Implementation of a Classification System of EEG Signals Based on FPGA

Víctor Asanza, Alisson Constantine, Stephany Valarezo, and Enrique Peláez
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral ESPOL
Guayaquil, Ecuador
Emails: {vasanza, alascons, valarezo, epelaez}@espol.edu.ec

***Abstract*— In the field of prosthetics, different technologies have been incorporated in recent years to improve their development and control, likewise the application of Field-Programmable Gate Arrays (FPGA) related to the Biomedicine field has increased due to its flexibility to perform multiple instructions in a reduced amount of time. This paper presents the implementation of a classification system based on FPGA capable of classifying characterized data, representing an imaginary motor task and a motor task in lower extremities. A three-layer feed-forward neural network was designed in Matlab, testing different architectures to assess the performance of the classifier, using methods such as the confusion matrix and the ROC curve.**

Keywords*— *Neural Networks, Electroencephalography, Embedded Systems, FPGA, Pattern Recognition.

I. Introduction

Field-Programmable Gate Arrays (FPGA) are electronic devices built as an array of configurable Logic Elements (LEs). Each LE can be configured to perform combinational or sequential functions. Modern FPGAs integrate other useful features, such as built-in multipliers, high-speed input/output (I/Os), data converters which includes analog-to-digital converters, large Random-Access Memories (RAM) arrays, and processors. All these features allow to create complex hardware in a System-on-Chip (SoC), which gives the option to create custom Central Processing Unit (CPU) for specific purposes, in order to perform multiple instructions [1].

Existing printed circuit boards (PCBs), such as the DE10 Standard, incorporate chips with System on Chip (SoC) and Hard Processor System (HPS) so they can create and implement two embedded systems in real time, each can work independently or together with their own processor (CPU). One of the advantages of communication between these two systems is that not only they can share physical resources, they can also verify response times and application execution [1].

Currently, the use of FPGA has increased quite high because it is simpler and cheaper than CPU / GPU implementations. It can be reprogrammed at any time to perform a different task than the one that was being executed initially. In addition, this allows the same algorithm to be modified to make it more robust and complex, in order to obtain better results.

The current trend on the use of SoC based-on FPGA is to program a simple and powerful microcontroller using C/C++ language, due to its flexibility and with the option of increasing the microcontroller capacity according to the application. FPGAs reduce time to market products and make the debugging process or adding features during the design phase easier [2]. Algorithms in C/C++ language have been established in all sectors of technology where embedded systems are used, and the trend is the use of machine learning algorithms for applications such as Embedded Vision, Speech Recognition, Industrial Robotics, Internet of Things, Big Data and Neural Networks (NN) that have contributed to strengthen Artificial Intelligence (AI) and Biomedical Signal Processing.

In this context, biomedical data such as electroencephalography (EEG), an electrophysiological technique to record the electrical activity arising from the human brain [3], can be used, along with Brain Computer Interfaces (BCI), to control electronic, robotic or prosthetic rehabilitation devices [4]. This processing methodology consists of pre-processing data, feature extraction and classification of EEG patterns for BCI usage. Hardware implementation using FPGA for real-time processing could help in the development of BCI devices, especially for those that require the analysis of large amounts of data.

In this work, EEG data recorded during the execution of the following tasks were used: motor activity of both feet, as well as the imaginary motor activity of both feet. Then a Neural Network was used to classify and determine the intention of the motor activity on test subjects using a Nios II processor. Section II describes related research done for detecting motor intentions; Section III details the dataset used; Section IV explains the methodology we have used in this research; Section V shows the results obtained. Finally, in section VI, we discuss the results and the conclusions of this work.

II. Related Work

An FPGA is a device that allows the modeling of a digital circuit using a specific language, being the most common Verilog Hardware Description Language (VHDL), which is then loaded in a matrix and created physically on a chip with a lot of gates and functions and other parts, such as the memory and input-output ports. The board, known as FPGA, can easily do multitasks, transmit and receive packets on high speed networks, so data communication is highly efficient, which can be used as the core of a variety of projects.

