2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A Real-Time Driver Identification System based on Artificial Neural Networks and Cepstral Analysis

Inés del Campo, Raúl Finker, Mª Victoria Martínez, Javier Echanobe, and Faiyaz Doctor

***Abstract*— The availability of advanced driver assistance systems (ADAS), for safety and well-being, is becoming increasingly important for avoiding traffic accidents caused by fatigue, stress, or distractions. For this reason, automatic identification of a driver from among a group of various drivers (i.e. real-time driver identification) is a key factor in the development of ADAS, mainly when the driver's comfort and security is also to be taken into account. The main focus of this work is the development of embedded electronic systems for in-vehicle deployment of driver identification models. We developed a hybrid model based on artificial neural networks (ANN), and cepstral feature extraction techniques, able to recognize the driving style of different drivers. Results obtained show that the system is able to perform real-time driver identification using non-intrusive driving behavior signals such as brake pedal signals and gas pedal signals. The identification of a driver from within groups with a reduced number of drivers yields promising identification rates (e.g. 3-driver group yield 84.6 %). However, real-time development of ADAS requires very fast electronic systems. To this end, an FPGA-based hardware coprocessor for acceleration of the neural classifier has been developed. The coprocessor core is able to compute the whole ANN in less than 4 μs.**

I. Introduction

Innovation in car safety over recent decades has undoubtedly contributed to a reduction in traffic accidents, even though the number of cars on the roads in the developed countries continues to rise. As a consequence of continuous technological advances, mainly in the areas of microelectronics and communications, new safety systems are being developed and incorporated into cars as standard equipment [1]-[3]. However, the main source of insecurity in a car is the driver himself, and many traffic accidents are wholly or partly caused by the driver. The availability of advanced driver assistance systems (ADAS), for safety and well-being, is becoming increasingly important in order to avoid traffic accidents caused by fatigue, stress, or distractions, especially since the driving population is getting older [4]-[5]. In this context, the ability to identify a driver and his/her driving behavior is the basis of many ADAS. In addition, being able to recognize the driver could be useful for security purposes (i.e. driver authentication) and comfort improvement in smart cars [6].

In the last decade there has been increasing research activity concerning driving behavior signals and their potential application in the development of ADAS [6]-[9]. These particular signals can be obtained in a non-intrusive manner, without disturbing the driver, as opposed video or audio signals which are the basis of some current ADAS. Driving behavior signals, mainly CAN bus signals, and sensor recordings (e.g. gas pedal pressure, brake pedal pressure, vehicle velocity, etc.) were used to develop models of drivers' behavior with the aim of identifying the driver and the driver's status under different cognitive conditions (e.g. distraction, and stress) [10]. The authors obtained satisfactory results by means of cepstral analysis and Gaussian mixture models (GMM). Cepstral feature extraction and cepstral filtering are well known techniques, commonly used in the digital processing of voice signals, and are suitable for efficient hardware implementation [11]. On the contrary, GMM are complex algorithms, with high computational demands [12]. These algorithms are unsuitable for in-vehicle embedded solutions with restrictive design specifications such as high performance, reduced size, and low power consumption.

To tackle the problem of the computational workload of statistical models such as GMMs, we investigated the suitability of artificial neural networks (ANN), combined with cepstral feature extraction techniques, for developing driver behavior models. The main aspects that support the proposal are the following:

1) Artificial neural networks have proven useful to model complex dynamic systems, in particular, human behavior in changing environments [13].
2) The learning capabilities of ANNs enable online adaptation of the models in demanding long-term applications.
3) The regular and parallel structure of typical ANNs is very suitable for developing high-speed hardware computation devices [14].

The automotive sector has ºrecently taken advantage of field programmable gate arrays (FPGA); this is mainly due to the high computational demands of this sector where a huge amount of signals have to be processed in real time by means of very fast electronic systems [15]-[17]. Currently, FPGAs

I. del Campo, R. Finker, M. V. Martínez, and J. Echanobe are with the Department of Electricity and Electronics, Faculty of Sciences and Technology, University of the Basque Country, Leioa, Vizcaya, 48940 Spain (e-mail: ines.delcampo@ehu.es).

F. Doctor is with the Faculty of Engineering and Computing, Coventry University, Priory Street, Coventry, CV1 5FB, UK (e-mail: faiyaz.doctor@coventry.ac.uk).

This work was supported in part by the Spanish Ministry of Science and Innovation and European FEDER funds under Grant TEC2010-15388 and by the Basque Country Government under Grants IT733-13, and S-PC12UN016.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

are used as single-chip embedded platforms (i.e. system-on-programmable chip: SoPC) or hardware coprocessors for algorithm acceleration, and as sensor interfaces (camera sensor interface, infrared or thermal camera interface, radar sensor interface, CAN bus interface, etc) [18], [19]. The main objective of this work is the development of embedded electronic systems for in-vehicle deployment of driver identification systems based on cepstral analysis and ANNs.

