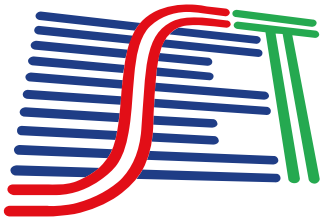
SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING

SET IJBE V. 8, 2022

ISSN print: 2446-9246
ISSN online: 2446-9432



SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING

SET IJBE V. 8, 2022

ISSN print: 2446-9246
ISSN online: 2446-9432

SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING

ISSN PRINT: 2446-9246 | ISSN ONLINE: 2446-9432

INDEXED IN:

Google's system that offers tools for search of academic literature.

International Cataloging Data in the Publication - CIP - Librarian Zoraide Gasparini CRB/9 1529

S517 SET International Journal of Broadcast Engineering —— vol. 8, (Dec. 2022). – São Paulo: Brazilian Society of Television Engineering —— SET, 2022

Annual frequency
ISSN Print 2446- 9246
ISSN online 2446- 9432
Available at: https://set.org.br/ijbe

Broadcasting – Periodic. 2. Transmission Engineering. 3. Digital TV. I. SET. II. Title.

CDD: 384.54

ANNUAL CIRCULATION : 500 EXEMPLARY

GRAPHIC DESIGN AND DIAGRAM: SOLANGE LORENZO

EDITORIAL BOARD

CORPORATE AUTHOR AND EDITOR

SET – Brazilian Society of Television Engineering, or, in Portuguese, SET – Sociedade Brasileira de Engenharia de Televisão.
Address: Av. Mario de Andrade, 252, suite 31 - Barra Funda District - São Paulo - SP - Brazil - Postal Code: 01156-001

SET BOARD OF DIRECTORS

Deliberative Council

2021 - 2022

President: Carlos Fini

Vice-President: Claudio Eduardo Younis

OFFICE HOLDER	SUBSTITUTE
Carlos Fini	Carlos Cauvilla
Luiz Bellarmino Polak Padilha	David Estevam de Britto
Claudio Eduardo Younis	José Carlos Aronchi de Souza
Claudio Alberto Borgo	Luis Otavio Marchezetti
Mauro Alves Garcia	Almir Antonio Rosa
Daniela Helena Machado e Souza	José Salustiano Fagundes de Souza
Vinicius Augusto da Silva Vasconcellos	Sergio Silva
Raymundo Costa Pinto Barros	Marcelo Santos Wance de Souza
Roberto Dias Lima Franco	Valderez de Almeida Donzelli
Emerson Weirich	Paulo Henrique Corona Viveiros de Castro
Sergio Eduardo di Santoro Bruzetti	Nelson Faria
José Eduardo Marti Cappia	Marco Tulio Nascimento
José Raimundo Lima da Cunha	Esdras Miranda de Araujo
Marcio Rogério Herman	Israel de Moraes Guratti
Cristiano Akamine	Marcelo Moreno
Rafael Duzzi de Oliveira	Fabio Ferraz
Jurandir Moreira Pitsch	Wagner Kojo

Fiscal Council

Nivelle Daou	Rafael Silviera Leal
José Chaves F. de Oliveira	Sandro Sereno
Marcos Paulo Teixeira	

Council of Former Presidents

Adilson Pontes Malta	Liliana Nakonechnyj
Carlos Eduardo Oliveira Capelão	Olímpio José Franco
Fernando Mattoso Bittencourt Filho	Roberto Dias Lima Franco
José Munhoz	

Regional representatives

North: Henrique Camargo e Eduardo Lopes
North East: Ronald Almeida e Gabriel Eskenazi
Midwest: Wender de Souza
Southeast: Geraldo Mello
South: Caio Klein e Alisson Heidemman

ABOUT THE JOURNAL

SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING

The SET IJBE, (SET International Journal of Broadcast Engineering) is an open access, peer-reviewed article--at-a-time international scientific journal whose objective is to cover knowledge about communications engineering in the field of broadcasting. The SET IJBE seeks the latest and most compelling research articles and state-of-the-art technologies.

Publishing schedule and schema

On-line version – Once an article is accepted and its final version approved by the Editorial Board, it will be published immediately on-line on a one article-at-a-time basis

Printed version – Once a year, all articles accepted and published on-line over the previous twelve months will be compiled for publication in a printed version.

Types of papers

Regular (Full) Papers: Traditional and original research [from 6 to 20 pages]

Tutorial Papers: Brand new OTT (Over-The-Top) detailed implementation and fully set up state-of-the art systems [4 – 6 pages]

Letters: Short notes and consideration about current and relevant techniques, technologies and implementations involving engineering solutions [1-3 pages]

Open Access Policy

If an article is accepted for publication, it will be made available to be read and re-used under a Creative Commons Attribution (CC-BY) license.

Editorial Office:

If you require any additional information, please contact the SET IBJE (SET International Journal of Broadcast Engineering) administration staff:

Address: Av. Mario de Andrade, 252, suite 31, Postal Code:01156-001, São Paulo – SP – Brazil; E-mail: IJBE@set.org.br

Aims and Scope include, but are not limited to:

Advanced audio technology and processing
Advanced display technologies
Advanced RF Modulation Technologies
Advanced technologies and systems for emerging broadcasting applications
Applying IT Networks in Broadcast Facilities
Broadcast spectrum issues – re-packing, sharing
Cable & Satellite interconnection with terrestrial broadcasters
Cellular broadcast technologies
Communication, Networking & Broadcasting
Content Delivery Networks – CDN
Digital radio and television systems: Terrestrial, Cable, Satellite, Internet, Wireless.
Electromagnetic compatibility issues between collocated services (e.g. broadcast and LTE)
General Topics for Engineers (Math, Science & Engineering)
Hybrid receiver technology
Interactive Technology for broadcast
IP Networks management and configuration
Metadata systems and management
Mobile DTV systems (all aspects, both transmission and reception)
Mobile/dashboard technology
Next-gen broadcast platforms and standards development
Non-real time (NRT) broadcast services
Ratings technology, second screen technology and services
Secondary service system design; mitigation of interference in primary services
Securing Broadcast IT Networks
Signal Processing & Analysis
Software Defined Radio – SDR Technologies
Streaming delivery of broadcast content
Transmission, propagation, reception, re-distribution of broadcast signals AM, FM, and TV transmitter and antenna systems
Transport stream issues – ancillary services
Unlicensed device operation in TV white spaces

We wish to inform you that the activities, events and publications of the Brazilian Society of Television Engineering – SET, including this one, enjoy international support, under formal agreements, from the following international organizations. We also take this opportunity to thank them and reiterate how proud we are that they support our work.

SUMMARY

SET IJBE v. 8, 2022, **74** pages, 7 articles.

	08	**Editorial**
Article 1	**10**	**Neural Network-Like LDPC Decoder for Mobile Applications** Fadi Jerji, Leandro Silva and Cristiano Akamine
Article 2	**22**	**Spectrum Availability for the Deployment of TV 3.0** Thiago Aguiar Soares, Paulo E. R Cardoso, Ugo Silva Dias
Article 3	**34**	**Advances in video compression: a glimpse of the long-awaited disruption** Thomas Guionnet, Marwa Tarchouli, Sébastien Pelurson and Mickaël Raulet
Article 4	**46**	**Implementation of Convolutional Neural Network in FPGA for Image Recognition** Victor Mendonça Aguirre and Fadi Jerji
Article 5	**56**	**Audience and complexity aware live video encoders orchestration** Abdelmajid Moussaoui, Thomas Guionnet, and Mickaël Raulet
Article 6	**65**	**Contributions to TV 3.0 using 5G-MAG Reference Tools** Wesley Henrique Silva de Souza and Cristiano Akamine
Article 7	**72**	**Comparison of the Physical-layer Performance between ATSC 3.0 and 5G Broadcast** Seok-Ki Ahn, Member, IEEE and Sung-Ik Park, Fellow, IEEE

EDITORIAL

This edition of the IJBE is dedicated to SET EXPO Conference and Trade Show. The best papers presented at the SET EXPO were published in this edition. This is the first in-person SET EXPO Conference and Trade Show after the Coronavirus (COVID-19) Pandemic.

This issue features several thematic articles covering aspects of adopted TV 3.0 technologies and candidate technologies for the physical layer.

The search for quality in developing technologies that make digital communication systems more efficient is the focus of researchers to whom the IJBE offers the opportunity to disseminate their studies, experiments, and research in the scientific and technological areas of production and distribution of informational content.

We hope you enjoy these articles and feel motivated to submit an article.