The neural networks (NN) is an algorithm architecture that contains a group of parameters that are constantly processed with machine learning as the goal. This is represented by a control system with feedback, where the output is being calculated from the raw input data as a reference on every stage of the whole system. Neural networks have been used to learn the set of parameters that define the function representing the training data [5]. This is a software architecture that contains a group of parameters, which are constantly learned using machine learning techniques. This NN represents a control system with feedback, where the output is being calculated from raw input data on every stage of the process.

Deep Neural Networks (DNN) such as the Convolutional Neural Networks (CNNs) are NNs focused on object

978-1-7281-5882-2/20/$31.00 © 2020 IEEE

classification problems and have multiple connected layers that can achieve lower error rates [5]. CNNs are more feasible for embedded implementations due to their reduced processing and storage demand, like the SqueezeNet model, which requires 480 KB of storage for its 1.2 million parameters [5]. They are compatible with cloud systems and their beauty is based on their small size, which allows them to be trained faster and have more efficient use of energy [5].

The motor cortical activity measured with EEG-BCI systems is in the μ frequency band of 8-13 Hz and in the β frequency band of 13-30 Hz [6]. It is common to use Power Spectral Density (PSD) to measure characteristics to determine the intention of motor activities in the frequency ranges of μ and β [7]. Yong Zhang et al. used Wavelet Coefficients, as characteristics for classifying mental tasks, contributing to the accuracy of the temporal resolution of these characteristics in the algorithm [8].

In machine learning, the knowledge of an expert is considered necessary to adjust the stages of preprocessing, choosing the hyperparameters and the activation functions. Classification of signals using NNs is done in 4 stages: Signals preprocessing, features extraction, features selection and then classification.

Before classifying the signals, data needs a preprocessing stage, commonly based on bandpass filters [9,10], and spatial filters used for removing averaged signals on all channels, typically with a Common Average Reference (CAR) filter [11,12,13]; then, at the features extraction stage, the NN looks for the features that allow us to obtain information in different domains that, depending on the application, let us have a better representation of the signals, such us: morphological [14], spectral [15], frequency time [13,16,17], temporary [15,9], representation [18,19] and entropy [20]. Some of the machine learning algorithms used to detect motor intentions are: Support Vector Machines (SVM) [21], Artificial Neural Networks (ANN) [21,22], Linear Discriminant Analysis (LDA) [9,21,23] and clustering algorithms [24]. In this research we have used Artificial Neural Networks (ANN) for the detection of motor activities.

III. Dataset

The dataset used was obtained from PhysioNet, which is shared by developers of the BCI2000 instrumentation system [22]. In our experiments, the dataset of EEG signals, acquired from healthy subjects, available on the website: www.physionet.org/physiobank/database/eegmmidb/ was used [22].

The dataset contains cortical brain activity signals, recorded from 8 subjects, using 64 superficial electrodes uniformly distributed following the international ten/ten standard system (excluding electrodes Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9, and P10) as shown in Fig. 1, with signals sampled at 160Hz for each subject; that is, 14 files in a European Data Format (EDF) [22] were recorded, which correspond to the following:

- Task 0: Rest position with eyes open and closed, both with a duration of one minute.
- Task 1: Open and close the left or right fist.
- Task 2: Imagine the opening and closing left or right fist.
- Task 3: Open and close both fists, or dorsi and plantar flexion of both feet.
- Task 4. Imagine the opening and closing of both fists or imagine the dorsi and plantar flexion of both feet.

The last four tasks were repeated three times for each topic with a duration of two minutes each. Task 0 represents the baseline and was repeated twice: with eyes open and closed.

