Research, Society and Development, v. 10, n. 12, e83101219181, 2021
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v10i12.19181

Pattern recognition on FPGA for aerospace applications

Reconhecimento de padrões em FPGA para aplicações aeroespaciais

Reconocimiento de patrones en FPGA para aplicaciones aeroespaciales

Received: 08/07/2021 | Reviewed: 08/15/2021 | Accept: 09/13/2021 | Published: 09/14/2021

Francisco de Assis Tavares Ferreira da Silva
ORCID: https://orcid.org/0000-0002-9958-8339
National Institute for Space Research, Brazil
E-mail: francisco.silva@inpe.br
Magno Prudêncio de Almeida Filho
ORCID: https://orcid.org/0000-0001-8018-429X
Federal University of Ceara, Brazil
E-mail: magnoprudencio@gmail.com
Antonio Macilio Pereira de Lucena
ORCID: https://orcid.org/0000-0003-1671-334X
National Institute for Space Research, Brazil
Fortaleza University, Brazil
E-mail: antonio.lucena@inpe.br
Alexandre Guirland Nowosad
ORCID: https://orcid.org/0000-0002-8687-3501
National Institute for Space Research, Brazil
E-mail: alexandre.nowosad@inpe.br

Abstract
This paper presents a low power near real-time pattern recognition technique based on Mathematical Morphology-MM implemented on FPGA (Field Programmable Gate Array). The key to the success of this approach concerns the advantages of machine learning paradigm applied to the translation invariant template-matching operators from MM. The paper shows that compositions of simple elementary operators from Mathematical Morphology based on ELUTs (Elementary Look-Up Tables) are very suitable to embed in FPGA hardware. The paper also shows the development techniques regarding all mathematical modeling for computer simulation and system generating models applied for hardware implementation using FPGA chip. In general, image processing on FPGAs requires low-level description of desired operations through Hardware Description Language-HDL, which uses high complexity to describe image operations at pixel level. However, this work presents a reconfiguring pattern recognition device implemented directly in FPGA from mathematical modeling simulation under Matlab/Simulink/System Generator environment. This strategy has reduced the hardware development complexity. The device will be useful mainly when applied on remote sensing tasks for aerospace missions using passive or active sensors.
Keywords: Intelligent satellites; Nanosatellites; Artificial intelligence in hardware; Computer vision; Machine learning; Mathematical morphology; Pattern recognition; Real time systems; Aerospace applications; Remote sensing.

Resumo
Esse trabalho apresenta uma técnica de reconhecimento de padrões baseada em Morfologia Matemática-MM, implementada em FPGA (Field Programmable Gate Array). A estratégia para o êxito dessa abordagem consiste na utilização das vantagens do paradigma de aprendizagem de máquina aplicado em operadores morfológicos de casamento de padrões invariantes à translação. Esse artigo mostra que a composição de simples operadores elementares da MM baseados em ELUTS (Elementary Look-Up Tables) são adequados para aplicações embarcadas em FPGA. Esse artigo também mostra as técnicas de desenvolvimento do sistema de reconhecimento de padrões, desde a modelagem matemática dos operadores morfológicos até a implementação do dispositivo eletrônico usando o software System Generator. Em geral, as operações para o processamento de imagens em FPGAs são implementadas em baixo nível de abstração das linguagens de descrição de hardware-HDL. Isto gera alta complexidade na implementação de operações em imagens ao nível de pixel. No entanto, esse trabalho apresenta um dispositivo reconfigurável aplicado ao reconhecimento de padrões implementado em FPGA, a partir da simulação da modelagem matemática usando o ambiente de software Matlab/Simulink/System Generator. Essa estratégia reduz a complexidade do desenvolvimento em hardware. O dispositivo apresentado deverá ser útil principalmente quando aplicado em tarefas de sensoriamento remoto para missões aeroespaciais através de sensores passivos ou ativos.
Palavras-chave: Satélites inteligentes; Nanosatélites; Inteligência artificial em hardware; Visão computacional; Aprendizagem de máquina; Morfologia matemática; Reconhecimento de padrões; Inteligência artificial em tempo real; Aplicações aeroespaciais; Sensoriamento remoto.

Resumen

El presente trabajo presenta una técnica de reconocimiento de patrones en tiempo real basada en Morfología Matemática-MM implementada en FPGA (Field Programmable Gate Array). La estrategia para la efectividad de este enfoque tiene que ver con las ventajas del paradigma de aprendizaje automática aplicada al modelo de correspondencia con la invariancia traslacional de operadores elementales da MM. El artículo muestra que las composiciones de operadores elementales simples de morfología matemática basadas en ELUT (tablas de consulta elementales) son adecuadas para integrarse en dispositivos FPGA. Este artículo también muestra técnicas de desarrollo de sistemas de reconocimiento de patrones, desde el modelado matemático de operadores morfológicos hasta la implementación del dispositivo electrónico utilizando el software System Generator. En general, las operaciones para el procesamiento de imágenes en FPGAs se implementan a un bajo nivel de abstracción de los lenguajes de descripción del hardware-HDL. Esto crea una gran complejidad en la implementación de operaciones en imágenes a nivel de píxeles. Sin embargo, este trabajo presenta un dispositivo reconfigurable de reconocimiento de patrones implementado directamente en FPGA a partir de simulación de modelado matemático en el software Matlab/Simulink/System Generator. Esta estrategia ha reducido la complejidad del desarrollo de hardware. El dispositivo será útil principalmente cuando se aplique en tareas de teledetección para misiones aeroespaciales utilizando sensores pasivos o activos.

Palabras clave: Satélites inteligentes; Nanosatélites; Inteligencia artificial en hardware; Visión artificial; Aprendizaje automática; Morfología matemática; Reconocimiento de patrones; Inteligencia artificial en tiempo real; Aplicaciones aeroespaciales; Detección remota.

1. Introduction

Space missions for deep space exploration or remote sensing using small satellites faces major difficulties regards energy consumption and communication downlinks for big volumes of data. These difficulties imposes severe restrictions regards automatic decisions on board satellites (Silva & Lucena, 2005a; Silva & Lucena, 2005b; Giuffrida et al., 2020; Rapuano et al., 2021; Furano et al., 2020; Nagel et al., 2020), mainly if they demand digital signal processing from high spatial and spectral resolution sensors (Silva & Lucena, 2005a; Silva & Lucena, 2005b; Giuffrida et al., 2020; Zhou & Kafatos, 2002; Felipe et al., 2006; Thompson et al., 2012; Fuchs et al., 2014; Chien et al., 2014).

