COMPUTATIONAL ARCHITECTURE FOR THE INFERENCE OF A QUANTIZED CONVOLUTIONAL NEURONAL NETWORK FOR THE DETECTION OF ATRIAL FIBRILLATION

ANDRÉS FELIPE JARAMILLO RUEDA
LAURA YURITZA VARGAS PACHECO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍA FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2020

COMPUTATIONAL ARCHITECTURE FOR THE INFERENCE OF A QUANTIZED CONVOLUTIONAL NEURONAL NETWORK FOR THE DETECTION OF ATRIAL FIBRILLATION

ANDRÉS FELIPE JARAMILLO RUEDA
LAURA YURITZA VARGAS PACHECO

Trabajo de grado presentado como requisito parcial para optar al título de ingeniero electrónico

Director
CARLOS AUGUSTO FAJARDO ARIZA
Doctor en Ingeniería

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍA FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2020

AGRADECIMIENTOS

Agradezco a mi madre Claudia Pacheco,
a mis abuelas Luz Marina Jarro
y Visitación Poveda, a mi familia
y amigos por el amor que me han dado,
por la mujer que me hicieron, los valores
que me inculcaron y por siempre
estar a mi lado brindando su
apoyo en cualquier situación. ***Laura Vargas***

Quiero agradecer de todo corazón a
mi madre Estella Rueda Molina que
forjó en mí la entrega y la pasión para
cada día esforzarme por hacer el bien.
También a todos mis amigos y familiares
que han fomentado la pasión
por mi crecimiento académico
y humano. ***Andrés Jaramillo***

Contents

List of Figures

List of Tables

RESUMEN

TITULO: ARQUITECTURA COMPUTACIONAL PARA LA INFERENCIA DE UNA RED NEURONAL CONVOLUCIONAL CUANTIZADA PARA LA DETECCIÓN DE LA FIBRILACIÓN AURICULAR*

AUTORES: ANDRÉS FELIPE JARAMILLO RUEDA** Y LAURA YURITZA VARGAS PACHECO**

PALABRAS CLAVE: FIBRILACIÓN AURICULAR, DETECCIÓN AUTOMÁTICA, IMPLEMENTACIÓN FPGA, RED NEURAL CONVOLUCIONAL CUANTIZADA.

DESCRIPCIÓN: Las arritmias cardíacas son una de las enfermedades cardíacas más comunes en todo el mundo, que se caracterizan por un ritmo cardíaco anormal que puede poner en peligro la vida. Recientemente, se han propuesto varias redes neuronales convolucionales para detectar diferentes arritmias cardíacas. Proponemos una arquitectura computacional para la inferencia de una red neuronal convolucional cuantificada (Q-CNN) que permite la detección de una arritmia cardíaca llamada fibrilación auricular (FA). La arquitectura computacional se implementó y probó en un FPGA Xilinx Artix-7. El diseño se basa en un procesador de matriz sistólica, que está optimizado para realizar tanto las capas convolucionales como las completamente conectadas. Los resultados experimentales se presentan con respecto al proceso de cuantización en diferentes números de bits, cantidad de hardware y precisión. Finalmente, se seleccionó un Q-CNN de 22 bits, que logra un 94% de precisión. Este trabajo pretende ser la base para la implementación futura de un dispositivo portátil, de bajo costo y alta confiabilidad para el diagnóstico de la FA.

*Trabajo de grado

**Facultad de Ingeniería Fisicomecánicas. Director Carlos A. Fajardo Ariza

ABSTRACT

TITLE: COMPUTATIONAL ARCHITECTURE FOR THE INFERENCE OF A QUANTIZED CONVOLUTIONAL NEURONAL NETWORK FOR THE DETECTION OF ATRIAL FIBRILLATION *

AUTHORS: ANDRÉS FELIPE JARAMILLO RUEDA** Y LAURA YURITZA VARGAS PACHECO**

KEY WORDS: ATRIAL FIBRILLATION, AUTOMATIC DETECTION, FPGA IMPLEMENTATION, QUANTIZED CONVOLUTIONAL NEURAL NETWORK.

DESCRIPTION: Cardiac arrhythmias are one of the most common heart diseases worldwide, which are characterized by an abnormal heartbeat rhythm that can be life-threatening. Recently, several convolutional neural networks have been proposed to detect different cardiac arrhythmias. We propose a computational architecture for the inference of a Quantized Convolutional Neural Network (Q-CNN) that allows the detection of a cardiac arrhythmia called Atrial Fibrillation (AF). The computational architecture was implemented and tested in a Xilinx Artix-7 FPGA. The design is based on a systolic array processor, which is optimized to perform both the convolutional and fully connected layers. Experimental results are presented regarding the quantization process at different number of bits, amount of hardware and accuracy. Finally, a 22-bits Q-CNN was selected, which achieves a 94% of accuracy. This work aims to be the basis for future implementation of a portable, low-cost and high-reliability device for the diagnosis of the AF.