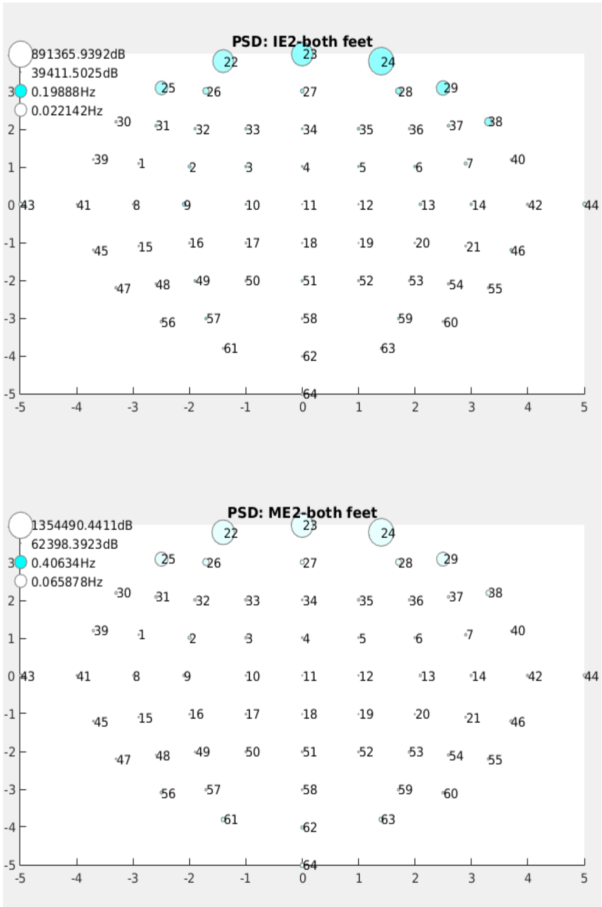


Fig. 1. Distribution and Spectral analysis of 64 surface electrodes seen in the international 10/10 system.

For this work Task 3 and Task 4 were used because it contained rich data that helped to detect the difference between the brain activity of the lower extremities during the execution of the motor task, as well as to imagine the motor activity [22].

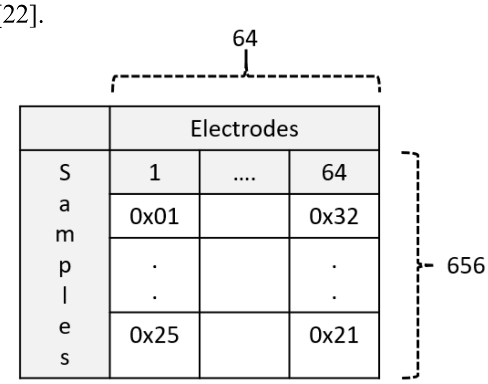


Fig. 2. Motor activity of both feet (ME2) file contain 4.599 rows or samples x 64 columns or surface EEG Electrodes

During the execution of Task 3 and Task 4, the subjects were instructed to randomly perform 30 events of 4,1 seconds each, as follows: Task 3 was composed of 7 repetitions of motor activity of both feet (ME2), with a total of 4.599 samples as shown in Fig. 2. To fulfill Task 4, the subjects were instructed to perform 9 repetitions of the imaginary motor activity of both feet (IE2) with a total of 6.008 samples. Each event was held after a resting position, labeled as event E0 [22]. In total, each task contained 30 pseudorandom events.

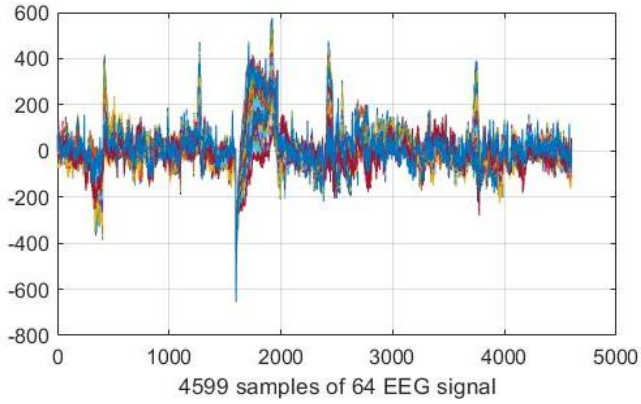


Fig. 3. Time-domain representation of 4599 samples of the EEG signal from motor activity of both feet (ME2)