One of the most promising proposals to overcome these difficulties is the development of new low power embedded intelligent devices that can provide smart compression method for images and/or that can recognize patterns near real-time using low power consumption on the spacecraft. These intelligent devices also are great promise for more efficiently use regards high-resolution sensors with the restrictions of size, energy consumption and communication bandwidth in space links (Rapuano et al., 2021; Furano et al., 2020; Giuffrida et al., 2020; Felipe et al., 2006; Dawood et al., 2002; Chien et al., 2004; Silva & Lucena, 2005a; Silva & Lucena, 2005b).

However embedded or standalone real-time image processing systems that require low power are not easily implemented using classical processors and sequential processing paradigm (Rapuano et al., 2021; Furano et al., 2020; Giuffrida et al., 2020; Arechiga et al., 2018; Brugger et al. 2015; Astua et al., 2014; Blake et al., 2009; Kalomiros & Lygouras, 2008; Akil & Zahirazami, 1998). Related works from specialized literature (Rapuano et al., 2021; Dinelli et al., 2020; Guo et al., 2018; Arechiga et al., 2018; Hentati et al., 2014; Hagiwara et al., 2011; Pell et al., 2013; Zakerhaghighi & Naji, 2013; Bekker et al., 2010; Kalomiros & Lygouras, 2008; Johnston et al., 2004) tried several image processing implementations using FPGA (Field Programmable Gate Array) to build low-power systems for real-time applications.

In the specialized literature there are also other important papers with contributions from Mathematical Morphology-MM and machine learning hybridization techniques (Nogueira et al., 2021; Franchi et al., 2020; Jouni et al., 2020; Shen et al., 2019; Mellouli et al., 2019; Hao et al., 2019). However, none of these works has implemented in hardware any morphological operators based on Elementary Look-Up Tables–ELUTs paradigm as proposed in Silva (1998) and summarized at Section 2.

Besides most of these devices implemented by FPGA hardware have used RTL schematics (Xilinx-RTL, 2011), Verilog or VHDL programing methodology to generate HDL codes (D'AMORE, 2005) as entry method to build configuration

files (bitstream) for hardware definition. The classical methodology used in these previous works requires hard effort to build scripts for hardware description in place of the use of functional models for description of tasks.

On the other hand, a signal-processing paradigm based on Look-Up Tables-LUTs is advantageous over FPGA architecture, considering LUTs are common blocks found in all modern FPGA chips. Due to their simplicity in terms of mathematical operations, in this work morphological operators for pattern recognition tasks as proposed by Silva (1998) have resulted in a first FPGA hardware device based in elementary operators from MM using LUTs paradigm applied to near real time patterns recognition tasks.

These operators are the mathematical basis to build adaptive controlled threshold template matching (Silva, 1998) without traditional convolution operations. These operators allowed the development of several different applications as described in previous works (Silva & Banon, 1999; Rempel & Silva, 2001; Silva & Silva, 2004; Souza et al., 2012; Souza et al., 2013; Filho et al., 2014; Filho et al., 2015). The present work proposes a new standalone device for near real time pattern recognition based on Mathematical Morphology-MM operators defined by ELUTs, taking advantages from an adaptive engine applied to morphological operators adapted from previous works (Silva, 1998; Silva, 2006; Filho et al., 2014). In this work, the device performs near real-time signal processing in pattern recognition on satellite images.

These morphological operators are part of the formalism of gray levels MM developed using ELUTs (Heijmans, 1991; Banon, 1995; Khosravi & Schafer, 1996; Banon & Faria, 1997; Banon, 2000; Banon, 1995; Silva, 1998). In this work, the hybridization of the MM and machine learning is a combination of the scheme proposed by Silva (1998) and Silva (2006) added to a learning engine adapted from Filho et al. (2014) presented at Sections 2 and 3.

Before the implementation in hardware, the mathematical model has been exhaustively tested on Matlab/Simulink (Matlab-Simulink, 2015) environment in different applications. The main results were show up in previous pattern recognition works (Silva, 1998; Silva & Banon, 1999; Silva & Silva, 2004; Filho et al., 2014; Filho et al., 2015) and in image/video compression tasks (Souza et al., 2012; Souza et al., 2013). However the present work shows the first hardware implementation of the morphological operators based on ELUTs using templates generated using machine learning scheme adapted from Silva (1998) and Silva (2006) and implemented on simulated form in previous works (Filho et al., 2014; Filho et al., 2015).

The main idea of this new approach is to use a training set to obtain a representative pattern for template matching with digital images and this representative pattern, implemented in FPGA hardware. In this work, the training set consisted of elements of the same target with small imaging variations like different remote sensing satellites imaging from revisiting of the same ROI (Region of interest). This strategy enables the morphological operators to use previously trained templates to perform exact or inexact and controlled template matching tasks, which are relevant for the hit or miss detection of patterns in real cases on satellite images as developed in Section 3.

The major contributions of this work are a simple and low power FPGA embedded pattern recognition scheme from simulated mathematical modeling for hardware design. The work shows a new hardware implementation of morphological operators based on ELUTs and the first results. The implementation methodology, presented at Section 2, is a very successful adaptation of the techniques developed in Silva et al. (2015). The device can do near real time pattern recognition for exact matching or inexact matching defined by similarity levels parameters. The processing scheme shows details for each image transformation and respective partial results of output images on each of the main FPGA blocks. The work also presents hardware specifications of the device, power consumption and performance during a pattern recognition task in a satellite image. Therefore, it can also be useful as reference to define different FPGA chips applied to similar tasks.

Section 2 presents the functional design of fixed-point codification of the patterns recognition morphological operators in hardware. It shows the functional blocks details to build MM elementary operators based on ELUTs embedded in FPGA using System Generator (Xilinx-SG, 2014). Details of the mathematical modeling methodology on FPGA using System

Generator (Xilinx-SG, 2014) and Matlab/Simulink environment (Matlab-Simulink, 2015) are in Silva et al. (2015). Section 3 shows the results for the device processing a Landsat satellite image with resources and performance from Xilinx Kintex 7 FPGA (XILINX-KC705, 2014). Final considerations are at Section 4.

2. Methodology

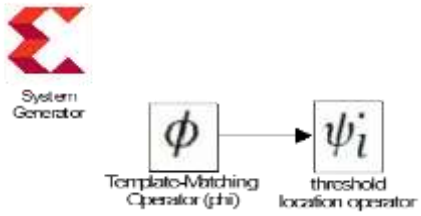
Before developing the device presented in this article, the authors followed steps:

a) Analysis of performance and energy consumption for the near real time pattern recognition tasks applied to aerospace missions, using the results of the previous studies and the references quoted at Introduction Section;
b) Study of low-power devices with potential to build near real-time pattern recognition artifacts, using the results of the references quoted at Introduction Section;
c) Research on robust mathematical modeling for pattern recognition hardware device implementations, using the results of the references quoted at Introduction Section;
d) Research on hardware devices suitable to meet the near real-time image processing requirements from (a) and (b), using the results of the references quoted at Introduction Section;
e) Research on techniques for implementing of the mathematical formalism (Silva, 1998) in FPGA hardware (Silva et al., 2015); and
f) Design of a low power device for near real time pattern recognition tasks in digital images, subject of this work.