*Bachelor Tesis

**Facultad de Ingeniería Fisicomecánicas. Director Carlos A. Fajardo Ariza

INTRODUCTION

Atrial fibrillation (AF) is an arrhythmia that presents irregular heartbeats and it is associated with an increase in heart rate due to a disorder in the electrical signals that activate the atria. This type of arrhythmia occurs asymptomatically, that is, there are no symptoms until the first acute episode.

On the one hand, several studies have proposed the convolutional neural networks (CNN) for the detection of atrial fibrillation with high levels of accuracy,. On the other hand, some researches have shown that custom hardware for the inference of CNNs could surpass the efficiency of general-purpose processor equivalents regarding both throughput and energy consumption.

Quantization is an effective strategy that allows reducing both precision of weights and activations (neuron outputs). The quantization of a CNN (Q-CNN) is a first step before to implement a CNN in a custom-hardware.

FPGAs have become striking to implement Q-CNNs due to their flexibility and high energy efficiency. However, there are still many challenges because the CNNs are known for demanding a huge amount of computational and memory resources.

CHUNGH, Sumeet S et al. "Worldwide epidemiology of atrial fibrillation: a Global Burden of Disease 2010 Study". In: *Circulation* 129.8 (2014), pp. 837–847.

YAO, Zhenjie; ZHU, Zhiyong; CHEN, Yixin. "Atrial fibrillation detection by multi-scale convolutional neural networks". In: *20th International Conference on Information Fusion, Fusion 2017 - Proceedings* (2017).

XIA, Yong et al. "Detecting atrial fibrillation by deep convolutional neural networks". In: *Computers in Biology and Medicine* 93.December 2017 (2018), pp. 84–92.

POURBABAEE, Bahareh; ROSHTKHARI, Mehrsan Javan; KHORASANI, Khashayar. "Deep Convolutional Neural Networks and Learning ECG Features for Screening Paroxysmal Atrial Fibrillation Patients". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.12 (2018), pp. 2095–2104.

GUO, Yunhui. "A survey on methods and theories of quantized neural networks". In: *arXiv preprint arXiv:1808.04752* (2018).

WANG, Erwei et al. "Deep Neural Network Approximation for Custom Hardware: Where We've Been, Where We're Going". In: *ACM Computing Surveys (CSUR)* 52.2 (2019), pp. 1–39.

Ibid.

Strategies to perform the inference process at the edge (low power) are currently a hot topic in hardware research. In is employed some loop optimization techniques to reuse data and manage data movement efficiently. In is proposed a Winograd transformation-based algorithm to optimize the convolution process, which uses a cross-layer strategy. The algorithm permits a reduction of over 90% in the transfer process of the intermediate data. In is proposed a Stacked Systolic Array (SSA) to reduce the impact of the timing violation caused by crossing-die critical paths. They achieve up to 85% performance improvement for different layers of DNN models.

In this work, we propose a computational architecture for the inference process of a quantized version of the Castillo-Granados CNN, which is based on. Our goal is to design a specific purpose processor, which carries out the inference process by using the minimum amount of computational and memory resources at high accuracy possible. We designed a systolic matrix processor architecture that is optimized to perform both the convolution and fully connected layers. This processor allows the inference of a 22-bits Q-CNN version of and achieves a 94% of accuracy.

This paper is organized as follows: Chapter I gives a description of the CNN used. Chapter II describes the quantization process of CNN. Chapter III describes the design of the computational architecture for the quantized CNN inference. Chapter IV summarizes the main results of this work. Finally, the article is closed with the conclusions in Chapter V.

MA, Yufei et al. “Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks”. In: *FPGA 2017 - Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (2017), pp. 45–54.

YU, Jincheng et al. “Instruction driven cross-layer CNN accelerator with winograd transformation on FPGA”. in: *2017 International Conference on Field-Programmable Technology, ICFPT 2017* 2018-Janua (2018), pp. 227–230.

WEI, Xuechao et al. “Automated Systolic Array Architecture Synthesis for High Throughput CNN Inference on FPGAs”. In: *Proceedings - Design Automation Conference* Part 12828 (2017), pp. 3–5.