Best wishes
SET IJBE Editors

Neural Network-Like LDPC Decoder for Mobile Applications

Fadi Jerji
Leandro Silva
Cristiano Akamine

CITE THIS ARTICLE

Jerji, Fadi; Silva, Leandro; Akamine, Cristiano; 2022. Neural Network-Like LDPC Decoder for Mobile Applications. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.1. Web Link: http://dx.doi.org/10.18580/seti-jbe.2022.1

Neural Network-Like LDPC Decoder for Mobile Applications

Fadi Jerji, Leandro Silva, and Cristiano Akamine, *Member, SET*

***Abstract*—This paper presents a low complexity iterative decoder for Low-Density Parity-Check (LDPC) codes for mobile applications using a Neural Network-like (NNL) structure and a modified Single-Layer Perceptron (SLP) training algorithm. The proposed approach allows for midrange decoding performance with a minimum gap to Shannon-limit of 3.19 dB at a frame error rate of 10^{-4} for the short frame and the code rate 13/15 of the next-generation Digital Terrestrial Television Broadcasting (DTTB) standard of the Advanced Television Systems Committee (ATSC), the "ATSC 3.0". The NNL decoder has a low decoding time, thus, it would be suitable for low power embedded systems, software-defined radio implementation tools, and software-based DTTB receptors.**

***Index Terms*—Channel coding, iterative decoding, Low-Density Parity-Check (LDPC) codes, Neural Networks.**

I. Introduction

SINCE the introduction of mobile phones in the 70s, they have become increasingly essential in our everyday lives, and with the introduction of the smartphone in the late 2000s, it started to replace many devices by combining their functionalities in one high-performance piece of hardware [1].

One of the main challenges of our modern smartphones design process is achieving a trade-off between the demand for higher processing power and multi-functionality and the cost, weight, power consumption and battery lifespan [2]. This only serves to increase the necessity for hardware and software optimization.

Most modern smartphones incorporate a variety of technologies to serve different purposes, such as the Wi-Fi, the Long Term Evolution (LTE) from the Third Generation Partnership Project (3GPP) and the under-development fifth-generation wireless technology for digital cellular networks (5G). While those technologies are vastly different in many aspects, they all share an important component, the Forward Error Correction Code (FEC) that is deployed using the Low-Density Parity-Check (LDPC) codes [3]–[5].

Another example of a technology that uses LDPC codes and is soon-to-be incorporated in smartphones is the Digital Terrestrial Television Broadcasting (DTTB) receivers, specifically the next-generation DTTB standard of the Advanced Television Systems Committee (ATSC), the "ATSC 3.0" [6].

The LDPC codes are chosen in many technologies due to their near-Shannon-limit performance but their high complexity decoders force their implementation in a specialized chipset [7].

Considering that each of those technologies has different requirements for their error correction performance and latency, several dedicated LDPC decoding chipsets have to be included in the smartphone, which increases energy requirements and price.

The remaining sections of this paper are organized as follows: the LDPC codes and the classical LDPC decoding methods in Section II, an overview of the perceptron, the Single-Layer Perceptron (SLP) and its training algorithms in Section III, the proposed Neural Networks-like (NNL) LDPC decoding method in Section IV, the mathematical complexity analysis and memory requirements of each of these decoders in Section V, the detailed results and discussions of the implemented experiments and numerical complexity calculation for each decoder in Section VI, and the conclusion in Section VII.

II. LDPC Codes

The LDPC error correction codes were introduced by Gallager in [8]. They are an attractive option for many technologies due to their near-Shannon-limit performance [7], but the LDPC code decoders are known for requiring a high processing power. Therefore, implementing these codes in software for smartphones requires highly optimized decoding algorithms, especially for the long codewords required for high-performance LDPC codes [9].

The LDPC code is represented by the notion (N,K), where N is the sum of the original information bits number K and the parity bits number P. The parity bits part $P = N - K$ is calculated and added to the original information bits, therefore generating a message that can be transmitted via a noisy channel.

An example LDPC code (7,4) defined by the matrix H is shown in (1) where the number of columns represents the total number of information bits and parity bits while the number of rows represent the number of parity bits.

The parity check matrix H can be represented by a Tanner graph representation and shown in Fig. 1 [10], where the variable nodes $V_1, V_2, \ldots, V_7$ are the values representing the message bits after passing through a noisy channel and the parity-check nodes C_1, C_2, C_3 are the calculated parity values used in error check and correction.

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

This work was supported in part by Coordination for the Improvement of Higher Education Personnel (CAPES), National Council of Technological and Scientific Development (CNPq) and MackPesquisa.

The authors are with the Postgraduate Program in Electrical and Computer Engineering (PPGEEC), Mackenzie Presbyterian University, São Paulo, Brazil.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

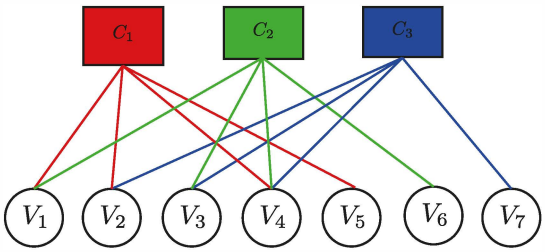


Fig. 1: Tanner Graph.

Many LDPC decoding methods were presented in the literature and generally, they can be divided into two families, high-complexity high-performance decoders and low-complexity low-performance decoders.

A. *High-complexity high-performance decoders*

The high-complexity high-performance family contains the sum-product algorithm (SPA), its simplified version the min-sum algorithm (MSA) and their variant decoding algorithms, the normalized min-sum algorithm (NMSA), the offset min-sum algorithm (OMSA) and the variable correction algorithm (VCMS) [11]–[15].

While the aforementioned decoding methods vary in performance and distance from Shannon-limit, they all share a high complexity [16]. This high complexity, when combined with a long codeword, makes a software implementation of any of them impractical, since the latency caused by the decoding process would cause a reception disturbance and make it unfeasible [17].

B. *Low-complexity Low-performance decoders*

In order to provide LDPC decoding methods with lower decoding complexity, several algorithms were introduced in the literature, such as the bit-flipping algorithm (BFA) and its variant the weighted bit-flipping algorithm (WBF) [8], [18], but their low performance makes them unattractive methods for industrial implementations.

1) Bit-Flipping decoding algorithm: The BFA algorithm proposed by Gallager [8] is the simplest LDPC decoder as it uses binary information and it does not require extensive calculation. It starts by applying (2) to calculate the parity-check nodes binary values C_{j_l} from the binary values of the variable node V_{i_l}, where i is the index of the variable node within the array of the variable nodes based on the matrix H that defines the LDPC code and j is the index of the check node within the array of the check nodes. Then for each variable node, the decoder counts the number of the unsatisfied parity-check nodes that are connected to it as in (3). Finally, the decoder flips the value of the variable node if that number was higher than a certain value X as in (4). The process is repeated until all parity-check nodes are satisfied or a maximum number of iterations is reached [8].

$$C_{j_l} = \text{XOR}_{i \in C(j)} \, V_{i_l} \tag{2}$$

$$FlipV_i = \sum_{j \in V(i)} C_{j_l} \tag{3}$$

$$V_{i_l} = \begin{cases} V_{i_l} & \text{if } FlipV_i < X \\ \text{NOT}(V_{i_l}) & \text{if } FlipV_i \geq X \end{cases} \tag{4}$$

Although this method can be easily implemented in software, its low error correction performance makes it an unattractive industrial solution in many cases [8].

2) Weighted Bit-Flipping decoding Algorithm: The WBF is an alternative LDPC decoding algorithm derived from the BFA by [18]. The decoding process starts by calculating the values of the parity-check nodes as it is done in BFA to detect any errors and terminating the decoding process if all the parity-check points were satisfied.

Otherwise, the input log likelihood ratio (LLR) the LLR_i are used to initialize the real value representation of each variable nodes V_{i_r}. Then, (5) is applied to find $|V_{i_r}|min_j$ that represents the lowest absolute real value among the variable nodes V_i that are connected to the parity-check nodes C_j. The logical representation of the parity-check nodes C_{j_l} is calculated using (6) where V_{i_l} is the logical representation of the variable nodes V_i, i and j are the indexes of the variable nodes and the check nodes respectively.

The real value E_{i_r} can be calculated using (7) then the flipping location can be determined by finding the highest E_{i_r} and flipping the logical value V_{i_l} that correspond to it using (8).

The operation is repeated until all the parity-check nodes are satisfied or the maximum number of iteration is reached.

$$|V_{i_r}|min_j = \min_{i \in C(j)} |V_{i_r}| \tag{5}$$

$$C_{j_l} = \text{XOR}_{i \in C(j)} \, V_{i_l} \tag{6}$$

$$E_{i_r} = \sum_{j \in V(i)} (2C_{j_l} - 1)|V_{i_r}|min_j \tag{7}$$

$$V_{i_l} = \begin{cases} \text{NOT}(V_{i_l}) & \text{if } E_{i_r} = \max_{i=1:N} E_{i_r} \\ V_{i_l} & otherwise \end{cases} \tag{8}$$

The WBF can perform better than the BFA in some cases, but its limitation of flipping only one bit per iteration limits its practical implementations.

III. The Single-Layer Perceptron Neural Networks

The perceptron was originally proposed by [19] and [20] as a mathematical representation of the human brain neuron and its synapses. It was developed to be able to learn the relation between its inputs $x_1, x_2, \ldots, x_n$ and its output y. According to Fig. 2, the perceptron has a bias of a fixed value of $+1$ that has a connection weight w_b. Also, each input has its own weight $w_1, w_2, \ldots, w_n$. The weights are initialized with random values at the beginning of the training and can be adjusted by the training algorithm.

