Performance analysis of a convolutional layer of a neural network synthesized on a FPGA using high-level tools

Adrián Rodríguez, Sebastián López, Yubal Barrios and Romén Neris
Instituto Universitario de Microelectrónica Aplicada
University of Las Palmas de Gran Canaria
Las Palmas, Gran Canaria
Email: {armolina, seblopez, ybarrios, rneris}@iuma.ulpgc.es

***Abstract*—When implementing on-board processing algorithms on satellites, it is common to use an FPGA as a platform, due to the performance they offer and the fact that some of them are certified as radiation tolerant. In this Master's Thesis, a High-Level Synthesis tool, Vitis HLS, has been used to analyze how the variation of the input parameters of one of the layers of the Mobilenet convolutional neural network affects the resources consumption and latency of the FPGA implementation. The starting point of this work is a layer code written in Python. This code was transcribed to generate two layer structures in C++, in order to obtain a hardware-friendly description, compatible with HLS tools. After modifying the input parameters of the layer, different configurations were obtained. From these results, it is concluded that a smaller input image size produces significantly better results in terms of latency and also the use of fixed point arithmetic operations significantly reduces resources consumption.**

***Index Terms*—High-Level Synthesis, FPGA, convolutional neural network, on-board video processing.**

I. Introduction

In the field of Remote Sensing, one of the most widely used devices to embark Earth observation sensors are orbital satellites. Data obtained by these devices have to be either transmitted from space to Earth to be processed or compressed to achieve the bandwidth requirements of space-to-Earth communication channel. To avoid this compression or to reduce the amount of data sent, on-board processing systems have to be developed.

This Master's Thesis is conducted within the VIDEO project [1] (Video Imaging Demonstrator for Earth Observation), that has received funding from the European Union's Horizon 2020 research and innovation program. One of the main goals of this project is to develop a highly-disruptive technology for an instrument that could be used for the video observation of Earth and the processing chain that will manage the data obtained by the instrument sensor.

The objective of this Thesis is to analyze the performance and resources consumption of the different configurations of the convolutional layer of a convolutional neural network that will be implemented on a FPGA. This implementation will be part of the processing chain in the VIDEO project. The different configurations have been generated by varying the input parameters of the convolutional layer. Synthesis results have been obtained using the Xilinx High-Level Synthesis tool, Vitis HLS.

II. Field Programmable Gate Arrays (FPGA)

An FPGA is a device whose internal logic can be programmed. Therefore, it is an integrated circuit whose internal connections can be modified to perform a specific task [2]. The performance that can be achieved by an FPGA device is better than the one obtained by other devices, like CPUs or GPUs, with lower power consumption. The main advantage of an FPGA is its flexibility to adapt the implemented functionality to new requirements that can appear during the system lifetime, without incurring in additional development costs. This is specially remarkable on space missions where, although there are Radiation Tolerant FPGAs specifically designed for working on space environment, radiation can produce a malfunction that can be solved thanks to FPGA reconfiguration capabilities.

This design has been synthesized using a Xilinx Kintex Ultrascale FPGA, whose part name is FPGA XCKU040-2FFVA1156E. This part is installed in a evaluation platform of Xilinx for debug purposes, called Kintex UltraScale FPGA KCU105 Evaluation Kit.

III. Convolutional Neural Network (CNN)

An artificial neural network is a processing chain formed by different nodes that try to simulate human neurons and perform a given activity in the computational process. The difference between different neural networks lies in the characteristics of these nodes, which are also known as layers.

A convolutional neural network, also known as CNN, is a type of neural network with specific layers, called convolutional layers. These type of neural networks offer great results in those activities in which the input data are images such as object detection or face emotion recognition [3].

A. Convolutional Layer

In a CNN there can be found one or more convolutional layers, which structure scheme is shown in figure 1. These

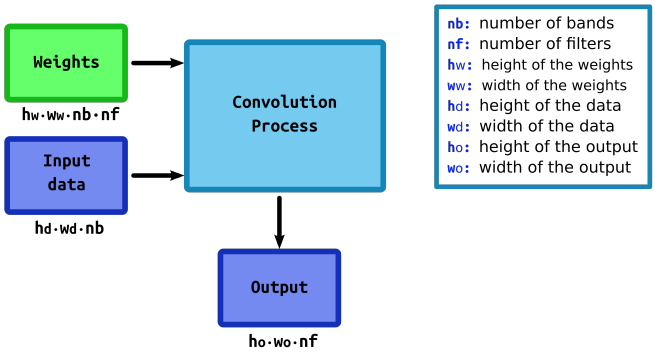

Figure 1: Structure of the convolutional layer

are the most significant layers of the network in which the relevant elements of the input data are highlighted using the weights obtained in the training of the network. The input parameters of a convolutional layer are:

- **Input Data:** the data that is going to be processed by the network, usually an image.
- **Padding:** this parameter indicates if the shape of the image will be modified by adding rows and columns of zeros around it.
- **Weights:** trainable parameters that operate on the input and allow to decide if a certain area of the image is relevant or not.
- **Strides:** the amount of elements the weights are being shifted on the image.