CASTILLO, Jeyson; GRANADOS, Yenny; FAJARDO, Carlos. “Patient-Specific Detection of Atrial Fibrillation in Segments of ECG Signals using Deep Neural Networks”. In: *Ciencia E Ingenieria Neogranadina* 30.1 (2020).

ACHARYA, U. Rajendra et al. “Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network”. In: *Information Sciences* 405 (2017), pp. 81–90.

CASTILLO; GRANADOS; FAJARDO, “Patient-Specific Detection of Atrial Fibrillation in Segments of ECG Signals using Deep Neural Networks”, op. cit.

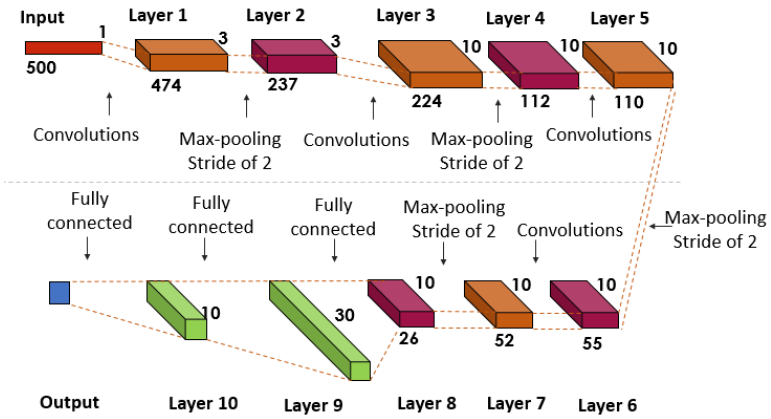
1. CONVOLUTIONAL NEURAL NETWORK

A typical CNN is made up of different layers. In each layer, there is a certain amount of connected filters that extract information for subsequent layers. An entry is provided, which passes through layers to generate a feature vector. Then, a classifier is used in the characteristic vector obtained to produce the result of the classification. There are mainly three types of layers in a CNN model: convolutional layers, grouping layers, and fully connected (FC) layers.

In this paper, the CNN Castillo-Granados will be implemented, which was trained for the detection of AF from ECG signals. These ECG signals were registered by the Einthoven triangle method and stored in a vector of 500 samples with a sampling rate of 250 [samples/s].

This CNN achieved an accuracy of 97.44% using a 64-bit double-float format. Figure 1 shows the architecture of the CNN. The network input has been designed to process vectors, which are the ECG segments taken in two seconds (500 samples).

Figure 1: Castillo-Grandos CNN Architecture



Source: CASTILLO, Jeyson; GRANADOS, Yenny; FAJARDO, Carlos. "Patient-Specific Detection of Atrial Fibrillation in Segments of ECG Signals using Deep Neural Networks". In: *Ciencia E Ingenieria Neogranadina* 30.1 (2020)

Ibid.

URIBE, William; DUQUE, Mauricio; MEDINA, Eduardo. "Electrocardiografıa y arritmias". In: *Clınica Medellın. Editorial PLA Export Bogotá DC* (2005), pp. 41–5.

CASTILLO; GRANADOS; FAJARDO, "Patient-Specific Detection of Atrial Fibrillation in Segments of ECG Signals using Deep Neural Networks", op. cit.

The CNN is made up of four convolution and three FC layers. Table 1 summarizes the characteristics of the layers. Note that, the network has a total of 9385 parameters.

Table 1: Architecture Parameters

Neural Network Architecture				
Layer type	Output dimension	Parameters	Kernel size	Stride
Input	(500,1)	-	-	-
Convolution	(474,3)	84	(27,3)	1
Max-pooling	(237,3)	-	-	2
Convolution	(224,10)	430	(14,10)	1
Max-pooling	(112,10)	-	-	2
Convolution	(110,10)	310	(3,10)	1
Max-pooling	(55,10)	-	-	2
Convolution	(52,10)	410	(4,10)	1
Max-pooling	(26,10)	-	-	2
Flatten	260	-	-	-
Fully-connected	30	7830	-	-
Fully-connected	10	310	-	-
Fully-connected	1	11	-	-