Combining several perceptrons, an SLP Neural Network (NN) can be formed as shown in Fig. 3. To train the SLP, first, the network weights are randomized to break symmetry, then the training algorithm is applied as a two-step algorithm, the forward step to calculate the outputs and the second step to adjust the weights of the network to minimize the total error. For the forward step, the individual perceptron in the SLP uses

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

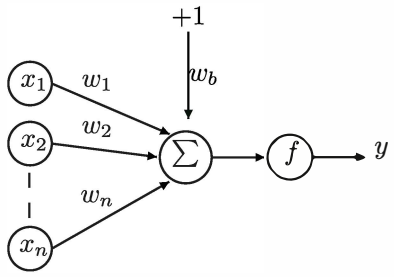


Fig. 2: The perceptron Adaline.

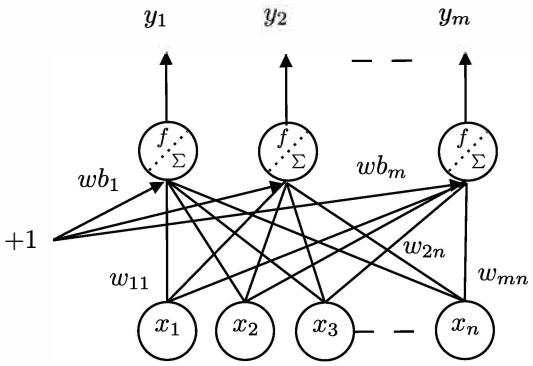


Fig. 3: Single-layer perceptron Neural Network.

a sum operation to produce its output as in (9), where x_i and yo_j are the input of the index i and the calculated output of the index j respectively.

$$yo_j = f(\sum_{i=1}^{n} x_i w_{ji} + wb_j) \tag{9}$$

The total error E is calculated using (10) where yo_j and yd_j are the calculated output and the desired output of the perceptron of the index j respectively [21].

$$E = \frac{1}{2}\sum_{j=1}^{m}(yo_j - yd_j)^2 \tag{10}$$

Then, the adjustment to the weight w_{ji}, Δw_{ji}, is calculated by the mean of the partial derivative of the total error with respect to the weight w_{ji} using (11)

$$\Delta w_{ji} = -\frac{\partial E}{\partial w_{ji}} \tag{11}$$

The final stage of the training iteration (t), is to calculate the weights for the next iteration $w_{ji}(t+1)$ applying the adjustment from the current iteration $\Delta w_{ji}(t)$ to the weights $w_{ji}(t)$ using the training rate η as in (12).

$$w_{ji}(t+1) = w_{ji}(t) + \eta \Delta w_{ji}(t) \tag{12}$$

The process is repeated until an acceptable total error or a maximum number of iterations is reached.

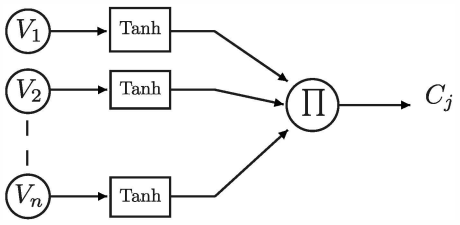


Fig. 4: The proposed XOR perceptron.

IV. The Proposed Neural Network-Like Algorithm

To provide an alternative to dedicated LDPC decoding chipsets, a configurable software implementation of the LDPC decoder for the smartphone can be used. By allowing some degradation in error correction performance, it becomes practical to create a flexible, cost-effective LDPC decoder with low power consumption.

Comparing the Tanner graph of Fig. 1 with the SLP structure in Fig. 3; the similarity can be easily spotted along with the possibility of applying the SLP NN training algorithm to decode the LDPC codes, although some modifications to the SLP and the training algorithm are required to fully match the Tanner graph.

An early attempt to implement a Multi-layer perceptron (MLP) decoder was done by [22] and revised by the same authors in [23] and while it provided a proof of concept, it was not able to deliver a stable and functional decoder for long code words due to several limitations such as the high number of multiplications and the overflow of memory register that resulted from it.

To modify the SLP structure of Fig. 3 to match the Tanner graph of Fig. 1, the bias weights should be omitted along with the activation function of each perceptron, in addition to omitting all the connections that correspond to ZERO in the matrix H and the Tanner graph. In addition to the modifications to the SLP structure, the training algorithm needs to be modified to adjust the inputs themselves instead of the weights of their connections.

The perceptron structure is not adequate for the LDPC decoder requirements since the relation between the Tanner graph inputs V and the outputs C is a logical XOR. The SLP training algorithm requires the derivation of the function XOR, which means that it should have Real numerical values as inputs and outputs instead of the Boolean XOR.

To solve the aforementioned issues, a new XOR perceptron is proposed as shown in Fig. 4.

Let real variables $\forall\ A_r,\ B_r \in \Re$, we define their logical representation A_l, B_l as the following:

$$A_l = \text{LOGICAL}(A_r) = \begin{cases} ONE & \text{if } A_r < 0 \\ ZERO & \text{if } A_r > 0 \end{cases} \tag{13}$$

Based on (13), the sign of A_r, $sgn(A_r)$, combined with its absolute value represent its probability. Thus, $A_r = 0$ is an equal probability of A_l being a logical ZERO and a logical ONE at the same time and in this case a small random value is forced. Therefore, we define:

$$\text{NOT}(A_l) = \text{LOGICAL}(-A_r)$$

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

and

$$A_l \,\mathrm{XOR}\, B_l = \mathrm{LOGICAL}(A_r \times B_r) \tag{14}$$

As per the Tanner graph representation of the LDPC code, each check node C_j that is connected to the variable nodes $V_1, V_2, \ldots, V_n$ has a real value C_{j_r} and a logical representation C_{j_l} and each variable node has a real value V_{i_r} and a logical representation V_{i_l}. The value of C_{j_l} is calculated as in (15).

$$C_{j_l} = V_{1_l} \,\mathrm{XOR}\, V_{2_l} \,\mathrm{XOR} \ldots \mathrm{XOR}\, V_{n_l} \tag{15}$$

From (14) and (15)

$$\begin{aligned} C_{j_r} &= V_{1_r} \times V_{2_r} \times \ldots \times V_{n_r} \\ &= \prod_{i=1}^{n} V_{i_r} \end{aligned}$$

In order to avoid a variable overflow due to the limitation of real numbers microprocessors representation such as float and double float we calculate C_{j_r} as shown by Fig. 4 and (16):

$$C_{j_r} = \prod_{i=1}^{n} \mathrm{Tanh}(V_{i_r}) \tag{16}$$

As for $\forall\ V_{i_r} \in \Re$ then

$$-1 < \mathrm{Tanh}(V_{i_r}) < 1$$

and

$$-1 < \prod_{i=1}^{n} \mathrm{Tanh}(V_{i_r}) < 1$$

thus, the aforementioned limitation for the algorithm implementation is removed.
For the LDPC code (N,K), (16) becomes (17)

$$C_{j_r} = \prod_{i \in C(j)} \mathrm{Tanh}(V_{i_r}) \tag{17}$$

By modifying the SLP training algorithm for the new XOR perceptron-Tanner graph structure, we obtain (18),

$$E = \frac{1}{2} \sum_{j=1}^{P} (e_j)^2 \tag{18}$$

where E is the total error and e_j is the error for the check node C_j and is calculated using (19),

$$e_j = Z - C_{j_r} \tag{19}$$

where Z is the desired output value and is a positive real number in the range (0.0 — 1.0) representing a logical ZERO. Then the partial derivative of the total error with respect to each variable node is calculated using (20),

$$\begin{aligned} \Delta V_{i_r} &= -\frac{\partial E}{\partial V_{i_r}} \\ &= -\sum_{j \in V(i)} \left(\frac{\partial E}{\partial e_j} \times \frac{\partial e_j}{\partial C_{j_r}} \times \frac{\partial C_{j_r}}{\partial V_{i_r}}\right) \end{aligned} \tag{20}$$

where

$$\frac{\partial E}{\partial e_j} = e_j$$

and

$$\frac{\partial e_j}{\partial C_{j_r}} = -1$$

and

$$\frac{\partial C_{j_r}}{\partial V_{i_r}} = \Big[\prod_{i' \in C(j) \setminus i} \mathrm{Tanh}(V_{i'_r})\Big][\mathrm{Tanh}'(V_{i_r})]$$

The last step of the iteration would be the application of the correction to V_i using (21)

$$V_{i_r}(t+1) = V_{i_r}(t) + \eta \Delta V_{i_r}(t) \tag{21}$$

where t, $V_{i_r}(t)$ and $\Delta V_{i_r}(t)$ are the current iteration, the value of V_{i_r} during the current iteration and the calculated value of ΔV_{i_r} in the current iteration respectively and where $t+1$ and $V_{i_r}(t+1)$ are the next iteration and the value of V_{i_r} during the next iteration respectively and η is a real number that represents the correction rate. This experimental study demonstrated that the proposed decoder has its best performance when η satisfied the condition $0 < \eta < 2$.

It is important to mention that to keep the Tanh functions in (16) functioning in the linear area of the Tanh curve, the input variables V_{i_r} need to be normalized to match that area. Since the data is normalized, the NNL decoder is not sensitive to channel estimation error.

V. Complexity Analysis and Memory Requirements

To compare the complexity of the proposed NNL decoder with the low-complexity decoders, the BFA and the WBF, we calculate the number of microprocessor cycles needed for each step of each decoder and its memory footprint.

A. Complexity analysis

Let I_a, I_m, I_c, I_x, I_n, I_t, I_s and I_{abs} be the number of microprocessor cycles needed for addition, multiplication, comparison, logical XOR operation, logical NOT operation, hyperbolic tangent, signal extraction operation and absolute, respectively, nI be the total number of microprocessor cycles needed for the current step of the decoding iteration.