IV. Methodology

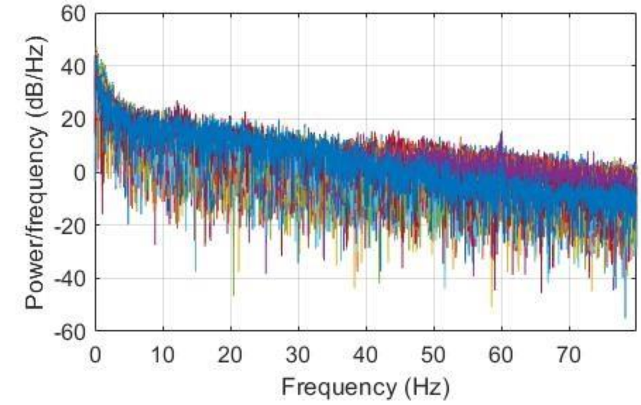
For the analysis of the data, Matlab 2015 was used, selecting the files corresponding to Task 3 and Task 4, and for each of the 8 subjects, the data of the events ME2 and IE2 were stored pseudorandomly in 16 Matlab files, containing a total of 128 Matlab files with EEG measurements. Fig. 3 shows the time domain EEG signal without Preprocessing.

Fig. 4 shows the spectrum of the 64 EGG signals where there is high power spectral density near zero Hz, which indicates that there is noise or low frequency artifacts, which is why a bandpass filter should be used later to allow attenuate the noise and allow the passage of signals in the frequency range that we are interested in analyzing, thus improving the signal to noise ratio.

Initially, the data was preprocessed using a bandpass filter from 8Hz to 30 Hz (including μ and β bands), this filter eliminated high frequency noise components, such as harmonics of electrical systems, and low frequency noise components, such as relative movements between electrode and scalp. The filter designed was a Butterworth Infinite Impulse Response (IIR) of order 200 and double pass to maintain a linear phase. Fig. 5 shows the spectrum of the 64 EEG signals after being filtered in the range of 8Hz to 30Hz, where the spectral density of the low frequency noise and the 60Hz noise produced by the power grid no longer predominates.

Fig. 4. Periodogram Power Spectral Density Estimate of the EEG signal from motor activity of both feet (ME2)

The characteristics from each electrode (64 in total) were extracted, using PSD in 5 seconds temporary windows, resulting in a new file containing 95.079 rows corresponding to the events, and 64 columns corresponding to the frequencies. To validate the results, two columns were added at the end with the identifier of the event to which each row belongs, that is IE2 [1 0] or ME2 [0 1]; and, the spectral analysis of EEG electrodes in the 10/10 system, as shown in Fig.1, which represents the average of all available subjects that perform the IE2 and ME2 activity.



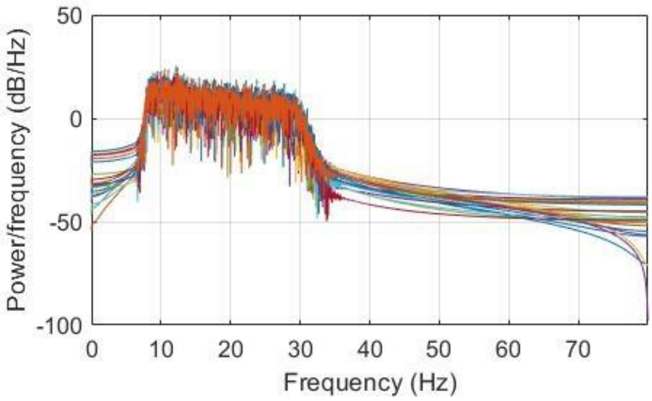


Fig. 5. Periodogram estimation of the Power Spectral Density, after bandpass filter of the EEG signal corresponding to the motor activity of both feet (ME2)

Based on the fact that FPGA is a limited resource hardware, a Multi-Layer Perceptron (MLP) Neural Network with three layers was decided to be used; an input layer with the characteristics of the events to be classified, a hidden layer with a number of neurons that will be calibrated to analyze the behavior of the classifier, and an output layer that will determine the class to which the event, that is being classified, belongs to.

The input layer of the neural network has 64 neurons corresponding to the 64 PSD features, calculated at 10 seconds of sampling of each EEG electrode. The hidden layer has a variable number of neurons that have been modified upwards from 5 to 100 neurons, based on the effects over the greatest difference in the accuracy of the classifier. Ten, thirty, sixty and ninety were subsequently evaluated with the Matlab Pattern Recognition App to be classified. The output layer has the same number of neurons according to the classes to which the events belong (IE2, ME2), that is, two neurons; the classifier, through this layer, will indicate the probability of belonging to classes IE2 and ME2 for the events evaluated by the classifier.