However, the focus of the current work are items (e) and (f), concerning the techniques for mathematical modeling implementation of morphological operators applied to near real time pattern recognition embedded in FPGA. As seen in Silva et al. (2015) the System Generator (Xilinx-SG, 2014) provides functional mathematical modeling and automatic FPGA code generation from Matlab/Simulink environment (Matlab-Simulink, 2015). Therefore, the techniques presented in Silva et al. (2015) were adapted for implementation of the morphological operators on FPGA. This approach permits design of a complete digital system like a functional scheme from mathematical formalisms implemented on System Generator as shown in the Figures 1, 2, 3, 4, 5, 6 and 11. In a second stage, this functional scheme automatically generates a configuration script as a bitstream file. Finally, the bitstream in the form of uploaded codes configures the Kintex 7 (KC705) hardware from Xilinx/Avnet (XILINX-KC705, 2014).

Figure 1 shows the developing scheme of morphological operator composition using System Generator (Xilinx-SG, 2014) running under Matlab/Simulink environment (Matlab-Simulink, 2015). This same configuration generates the final bitstream uploaded into FPGA.

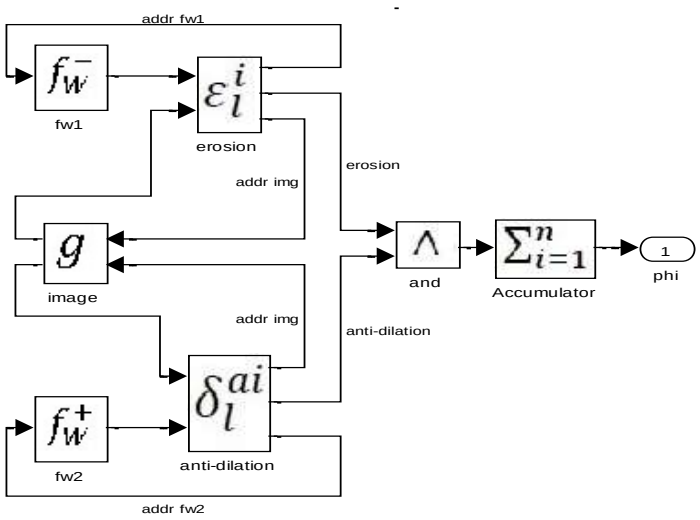
Figure 1. The composition of the operators (Silva, 1998), from equations (6) and (7), on System Generator environment.



Source: Authors.

The ϕ operator (equation (6)) executes the summation of the intersections between erosion and anti-dilation operations, as shown in Figure 2.

Figure 2. System Generator scheme to make ϕ.



Source: Authors.

In this work mathematical representations are used with gray levels images, for domain D (pixels positions set) and K_m scale gray levels. The set of mappings from D to K_m is denoted as K_m^D. Therefore, the digital images in D domain are represented by K_m^D. In the specific case K_m^W denotes the set of mappings from a window W to K_m (Banon, 1995; Silva, 1998). If $f \in K_m^D$, and m is equal to 1, so f is a binary image, otherwise, f is a gray level image.

In Figure 2, the blocks "fw1", "fw2" and "image" are memory buffers that store the "g" image, and the f_W^- and f_W^+ patterns are obtained using equations (1) and (2) presented below.

$$f_W^-(x) \triangleq max(0, min\ (m,\ f_W(x) + c_1)), \quad (1)$$

$$f_W^+(x) \triangleq max(0, min\ (m,\ f_W(x) + c_2)), \quad (2)$$

where $x \in W$, $f_W \in K_m^W$ (representative patterns), $c_1, c_2 \in \boldsymbol{Z}$ (the integer numbers set), $c_1 \leq c_2$, and $f_W^-, f_W^+ \in K_m^W$ define two slack images. These two slack images define the intervals around the f_W pattern to perform an inexact matching. Of course if $c_1 = c_2 = 0$ it will perform an exact matching condition.

The blocks "erosion" and "anti-dilation", Figures 3 and 4, control access to buffers in which the images are stored and used in erosion and anti-dilation operations (equations (3) and (4)).

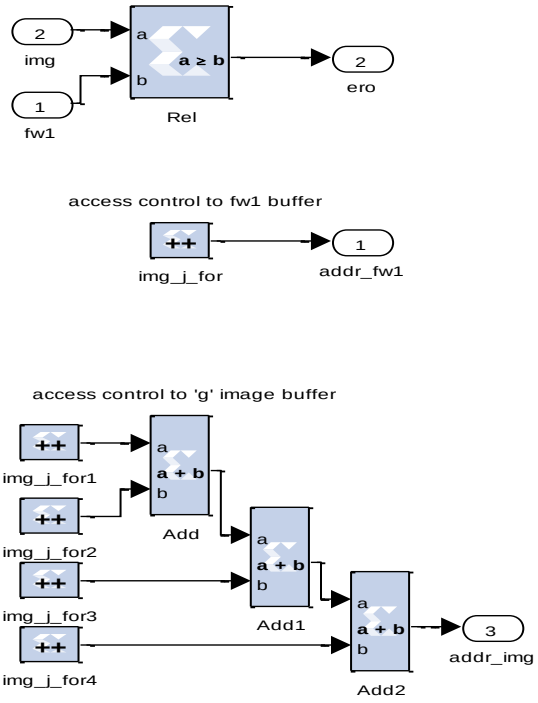
$$\varepsilon_l^i(g)(x) \triangleq \begin{cases} 1, & if\ g(x + W_i) \geq l \\ 0, & otherwise, \end{cases} \quad (3)$$

$$\delta_l^{ai}(g)(x) \triangleq \begin{cases} 1, & if\ g(x + W_i) \leq l \\ 0, & otherwise, \end{cases} \quad (4)$$

where l is an integer number in K_m and $i = 1, \ldots, n$, and $n = \#W$ (number of pixels of W), $g \in K_m^D$ and $x \in E$, for $E = D \ominus W$ ($\ominus$ represents a Minkowski subtraction (Banon & Barrera, 1998)). So the operator ε_l^i is an erosion and the δ_l^{ai} is an anti-dilation from K_m^D to K_1^E.