For the LDPC code (N,K) where O is the number of nonzero elements of the sparse matrix H of N columns and P rows, the value of O is equal to the total number of connections in the corresponding Tanner graph. The value N_{errors} is the number of erroneous bits.

For the same LDPC code (N,K), nV_i and nC_j are the number of the check nodes connected to the variable node V_i and the number of variable nodes connected to the check node Cj respectively.

1) BFA complexity: To calculate the complexity of BFA, we start by calculating the complexity of (2) that is represented by (22).

$$\begin{aligned} nI &= \sum_{j=1}^{P} nC_j \times I_x \\ &= O \times I_x \end{aligned} \tag{22}$$

The complexity of (3) is represented by (23).

$$\begin{aligned} nI &= \sum_{i=1}^{N} nV_i \times I_a \\ &= O \times I_a \end{aligned} \tag{23}$$

The complexity of (4) is represented by (24).

$$\begin{aligned} nI &= \sum_{i=1}^{N} I_c + N_{errors} \times I_n \\ &= N \times I_c + N_{errors} \times I_n \end{aligned}$$

$$\text{as } 0 \geq N_{errors} \geq N \text{ then}$$

$$N \times I_c \geq nI \geq N \times (I_c + I_n) \tag{24}$$

2) WBF complexity: To calculate the complexity of WBF, it is necessary to calculate the complexity of each step of its iteration. The complexity of (5) is represented by (25).

$$\begin{aligned} nI &= \sum_{j=1}^{P} nC_j \times (I_c + I_{abs}) \\ &= O \times (I_c + I_{abs}) \end{aligned} \tag{25}$$

The complexity of (6) is represented by (26).

$$\begin{aligned} nI &= \sum_{j=1}^{P} nC_j \times I_x \\ &= O \times I_x \end{aligned} \tag{26}$$

The complexity of (7) is represented by (27).

$$\begin{aligned} nI &= \sum_{i=1}^{N} nV_i \times (2 \times I_m + I_a) \\ &= O \times (2 \times I_m + I_a) \end{aligned} \tag{27}$$

The complexity of (8) is represented by (28).

$$nI = N \times I_x + I_n \tag{28}$$

3) NNL complexity: The complexity of the proposed NNL decoder is provided by the sum of the complexity of all its steps, starting with the calculation of the complexity of (17) that is represented by (29).

$$\begin{aligned} nI &= \sum_{j=1}^{P} [nC_j \times (I_m + I_t)] \\ &\text{Ignoring the repetitive calculations of } \mathrm{Tanh}(V_{i_r}) \\ &= I_m \times \sum_{j=1}^{P} nC_j + N \times I_t \\ &= O \times I_m + N \times I_t \end{aligned} \tag{29}$$

The complexity of (18) is represented by (30).

$$nI = P \times (2 \times I_a + I_m) + I_m \tag{30}$$

The complexity of (20) is represented by (31).

$$\begin{aligned} nI =& \sum_{i=1}^{N} \{I_a + \sum_{j \in V(i)} [I_a + (nC_j - 1) \times (I_m + I_t) \\ &+ 2 \times I_a + 3 \times I_m + I_t]\} \\ =& (N + 3 \times O) \times I_a + O \times (3 \times I_m + I_t) \\ &+ (I_m + I_t)(-O + \sum_{j=1}^{P} nC_j^2) \end{aligned} \tag{31}$$

However, since $\mathrm{Tanh}(V_{i_r})$ was pre-computed in (16), (31) becomes (32):

$$= (N + 3 \times O) \times I_a + I_m \times (2 \times O + \sum_{j=1}^{P} nC_j^2) \tag{32}$$

The complexity of (21) is represented by (33)

$$nI = N \times (I_a + I_m) \tag{33}$$

B. Memory requirements

As all methods require storing both V and C matrices that have the sizes $N \times 1$ and $P \times 1$ respectively, then we analyze the memory space that is required by each method in addition to the basic $N + P$ locations. We ignore single variables as they require a negligible memory space in comparison to the matrices used in the decoding methods.

To calculate the number of memory accesses per iteration for each decoding method, it is necessary to calculate the number of memory accesses for each step represented by nMa where Ma is the memory access per variable.

1) BFA memory requirements: The BFA requires storing the matrix $FlipV_i$ with the size $N \times 1$. Thus, the total memory requirement is: $2 \times N + P$.

The memory access of (2) is represented by (34).

$$\begin{aligned} nMa &= \sum_{j=1}^{P} Ma + \sum_{j=1}^{P} nC_j \times Ma \\ &= (P + O) \times Ma \end{aligned} \tag{34}$$

The memory access of (3) is represented by (35).

$$\begin{aligned} nMa &= \sum_{i=1}^{N} Ma + \sum_{i=1}^{N} nV_i \times Ma \\ &= (N + O) \times Ma \end{aligned} \tag{35}$$

The memory access of (4) is represented by (36).

$$\begin{aligned} nMa &= \sum_{j=1}^{N} 2 \times Ma \\ &= 2 \times N \times Ma \end{aligned} \tag{36}$$

2) WBF memory requirements: The WBF requires storing the matrices $|V_{i_r}|min_j$ and C_{j_l} with the size $P \times 1$ each. Thus, the total memory requirement is: $N + 3 \times P$.

The memory access of (5) is represented by (37).

$$\begin{aligned} nMa &= \sum_{j=1}^{P} Ma + \sum_{j=1}^{P} nC_j \times Ma \\ &= (P + O) \times Ma \end{aligned} \tag{37}$$

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

The memory access of (6) is represented by (38).

$$\begin{aligned} nMa &= \sum_{j=1}^{P} Ma + \sum_{j=1}^{P} nC_j \times 2 \times Ma \\ &= (P + 2 \times O) \times Ma \end{aligned} \tag{38}$$

The memory access of (7) is represented by (39).

$$\begin{aligned} nMa &= \sum_{i=1}^{N} Ma + \sum_{i=1}^{N} nV_i \times (2 \times Ma) \\ &= (N + 2 \times O) \times Ma \end{aligned} \tag{39}$$

The memory access of (8) is represented by (40).

$$nMa = N \times Ma \tag{40}$$

3) NNL memory requirements: The NNL method requires storing the matrix $\mathrm{Tanh}(V_{i_r})$ with the size $N \times 1$, and the storage of the matrix ΔV_{i_r} can be omitted since only the current ΔV_{i_r} is needed, and it does not need to be stored between iterations. Thus, the total memory requirement is: $2 \times N + P$.

The memory access of (17) is represented by (41).

$$\begin{aligned} nMa &= \sum_{j=1}^{P} Ma + \sum_{j=1}^{P} [nC_j \times Ma] \\ &= (P + O) \times Ma \end{aligned} \tag{41}$$

The memory access of (18) is represented by (42).

$$\begin{aligned} nMa &= \sum_{j=1}^{P} Ma + 1 \\ &= P \times (Ma + 1) \end{aligned} \tag{42}$$

The memory access of (20) is represented by (43).

$$\begin{aligned} nMa &= \sum_{i=1}^{N} Ma + \sum_{i=1}^{N} \sum_{j \in V(i)} (nC_j + 1) \times Ma \\ &= N \times Ma + O \times Ma + Ma \times \sum_{j=1}^{P} nC_j^2 \\ &= (N + O + \sum_{j=1}^{P} nC_j^2) \times Ma \end{aligned} \tag{43}$$

The memory access of (21) is represented by (44)

$$\begin{aligned} nMa &= \sum_{i=1}^{N} Ma + \sum_{i=1}^{N} (2 \times Ma) \\ &= 3 \times N \times Ma \end{aligned} \tag{44}$$

VI. Experimental Results and Discussions

To numerically compare the proposed NNL decoder with the classical LDPC decoders, both a performance simulation and a complexity calculation are necessary.

A group of LDPC codes was selected for comparative analysis. In this paper, a case study was done using the ATSC 3.0 LDPC codes.

A. Case study: ATSC 3.0 LDPC Codes

The ATSC 3.0 has been under development as the next-generation DTTB standard by the ATSC since early 2013 [24], [25]. The ATSC 3.0 offers higher data rates, allowing higher image quality and higher robustness in comparison to the earlier ATSC 1.0 A/53 standard that was developed over a decade earlier [26].

In order for the new ATSC 3.0 standard to ensure higher robustness than its predecessors, several new techniques were adopted, such as the Bit-interleaved Coded Modulation (BICM) [27]. The BICM deploys two concatenated FEC layers with an inner LDPC code and an outer Bose-Chaudhuri-Hocquenghem (BCH) code [28]. The outer code can be replaced with a Cyclic Redundancy Check (CRC) or it can be omitted.

The ATSC 3.0 standard uses a systematic LDPC coding with two different values for its frame size N, a normal frame of 64800 bits and a short frame of 16200 bits, given that the normal frame provides a better error correction performance and the short frame a lower latency for latency-sensitive applications [29]. Ten different Code Rates (CR) from (2/15) up to (13/15) for each frame size cause a variance in the length of information part K of the LDPC message.