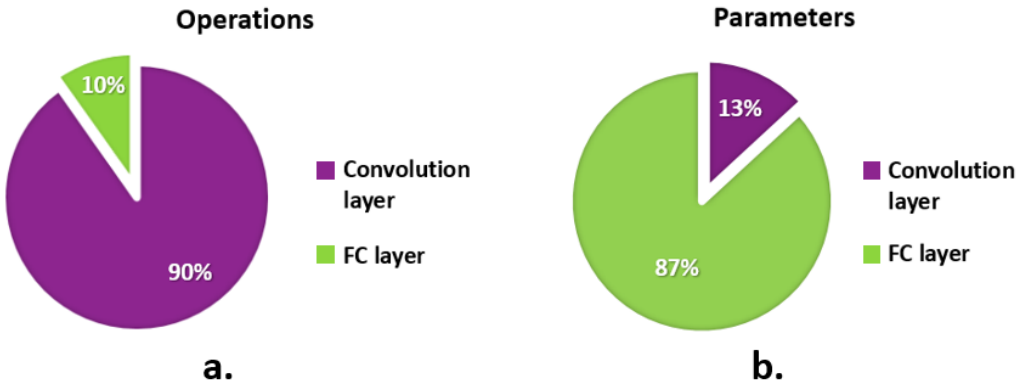
Adapted from source: CASTILLO, Jeyson; GRANADOS, Yenny; FAJARDO, Carlos. “Patient-Specific Detection of Atrial Fibrillation in Segments of ECG Signals using Deep Neural Networks”. In: *Ciencia E Ingenieria Neogranadina* 30.1 (2020)

Figure 2 shows the number of operations and parameters for both convolutional and FC layers. Figure 2a. shows the distribution percentages of the number of operations made in the convolutional and FC layers. Figure 2b. shows the distribution percentages of the number of parameters required in the convolutional and FC layers.

Note that, on the one hand, the convolutional layers perform the highest percentage of operations (90% vs. 10%). On the other hand, the FC layers require the highest percentage of parameters (87% vs. 13%).

Ibid.

Figure 2: Percentage of operations and parameters for convolutional and fully connected layers

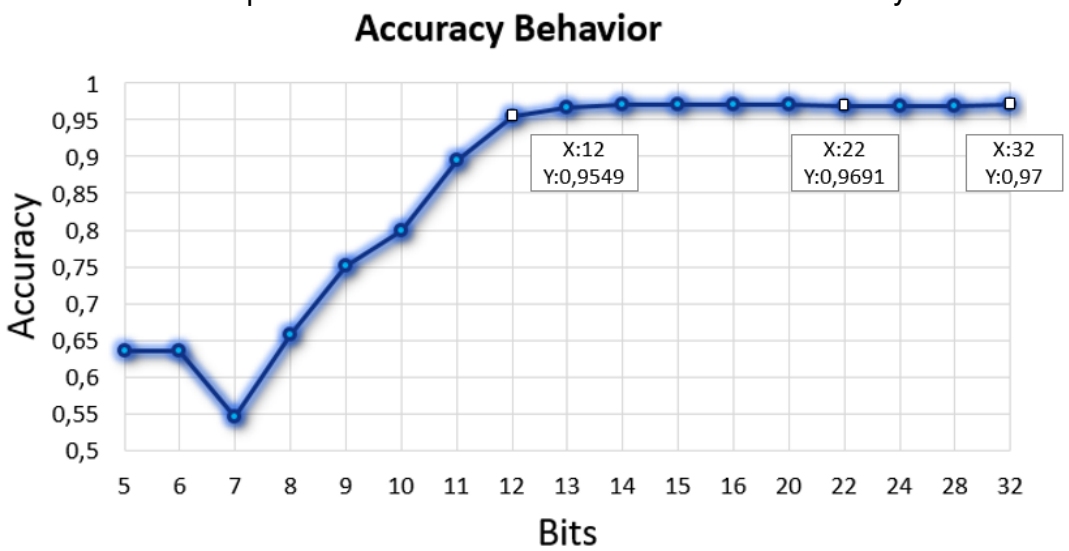


2. QUANTIZATION PROCESS

The implementation of the inference process in a custom hardware requires a quantization process. This process allows the change of 64-bit double-precision format for a reduced number of bits in fixed-point format. This change reduces considerably the amount of computational and memory resources.

Figure 3 shows the results of the fake quantization process that was carried out using Matlab ® . Note that by using just 12 bits an accuracy of 95% is achieved. Also, note that from 12 bits onwards there is no a considerable increase in the accuracy. However, in the hardware implementation, there are some issues related to the truncation error because of the reduction in the number of bits, which will be analyzed in Chapter 4.

Figure 3: Relationship between the number of bits and the accuracy



Source: False quantization for convolutional neural networks - Universidad Industrial de Santander -Grupo CPS Material without publishing.

The fake quantization process is out of the scope of this project. The Quantized CNN was provided for the CPS research group.