The paper is organized as follows: Section II presents the data base used in this work (i.e. Uyanik corpus), and the main characteristics of the selected driving behavior signals. In Section III the proposed model is presented, and representative simulation results are discussed. Section IV addresses the development of the driver identification systems, and provides details of their FPGA-based implementations. Finally, Section V presents some concluding remarks.

II. DRIVING BEHAVIOR SIGNALS

The aim of this work is to model individual differences in driving behavior of a group of drivers, and identify the driver in real-time by using the developed models. The main characteristics of the data collection are then introduced and the selection of signals, from the whole set of driving behavior signals, is justified.

A. Data Collection

The driving behavior data collection was supplied by the "Drive-Safe Consortium". It was collected in Istanbul with an instrumented car called *Uyanik*, which is a sedan car equipped with different sensors [7]-[8]. The complete data set (84 male and 17 female) includes audio and video recordings, CAN-bus signals, pedal-sensor recordings, 180º laser range finder, and XYZ accelerometer recordings.

The car route is around 25 km (about 40 minutes), and includes different kinds of sections: city, very busy city, highway, highway with less traffic, a university campus, etc. The route is the same for all drivers; however, the road conditions differ depending on traffic and weather. Approximately half of the driving sessions include driving while completing specific tasks with the aim of disturbing the attention of the drivers: signboard and plate reading, different types of dialogs on mobile phones, and conversations with passengers. To avoid additional noise sources, these driving periods were not considered.

B. Signal Selection

Firstly, the most suitable signals for performing driver identification in a non-intrusive manner were selected. The data collection was analyzed using data mining techniques with the aim of categorizing the data, finding similar characteristics across a large number of observations, and identifying potential useful signals. As a result of this task, also of some preliminary experiments, two signals were selected: gas pedal pressure (GP), and brake pedal pressure (BP). Both signals GP and BP are continuously sampled at 32 Hz.

Illustrative histograms of GP and BP signals, obtained from five randomly selected drivers, are shown in Fig. 1. As can be seen, each driver has his/her own driving style. The first driver on the top makes little use of the brake pedal. On the contrary, the fourth driver presses the brake pedal with much more strength (note that a different X-axis scale has been used for this driver in Fig. 1). The same consideration applies to the gas pedal. Moreover, the particular driving style of driver four is easier to identify than the other drivers.

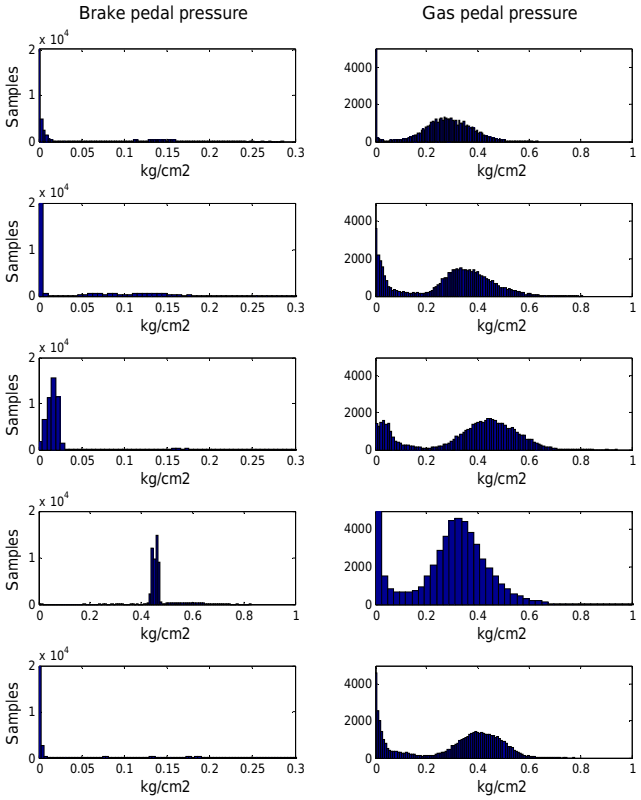

Fig. 1. Histograms of the selected driving behavior signals, sampled at 32 Hz over 30 minutes, for five randomly selected drivers. The histograms of brake pedal pressure are shown on the left side of the figure, while the histograms of gas pedal pressure are shown on the right side. It is worth noting that Y-axis scale is limited to $2x10^4$ and $5x10^3$ samples for BP and GP, respectively.

III. DRIVER IDENTIFICATION MODEL

The driver identification system proposed in this work is based on cepstral analysis and ANNs. Firstly, cepstral analysis is used to extract the most relevant features of the driving behavior signals, and then an ANN classifies the drivers according to their driving style. Let us briefly introduce both techniques.

A. Cepstral Analysis

Cepstral analysis is a nonlinear signal processing technique [11]. It was originally designed for characterizing the seismic echoes associated with earthquakes. However, at present, the most fruitful application area is concerned with the digital

processing of voice signals (e.g. speech recognition and speaker recognition). It has also been used to analyze radar signal returns, and to evaluate machinery vibration. Recently, encouraging results have been obtained by applying Cepstral feature extraction to driving behavior signals [10].