In order for the new ATSC 3.0 standard to achieve high robustness, two different LDPC structures are used, the irregular repeat accumulate (IRA) structure that has a high performance in medium and high CR but with the disadvantage of low performance in low CR [25], [30]. Therefore, the multi-edge type (MET) structure is used for low CR [6].

B. Performance simulation

To provide a comparative analysis, the proposed NNL was implemented and tested over an additive white Gaussian noise (AWGN) channel using quadrature phase-shift keying (QPSK) modulation [31]. Each message was decoded using a maximum of 50 iterations. An extensive computer simulation was executed to obtain the signal to noise ratio per symbol (E_s/N_0) that corresponded to the threshold of 10^{-4} frame error rate (FER) since it is always assumed that the outer coding is set to BCH. A performance of 10^{-4} FER by the LDPC inner code will guarantee an overall performance of 10^{-6} FER after applying the outer code, which is sufficient for terrestrial broadcasting services [16].

The value η was chosen to be equal to 1.0 for all code lengths and CR. For code length of 64800 and CR of (2/15 — 6/15), the normalization range and the value Z were chosen to be equal to (-2.0 — 2.0) and 0.60 respectively, and for the CR of (7/15 — 13/15) the respective equal values chosen were (-4.0 — 4.0) and 0.99. For code length of 16200 and CR of (2/15 — 4/15) and (6/15 — 8/15) the normalization range and the value Z were chosen to be equals to (-2.0 — 2.0) and 0.60 respectively, and for the CR of (5/15 and 9/15 — 13/15) the respective equal values chosen were (-4.0 — 4.0) and 0.99.

Fig. 5 demonstrates the performance of various ATSC 3.0 LDPC decoders with a code length of 16200 and CR of 13/15. In this case, the simulation was run until the total number of frames are processed regardless of the FER value, therefore

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

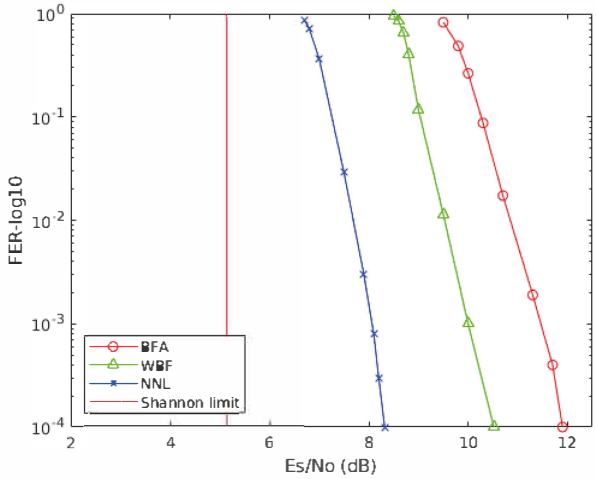


Fig. 5: ATSC 3.0 LDPC codes performance using various decoders (code length = 16200, CR = 13/15).

TABLE I: Performance comparison for ATSC 3.0 LDPC decoders (code length = 64800) (QPSK)

CR	Structure	Shannon limit (Es/N_0 ,dB)	Required Es/N_0 for FER=10^{-4} (dB)		
			NNL	WBF [32]	BFA [32]
2/15	MET	-6.92	3.81	10.32	6.80
3/15	MET	-4.94	4.90	10.32	7.90
4/15	MET	-3.47	6.02	10.34	8.41
5/15	MET	-2.26	6.70	10.36	9.31
6/15	IRA	-1.21	7.01	10.50	10.11
7/15	MET	-0.26	8.09	10.40	10.30
8/15	IRA	0.62	7.95	10.47	10.40
9/15	IRA	1.47	8.15	10.44	10.37
10/15	IRA	2.31	8.11	10.49	10.50
11/15	IRA	3.17	8.14	10.50	10.74
12/15	IRA	4.08	8.27	11.00	11.56
13/15	IRA	5.13	8.51	10.91	12.00

TABLE II: Performance comparison for ATSC 3.0 LDPC decoders (code length = 16200) (QPSK)

CR	Structure	Shannon limit (Es/N_0 ,dB)	Required Es/N_0 for FER=10^{-4} (dB)		
			NNL	WBF [32]	BFA [32]
2/15	MET	-6.92	4.04	9.21	8.70
3/15	MET	-4.94	4.90	9.13	8.16
4/15	MET	-3.47	6.32	9.20	9.03
5/15	MET	-2.26	6.50	9.17	9.10
6/15	IRA	-1.21	6.42	9.50	9.91
7/15	IRA	-0.26	7.00	10.00	9.70
8/15	IRA	0.62	6.94	10.00	10.00
9/15	IRA	1.47	7.31	9.71	10.02
10/15	IRA	2.31	7.37	10.00	10.30
11/15	IRA	3.17	7.60	10.20	10.52
12/15	IRA	4.08	7.90	10.23	10.93
13/15	IRA	5.13	8.32	10.52	11.90

it can be verified that the proposed NNL decoder does not present an error floor.

The experimental results for each coding rate are presented in Tables I and II where the NNL superior performance to both the WBF and the BFA decoding algorithm can be seen in all code rates and code lengths.

It can be seen that the limitation of the WBF of only correcting one error per iteration is clear in the results as the WBF decoder does not have a decoding performance that can achieve BER of 10^{-4} for a value of $E_s/N_0 < 9.13$ dB for any of the code rates.

The BFA decoding algorithm limitation due to its dependency on binary LLR_i values is shown in the results as well, especially at high code rates.

The proposed NNL decoding algorithm demonstrated a mid-range decoding performance getting closer to the theoretical limit in higher code rates.

C. Complexity calculation

As the hyperbolic tangent function requires one exponential function to be calculated in approximately 60 microprocessor clock cycles, while the inverse hyperbolic tangent function requires one logarithmic function thus approximately 52 processor clock cycles; most decoders use a work-around for this issue by tabling the hyperbolic tangent function to reduce the required cycles [33]. Most modern microprocessors utilize a hardware float point unit (FPU) that is capable of executing addition and multiplication in one to two cycles in addition to the logical and comparison functions in a single cycle [34]. Therefore, we can approximately calculate the number of cycles required to decode an ATSC 3.0 LDPC message of different CR using different decoders.

Considering that all the mathematical and logical operations mentioned in Section V would be executed in one processor's cycle except for the multiplication I_m that needs two cycles and the hyperbolic tangent I_t that requires 60 cycles, it is possible to calculate the number of cycles per iteration for each decoder by applying the aforementioned values in the complexity equations demonstrated in Section V. The implementation overhead caused by the various loops required to execute each step of the decoding process is estimated by counting the required operations and the number of memory accesses.

TABLE III: ATSC 3.0 LDPC decoders number of instructions and memory accesses (code length = 64800)

CR	NNL ($\times 10^6$)		WBF ($\times 10^6$)		BFA ($\times 10^6$)	
	nI	nMa	nI	nMa	nI	nMa
2/15	5.62	2.37	2.10	1.51	0.61	0.76
3/15	6.99	3.01	2.22	1.58	0.64	0.79
4/15	8.58	3.79	2.30	1.62	0.66	0.80
5/15	11.91	5.41	2.41	1.68	0.68	0.83
6/15	7.04	3.01	2.32	1.61	0.66	0.80
7/15	22.03	10.43	2.55	1.75	0.72	0.85
8/15	8.34	3.65	2.35	1.62	0.67	0.79
9/15	9.18	4.07	2.34	1.61	0.67	0.79
10/15	10.18	4.58	2.31	1.58	0.66	0.78
11/15	11.85	5.42	2.30	1.56	0.66	0.77
12/15	15.35	7.16	2.34	1.58	0.67	0.78
13/15	20.93	9.96	2.30	1.55	0.66	0.76

It can be noted that as the complexity of BFA changes based on the number of the erroneous bit, as demonstrated in (24), the presented BFA complexity calculation is done with the assumption that half of the received bits are erroneous and thus provides an average decoding time.

Tables III and IV demonstrate the complexity values calculated using the method described in Section V represented by the total number of instructions and the total number of

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

TABLE IV: ATSC 3.0 LDPC decoders number of instructions and memory accesses (code length = 16200)

CR	NNL ($\times 10^6$)		WBF ($\times 10^6$)		BFA ($\times 10^6$)	
	nI	nMa	nI	nMa	nI	nMa
2/15	0.83	0.35	0.38	0.29	0.12	0.15
3/15	1.32	0.56	0.50	0.36	0.15	0.18
4/15	1.45	0.63	0.50	0.36	0.14	0.18
5/15	2.11	0.93	0.57	0.40	0.16	0.20
6/15	1.81	0.77	0.59	0.41	0.17	0.20
7/15	1.98	0.86	0.60	0.41	0.17	0.20
8/15	2.27	1.00	0.62	0.42	0.17	0.21
9/15	2.13	0.94	0.56	0.38	0.16	0.19
10/15	2.86	1.29	0.62	0.42	0.17	0.20
11/15	2.76	1.26	0.55	0.38	0.16	0.19
12/15	3.72	1.73	0.57	0.39	0.16	0.19
13/15	5.47	2.60	0.59	0.39	0.17	0.19

memory accesses required in a single decoding iteration of the BFA, WBF and the proposed NNL decoding algorithms for the ATSC 3.0 codes.