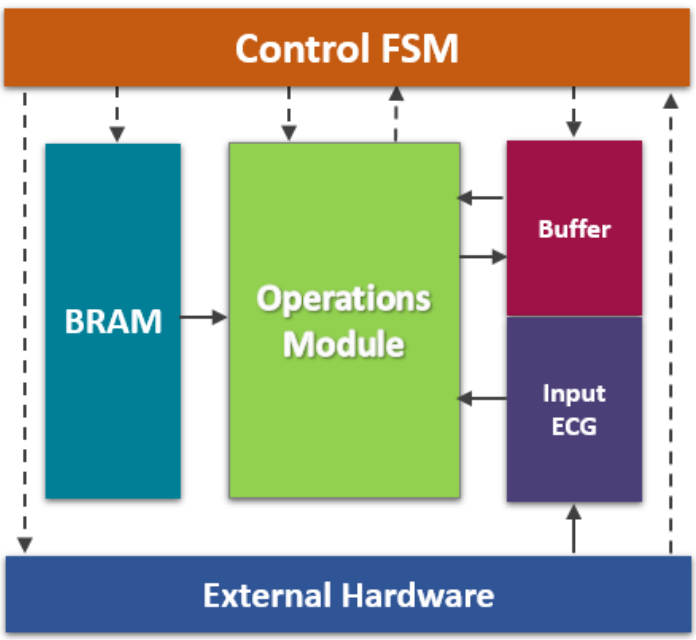
3. DESIGN DESCRIPTION

Figure 4 shows a block diagram of the computational architecture designed for the inference of the CNN of Figure 1.

The design has an operation module that computes the input data with the parameters of each layer. This module is controlled by a state machine (FSM control) that addresses the computational resources that carry out the mathematical operations.

We have designed a computational strategy that allows us to use a single *operation module* to perform both the convolutional and FC layers. This strategy demands the use of buffers to temporarily store output results. Thus, the proposed architecture achieves a considerable reduction in the use of computational resources. However, this strategy penalizes the throughput because the reuse strategy does not allow a pipeline implementation.

Figure 4: Block diagram of computational architecture. ***Dashed line:** Control signal. **Continuous line:**Data



A functional description of the modules on Figure 4 is given below :

- *Control FSM*: State machine addresses the flow of data processed in each module. Also, the FSM controls the *Operation Module* to perform all layers. Finally, the FMS is in charge of write/read memory process.
- *BRAM*: Memory to store all parameters of the CNN.
- *Operations Module*: adaptive module that computes convolution or FC operations.
- *Buffer*: Set of two memories that store the temporary outputs of each layer, alternating writing and reading, the read data is returned as inputs of the next layer.
- *Input ECG*: Memories that stores the ECG segments. In this design, there are two Input ECG memories, which allow us to read a new segment while a previous segment is being processing.
- *External Hardware*: ECG signals are acquired through External Hardware, this is a module that was designed to communicate the FPGA with an ADC using SPI protocol. *External hardware* provides the data to *Input ECG* in groups of 500 samples with a sampling frequency of 250 [samples/s].

The operations module is one of the most important, below is the design made.

3.1. DESIGN OF THE OPERATIONS MODULE

The operations processing module has been designed based on the convolution layers since these contain the largest number of operations in the architecture (Fig.2).

The *Operations Module* uses a loop unrolling strategy for the kernels in the convolu-

tion layers. This strategy is carried out by a custom systolic array processor, which contains 27 multipliers, 27 adders, and 27 shift registers. This custom processor allows the reuse of the hardware for all layers.