B. Mobilenet

In this work the CNN architecture that will be implemented on the FPGA is called Mobilenet [4]. The first layer of this network has been analyzed in this Master's Thesis. Using the Python code of this layer, two similar C++ layer structures have been developed: one named *line*, which process each line as it is received; and the other as *block*, which stores first as many lines as rows have the weights and then processes the entire memory block.

IV. Vitis HLS Synthesis

Vitis HLS is a High-Level Synthesis tool developed by Xilinx. This tool allows the user to generate a timing and resources consumption estimation of a hardware model wrote in a compatible language, such as C/C++. Using a Python script, all the required files to synthesize the two developed layers with different input parameters have been generated. These parameters are shown in table I. In addition, the data-type of both layers in the solution 5 and 6 has been evaluated using both floating-point and fixed-point precision, using a length of 32 bits and 5 bits for the integer value.

V. Results

By using Vitis HLS, results of latency and resources consumption have been obtained. In table II the best and the worst results that have been obtained with the solution and layer used

Table I: Input parameters tested in this work and its labels

Label	Image Size	Weights Size	Strides	Padding
1 - 2	128·128·3	3·3·3·8	2·2	No - Yes
3 - 4	256·256·3	3·3·3·8	2·2	No - Yes
5 - 6	512·512·3	3·3·3·8	2·2	No - Yes
7	512·512·3	4·4·3·8	2·2	No
8	512·512·3	7·7·3·8	1·1	No
9	512·512·3	7·7·3·8	2·2	No
10	512·512·3	7·7·3·8	3·3	No
11	512·512·3	7·7·3·8	4·4	No

are summarized. Each change in the image size increases the execution time by a factor of 4. After this, the largest increase recorded was between solution 8 and 9 in the *line* layer, whose execution time increases by a factor of 3.54.

Table II: Most relevant synthesis results

Resource		Value	Layer	Solution
Execution Time	Min	3.877 ms	*line*	1
	Max	1454.348 ms	*line*	8
BRAM	Min	13	*block*	1
	Max	38	*line*/*block*	8
DSP	Min	25	*line*/*block*	1 - 6
	Max	72	*line* (fixed point)	5 - 6
FFs	Min	899	*block* (fixed point)	5
	Max	6159	*line*	10
LUTs	Min	2270	*block* (fixed point)	6
	Max	6099	*line*	10

Comparing both generated layers, the *line* layer has better latency results, achieving as much as 43.24% in solution 6, while the *block* layer consumes less resources overall, reaching the less resources consumption in the solution 5 and 6 using fixed-point arithmetic precision, as is shown in figure 2.

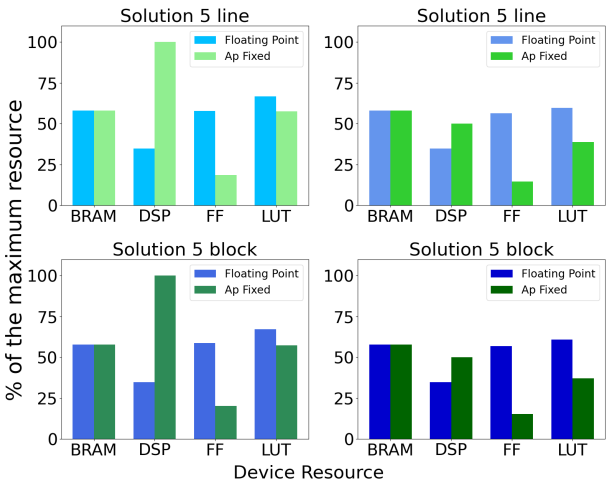

Figure 2: Comparison between solution with floating point and fixed point.

VI. Conclusions

In this Master's Thesis, several configurations of a convolutional layer have been synthesized in order to find out which parameters of the layer have the greatest impact on the resources consumption and latency of the implementation on the target FPGA. The obtained results have shown that the size of the input image has the biggest impact on the latency of the solution. On the other hand, changing the data-type of the layer from floating-point to fixed-point precision seems to reduce the logic consumption of the layer, at the expense of a penalty in the layer results around the 0.00001%.

References

[1] C. EU, *Video imaging demonstrator for earth observation*, access date: 03-03-2021, 2019. [Online]. Available: https://cordis.europa.eu/project/id/870485.

[2] H. Amano, *Principles and Structures of FPGAs*. Springer, 2018.

[3] A. Verma, P. Singh, and J. S. Rani Alex, "Modified convolutional neural network architecture analysis for facial emotion recognition," in *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2019, pp. 169–173. DOI: 10.1109/IWSSIP.2019.8787215.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.