing the use of Deep Learning approaches on our automotive board based on Accordo 5 processor, alongside reducing the computational cost for the development of a useful tool for the automotive industry.

REFERENCES

[1] S. Battiato, G. M. Farinella, G. Gallo, and O. Giudice, "On-board monitoring system for road traffic safety analysis," *Computers in Industry*, vol. 98, pp. 208–217, 2018.

[2] "Road accidents report," https://www.nsc.org/road-safety/safety-topics/fatality-estimates, accessed: 2020-08-06.

[3] F. Rundo, A. Ortis, S. Battiato, and S. Conoci, "Advanced bio-inspired system for noninvasive cuff-less blood pressure estimation from physiological signal analysis," *Computation*, vol. 6, no. 3, p. 46, 2018.

[4] F. Trenta, S. Conoci, F. Rundo, and S. Battiato, "Advanced motion-tracking system with multi-layers deep learning framework for innovative car-driver drowsiness monitoring," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019, pp. 1–5.

[5] F. Rundo, S. Conoci, A. Ortis, and S. Battiato, "An advanced bio-inspired photoplethysmography (ppg) and ecg pattern recognition system for medical assessment," *Sensors*, vol. 18, no. 2, p. 405, 2018.

[6] F. Rundo, S. Rinella, S. Massimino, M. Coco, G. Fallica, R. Parenti, S. Conoci, and V. Perciavalle, "An innovative deep learning algorithm for drowsiness detection from eeg signal," *Computation*, vol. 7, no. 1, p. 13, 2019.

[7] V. Vinciguerra, E. Ambra, L. Maddiona, M. Romeo, M. Mazzillo, F. Rundo, G. Fallica, F. di Pompeo, A. M. Chiarelli, F. Zappasodi *et al.*, "Ppg/ecg multisite combo system based on sipm technology," in *Convegno Nazionale Sensori*. Springer, 2018, pp. 353–360.

[8] M. Mazzillo, L. Maddiona, F. Rundo, A. Sciuto, S. Libertino, S. Lombardo, and G. Fallica, "Characterization of sipms with nir long-pass interferential and plastic filters," *IEEE Photonics Journal*, vol. 10, no. 3, pp. 1–12, 2018.

[9] G. Grasso, P. Perconti, and A. Plebe, "Assessing social driving behavior," in *International Conference on Intelligent Human Systems Integration*. Springer, 2019, pp. 111–115.

[10] G. M. Grasso, C. Lucifora, P. Perconti, and A. Plebe, "Evaluating mentalization during driving." in *VEHITS*, 2019, pp. 536–541.

[11] "Stmicroelectronics accordo 5 automotive microcontroller," https://www.st.com/en/automotive-infotainment-and-telematics/automotive-infotainment-socs.html?icmp=tt4379_gl_pron_nov2016, accessed: 2020-08-06.

[12] A. A. Lenskiy and J.-S. Lee, "Driver's eye blinking detection using novel color and texture segmentation algorithms," *International journal of control, automation and systems*, vol. 10, no. 2, pp. 317–327, 2012.

[13] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[14] D. Sarkar and A. Chowdhury, "A real time embedded system application for driver drowsiness and alcoholic intoxication detection," *International Journal Of Engineering Trends and Technology (IJETT)*, vol. 10, no. 9, pp. 461–465, 2014.

[15] A. Chellappa, M. S. Reddy, R. Ezhilarasie, S. K. Suguna, and A. Umamakeswari, "Fatigue detection using raspberry pi 3," *International Journal of Engineering & Technology*, vol. 7, no. 2.24, pp. 29–32, 2018.

[16] A. Torrisi, G. M. Farinella, G. Puglisi, and S. Battiato, "Selecting discriminative clbp patterns for age estimation," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–6.

[17] L. K. McIntire, R. A. McKinley, C. Goodyear, and J. P. McIntire, "Detection of vigilance performance using eye blinks," *Applied ergonomics*, vol. 45, no. 2, pp. 354–362, 2014.

[18] J. Cech and T. Soukupova, "Real-time eye blink detection using facial landmarks," *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague*, pp. 1–8, 2016.

[19] P. Viola, M. Jones *et al.*, "Robust real-time object detection," *International journal of computer vision*, vol. 4, no. 34-47, p. 4, 2001.

[20] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685–1692.

[21] "Yocto overview manual," https://www.yoctoproject.org/docs/3.1.2/overview-manual/overview-manual.html, accessed: 2020-05-15.

[22] A. Plebe and G. Grasso, "The unbearable shallow understanding of deep learning," *Minds and Machines*, vol. 29, no. 4, pp. 515–553, 2019.