Figure 5: Data flow in *Operations Module*

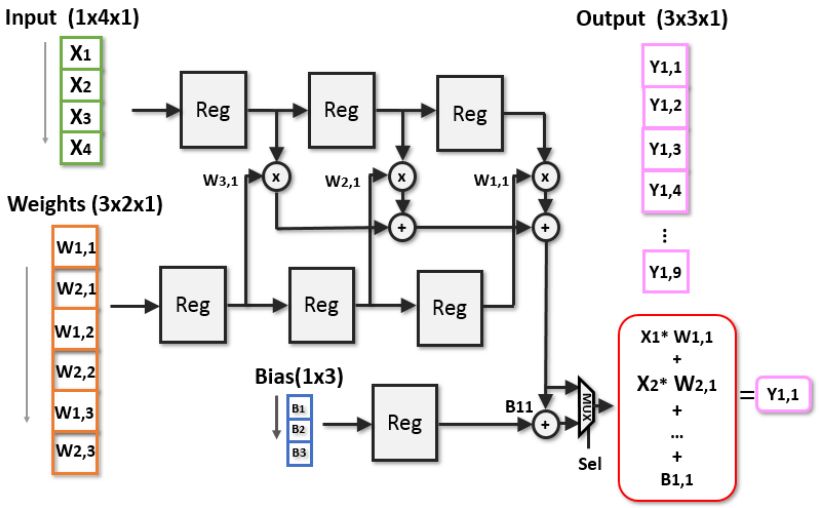


Figure 5 illustrates the configuration of the logistic resources used for the execution of operations. Note that Kernel and input data (Chapter 1) flow from left to right in each clock cycle until all 27 registers are filled. Once the first 27 data are saved on the registers, a first temporary data out is obtained. Then, the input data is 1-left shifted and a second temporary data is obtained and so on. All temporary data are accumulated in a specific position of the Buffer memory.

It is important to note that the dimensions change for one layer to another, so the bias is added in the last tensor dimension.

The data flow is modified using control signals from the state machine, the design in figure 4 is configured to calculate the FC layers, performing point product operations. This strategy saves the use of logical resources for the description of layers that execute different operations.

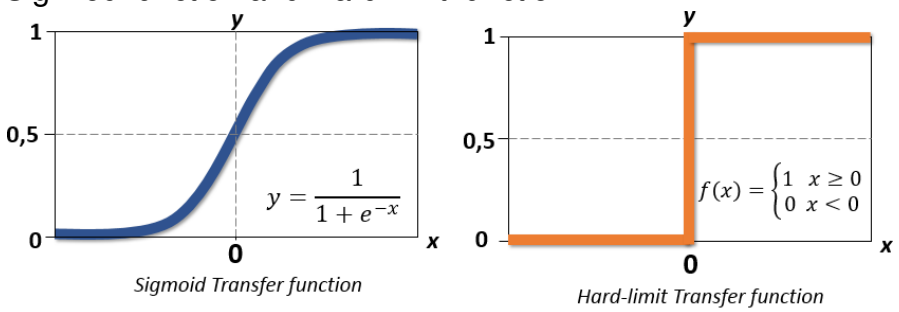
LI, Huimin et al. “A high performance FPGA-based accelerator for large-scale convolutional neural networks”. In: *FPL 2016 - 26th International Conference on Field-Programmable Logic and Applications* (2016), pp. 1–9.

If better latency is required, more parallelism can be applied (more than 27 operations per clock cycle) and more than one kernel at a time.

3.2. HARD-LIMIT TRANSFER FUNCTION

The original design of the neural network was performed with the *Sigmoid* activation function (Figure 6). This function is applied after the last FC layer. In order to reduce computational resources, we replaced the *Sigmoid* function with a Hard-limit function, which was implemented in hardware by using a simple not gate. This gate was connected signed bit of the final output.

Figure 6: Sigmiod function and Hard-limit function



Our results suggest that the design can be implemented with a *Hard-limit* function without affecting the accuracy of the network.

TISAN, A et al. “Digital Implementation of The Sigmoid Function for FPGA Circuits”. In: *Acta Technica Napocensis* 50.2 (2009), pp. 15–20.

4. RESULTS

The computational architecture for inference of CNN was implemented on the Basys 3 Development Board which is based on the latest Artix-7 FPGA from Xilinx. The synthesis, simulation and debugging was done using the Xilinx Vivado Design Suite software with the 2019.1 version.

The design was tested with a set of 1000 ECG signals of the MIT BIH Atrial Fibrillation database. These signals were quantized from 12 to 32 bits by using Matlab (Chapter 2). Several tests were developed to validate intermediate and final results. The intermediate results were validated by *ILA Tool* from Vivado.

The percentage truncation error (E_t) is generated for the reduction in the number of bits and calculated by Equation 4.1.

$$E_t = \left| \frac{CHR - SR}{SR} \right| \times 100\% \tag{4.1}$$

Where CHR is the Custom Hardware Result and SR is Software Result (Matlab). The E_t depends on the number of bits. The error increases when the number the bits is reduced. Besides, this error is propagated through all layers. Thus the bigger E_t is found in the last layer. For example, the E_t, for 22 bits, in the last layer was around 0.79%.