The real *cepstrum* for a long-time sequence $x(n)$ is defined as

$$c_x(n) = \mathcal{F}^{-1}\{\log|\mathcal{F}(x(n))|\}, \quad (1)$$

in which $\mathcal{F}$ denotes the discrete-time Fourier transform (DTFT), and $\mathcal{F}^{-1}$ denotes its inverse (IDTFT). Usually, the natural or base 10 logarithm is computed, but any base can be used.

An important property of the cepstrum is that the logarithm operation transforms the magnitude spectrum of a signal, where the components of the signal are generally not separable, to a linear combination (sum) of these components. The separation is done by taking the IDFT of the linearly combined logarithm spectra (e.g. separation of excitation and vocal tract system in speech signals, or separation of a low frequency signal from high-frequency noise). The IDFT of linear spectra transforms back to the time domain, but the IDFT of logarithm spectra transforms to the so-called *quefrency* domain or the *cepstral* domain.

In the case of practical signal processing applications, short terms or frames of the signals have to be used [20]. To select a desired frame of the original signal $x(n)$, the signal is multiplied by a finite length window $w(n)$. Commonly used window sequences are smooth bell-shaped functions, symmetric about the time $(T\text{-}1)/2$, where T is the duration of the frame (e.g. Hamming window). This kind of window is useful to reduce the edge effects caused by data segmentation [20].

B. Data preprocessing

The selected signals, gas pedal pressure (GP) and brake pedal pressure (BP), were sampled at 32 Hz and subdivided into frames of T=2s duration (64 samples). The overlapping between consecutive windows is of 60 samples. That is to say, a new frame begins every 125 ms (i.e. 4-sample frame shift). For each frame k the short term real cepstrum is evaluated, and K cepstral features f_k are extracted as follows

$$f_k = \mathcal{F}^{-1}\{\mathrm{BPF}\{\log_2|\mathcal{F}(x_w(n+kT))|\}\}, \quad (2)$$

where $x_w(n+kT)$ is the frame signal multiplied by the window function. The fast Fourier transform (FFT) was used to compute the DTFT and its inverse, and finite impulse response (FIR) filters were developed to perform high frequency noise filtering. The band-pass filter (BPF) separates noise from driving behavior signals. Two BPFs, with different cut-off frequencies, were implemented for BP and GP signal, respectively. As suggested in [10], 1-13 Hz cut-off frequencies were selected for BP signals, while 1-6.5 Hz frequencies were used for GP signals. Moreover, a base 2 logarithm was used to simplify further hardware implementation; this base is more suitable for efficient digital hardware implementation.

C. Neural Classifier

The kernel of the driver identification system is a multi-layer perceptron (MLP). Concerning the topology selection, a four-layer interconnected network (i.e. two hidden layers) has been devised (see Fig. 4). The size of the input layer is equal to the product of the number of driving behavior signals S, and the number of cepstral features K. The size of the hidden layers (i.e. number of hidden neurons) is a critical design parameter as it has a high impact on the modeling capability of the neural network. It is well known that too few hidden neurons result in poor performances, while an excess of hidden neurons could weaken the generalization capability of the network. Moreover, since our goal is to develop a single-chip hardware solution, too complex architectures should be avoided.

The best trade-off between complexity and performance was obtained with the same number of hidden neurons per layer as the inputs. Finally, the output layer has d neurons, where d is the number of drivers in the group. For example, the topology of a neural classifier based on two driving behavior signals (S=2), using 10 cepstral features (K=10) for a 3-driver group (d=3) is: 20-20-20-3. The driver of the group identified by the MLP is that associated with the neuron in the output layer which achieves the maximum activation.

D. Experimental results

The proposed neural classifier was tested using the Uyanik data set. A preprocessing stage based on cepstral feature extraction, like the one described in Subsection *B*, was included. The ANN was trained for groups of three, four, and five drivers, as these are typical number of drivers in real-life situations (e.g. family cars used by various drivers, or a fleet of vehicles with frequent driver reassignment). This task was accomplished by means of the back-propagation gradient descent method (GDM), while the mean squared error (MSE) was selected as the error function. 30% of available data was used in the learning phase for training and validation (i.e. approximately 8 minutes of data collection), while the remaining 70% were used for testing purpose (driver recognition). Three randomly selected sets of data were used in each case, and the mean ratio of success percentage was computed.

For the cepstral feature extraction stage, the number of features was set to K=10, as no additional improvement of the classifier was observed by increasing the number of features. The MSE was used to evaluate the training performance of the ANNs, and the percentage of successful driver identification was used to check the performance of the models. As can be seen in Table I, the fusion of GP and BP provides the best training performance.