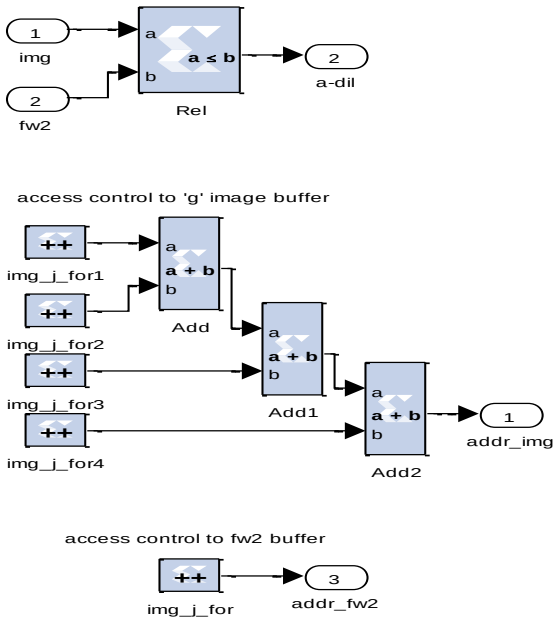
Figures 3 and 4 show the respective block implementations of the equations (3) and (4).

Figure 3. "Erosion" block.



Source: Authors.

Figure 4. "Anti-dilation" block.



Source: Authors.

The previous outputs, from erosion and anti-dilation blocks, are processed on "Accumulator" block (Figure 5), using equations (5) and (6).

$$\lambda^i \triangleq \varepsilon^i_{f^-_{W(w_i)}} \wedge \delta^{ai}_{f^+_{W(w_i)}}, \qquad (5)$$

where ε_l^i is an erosion and the δ_l^{ai} is an anti-dilation from $K\,{}_m^D$ to $K\,{}_1^E$, f_W^- and $f_W^+ \in K_m^W$, from equations (1), (2), (3) and (4).

The result from equation (5), for each pixel, is accumulated in "Accumulator" block (Figure 5) using the equation (6). This Accumulator block performs the sum ϕ of sup-generating morphological operators λ^i (Banon & Faria, 1997).

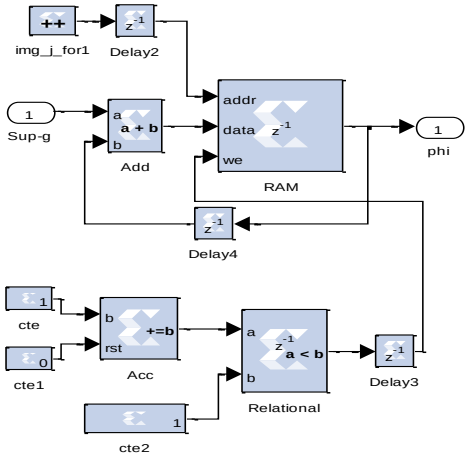
The ϕ operator from $K\,{}_m^D$ to $K\,{}_1^E$ is a Template-Matching Operator-TMO (Banon & Faria, 1997) as summarized below:

$$\phi \triangleq \sum_{i=1}^{n} \lambda^i, \tag{6}$$

where the λ^i are n operators from $K\,{}_m^D$ to $K\,{}_1^E$.

The result of the sup-generation operation for each pixel accumulated in "Accumulator" block executes the sum of sup-generating morphological operators and executes a kind of correlation measure. Figure 5 shows the "Accumulator" block implementation.

Figure 5. "Accumulator" block.



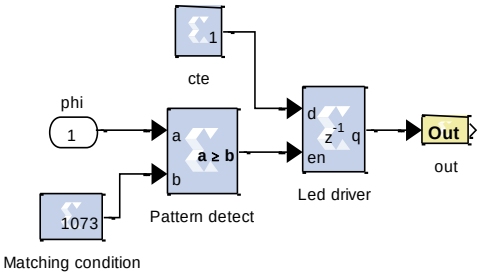
Source: Authors.

Finally, Figure 6 shows how the $\psi_l^{\bullet}$ operator (equation (7)) finds the predefined similarity "*l*" greater than or equal to matching condition.

$$\psi_l^{\bullet}(f) \triangleq \begin{cases} 1, & if\ \exists x \in E, f(x) \geq l \\ 0, & otherwise, \end{cases} \tag{7}$$

where $l \in K_m$ and f is the output result from ϕ operator. The operator $\psi_l^{\bullet}$ from $K\,{}_n^E$ to K_1 defines an especial erosion (Silva, 1998) that can identifies values greater than or equal to l (matching condition).

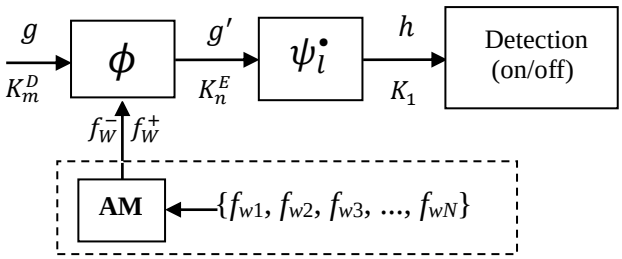
Figure 6. The System Generator composition for $\psi_i^{\bullet}$ operator (Silva, 1998) identifies the equal or predefined similarity values for matched condition.



Source: Authors.

A machine-learning engine called Thresholded Template-Matching Operator (TTMO) adapted from Silva (1998), Silva (2006) and Filho et al. (2014) generates the operator templates. Figure 7 shows the TTMO block diagram similar to the one shown in Filho et al. (2014), where in Adaptive Machine-AM: $f_{Wj} \in K_m^W$ and $j = 1, \ldots, N$; in ϕ: $g \in K_m^D$, f_W^- and $f_W^+ \in K_m^W$; in $\psi_i^{\bullet}$: $g' \in K_n^E$ and $h \in K_1$. The AM processes the *N* variants of a pattern contained in the training set, ϕ processes the g image and the representative f_W^- and f_W^+ pattern from AM, and h is the output result from $\psi_i^{\bullet}$ presented on Detection block to indicate if the pattern f_W is present (on) or not (off) on image g (Figure 7).

Figure 7. Blocks diagram of TTMO.



Source: Adapted from Filho et al. (2014).

The dashed part of Figure 7 is part of the training process of the representative pattern. This learning scheme takes advantage of an AM, which uses a training rule shown in Filho et al. (2014), applied to the training dataset. The Adaptive Machine does the following:

Algorithm 1. Algorithm used by AM learning engine.

1. Choose a sample from the training set;
2. Apply the learning rule;
3. Repeat (1, 2) until a stop conditions is satisfied;
4. Return the representative pattern;
5. End.

Source: Adapted from Filho et al. (2014).