For training the neural network, 15% of the data was used for validation, 15% for testing and 70% for training, chosen randomly. For training we used a Scaled Conjugate Gradient

Backpropagation (CGB) technique and the performance of the classification model was measured by the Receiver Operating Characteristic (ROC) curve.

As part of the preprocessing stage the input data was normalized between the minimum and maximum values of each row, in order to have them in a specific range, in this case, [-1,1]. Then, from the input layer, the weights and bias are added to the hidden layer, as can be seen in Fig. 6. For the hidden layer a Hyperbolic Tangent Sigmoid transfer function was used, which calculates the output of the neurons from their net input. Then, a layer weight is connected to the output layer from the hidden layer. The output layer has a similar procedure, but instead a Smooth Maximum transfer function was used.

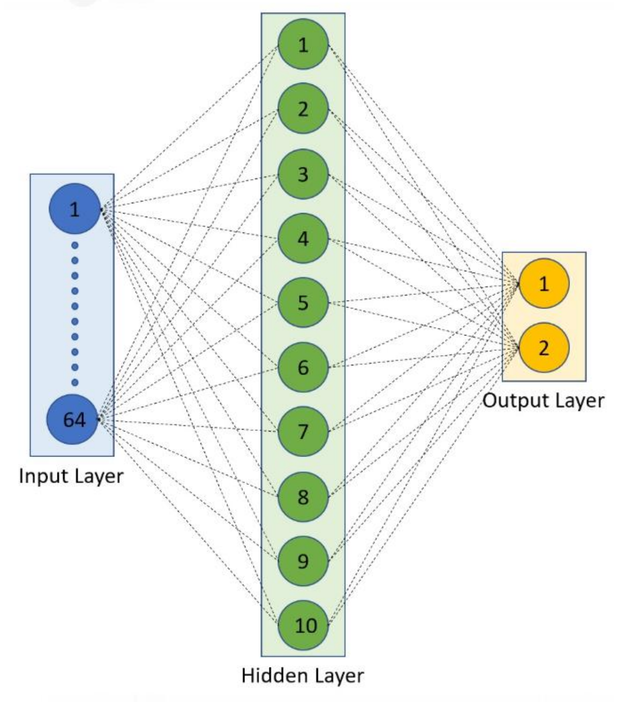


Fig. 6. Diagram of the pattern recognition function of neural networks in Simulink.

In order to test the program on the NIOS II processor, an equivalent to the Matlab code for NN was created in C. The weight and biases for the trained Neural Network were obtained from Matlab and converted into arrays in the C program.

The FPGA used in this research is a NIOS II/e processor implemented on the Cyclone V FPGA of an Altera DE10 Standard. The processor ran directly from the DE10's 50MHz clock, and it included an Interval Timer, JTAG UART and SD_Card Interface. The memory used was the DE10's 64 MB SDRAM chip, running at 50 MHz, which was large enough to fit the weights parameters and bias vectors needed by the NN and libraries used in the code developed. All the other parameters were left as their defaults.

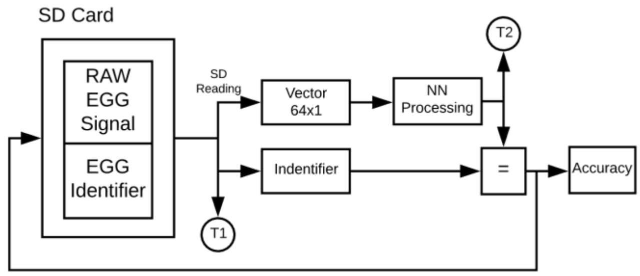


Fig. 7. Block diagram of data processing in FPGA

The EEG data recorded from the 64 surface electrodes and their respective identifier were stored on a 2GB FAT16 format SD card, using a CSV (Comma separated values) file format, connected to the FPGA through GPIO pins and then read by the code developed in C, as detailed in Fig. 7. The values of the electrodes were stored in a 64 columns vector, and then processed by the NN, then the result obtained was compared with the real identifier of the motor intention performed by the subject, to assess the accuracy of the result, this process was executed for each of the data saved in the SD card. Through the Interval Timer, the execution time of each of the processes were calculated. The process follows three steps: Open the SD card file, read the data from the SD card (T1) and process the data through the ANN (T2).