TABLE I. TRAINING PERFORMANCE: AVERAGE MSE

Driving behavior signals	***3 drivers***	***4 drivers***	***5 drivers***
Gas pedal (GP)	0.162	0.144	0.105
Brake pedal signal (BP)	0.105	0.097	0.082
BP + GP signals	0.071	0.072	0.072

Figure 2 shows the average driver identification rates for the three groups of drivers using single signal models (S=1) for GP and BP, and 2-signal models (S=2), for the fusion of GP and BP. For the case of single signal models, BP is able to provide a 75% success rate among three drivers, while GP achieves only 61% for the same group. The fusion of both GP and BP signals provides the best result, an 84% success rate. The results obtained with the fusion of GP and BP are similar to those obtained in [10] by means of a more complex statistical model (i.e. Gaussian mixture model) using the same data set. However, in the case of single signals, GMM yields slightly better results, mainly for the gas pedal signal.

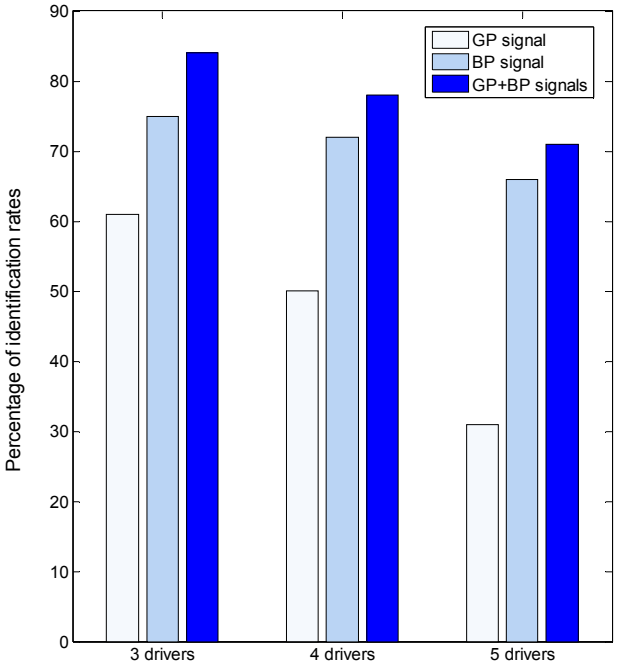

Fig. 2. Comparison of driver identification rates obtained with driving behavior signals (GP: gas pedal pressure, and BP: brake pedal pressure) using the developed model for a different number of drivers.

Several experiments have also been performed using Extreme Learning Machine (ELM) method [21]. The main advantage of this technique is that it provides very fast adaptation. The obtained accuracy was similar to that obtained using GDM. However, the size of the networks (i.e. the number of neurons) was considerably greater than that discussed above. For this reason, a traditional GDM adaptation was used in order to reduce digital hardware complexity.

IV. IMPLEMENTATION OF THE DRIVER IDENTIFICATION SYSTEM

An embedded system is a special-purpose computing platform designed to perform one or several dedicated functions. They are often designed for a particular kind of activity that is required to work under certain constraints, such as low power consumption, real-time operation, processing capacity, dependability, security, etc. In addition, low cost, and small size/weight are also typical requirements for these computing platforms.

In the progress towards a more autonomous and flexible lifestyle, with new levels of comfort, safety and productivity in all areas, many embedded platforms have emerged in the market and are in use in our daily activities. They can be found everywhere in a variety of application areas, from control systems in automotive sectors, to consumer and multimedia products, among others.

Field programmable gate arrays (FPGAs) have appeared as a suitable means for the development of embedded systems [22]. A milestone in the evolution of reconfigurable hardware has been to combine the logic blocks and interconnections of traditional FPGAs (logic fabric) with embedded microprocessors (e.g. standard PowerPC or ARM) and related peripherals to form a system-on-programmable chip (SoPC) or a multiprocessor SoPC (MSoPC). A similar approach consists in using soft-processor cores instead of hard-cores that are implemented within the FPGA logic such as for example Xilins's MicroBlaze [23].

The development of efficient SoPC-based embedded systems involves the use of hardware/software (HW/SW) co-design techniques. HW/SW co-design proposes the partition of the computation algorithms into HW and SW blocks by searching for the partition that optimizes the performance parameters of the whole system. This approach provides an optimal solution for many systems where a trade-off between versatility and performance is required, as for example, many applications in the ever-competitive automotive sector. In this context, the implementation of efficient real-time electronic systems for ADAS, using FPGA-based embedded systems for in-vehicle integration, is an issue of great interest.

The FPGA selected to implement the driver identification core is the XC7k325T device of Xilinx's KINTEX-7 family [24]. The device is one of the smallest of this family. It has 50950 Slices (each Slice contains four 6-input look-up table (LUTs), and eight flip-flops), 840 digital signal processing (DSP) blocks (each DSP consists of a multiplier, an adder, and an accumulator), and 445 RAM memory blocks of 36 Kbits each. This device family provides a scalable architecture for mid-range applications. It enables upward scalability to a Virtex-7 FPGA for greater performance, or downward scalability to an Artix-7 FPGA for further reductions in power consumption [24].