By analyzing the values in Tables III and IV, one can assume that the NNL decoder decoding time would be higher than the WBF decoding time, but this assumption would not be accurate even when considering the additional implementation overhead [17]. Such an assumption would be discarding two important factors, the first is the cache memory available in all practical processors that, combined with the small memory footprint of the three decoders, eliminates the need for a high number of memory accesses by storing the C and V matrices of the size P and N respectively in the cache memory thus speeding up the over-all processing [35].

The second factor is that the instructions are not necessarily executed sequentially and that currently used processors deploy the superscalar pipelining that allows for the execution speedup by benefiting from the intrinsic parallelism of the algorithm, therefore executing multiple instructions at the same time, nevertheless, any dependency in the code can limit the superscalar, especially conditional operations [36].

The proposed NNL decoding algorithm was designed with superscalar in mind by eliminating of conditional operations to avoid any branch penalties and allow for a considerable performance enhancement while the WBF and the BFA still require a high number of conditional operations relative to the total number of operations, in both cases the benefit of the superscalar pipelining is limited.

The result of both cache memory speedup and the superscalar pipelining speedup is clear in the complexity of an implementation of the three decoding methods with a clear additional speedup in the case of the proposed NNL decoder.

To calculate the decoding time for each of the decoders three states of the art mobile phones were selected, the iPhone Xs from Apple, the Galaxy S10 from Samsung and the Pixel 3XL from Google. The cycle time and memory access time is calculated based on the mobile phone specification and the values are used to calculate the decoding time for the decoders.

Although the mobile phone companies don't usually announce the specification of the processor and the memory used in their products, those specifications have been identified by tech enthusiasts. The iPhone XS deploys an A12 Bionic 2.49 GHz processor and a 64-bit single-channel 2133 MHz LPDDR4X memory while the Samsung Galaxy S10 uses a Samsung Exynos 9820 2.7 GHz processor and a Samsung K3UH7H70AM-AGCL 2133 MHz memory. The Google Pixel 3XL uses a Qualcomm Kryo 385 2.8 GHz processor and an MT53D512M64D4RQ-053_WT_E LPDDR4 1866MHz memory [37]–[39].

The values of the decoding throughput in kbps, for 50 interactions and after removing the parity bits, for the LDPC decoders are shown in Tables V and VI.

These results confirm the additional speedup that the proposed NNL decoding method gained from the superscalar pipelining in addition to the cache memory speedup that all three decoders gained.

The results for decoders throughput in kbps, for 50 interactions and after removing the parity bits, on the Samsung Galaxy S10 are shown in Fig. 6 and 7. In addition, Fig 8 and 9 demonstrate the decoders' throughput in kbps in relation to their decoding performance represented by the required E_s/N_0 for FER=10^{-4}, these figures demonstrate that even in the cases where a low code-rate BFA decoder would have a better performance than a high code-rate NNL decoder, the NNL decoder would still provide higher throughput.

Only when the E_s/N_0 is around 10 (dB), does the BFA start being more advantageous when the QPSK modulation is used, but in this case, the proposed NNL decoder might not require all 50 iterations to correct the channel-induced errors and its throughput would be higher as well. This would be a dynamic throughput value that would change according to the number of required decoding iterations.

VII. Conclusion

In this paper, we proposed an NNL low complexity LDPC decoder derived from NN and SLP training algorithm and supported with mathematical proof and computer simulation results. We demonstrated the difference in the computational complexity and memory requirements for the proposed midrange NNL, BFA, WBF decoders.

The experimental results showed that the proposed NNL outperformed both the BFA and the WBF algorithms in all of the tested code rates and code lengths of the ATSC 3.0 LDPC. In addition, the mathematical calculation along with the experimental results highlighted the speedup that the NNL decoder benefited from due to superscalar and cache memory.

The proposed NNL decoder demonstrates a low complexity and a mid-range performance; since its error correction performance is superior to the BFA and WBF while having a memory requirement and a complexity that is lower than the WBF and, in some cases, the BFA decoder. These characters makes it a very suitable decoder for software implementations for mobile applications, low-cost DTTB receivers and embedded systems.

Acknowledgment

The authors would like to thank their colleagues at the Postgraduate Program in Electrical Engineering and Computing Department and the Digital TV Research Laboratory at Mackenzie Presbyterian University.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

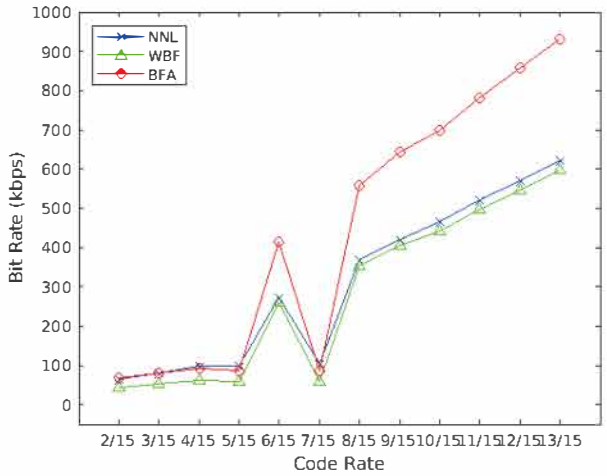


Fig. 6: ATSC 3.0 LDPC decoders throughput in kbps on Samsung Galaxy S10, (code length = 64800).

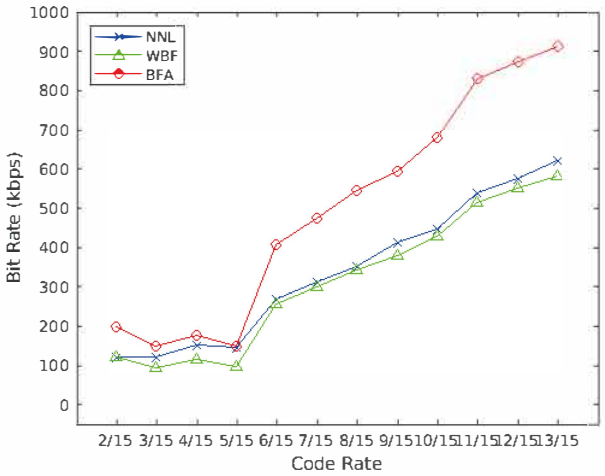


Fig. 7: ATSC 3.0 LDPC decoders throughput in kbps on Samsung Galaxy S10, (code length = 16200).

TABLE V: ATSC 3.0 LDPC decoders throughput in kbps (code length = 64800).

CR	Device	NNL	WBF	BFA
2/15	iPhone XS	58.8	40.5	62.5
	Galaxy S10	63.4	43.7	67.2
	Pixel 3XL	64.9	44.6	69.2
3/15	iPhone XS	75.3	48.8	74.9
	Galaxy S10	81.1	52.6	80.9
	Pixel 3XL	83.3	54.0	83.0
4/15	iPhone XS	90.2	56.3	86.1
	Galaxy S10	97.5	60.8	93.2
	Pixel 3XL	100.1	62.5	95.9
5/15	iPhone XS	91.3	53.3	81.1
	Galaxy S10	98.6	57.6	87.7
	Pixel 3XL	101.4	59.3	90.5
6/15	iPhone XS	254.4	243.4	386.5
	Galaxy S10	273.6	261.0	415.0
	Pixel 3XL	278.2	262.3	421.9
7/15	iPhone XS	97.0	53.1	80.4
	Galaxy S10	104.9	57.4	87.0
	Pixel 3XL	108.0	59.2	89.9
8/15	iPhone XS	340.9	327.7	519.2
	Galaxy S10	366.8	351.6	557.9
	Pixel 3XL	372.9	353.4	562.5
9/15	iPhone XS	391.4	379.7	597.9
	Galaxy S10	419.5	406.1	643.5
	Pixel 3XL	426.6	410.5	654.6
10/15	iPhone XS	434.9	413.6	649.0
	Galaxy S10	466.2	441.8	697.3
	Pixel 3XL	474.0	446.4	709.0
11/15	iPhone XS	483.4	461.8	725.1
	Galaxy S10	521.4	496.3	779.9
	Pixel 3XL	530.4	501.7	793.3
12/15	iPhone XS	527.3	508.8	797.2
	Galaxy S10	568.8	544.4	858.1
	Pixel 3XL	578.6	550.3	872.8
13/15	iPhone XS	580.4	556.8	870.5
	Galaxy S10	623.2	596.1	929.6
	Pixel 3XL	634.0	602.7	945.6

TABLE VI: ATSC 3.0 LDPC decoders throughput in kbps (code length = 16200).