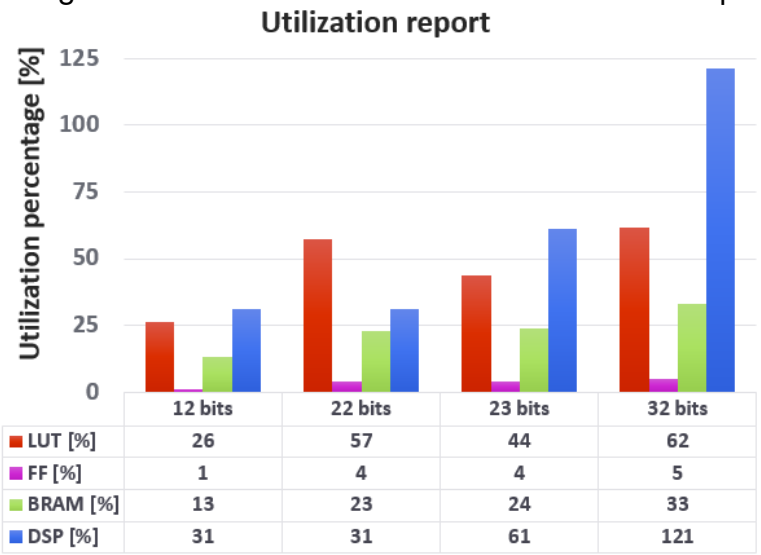
MOODY, G. B.; MARK, R. G. *A new method for detecting atrial fibrilltion using R-R invervals*. 1983.

Xilinx Inc. “Integrated Logic Analyzer v6.2 - LogiCORE IP Product Guide”. In: (2016), p. 31.

4.1. HARDWARE RESOURCE UTILIZATION

Figure 7 shows the percentages (with respect to the total available in the FPGA) of resource utilization for a different number of bits. It can be observed that between 12 and 22 bits there is no change in the percentage of DSP utilization.

Figure 7: Percentage of utilization for different amounts of bits of quantization



4.2. ACCURACY REGARDING THE NUMBER OF BITS

A test was performed using the set of 1000 ECG signals, which 500 corresponds to *Fibrillation signals* and the other 500 with *Not fibrillated signals*. Table 2 summarizes the results.

Table 2: Accuracy on the inferences process for 12 and 22 quantization bits

Bit quantity	Accuracy
12-bits	88 [%]
22-bits	94 [%]

Note that for 12 bits there is an important reduction in the accuracy, which is due to the truncation error. Taking into account the accuracy and the amount of resources required, a 22-bits quantization is adopted.

4.3. PERFORMANCE

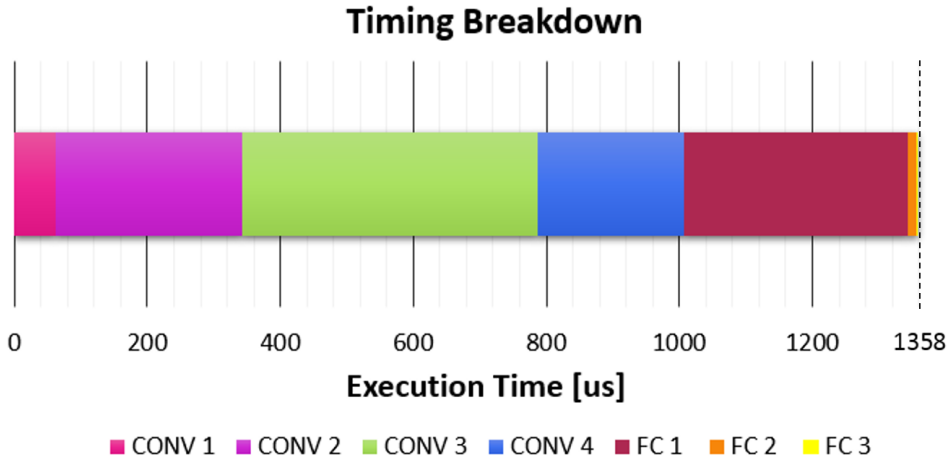
A clock frequency of 34.6 [KHz] is implemented that satisfies the design application that requires a throughput of an inference every two seconds. Table 3 summarizes the main results to obtain maximum performance on FPGA.

Table 3: Feature performance summary

Feature performance	
Throughput required	0.5 [inferences/s]
Maximum clock frequency (MCF)	25,5 [MHz]
Latency at MCF	1,358 [ms]
Throughput at MCF	736 [inferences/s]

Figure 8 shows the breakdown of the execution time to calculate each CNN layer. Note that convolution operations are the ones that consume the most time, therefore, if better latency is required, parallelism techniques can be applied.