Hardware/Software partition

The driver identification system architecture was designed

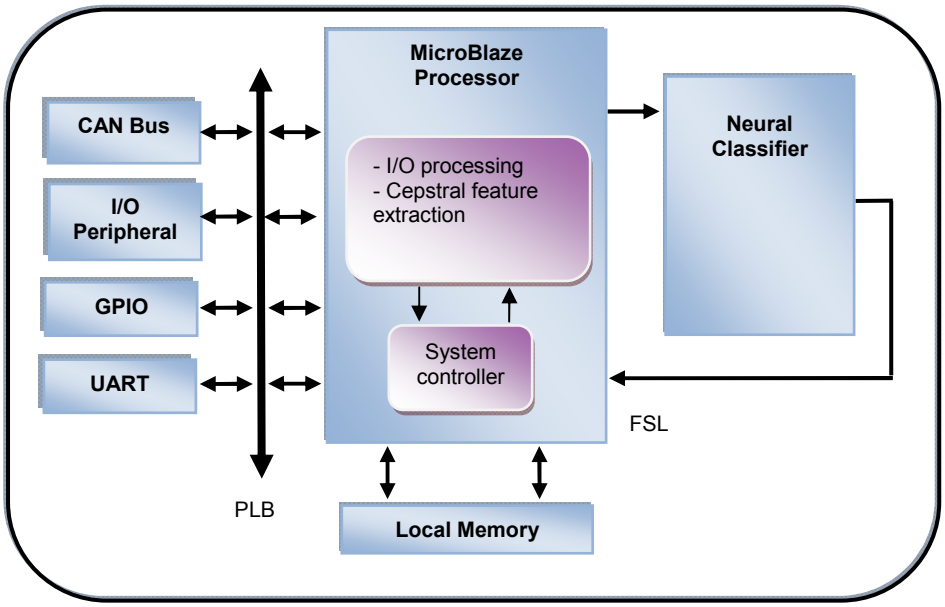

Fig. 3. Block diagram of the proposed hardware/software architecture for efficient implementation of the driver identification system. The software partition is developed on a MicroBlaze processor, while the neural classifier is implemented in the hardware partition with the aim of accelerate real-time data processing of the most time consuming task.

to enable automatic identification of a driver, among a group of various drivers, by recognizing his/her driving style. The system has been partitioned into three main modules: the input/output (I/O) management, the computation of the cepstral features (2), and the evaluation of the four-layer MLP (feed-forward network). In the proposed architecture (see Fig. 3) the first two modules are included in the software partition, while the latter is developed in the hardware partition. Although the sequential computation of the MLP is a time-consuming task, it exhibits a regular and highly parallelizable structure, so its location in the HW partition is appropriate.

On the other hand, the feature extraction preprocessing algorithm is not so critical because it is computed only for the selected signals (S=1 or S=2). However, in future works we intend to develop special purpose hardware in order to accelerate the cepstral feature extraction stage. As has been explained in Section III, cepstral analysis involves the computation of typical digital signal processing algorithms which can be efficiently implemented on FPGA devices.

A. Neural Classifier

The neural classifier, implemented in the hardware partition, computes an f-input d-output feed-forward network with two hidden layers (i.e. a four-layer MLP), being f=SxK. This coprocessor communicates with the microprocessor by means of a Fast Simplex Link (FSL) Bus (see Fig. 3). It is a VHDL module that can be sized in several dimensions by means of GENERIC parameters (i.e. word-length, number of inputs, number of outputs, and number of neurons).The coprocessor architecture exploits the high degree of parallelism inherent in neural networks. It is optimized for high-speed processing and is able to provide real-time response for advanced driving assistance systems.

Fig. 4 depicts a block diagram of the coprocessor core. The main modules of the core are the Input Layer, the Hidden Layers, the Output Layer, the RAM module where the neuron weights and biases are stored, and the Core Controller. The MLP was previously trained (off-line training), as has been explained in previous sections. In real-time operation mode, the Input Layer reads the inputs provided by the FSL bus and pushes them into the parallel data path. Then, the Hidden Layers perform the computation of all hidden neurons in parallel.

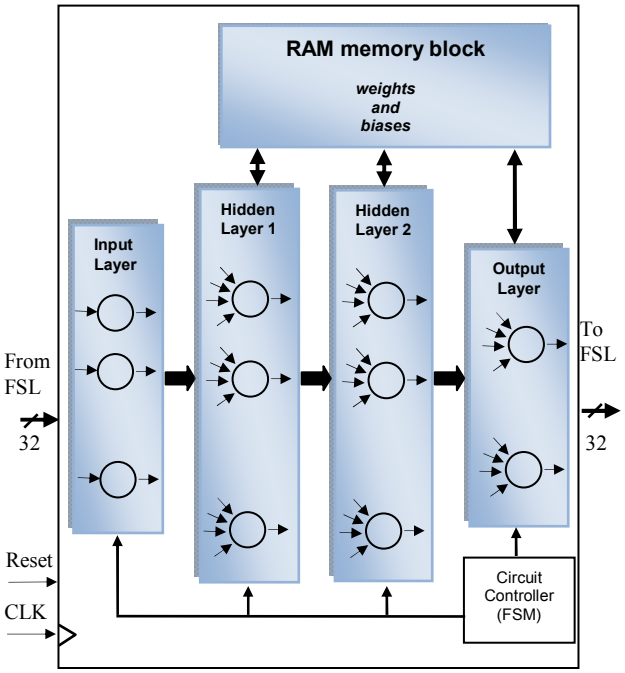

Fig. 4. Internal architecture of the neural classifier.