V. Results

As seen in the Confusion matrix, the NN was able to perform the classification between motor activity (ME2) versus imaginary motor activity (IE2) of lower extremities, with 10 seconds of EEG samples (1600 samples).

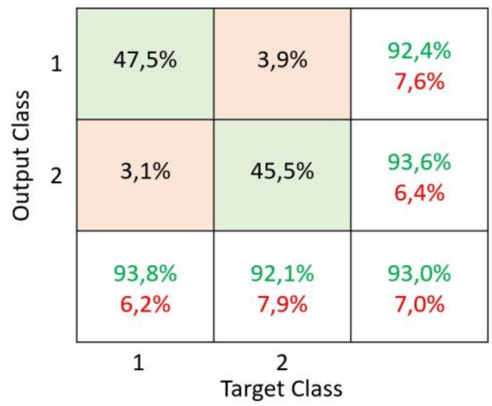


Fig. 8. Confusion Matrix of the classification of all events

Fig. 8 shows the confusion matrix of all EEG signals from the sixty-four superficial electrodes contrasting the prediction of false positives versus false negatives for each of the tasks 1-IE2, 2-ME2. The probability of detecting a true positive, when ME2 event is classified as a ME2 motor activity, is 92.1%; the probability of detecting a false positive, when ME2 event is classified as an IE2 imaginary motor activity, is 7.9%. Likewise, the probability of detecting a true negative, when IE2 event is classified as an IE2 imaginary motor activity, is 93.8%; the probability of detecting a false negative, when IE2 event is classified as an ME2 motor activity, is 6.2%. That is, the neural network correctly predicts ME2 events with 92,1% accuracy, and IE2 events with 93,8%.

Fig. 9 shows the ROC curve of the classes IE2 y ME2 with 809 epochs; the neural network has a good sensitivity in the detection of true positives and true negatives in both classes, based on this criteria, we select the neural network with a size of 30 neurons for the hidden layers.

When testing the neural network of 3 layers with the following number of neurons in the hidden layer (10, 30, 60 and 100) we obtained the following accuracies: 85.5%, 93%, 91.6% and 80.2% respectively. The network configured with 30 neurons in the hidden layer had the greater precision in this FPGA based classification system, and it was used in this research.

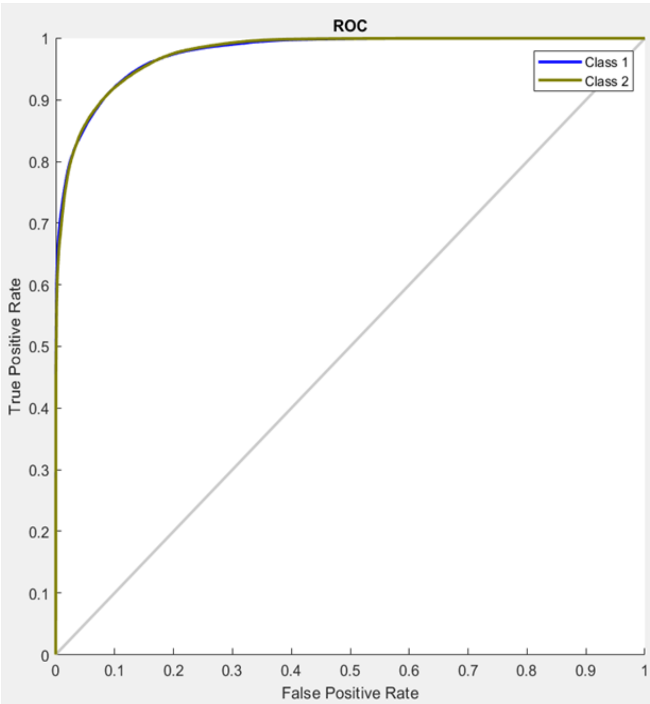


Fig. 9. Receiver Operating Characteristics of the classification of all events

Table 1 shows the resources used by the FPGA for the implementation of the neural network and the reading of the electrode data through the SD card

TABLE I. RESOURCES USED BY THE FPGA

Logic utilization (in ALMS)	1,303 / 41,910 (4%)
Total block memory bits	47,360 /5,662,720 (<1%)
Total pins	45/499 (9%)

The execution times of the process in the FPGA with the mentioned configurations are detailed in Table II.