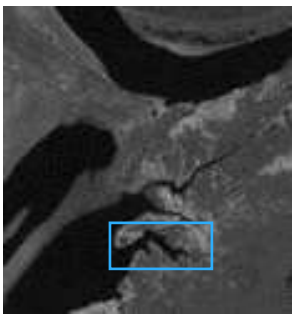
3. Results and Discussion

This section shows some results of preliminary experiments using the implemented FPGA device applied to pattern recognition in a Landsat satellite image. Landsat TM images, acquired free from Instituto Nacional de Pesquisas Espaciais – INPE (www.dgi.inpe.br).

For these first experiments, we used satellite images in ".jpg" format as shown in the Figure 8. Figure 8 shows a scene of Parintins-AM/Brazil, at the vicinity of Amazon River, located at Lat: -2.89303; Long: -57.47650.

The goal is to find in the 116 x 131 pixels image (Figure 8), a 48 x 26 pixels representative pattern (Figure 9), represented by the set f_W^- and f_W^+ target patterns, previously trained by the adaptive engine with the help of the training set. In this case, the training set consisted of models of pattern images selected from other passages of the satellite, different from those acquired in the passage shown in the Figure 8.

Figure 8. Satellite image ".jpg" format.



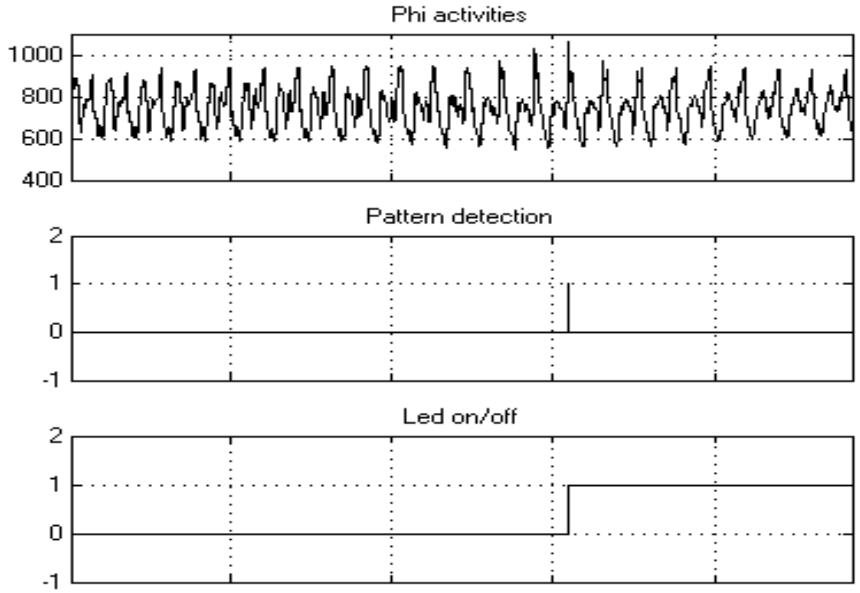
Source: Adapted from Landsat satellite image.

Figure 9. Representative pattern (extended as *zoom in* from original pattern) compound by twelve samples of the same scene from imaging using different LandSat revisits.

Source: Authors (from Algorithm 1).

Figure 10 shows the signals generated in ϕ and $\psi_l^\bullet$ Blocks. The first graph, named "Phi activities", shows the serialized values of the sum of sup-generating morphological operator. When one of these values is greater than the predefined matching conditions, it means that the pattern image is included in the search image, as verified on the second graph, "Pattern detection". The third graph "Led on/off" shows a signal that turn on/off one Led terminal from KC705 board (XILINX-KC705, 2014) to indicate to the user if the pattern is present or not. The level of similarity between the ROI, blue polygon in Figure 8, and the representative pattern in Figure 9, used in this case for definition of matching condition, was equal to 86%.

Figure 10. Signals generated in ϕ and $\psi_i^{\bullet}$ operators from hardware implementation.



Source: Authors (from FPGA device).

Table 1 shows a summary of the resource use and power consumption of the model implemented in an FPGA chip KC7K325T-2FFG900C Kintex 7 (XILINX-KC705, 2014). The first line shows the amount of registers used mainly for temporary memory for images, templates and intermediary values between functional blocks as buffers. The second line shows the amount of total LUTs used for the morphological operators, access control and relational logic used in the accumulator block.

Table 1. Resource utilization of system.

	Kintex 7 (KC705)
Number of Slice Registers	13064 from 407600 (3%)
Number of Slice LUTs	13140 from 203800 (6%)
Number of occupied Slices	3363 from 50950 (6%)
Inference power consumption	416 mW

Source: Authors (from System Generator-SG).

For all processing, it is necessary to perform a scan in the input image to search the target pattern. The amount of clock cycles (cycles in equation (8)) required for the entire process to be completed is:

$$cycles = (img_i - w_i + 1)\big(img_j - w_j + 1\big)(w_i w_i + 1), \quad (8)$$

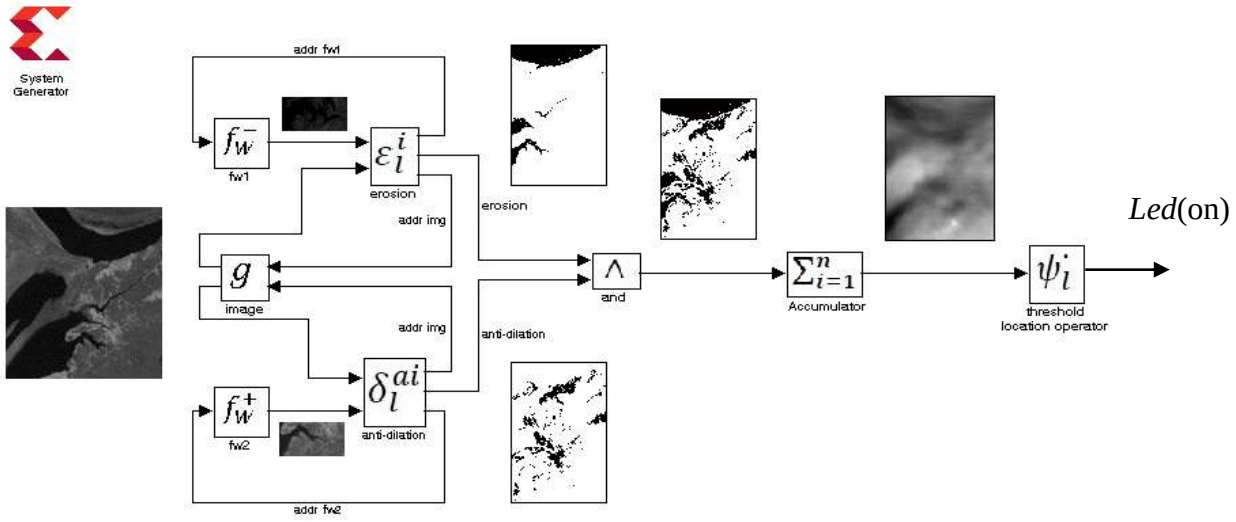
where img_i, img_j, w_i *e* w_j are the number of rows and columns of the image to be processed and of the pattern to be identified.