Finally, the Output Layer (which is similar to the Hidden Layers) includes as many neurons as possible drivers, as well as a multiplexer (MUX) that sequences the transfer of the core outputs to the MicroBlaze by means of the FSL bus. The Core Controller is a simple finite state machine (FSM) responsible for the data pipelining through the data path.

B. Hidden Layers

The Hidden Layers are organized into *h* parallel neurons. Each neuron in these layers computes the sum of as many products (SOPs) as neural network inputs (*f*) (see Fig. 5). Then, the SOPs are passed through a high-precision sigmoid filter. When the neuron is enabled, the adder accumulator is initialized with the neuron bias, followed by a burst of *h* products (i.e. inputs and weights) which are sequentially added. The computation of the SOPs lasts (*f*+1) clock cycles. After the computation of SOPs (i.e. the linear part of the neuron), the sigmoid filter is activated. The Output Layer is similar to the Hidden Layer. The number of output neurons is equal to the number of possible drivers, *d*, while the computation time is (*d*+1) clock cycles.

C. Sigmoid Module

As can be seen in Fig. 6, the input to the sigmoid Filter is the result of the computation of the linear part of the neuron (SOP). This module is based on a controlled accuracy approximation of the sigmoid function [25], [26]. It implements the sigmoid function with a maximum approximation error ε=6x10^{-4}, using a second order Taylor scheme.

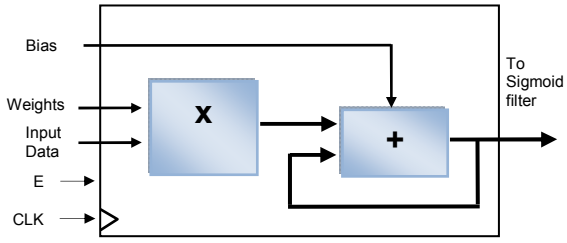

Fig. 5. Diagram of the linear part of a hidden/output neuron (SOP module).

The approximation scheme divides the input range into two kinds of regions, the so called saturation regions and the Taylor regions. The saturation region is that region where the first derivative of the sigmoid function is close to zero; the starting point of this region depends on the required precision. The Taylor regions, in turn, are split into a number of intervals *I* of width 2*r* containing *a*, $I=(a-r,a+r)$, where a local approximation of the sigmoid function *f(x)* is computed as follows

$$f(x)\approx f(a)+f'(a)(x-a)+\frac{f''(a)}{2!}(x-a)^2, \qquad (3)$$

The main computation unit of the sigmoid module is a typical DSP core. These embedded blocks, available in most current FPGA families, provide high-performance with low-power consumption. Two Read Only Memory (ROM) modules are used to store the Taylor coefficients in (3), ROM1 and ROM2. Both memories are addressed by means of the most significant bits of the SOP. The circuit performs the computation of the sigmoid approximation in only 5 clock cycles.

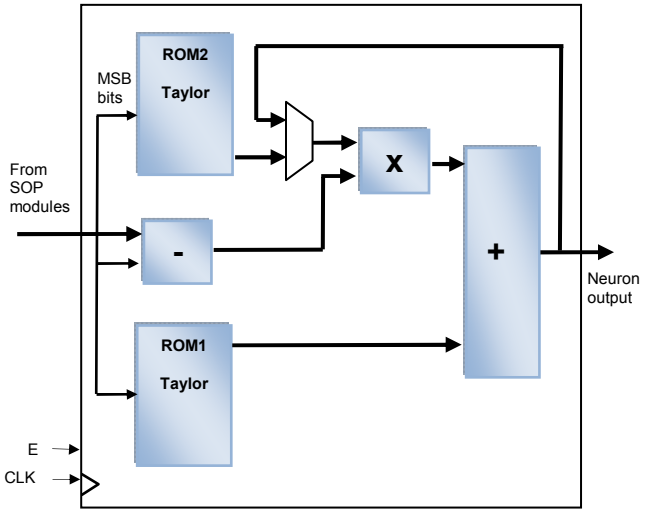

Fig. 6. Diagram of the Sigmoid circuit. It is based on a second order Taylor approximation of the function.