TABLE II. EXECUTION TIME

Time to look for the file in the SD	21,26 [us]
Time to open the file in the SD	22,30 [us]
Processing time of the neural network	27,36 [us]

VI. DISCUSSION AND CONCLUSIONS

FPGAs are a powerful tool that can be used as a type of microcontroller. Although, they have limited capacity FPGAs have a wide variety of components that allows them to perform as well as a PC or even as a server.

For the analysis and deployment of innovative solutions, it is easy and simple to start carrying out tests using FPGA development cards, they are flexible when correcting logical modules is needed, and allows the possibility of multi code reloads, which represents hardware and software updates for the device [24].

The use of the micro SD card adds a time of 43.56 µs before the classification process, this is because these memory devices are external to the designed FPGA system and requires an adaptation in the voltage levels.

Neural Networks could be a complex concept to understand that requires a whole new knowledge with computing algorithms and more statistical analysis. When using data from temporary EEG signals, it is important to select the appropriate characteristics that allow an adequate representation of the events to be classified, but if the application uses limited hardware resources, the features must not have a high computational cost to be extracted from raw data. In this case, the three-layer neural network with 30 neurons in the hidden layer, had a better performance detecting IE2 (93.8%) vs ME2 (92.1%) events in this FPGA based classification system; this will allows us to use data of imaginary motor tasks to control prostheses for people who have lost some limb, or due to accidents, have suffered damage to the spinal cord and can no longer use other assistance limbs; in addition, we could control assistive equipment rehabilitation, in subjects with residual motor skills.

The processing times and the total resources used in the FPGA are optimal enough to execute neural networks of characterized data, and then being able to classify them in almost real time; however, the characterization time of the data with PSD makes the total classification time of the network increase to 5 seconds.

For future work, in the implementation of FPGA-based classification systems, it is proposed to change the SD memory for a Double Data Rate 3 Synchronous Dynamic Random-Access (DDR3-SDRAM) memory, and test with different configuration to reduce the access time; in the algorithmic part it is proposed to reduce the extraction time of the characteristics using time-frequency characteristics. In addition, it is proposed to analyze the capabilities of the FPGA with deep neural networks where it is not necessary to extract features.

REFERENCES

[1] Harris, S., & Harris, D. (2015). Digital design and computer architecture: arm edition. Morgan Kaufmann.

[2] Reaz, M. B. I, Hussain, M. S. and Mohd-Yasin, F., "Techniques of EMG signal analysis: detection, processing, classification and applications", Biological Procedures Online, vol. 8, no. 1, 2006.

[3] Britton JW, Frey LC, Hopp JLet al., authors; St. Louis EK, Frey LC, editors. Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants [Internet]. Chicago: American Epilepsy Society; 2016

[4] Gordleeva, S. Y., Lukoyanov, M. V., Mineev, S. A., Khoruzhko, M. A., Mironov, V. I., Kaplan, A. Y., & Kazantsev, V. B. (2017).

Exoskeleton control system based on motor-imaginary brain–computer interface. Современные технологии в медицине, 9(3 (eng)).

[5] Shin, Y. K., Lee, D. R., Hwang, H. J., You, S. J. H., & Im, C. H. (2012). A novel EEG-based brain mapping to determine cortical activation patterns in normal children and children with cerebral palsy during motor imagery tasks. NeuroRehabilitation, 31(4), 349-355.

[6] Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448.

[7] Zhang, Y., Liu, B., Ji, X., & Huang, D. (2017). Classification of EEG signals based on autoregressive model and wavelet packet decomposition. Neural Processing Letters, 45(2), 365-378.