CR	Device	NNL	WBF	BFA
2/15	iPhone XS	114.0	114.0	183.4
	Galaxy S10	124.1	124.1	200.9
	Pixel 3XL	125.3	123.8	200.0
3/15	iPhone XS	113.0	89.1	140.6
	Galaxy S10	121.7	95.9	150.7
	Pixel 3XL	124.6	979	153.4
4/15	iPhone XS	143.0	106.8	165.4
	Galaxy S10	153.4	115.6	179.5
	Pixel 3XL	157.2	117.9	183.6
5/15	iPhone XS	135.2	90.9	138.8
	Galaxy S10	146.5	97.7	150.7
	Pixel 3XL	149.4	100.2	154.6
6/15	iPhone XS	253.1	243.4	383.5
	Galaxy S10	269.3	258.3	408.3
	Pixel 3XL	275.4	260.6	417.2
7/15	iPhone XS	289.5	278.6	447.4
	Galaxy S10	314.2	301.3	476.3
	Pixel 3XL	319.7	302.4	483.0
8/15	iPhone XS	330.9	318.4	511.4
	Galaxy S10	351.6	344.4	544.4
	Pixel 3XL	361.4	345.1	540.9
9/15	iPhone XS	387.4	358.2	558.4
	Galaxy S10	412.7	379.7	593.3
	Pixel 3XL	424.1	386.4	612.5
10/15	iPhone XS	413.6	398.0	639.2
	Galaxy S10	448.8	430.5	680.4
	Pixel 3XL	455.3	432.7	687.3
11/15	iPhone XS	504.4	483.4	773.4
	Galaxy S10	539.6	515.6	828.7
	Pixel 3XL	551.0	524.9	831.8
12/15	iPhone XS	538.6	516.6	791.0
	Galaxy S10	575.3	550.3	872.8
	Pixel 3XL	584.2	554.7	876.5
13/15	iPhone XS	571.3	537.7	856.9
	Galaxy S10	623.2	583.4	914.1
	Pixel 3XL	633.3	587.5	924.7

REFERENCES

[1] I. Lin, B. Jeff, and I. Rickard, "Arm platform for performance and power efficiency — hardware and software perspectives," in *2016 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–5, April 2016.

[2] J. Cho, Y. Woo, S. Kim, and E. Seo, "A battery lifetime guarantee

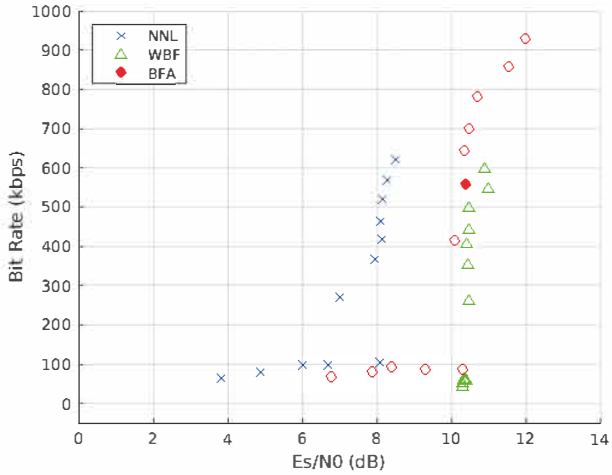


Fig. 8: ATSC 3.0 LDPC decoders throughput in kbps on Samsung Galaxy S10 in relation to their decoding performance, the required Es/N_0 for FER=10^{-4} (dB), (code length = 64800).

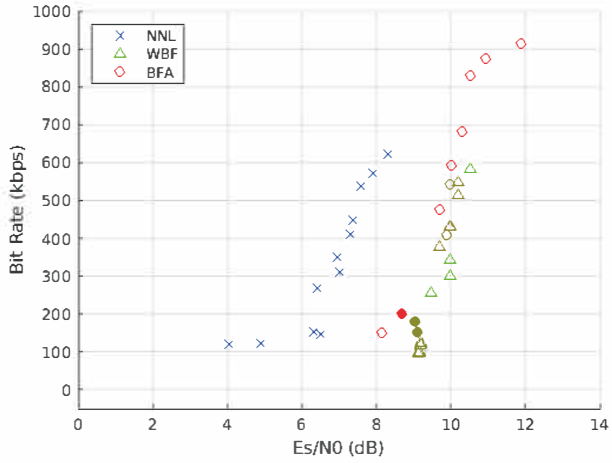


Fig. 9: ATSC 3.0 LDPC decoders throughput in kbps on Samsung Galaxy S10 in relation to their decoding performance, the required Es/N_0 for FER=10^{-4} (dB), (code length = 16200).

scheme for selective applications in smart mobile devices," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 155–163, 2014.

[3] IEEE-Std, "Ieee standard for information technology– local and metropolitan area networks– part 11, amendment 5," *IEEE Std 802.11n-2009*, pp. 1–565, Oct 2009.

[4] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5g new radio," *IEEE Communications Magazine*, vol. 56, pp. 28–34, MARCH 2018.

[5] G. Indumathi and D. A. Joe, "Design of optimum physical layer architecture for a high data rate lte uplink transceiver," in *2013 International Conference on Green High Performance Computing (ICGHPC)*, pp. 1–8, March 2013.

[6] K. Kim, S. Myung, S. Park, J. Lee, M. Kan, Y. Shinohara, J. Shin, and J. Kim, "Low-density parity-check codes for atsc 3.0," *IEEE Transactions on Broadcasting*, vol. 62, pp. 189–196, March 2016.

[7] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, pp. 457–458, Mar 1997.

[8] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, Jan 1962.

[9] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 399–431, Mar 1999.

[10] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, pp. 533–547, Sep 1981.

[11] F. R. Kschischang, B. J. Frey, and H. . Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb 2001.

[12] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Transactions on Communications*, vol. 47, pp. 673–680, May 1999.

[13] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Transactions on Communications*, vol. 50, pp. 406–414, March 2002.

[14] J. Chen and M. P. C. Fossorier, "Density evolution for two improved bp-based decoding algorithms of ldpc codes," *IEEE Communications Letters*, vol. 6, pp. 208–210, May 2002.

[15] C. Chen, Y. Xu, H. Ju, D. He, W. Zhang, and Y. Zhang, "Variable correction for min-sum ldpc decoding applied in atsc3.0," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–5, June 2018.

[16] S. Myung, S. Park, K. Kim, J. Lee, S. Kwon, and J. Kim, "Offset and normalized min-sum algorithms for atsc 3.0 ldpc decoder," *IEEE Transactions on Broadcasting*, vol. 63, pp. 734–739, Dec 2017.

[17] F. Jerji and C. Akamine, "Advanced isdb-t and atsc 3.0 ldpc codes performance and complexity comparison," *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 254–262, 2022.

[18] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, pp. 2711–2736, Nov 2001.

[19] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[20] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Jun 1960.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct 1986.

[22] A. R. Karami, M. A. Attari, and H. Tavakoli, "Multi layer perceptron neural networks decoder for ldpc codes," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Sept 2009.

[23] A. Karami and M. A. Attari, "Novel ldpc decoder via mlp neural networks," 2014.

[24] L. Fay, L. Michael, D. Gómez-Barquero, N. Ammar, and M. W. Caldwell, "An overview of the atsc 3.0 physical layer specification," *IEEE Transactions on Broadcasting*, vol. 62, pp. 159–171, March 2016.

[25] ATSC, "Physical Layer Protocol Standard, A/322:2017," standard, The Advanced Television Systems Committee, USA, June 2017.

[26] ATSC, "ATSC Digital Television Standard, A/53, Part 1:2007," standard, The Advanced Television Systems Committee, USA, Jan. 2007.

[27] L. Michael and D. Gómez-Barquero, "Bit-interleaved coded modulation (bicm) for atsc 3.0," *IEEE Transactions on Broadcasting*, vol. 62, pp. 181–188, March 2016.

[28] R. Bose and D. R. Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68 – 79, 1960.

[29] L. Michael and D. Gómez-Barquero, "Modulation and coding for atsc 3.0," in *2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1–5, June 2015.

[30] DVB, "Part 1: DVB-S2, ETSI EN 302 307-1 V1.4.1," standard, Digital Video Broadcasting, FRANCE, Nov. 2014.

[31] A. Ghazel, E. Boutillon, J. . Danger, G. Gulak, and H. Laamari, "Design and performance analysis of a high speed awgn communication channel emulator," in *2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE Cat. No.01CH37233)*, vol. 2, pp. 374–377 vol.2, Aug 2001.

[32] F. Jerji and C. Akamine, "Gradient bit-flipping ldpc decoder for atsc 3.0," in *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–4, Jun 2019.

[33] J. Harrison, T. Kubaska, S. Story, M. S. Labs, and I. Corporation, "The computation of transcendental functions on the ia-64 architecture," *Intel Technology Journal*, vol. 4, pp. 234–251, 1999.

[34] A. Nannarelli, "A multi-format floating-point multiplier for power-efficient operations," in *2017 30th IEEE International System-on-Chip Conference (SOCC)*, pp. 351–356, Sept 2017.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

[35] S. Ristov and M. Gusev, "Superlinear speedup for matrix multiplication," in *Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces*, pp. 499–504, 2012.
[36] R. Khanna, S. Verma, R. Biswas, and J. Singh, "Implementation of branch delay in superscalar processors by reducing branch penalties," in *2010 IEEE 2nd International Advance Computing Conference (IACC)*, pp. 14–20, 2010.
[37] D. Yang and S. Wegner, "Apple iphone xs max teardown," 2018.
[38] M. Alarcon, D. Yang, S. Wegner, and A. Cowsky, "Samsung galaxy s10+ teardown," 2019.
[39] D. Yang and S. Wegner, "Google pixel 3 xl teardown," 2018.

FADI JERJI (S'18) received a B.S. degree in computer engineering in 2010 and an M.S. degree in electrical and computation engineering from Mackenzie Presbyterian University, São Paulo, Brazil, in 2019. He is currently pursuing a Ph.D. degree in electrical and computation engineering at Mackenzie Presbyterian University. Since 2017 he has been a post-grad Researcher with the Digital TV Research Laboratory at Mackenzie Presbyterian University.