The time required (inference time in equation (9)) for the proposed image processing applied on input image (Figure 8) to detect the pattern in Figure 9, assuming 200 MHz FPGA clock, was approximately 45.68 *ms*, using equations (8) and (9):

$$inference\ time \cong \frac{cycles}{clock} \cong \frac{9135186}{200\ MHz} \cong 45{,}68\ ms \quad (9)$$

Figure 11 shows the processing details step by step for each image transformation and respective partial result with the output images in each of the main processing blocks.

Figure 11. The complete scheme of the device tasks with the respective outputs in each block.



Source: Authors.

4. Final Considerations

This work has shown a new hardware implementation for ELUTs based morphological operators developed for pattern recognition in future space missions. The final bitstream file generated directly from simulated mathematical modeling running on Matlab/Simulink environment (Matlab-Simulink, 2015) reduces the abstraction gap between the mathematical modeling and the finished hardware. The Matlab/Simulink environment to simulate the mathematical model and generates FPGA bitstreams using the System Generator toolbox (Xilinx-SG, 2014) worked efficiently, similar to the presented in Silva et al. (2015).

As shown in Section 3, the device took only 6% of the resources of the FPGA Kintex7 without any sophisticated feature as DSP blocks, i.e. as seen in Table 1; the device only demanded LUTs and registers, so the whole scheme also can run in an on-orbit reconfigurable radiation tolerant FPGA chip as XQRKU060 Kintex (XILINX-RTK, 2021).

It may be especially useful when applied on detection of environmental changes using small satellites like remote sensing nanosatellites, because this new high performance device can overcome the major limitation of size and low power restrictions from small satellite projects.

In addition to the tasks of pattern recognition proposed in this paper, we expect that the same operators implemented in FPGA can also be adapted for image compression tasks from previous works (Souza et al., 2012; Souza et al., 2013) in a hardware version. It may be an option applied to a future generation of remote sensing satellites, UAVs and deep-space crafts.

This first prototype version has prioritized the applicability of this mathematical modeling techniques and the bitstream generation directly from this simulated mathematical model applied to satellite imagens. The authors suggest a new version to be an option to solve template-matching tasks between previous aerial photography and satellite images and real time acquired images from unmanned aerial vehicle (UAV). The main idea is to take advantage of inexact template matching operators to find terrestrial references as waypoints obtained from previous aerial photography or satellite images. It can assist UAV navigation under GNSS jamming or GNSS failures. We hope this new device also may be a solution as embedded device to assist semi-automatic navigation tasks applied to UAVs missions as improvement to the ones shown in previous works (Castelli et al., 2016; Ramos et al., 2016; Lange et al., 2008; Chowdhary et al., 2013; Beul et al., 2015; Belmonte et al., 2019).

In this first hardware implementation, the work presented only a simple set of morphological operators as a single canonical artificial neuron model (Silva, 1998) when compared to a full deep learning paradigm as the morphological artificial neural networks shown in previous works. (Nogueira et al., 2021; Franchi et al., 2020; Jouni et al., 2020; Shen et al., 2019; Mellouli et al., 2019; Hao et al., 2019). Currently the authors are working on the parallel graph analysis to find main bottlenecks for each operator to increase image access and multiplicity of operators used in order to explore both spatial and temporal parallelism forms (Downton & Crookes, 1998; Johnston et al., 2004). The authors intend to build this new one approach toward a full parallel model suitable for deep learning paradigm. However differently from the previous works (Nogueira et al., 2021; Gianni et al., 2020; Shen et al., 2019; Mellouli et al., 2019; Hao et al., 2019; Jouni et al., 2020) the authors intend to build a new morphological neural networks based on ELUTs.

The full parallelism approach, implemented in hardware, may also be useful when embedded in the new generation of smart satellites for earth observation. However before trying a satellite embedded version, the authors intend it to perform efficient near-real time pattern recognition tasks in wildfire monitoring taking as reference the strategy presented in (Ban et al., 2020).

The device permits exact or inexact matching so it can be also useful for pattern recognition in digital signals from different kinds of sensors such as for example hyperspectral imaging sensors or active sensors as LIDAR, Radar, SAR or InSAR.

For future applications using this device, the authors suggest testing the device to detect changes on earth surface such as for example due to forests under wildfire or deforestation, soil and water contamination, land degradation, soil erosion and landslides.

For future research work, the authors suggest increasing the parallelism of the device to build new artificial neural networks models suitable for deep learning techniques or new paradigms of computer vision with a strong biological inspiration (Silva, 2005).

Acknowledgments

This work was supported in part by Agência Espacial Brasileira (AEB) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) through Edital MCT/CNPq/AEB nº 33/2010.

References

Akil, M., & Zahirazami, S. (1998). Multi-FPGA Processor for Gray Scale Morphology, European Association for Signal Processing, Eusipco-98, IX European Signal Processing Conference, Island of Rhodes, Greece, 8-11 September.

Arechiga, A. P., Michaels, A. J., & Black, J. T. (2018). "Onboard Image Processing for Small Satellites," NAECON 2018 - IEEE National Aerospace and Electronics Conference, 234-240, 10.1109/NAECON.2018.8556744.

Astua, C., Crespo, J., & Barber, R. (2014). Detecting Objects for Indoor Monitoring and Surveillance for Mobile Robots, Emerging Security Technologies (EST), in Fifth International Conference, 124-129.

Ban, Y., Zhang, P., Nascetti, A., Bevington, A. R., & Wulder M. A. (2020). Near Real-Time Wildfire Progression Monitoring with Sentinel-1 SAR Time Series and Deep Learning, Nature Scientific. 10(1322).

Banon, G. J. F., (1995). Characterization of translation invariant elementary morphological operators between gray-level images, Research report, INPE-5616-RPQ/671, INPE.

Banon, G. J. F., & Barrera, J. (1998). Bases da morfologia matemática para análise de imagens binárias, (2nd ed.), MCT/INPE, INPE-6779-RPQ/682.

Banon, G. J. F. (2000). Formal introduction to digital image processing. Technical Report: INPE-7682-PUD/43, São José dos Campos, Brasil., 2000. dpi.inpe.br/ banon/2001/01.11.16.04.

Banon, G. J. F., & Faria, S. D. (1997). Morphological approach for template matching, SIBGRAPI'97, IEEE Computer Society, 171-178.

Bekker, D. L., Werne, T. A., Wilson, T. O., Pingree, P.J., Dontchev, K., Heywood, M., Ramos, R., Freyberg, B., Saca, F., Gilchrist, B., Gallimore, A., & Cutler, J. (2010). A CubeSat design to validate the Virtex5 FPGA for spaceborne image processing, Aerospace Conference, 2010 IEEE, pp.1-9.