D. Timing Considerations and Resource Utilization

Table II presents post place and route timing results for different MLP topologies. As can be seen, a 10 feature classifier (i.e. only one driving behavior signal, BP or GP) is able to perform the network computation in less than 2 µs (e.g. a 3-driver classifier requires 1.4 µs, while a 5-driver classifier needs 1.64 µs). Concerning the topology that achieves the best recognition rates (i.e. two driving behavior signal, BP and GP), the 20 feature core requires only 3.70 µs to evaluate a 3-driver classifier, 4.07 µs to evaluate a 4-driver classifier, and a similar result, 3.94 µs to compute a 5-driver topology. This performance allows true real-time driver identification. In contrast, an embedded system based on a whole software implementation of the MLP would have increased the computation time by several magnitude orders.

As can be seen in Table II, the maximum computation frequency is greater than 100 MHz when only 10 cepstral features are used (i.e. one driving behavior signal), but slightly less than 100 MHz when a 20-feature topology is required. This performance could be improved by using distributed RAM memories instead of a single RAM module to store weights and biases of the whole ANN. Each neuron would have its own storage module with the aim of making the neural architecture more flexible, and reducing signal delays.

TABLE II. TIMING PERFORMANCE

Topology of the MLP (2 hidden layers)	*Achieved frequency (MHz)*	*Computation time (μs)*
10-10-10-3	126	1.40
20-20-20-3	91	3.70
10-10-10-4	121	1.50
20-20-20-4	84	4.07
10-10-10-5	114	1.64
20-20-20-5	88	3.94

Concerning resource utilization, Table III summarizes the implementation requirements of different MLP topologies. The most representative FPGA primitives have been considered (i.e. LUTs, registers or flip-flops, and DSP modules). As can be seen, the percentage of resource utilization, on average, is less than 7% of the total resources in the Xilinx KINTEX-7 device used in this work -it is the third in size of this family. Therefore, it can be concluded that the resource demands of the classifier core is small enough to allow full implementation of more complex topologies. In addition, the remaining resources, 93 % of the device, could be dedicated to adding new algorithms and strategies for real-time ADAS implementation. Alternatively, a smaller device of this family could be selected with the aim of reducing cost, size, and power consumption of the neural classifier.

TABLE III. RESOURCE UTILIZATION

Topology of the MLP (2 hidden layers)	*LUTs*	*Flip-flops*	*DSP Blocks*	*Mean resource utilization*
10-10-10-3	3896	6468	24	2.2 %
20-20-20-3	17623	20834	44	6.4 %
10-10-10-4	4045	6779	25	2.3 %
20-20-20-4	19558	21292	45	6.8 %
10-10-10-5	4373	7044	26	2.4 %
20-20-20-5	21555	21765	46	7.2 %

V. CONCLUSION

The availability of advanced driver assistance systems (ADAS), for safety and well-being, is becoming increasingly important in order to avoid traffic accidents caused by fatigue, stress, distractions or chronic diseases. This work contributes to the development of ADAS with a driver-centred perspective which aims at improving the driver's awareness and driving performance in a personalized way.

A new approach to the problem of real-time driver identification is presented. The proposed solution is based on artificial neural networks and cepstral analysis. Obtained results show that the model is able to recognize different driving styles using non-intrusive driving behavior signals (gas pedal signal and brake pedal signal). The driver is then identified through his/her driving style.

Real-time development of ADAS requires very fast electronic systems. To fulfill this requirement, an FPGA-based hardware coprocessor for acceleration of the neural classifier has been developed. The coprocessor core is able to compute the whole ANN in less than 4 μs. In addition, the resource demand is small enough to allow full implementation of more complex topologies.

In future works we are going to improve the identification performance of the neural classifier by adding new driving behavioral signals. To this aim, we will investigate the fusion of two additional CAN bus signals, the vehicle speed and the engine revolutions per minute. In addition, the performance of the FPGA-based system will be improved in order to enable on-line training. This new capability of the the system would allow the adaptation of the reference driving style models in the long term.

ACKNOWLEDGMENT

The authors would like to thank Dr Hüseyin Abut and the researchers of the Drive-Safe Consortium in Istanbul (Turkey) for providing the "Uyanik" data set used to perform the experimentation.

REFERENCES

[1] A. Schmidt, J. Paradiso, and B. Noble, "Automotive Pervasive Computing," Guest Editors' introduction, *PERVASIVE Computing*, pp.12-13, 2011.

[2] E Shang,J. Li, X. An, and H. He, "A Real-time Lane Departure Warning System Based on FPGA," in *Proc. 14th International IEEE Conference on Intelligent Transportation Systems*, Washington DC, USA, 2011, pp.1243-1248.

[3] Z. Stamenkovic, K. Tittelbach-Helmrich, J. Domke, C. Lörchner-Gerdaus, J. Anders, V. Sark, M. Eric, and N. Šira, "Rear View Camera System for Car Driving Assistance," *Proc. 28th International Conference on Microelectronics (MIEL 2012)*, NIŠ, Serbia, 2012, pp.383-386.

[4] J.F. Coughlin, B. Reimer, and B. Mehler, "Monitoring, Managing, and Motivating Driver Safety and Well-Being," *PERVASIVE Computing*, pp.14-21, 2011.

[5] A. Eskandarian, and A. Mortazabi, "Unobtrusive driver drowsiness detection system and method," US patent 2010/0109881 A1, May 6, 2010.