[8] Subasi, A., & Gursoy, M. I. (2010). EEG signal classification using PCA, ICA, LDA and support vector machines. Expert systems with applications, 37(12), 8659-8666.

[9] Bastos-Filho, T., Floriano, A., Couto, E., & Godinez-Tello, R. J. (2018). Towards a system to command a robotic wheelchair based on independent SSVEP–BCI. In Smart Wheelchairs and Brain-Computer Interfaces (pp. 369-379).

[10] Becerra, M. A., Londoño-Delgado, E., Pelaez-Becerra, S. M., Castro-Ospina, A. E., Mejia-Arboleda, C., Durango, J., & Peluffo-Ordóñez, D. H. (2018, September). Electroencephalographic Signals and Emotional States for Tactile Pleasantness Classification. In International Workshop on Artificial Intelligence and Pattern Recognition (pp. 309-316). Springer, Cham.

[11] Tello, R. M., Müller, S. M., Hasan, M. A., Ferreira, A., Krishnan, S., & Bastos, T. F. (2016). An independent-BCI based on SSVEP using Figure-Ground Perception (FGP). Biomedical Signal Processing and Control, 26, 69-79.

[12] Iandola, F., & Keutzer, K. (2017, October). Small neural nets are beautiful: enabling embedded systems with small deep-neural-network architectures. In Proceedings of the Twelfth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis Companion (p. 1). ACM.

[13] Rodríguez-Sotelo, J., Osorio-Forero, A., Jiménez-Rodríguez, A., Cuesta-Frau, D., Cirugeda-Roldán, E., & Peluffo, D. (2014). Automatic sleep stages classification using EEG entropy features and unsupervised pattern analysis techniques. Entropy, 16(12), 6573-6589.

[14] Lisi, G., Noda, T., & Morimoto, J. (2014). Decoding the ERD/ERS: influence of afferent input induced by a leg assistive robot. Frontiers in systems neuroscience, 8, 85.

[15] Castillo-Garcia, J. F., Caicedo-Bravo, E. F., & Bastos, T. F. (2018). Adaptive Spontaneous Brain-Computer Interfaces Based on Software Agents. Advances in Data Science and Adaptive Analysis.

[16] Gao, Z. K., Cai, Q., Yang, Y. X., Dong, N., & Zhang, S. S. (2017). Visibility graph from adaptive optimal kernel time-frequency representation for classification of epileptiform EEG. International Journal of Neural Systems, 27(04), 1750005.

[17] Gordleeva, S. Y., Lukoyanov, M. V., Mineev, S. A., Khoruzhko, M. A., Mironov, V. I., Kaplan, A. Y., & Kazantsev, V. B. (2017). Exoskeleton control system based on motor-imaginary brain–computer interface. Современные технологии в медицине, 9(3 (eng)).

[18] Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., ... & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. Human brain mapping, 38(11), 5391-5420.

[19] Becerra, M. A., Londoño-Delgado, E., Pelaez-Becerra, S. M., Serna-Guarín, L., Castro-Ospina, A. E., Marin-Castrillón, D., & Peluffo-Ordóñez, D. H. (2018, September). Odor Pleasantness Classification from Electroencephalographic Signals and Emotional States. In Colombian Conference on Computing (pp. 128-138). Springer, Cham.

[20] Movahedi, F., Coyle, J. L., & Sejdić, E. (2018). Deep belief networks for electroencephalography: A review of recent contributions and future outlooks. IEEE journal of biomedical and health informatics, 22(3), 642-652.

[21] Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1), 115.

[22] Murad, A., & Pyun, J. Y. (2017). Deep recurrent neural networks for human activity recognition. Sensors, 17(11), 2556.

[23] Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R. BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. IEEE Transactions on Biomedical Engineering 51(6):1034-1043, 2004.

[24] Kenny, R., & Watt, J. (2016). The Breakthrough Advantage for FPGAs with Tri-Gate Technology. URL: https://www.altera.com/en_US/pdfs/literature/wp/wp-01201-fpga-tri-gate-technology.pdf:12.10.2017.