Belmonte L. M, Morales R., & Fernández-Caballero A. (2019). Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study. Applied Sciences. 2019, 9(15):3196. https://doi.org/10.3390/app9153196

Beul, M., Krombach, N., Zhong, Y., Droeschel, D., Nieuwenhuisen, M., & Behnke, S. (2015). A high-performance MAV for autonomous navigation in complex 3D environments. In Unmanned Aircraft Systems (ICUAS), International Conference on, Denver, USA, 1241-1250. IEEE.

Blake, G., Deslinski, R.G., & Mudge, T. (2009). A survey of multicore processors: A review of their common attributes, IEEE Signal Processing Magazine, vol. 26, no. 6, pp. 26–37.

Brugger, C., Dal'Aqua, L., Varela, J., De Schryver, C., When, N., Klein, M., & Siegrist, M. (2015). A Quantitative Cross-Architecture Study of Morphological Image Processing on CPUs, GPUs, and FPGAs. In Proceedings of the IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), April 2015.

Castelli, T., Sharghi, A., Harper, D., Tremeau, A., & Shah, M. (2016). Autonomous navigation for low-altitude UAVs in urban areas. arXiv preprint arXiv:1602.08141.

Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Lee, R., M, D., Frye, S. Trout, B. Hengemihle, J., Shulman, S., Ungar, S., & Brakke, T. (2004). The EO-1 autonomous science agent.

Chien, S., Bue, B., Castillo-Rogez, J., Gharibian, D., Knight, R., Schaffer, S., Thompson, D., & Wagstaff, K. (2014). Agile Science: Using Onboard Autonomy for Primitive Bodies and Deep Space Exploration.

Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., & Shein, A. (2013). GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. Journal of Field Robotics, 30(3), 415-438.

D'AMORE, R. (2005). VHDL: Descrição e Síntese de Circuitos Digitais. Editora LTC, 276 p.

Dawood, A., Visser, S., & Williams, J. (2002). Reconfigurable FPGAs for real time image processing in space. In Digital Signal Processing. IEEE 14th International Conference on, 2. 845–848.

Dinelli, G., Meoni, G., Rapuano, E., Pacini, T., & Fanucci, L. (2020). MEM-OPT: A Scheduling and Data Re-use System to Optimize On-chip Memory Usage for CNNs On-board FPGAs. IEEE J. Emerg. Sel. Top. Circuits Syst. 2020, 10, 335–347.

Downton, A., & Crookes, D. (1998). Parallel Architectures for Image Processing. Electronics & Communication Engineering Journal, 10, 139-151.

Felipe, I., Dohm, J. M., Baker, V. R., Doggett, T., Davies, A. G., Castano, R., Chien, S., Cichy, B., Greeley, R., Sherwood, R., Tran, D., & Rabideau, G. (2006). Flood detection and monitoring with the Autonomous Sciencecraft Experiment onboard EO-1: Remote Sensing of Environment, 101(4), 463-481.

Filho, A., M. P., Silva, F. A. T. F., & Braga, A. P. S. (2014). Machine learning and adaptive morphological operators. In: XI Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2014). São Carlos, Brasil. Annals of XI Encontro Nacional de Inteligência Artificial e Computacional.

Filho, A., M. P., Silva, F. A. T. F., & Braga, A. P. S. (2015). Operador morfológico adaptativo de casamento de padrões – proposta e aplicação na análise de imagens de satélite. In: Simpósio Brasileiro de Automação Inteligente (SBAI 2015). Natal, Brasil.

Franchi, G., Fehri, A., & Yao, A. (2020). Deep morphological networks, Pattern Recognition, Volume 102.

Fuchs, T., Bue, B., Rogez, J. C., Wagstaff, K., & Thompson, D. (2014). Autonomous Onboard Surface Feature Detection for Flyby Missions, International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), June 2014.

Furano, G., Meoni, G., Dunne, A., Moloney, D., Ferlet-Cavrois, V., Tavoularis, A., Byrne, J., Buckley, L., Psarakis, M., Voss, K. O., et al. (2020). Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities. IEEE Aerosp. Electron. Syst. Mag. 2020, 35, 44–56.

Giuffrida, G., Diana, L., de Gioia, F., Benelli, G., Meoni, G., Donati, M., & Fanucci, L. (2020). CloudScout: A Deep Neural Network for On-Board Cloud Detection on Hyperspectral Images. Remote. Sens. 2020, 12, 2205.

Guo, K., Sui, L., Qiu, J., Yu, J., Wang, J., Yao, S., Han, S., Wang, Y., & Yang, H. (2018). Angel-Eye: A Complete Design Flow for Mapping CNN Onto Embedded FPGA. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 2018, 37, 35–47

Hagiwara, H., Asami, K., & Komori, M. (2011). FPGA Implementation of Image Processing for Real-time Robot Vision System. Proc. of International Conference on Convergence and Hybrid Information Technology, pp.134-141, Daejeon, Korea, September.

Hao, R., Wang, X., Zhang, Liu, J. J., Du, X., & Liu, L. (2019). Automatic Detection of Fungi in Microscopic Leucorrhea Images Based on Convolutional Neural Network and Morphological Method. IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (2019): 2491-2494.

Heijmans, H. J. A. M. (1991). Theoretical Aspects of Gray-Level Morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, v.13, pp. 568-582.

Hentati, R., Hentati, M., Aoudni, Y., & Abid, M. (2014). The implementation of basic morphological operations on FPGA using partial reconfiguration. IEEE IPAS'14: INTERNATIONAL IMAGE PROCESSING APPLICATIONS AND SYSTEMS CONFERENCE 2014.

Johnston, C. T., Gribbon, K. T., & Bailey, D. G. (2004). Implementing Image Processing Algorithms on FPGAs. In: Proc. Electronics New Zealand Conference, Palmeston North, New Zealand, pp. 118-123.

Jouni, M., Mura, M. D., & Comon, P. (2020). Hyperspectral Image Classification Based on Mathematical Morphology and Tensor Decomposition, Mathematical Morphology - Theory and Applications, 4(1), 1-30.

Kalomiros, J. A., & Lygouras, J. (2008). Design and evaluation of a hardware/software FPGA based system for fast image processing, Microprocessors and Microsystems, pp. 95-106.

Khosravi, M., & Schafer, R. W. (1996). Template matching based on a grayscale hit-or-miss transform. Image Processing, IEEE Transactions on, 5(6), pp. 1060-1066.

Lange, S., Sünderhauf, N., & Protzel, P. (2008). Autonomous Landing for a Multirotor UAV Using Vision. In Workshop Proceedings of SIMPAR 2008 Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robts, Venice, Italy, 3-4, 482-491.