[6] J.H.L. Hansen, P. Boyraz, K. Takeda, and H. Abut, eds., "Digital Signal Processing for In-vehicle Systems and Safety," Springer, 2012.

[7] C. Miyajima, T. Kusakawa, T. Nishino, N. Kitaoka, K. Itou, K. Takeda, *"On-Going Data Collection of Driving Behavior Signals*," in *Corpus and Signal Processing for Driver Behavior*, K. Takeda, J. H.L. Hansen, H. Erdoğan, and H. Abut, (eds.) Springer Business-Science, 2008, ch4.

[8] P. Angkititrakul, J.H.L. Hansen, "UTDrive: The smart vehicle project," in *Corpus and Signal Processing for Driver Behavior*, K. Takeda, J. H.L. Hansen, H. Erdoğan, and H. Abut, (eds.) Springer Business-Science, 2008, ch5.

[9] H. Abut, H. Erdogan, A. Ercil, et al., "Data collection with "UYANIK: too much pain; but gains are coming," in *Corpus and Signal Processing for Driver Behavior*, K. Takeda, J. H.L. Hansen, H. Erdoğan, and H. Abut, (eds.) Springer Business-Science, 2008, ch3.

[10] E. Öztürk, and E. Erzin, "Driver Status Identification from Driving Behavior Signals," in *Digital Signal Processing for In-vehicle Systems and Safety* J.H.L. Hansen, P. Boyraz, K. Takeda, and H. Abut, eds, Springer, 2012, ch.3.

[11] A.V. Oppenheimer, and R.W. Schafer, "From Frequency to Quefrency: A History of the Cepstrum", *IEEE Signal Processing Magazine*, pp.95-106, 2006.
[12] G.J. McLachlan, and David Peel, *Finite Mixture Models*, Wiley Series in Probability and Statistics, 2000.
[13] I. del Campo, K. Basterretxea, J. Echanobe, G. Bosque, and F. Doctor, "A System-on-Chip Development of a Neuro-Fuzzy Embedded Agent for Ambient-Intelligence Environments," *IEEE Transactions on System, Man, and Cybernetics, Part B*, vol.42, pp.501-512, 2012.
[14] R. Finker, I. del Campo, J. Echanobe and F. Doctor, "Multilevel Adaptive Neural Network Architecture for Implementing Single-Chip Intelligent Agents on FPGAs," *Proc. Of IJCNN 2013 International Joint Conference on Neural Networks*, Dallas (USA), 2013.
[15] Altera Corp., "Automotive. Accelerate Innovation with Programmable Logic", Available: http://www.altera.com/end-markets/auto/aut-index.html.
[16] Xilinx Inc., "All Programmable Innovation in Automotive", Available: http://www.xilinx.com/applications/automotive/index.htm.
[17] Atmel Corp., "30 Years of Automotive Electronics Design Experience: Designing a Safer, Cleaner, More Reliable Vehicle", Available: http://www.atmel.com/products/automotive/default.aspx.
[18] Khan, S. Niar, M. Saghir, Y. El-Hillali, and A. Rivenq, "Driver Assistance System Design and its Optimization for FPGA Based MPSoC," *Proc. Of IEEE 7th Symposium on Application Specific Processors (SASP),* San Francisco, USA, 2009, pp.62-65.
[19] E Shang,J. Li, X. An, and H. He, "A Real-time Lane Departure Warning System Based on FPGA", *Proc. Of 14th International IEEE Conference on Intelligent Transportation Systems*, Washington DC, USA, 2011, pp.1243-1248.
[20] J.R. Deller, J.H.L. Hansen, and J.G. Proakis, "Discrete-Time Processing of Speech Signals," IEEE-Wiley-Interscience, USA, 1993.
[21] G.-B. Huang, et al., « Extreme Learning Machine : Theory and Applications," *Neurocomputing*, vol.70, pp.489-501, 2006.
[22] L. Hózwiak, N. Nedjah, and M. Figueroa, "Modern Development Methods and Tools for Embedded Reconfigurable Systems: a Survey," *INTEGRATION, The VLSI Journal*, vol. 43, pp.1-33, 2010.
[23] Xilinx Inc., "Microblaze Processor Reference Guide" (ver 14.1, 2012). Available: http://www.xilinx.com/tools/microblaze.htm.
[24] Xilinx Inc. "Kintex-7 FPGA Family". Available: http://www.xilinx.com/products/silicon-devices/fpga/kintex-7.
[25] Basterretxea, K., Tarela, J.M., del Campo, I., and Bosque, G.: 'An experimental study on nonlinear function computation for neural/fuzzy hardware design, *IEEE Trans. Neural Networks*, 2007, pp. 266–283.
[26] I. del Campo, R. Finker, J. Echanobe, and K. Basterretxea, "Controlled accuracy approximation of sigmoid function for efficient FPGA-based implementation of artificial neurons," *Electronics Letters*, pp.1598-1600, 2013.