Figure 8: Timing breakdown for the inference process



5. CONCLUSIONS

In this work, a computational architecture was proposed to carry out the inference process of a 22-bits Q-CNN, which allows the detection of Atrial Fibrillation. The design is based on a systolic array processor which is optimized for both convolutional and FC layers, which allows the reduction in the amount of computational and memory resources. The design has a throughput of an inference every two seconds, i.e. it works at 34.6 [KHz], However, the design could achieve a throughput of 736 [inferences/s] at its is maximum design frequency (25.5[Mhz]). The tests show accuracy in the inference of 94%, which moves approximately 2,97% away from the inference in 64-bit software. We aim to use this design in future work, which will focus on the implementation of a Q-CNN-based portable device for automatic detection of AF.

BIBLIOGRAPHY

ACHARYA, U. Rajendra; FUJITA, Hamido; LIH, Oh Shu; HAGIWARA, Yuki; TAN, Jen Hong; ADAM, Muhammad. “Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network”. In: *Information Sciences* 405 (2017), pp. 81–90.

CASTILLO, Jeyson; GRANADOS, Yenny; FAJARDO, Carlos. “Patient-Specific Detection of Atrial Fibrillation in Segments of ECG Signals using Deep Neural Networks”. In: *Ciencia E Ingenieria Neogranadina* 30.1 (2020).

CHUNGH, Sumeet S et al. “Worldwide epidemiology of atrial fibrillation: a Global Burden of Disease 2010 Study”. In: *Circulation* 129.8 (2014), pp. 837–847.

GUO, Yunhui. “A survey on methods and theories of quantized neural networks”. In: *arXiv preprint arXiv:1808.04752* (2018).

LI, Huimin; FAN, Xitian; JIAO, Li; CAO, Wei; ZHOU, Xuegong; WANG, Lingli. “A high performance FPGA-based accelerator for large-scale convolutional neural networks”. In: *FPL 2016 - 26th International Conference on Field-Programmable Logic and Applications* (2016), pp. 1–9.

MA, Yufei; CAO, Yu; VRUDHULA, Sarma; SEO, Jae Sun. “Optimizing loop operation and dataflow in FPGA acceleration of deep convolutional neural networks”. In: *FPGA 2017 - Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (2017), pp. 45–54.

MOODY, G. B.; MARK, R. G. *A new method for detecting atrial fibrilltion using R-R invervals*. 1983.

POURBABAEE, Bahareh; ROSHTKHARI, Mehrsan Javan; KHORASANI, Khashayar. “Deep Convolutional Neural Networks and Learning ECG Features for Screening Paroxysmal Atrial Fibrillation Patients”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.12 (2018), pp. 2095–2104.

TISAN, A; ONIGA, S; MIC, D; BUCHMAN, A. “Digital Implementation of The Sigmoid Function for FPGA Circuits”. In: *Acta Technica Napocensis* 50.2 (2009), pp. 15–20.

URIBE, William; DUQUE, Mauricio; MEDINA, Eduardo. “Electrocardiografıa y arritmias”. In: *Clınica Medellın. Editorial PLA Export Bogotá DC* (2005), pp. 41–5.

WANG, Erwei; DAVIS, James J; ZHAO, Ruizhe; NG, Ho-Cheung; NIU, Xinyu; LUK, Wayne; CHEUNG, Peter YK; CONSTANTINIDES, George A. “Deep Neural Network Approximation for Custom Hardware: Where We’ve Been, Where We’re Going”. In: *ACM Computing Surveys (CSUR)* 52.2 (2019), pp. 1–39.

WEI, Xuechao; YU, Cody Hao; ZHANG, Peng; CHEN, Youxiang; WANG, Yuxin; HU, Han; LIANG, Yun; CONG, Jason. “Automated Systolic Array Architecture Synthesis for High Throughput CNN Inference on FPGAs”. In: *Proceedings - Design Automation Conference* Part 12828 (2017), pp. 3–5.

XIA, Yong; WULAN, Naren; WANG, Kuanquan; ZHANG, Henggui. “Detecting atrial fibrillation by deep convolutional neural networks”. In: *Computers in Biology and Medicine* 93.December 2017 (2018), pp. 84–92.

XILINX INC. “Integrated Logic Analyzer v6.2 - LogiCORE IP Product Guide”. In: (2016), p. 31.

YAO, Zhenjie; ZHU, Zhiyong; CHEN, Yixin. “Atrial fibrillation detection by multi-scale convolutional neural networks”. In: *20th International Conference on Information Fusion, Fusion 2017 - Proceedings* (2017).

YU, Jincheng; HU, Yiming; NING, Xuefei; QIU, Jiantao; GUO, Kaiyuan; WANG, Yu; YANG, Huazhong. “Instruction driven cross-layer CNN accelerator with winograd transformation on FPGA”. In: *2017 International Conference on Field-Programmable Technology, ICFPT 2017* 2018-Janua (2018), pp. 227–230.