Matlab-Simulink, (2015). http://www.mathworks.com.

Mellouli, D., Hamdani, T. M., Sanchez-Medina, J. J., Ayed, M. B., & Alimi, A. (2019). Morphological Convolutional Neural Network Architecture for Digit Recognition. IEEE Transactions on Neural Networks and Learning Systems, 30, 2876-2885.

Nagel, G. W., de Novo, E. M. L. M., & Kampel, M. (2020). Nanosatellites applied to optical Earth observation: A review.Rev. Ambient. Água 2020, 15.

Nogueira, K. J. Chanussot, Mura, M. D., & Santos, J. A. D. (2021). "An Introduction to Deep Morphological Networks," in IEEE Access, 9, 114308-114324, 2021, 10.1109/ACCESS.2021.3104405.

Pell, O., Mencer, O., Tsoi, K.H., & Luk, W. (2013). Maximum performance computing with dataflow engines, in High-Performance Computing Using FPGAs, ed: Springer, pp. 747-774.

Ramos, R., A., Sampedro, C., Carrio, A., Bavle, H., Fernández, R. A. S., Miloševic, Z., & Campoy, P. (2016). A monocular pose estimation strategy for uav autonomous navigation in gnss-denied environments. In International micro air vechicle competition and conference 2016, Beijing, PR of China, pp. 22-27.

Rapuano, E., Meoni, G., Pacini, T., Dinelli, G., Furano, G., Giuffrida, G., & Fanucci, L. (2021). An FPGA-Based Hardware Accelerator for CNNs Inference on Board Satellites: Benchmarking with Myriad 2-Based Solution for the CloudScout Case Study. Remote Sens. 2021, 13, 1518. https://doi.org/10.3390/rs13081518.

Rempel, E. L., & Silva, F. A. T. F. (2001). Reconhecimento de Padrões Invariante a Rotação Utilizando uma Rede Neural Morfológica Não Supervisionada. In: V Brazilian Conference on Neural Networks, 2001, Rio de Janeiro. Proceedings of V Brazilian Conference on Neural Networks, 2001. 109-112.

Shen, Y., Zhong, X., & Shih, F. (2019). Deep Morphological Neural Networks. ArXiv, abs/1909.01532.

Silva, F. A.T. F. (1998). Rede Morfológica Não Supervisionada-RMNS. Thesis (Graduate Program in Applied Computing-CAP of the National Institute for Space Research-INPE), INPE-8759-TDI/800, Instituto Nacional de Pesquisas Espaciais-INPE, São José dos Campos, 1998. http://gjfb0520.sid.inpe.br/col/dpi.inpe.br/banon/1998/07.30.19.40/doc/tese.pdf

Silva, F. A. T. F. & Banon, G. J. F. (1999). Rede Morfológica Não Supervisionada., Proceedings of the IV Brazilian Conference on neural Networks. ITA- São José dos Campos, pp. 400-405.

Silva, F. A. T. F., & Silva, L. A. (2004). Detecção de múltiplos padrões usando redes neurais morfológicas. Anais do Congresso Brasileiro de Automática, A8-Sistemas Inteligentes, S0801, 1-6, RS, Brasil.

Silva, F. A. T. F., & Lucena, A. M. P. (2005a). Satélites inteligentes aplicados ao monitoramento ambiental em tempo real. 1-6. 10.21528/CBRN2005-068.

Silva, F. A. T. F., & Lucena, A. M. P. (2005b). Processamento Inteligente de Sinais aplicado ao Monitoramento Ambiental em Tempo Real. Anais XII Simpósio Brasileiro de Sensoriamento Remoto, Goiânia, Brasil, 16-21 abril 2005, INPE,. 3341-3348.

Silva, F. A. T. F. (2005). Neurônios morfológicos: Uma introdução às células on-off artificiais. Anais do VII CONGRESSO BRASILEIRO DE REDES NEURAIS – CBRN, São José dos Campos, Brasil, 2005, 1-6. https://sbic.org.br/eventos/cbrn_2005/CBRN2005_041/

Silva, F. A. T. F. (2006). Operadores morfológicos adaptativos. In: Congresso de Matemática e suas Aplicações, Foz do Iguaçu, Anais do Congresso de Matemática e suas Aplicações, pp.1-2.

Silva, F. A. T. F., Filho, P. M., Moreira, N. A., Rios, C. S. N., Oliveira, P. D. L., Camurca, P. J., & Lucena, A. M. P. (2015). Modelagem matemátia em microeletrônica reconfigurável: Estudo de caso sobre modulares BPSK. Relatório de Pesquisa INPE, sid.inpe.br/mtc-m21b/2015/05.28.17.26-RPQ, 2015. http://mtc-m21b.sid.inpe.br/col/sid.inpe.br/mtc-m21b/2015/05.28.17.26/doc/publicacao.pdf?languagebutton=pt-BR

Souza, O., Cortez, P. C., & Silva, F. A. T. F. (2012). Grayscale images and RGB video: compression by morphological neural network. In: Artificial Neural Networks in Pattern Recognition. Springer Berlin Heidelberg, 213-224.

Souza, O., Cortez, P. C., & Silva, F. A. T. F. (2013). Artificial Neural Networks for Compression of Gray Scale Images: A Benchmark. In: X Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2013). Fortaleza, Brasil. Anais of X Encontro Nacional de Inteligência Artificial e Computacional.

Thompson, D. R, Rogez, J. C., Chien, S., Doyle, R., Estlin, T., & McLaren, D. (2012). Agile science operations: A new approach for primitive bodies exploration. In Proceedings of SpaceOps, Stockholm, Sweden, June 2012.

Zakerhaghighi, M. R., & Naji, H. R. (2013). Whale algorithm for image processing, a hardware implementation, Machine Vision and Image Processing (MVIP), in 8th Iranian Conference, 355-359.

Zhou, G., & Kafatos, M. (2002). Future intelligent Earth observing satellites. Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS 2002 Conference Proceedings.

Xilinx-RTL, (2011). Technology Schematic Viewers Tutorial, UG685, v13.1 (v2011).

Xilinx-SG, (2014). Vivado Design Suite: Model-Based DSP Design using System Generator, UG897 (v2014.1).

XILINX-KC705, (2014). Hardware User Guide. v.1.6. USA: XILINX, 2014. 110 p. <http://www.Xilinx.com/support/documentation/boards_a nd_ki ts/kc705/ug810_KC705_Eval_Bd.pdf>.

Xilinx-RTK, (2021). Radiation Tolerant Kintex UltraScale XQRKU060 FPGA. https://www.xilinx.com/support/documentation/data_sheets/ds8 82-xqr-kintex-ultrascale.pdf.