LEANDRO A. SILVA is a Computer Engineer with a Ph.D. in Systems Engineering from the School of Engineering of the University of São Paulo, Brazil. He is currently an Adjunct Professor at the School of Computing and Informatics at Mackenzie University. He works on artificial neural networks, pattern recognition, data mining, machine learning, and big data analytics.

CRISTIANO AKAMINE received a Ph.D. degree in electrical engineering from the State University of Campinas, Brazil, in 2011. He is a Professor at Mackenzie Presbyterian University, where he is a Coordinator of the Digital TV Research Laboratory. He is a member of the Board of the Brazilian Digital Terrestrial Television Forum and Society of Brazilian Broadcast Engineers (SET). He works with the ISDB-TB broadcasting standardization and holds several patents, intellectual property licenses. He also has published numerous articles and has a Brazilian scientific grant of Productivity and Technological Development and Innovative Extension—Level 2 from the National Council of Technological and Scientific Development. He has also served as a reviewer for several periodicals and conferences and has participated as a Guest Editor in the Special Issue Point-to-Multipoint Communications and Broadcasting in 5G of IEEE Communications Magazine.

Received in 2022-09-26 | Approved in 2022-12-21

http://www.set.org.br/ijbe/ doi:10.18580/setijbe.2022.1 Web Link: http://dx.doi.org/10.18580/setijbe.2022.1

Spectrum Availability for the Deployment of TV 3.0

Thiago Aguiar Soares
Paulo E. R Cardoso
Ugo Silva Dias

CITE THIS ARTICLE

Soares, Thiago Aguiar; Cardoso, Paulo E. R ; Dias, Ugo Silva; 2022. Spectrum Availability for the Deployment of TV 3.0. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

Spectrum Availability for the Deployment of TV 3.0

Thiago Aguiar Soares, Paulo E. R Cardoso, Ugo Silva Dias

***Abstract*—In this paper, we study the current and future spectrum availability of the VHF and UHF bands in Brazil for the deployment Next-Generation Digital Terrestrial Television Systems, which are being studied under the "TV 3.0 Project" initiative, coordinated by The Brazilian Digital Terrestrial Television System Forum (SBTVD Forum). Coverage simulations of all expected operating television stations were computed in different scenarios to estimate the spectrum usage over the Brazilian territory. Results indicate that even after the analog TV switch-off there will be no spectrum availability in the main metropolitan regions for simulcast transmissions between the current ISDB-Tb System and the future TV 3.0. Hence, hybrid approaches should be implemented to smoothly introduce a new digital television system in Brazil.**

***Index Terms*—Digital Terrestrial Television (DTT), Next-Generation Digital Terrestrial Television Systems, Regulation, Spectrum Policies, TV 3.0.**

I. Introduction

Digital terrestrial television (DTT) plays an important role worldwide in providing free-to-air audiovisual content with better picture and sound quality. DTT is being introduced in the VHF/UHF bands by administrations from 1997. The first-generation DTT systems were standardized by International Telecommunications Union (ITU) in Recommendation ITU-R BT.1306 - Error correction, data framing, modulation, and emission methods for digital terrestrial television broadcasting [1]. Four systems are currently standardized in the above-mentioned recommendation:

- ATSC - Advanced Television Systems Committee;
- DVB-T - Digital Video Broadcasting - Terrestrial;
- ISDB-Tb - Integrated Services Digital Broadcasting - Terrestrial;
- DTMB - Digital Television Terrestrial Multimedia Broadcasting

Since their standardization, DTT Systems have been widely implemented worldwide and several countries have started switching-off analog TV services. In many countries, this process has still been completed, mainly in developed countries in North America, Europe, and Asia [2].

Driven by the rapid improvement of TV sets definition and the necessity of higher quality and connectivity, DTT systems continue their technological evolution. In November 2010, ITU started to standardize second-generation systems with the approval of Recommendation BT.1877: Error-correction, data framing, modulation, and emission methods and selection guidance for second-generation digital terrestrial television broadcasting systems [3]. In its recent version there are three standardized systems:

- ASTC 3.0 - Advanced Television Systems Committee 3.0;
- DVB-T2 - Digital Video Broadcasting - Terrestrial 2;
- DTMB-A - Digital Television Terrestrial Multimedia Broadcasting

These recommended systems have already been implemented around the world and new systems are also being developed for future operation. Advanced Integrated Services Digital Broadcasting - Terrestrial, an evolution of ISDB-Tb System [4] and 5G Broadcasting, based on a set of 3GPP specifications [5], are well-known candidates for the next standardized Second Generation DTT Systems.

The second generation of digital terrestrial television broadcasting transmission systems is meant as systems offering higher bit rate capacity per Hz and better power efficiency in comparison to the systems described in Recommendation ITU-R BT.1306 and there is no general requirement for backward compatibility with first-generation systems [3]. So, transitioning from first to second-generation DTT systems will require spectrum availability.

In Brazil, studies for Next-Generation Digital TV Systems have already been initiated. In July 2020, The Brazilian Digital Terrestrial Television System Forum (SBTVD Forum) released a Call for Proposals (CfP) seeking input from interested organizations for Brazil's next-generation Digital Television system components and sub-components [6] . The initiative is called "TV 3.0 Project". The CfP received in total, considering its 6 system components (Over-the-air Physical Layer, Transport Layer, Video Coding, Audio Coding, Captions, and Application Coding), 36 responses from 21 different organizations worldwide [1].

The TV 3.0 Project Phase 2 "Testing and Evaluation of the candidate technologies" was carried out from July 2021 to December 2021. Considering the test results, as well as the market and intellectual property aspects of the candidate technologies, some components have already been defined[1]. Complementary tests for selection of the physical layer technology, development of the necessary adaptations and extensions to the transport layer specification, subjective assessment of the video coding quality (determination of the necessary bitrate), development of adaptations and extensions of the Application Coding, and other activities are expected to be developed until 2023 [7].

Nonetheless, the availability of spectrum resources for DTT Services is declining worldwide, especially in the UHF Band. The 700 MHz band is already globally harmonized for the deployment of IMT Services and the 600 MHz band is also being considered in many countries for services rather than DTT. Furthermore, there is an intense debate for the next ITU-R World Radiocommunication Conference over the full review

T. A . Soares is with the Ministry of Communications (MCom) and University of Brasilia (UnB) (thiago.soares@mcom.gov.br).

Paulo E. R Cardoso is with the National Telecommunications Agency (Anatel) (perc@anatel.gov.br).

U. S. Dias is with the Department of Electrical Engineering, University of Brasilia (UnB) (udias@unb.br).

The information and views set out in this paper are those of the authors and do not necessarily reflect the official opinion of the institutions with which they are affiliated.

[1]The information about candidate technologies can be found at https://forumsbtvd.org.br/tv3_0/

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

of the frequency band 470-694 MHz in ITU Region 1 [8], which includes Europe, Africa, the northern part of Asia, and part of the Middle East.

This paper analyzes the current and the future spectrum usage of television services in Brazil to develop realistic transition models approaches for the deployment of Next Generation Digital Television Services. In Section II, we begin by analyzing the current spectrum usage of digital television in Brazil. Next, in Section III, we discuss tendencies for UHF and VHF Band usage for DTT Systems in near future and we discuss some transition policies for next-generation digital television systems. Lastly, in Section IV, we draw some conclusions.

II. Current Spectrum Usage of Television Services in Brazil

The first phase of the transition from analog to digital terrestrial television broadcasting in Brazil was successfully completed in January 2019. It has included 1,379 municipalities (129.6 million inhabitants, 62.6% of the Brazilian population), including all state capitals, metropolitan areas, and other areas where the analog switch-off was required to clear the 700 MHz band. For the rest of the country (77.3 million inhabitants, 37.4% of the population, distributed in 4,191 municipalities), it is expected that the analog television switch-off will be completed at the end of 2023[9] [10].

Since the introduction of DTT services in Brazil, in the middle 2007, about 19,721 digital channels were planned and included in the Brazilian Master Register for Digital Television Channels, a database for broadcasting channels managed by the National Agency of Telecommunications (Anatel) [11]. Figure 1 shows the current distribution of DTT channels in Brazil.

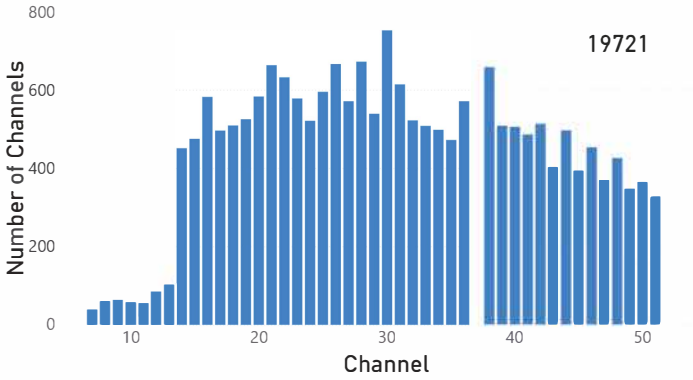


Fig. 1: Distribution of Digital TV Channels in Brazil.

To evaluate spectrum usage of TV Services in the Brazilian territory, simulations were made to estimate the coverage of all operating channels. The applied methodology is described in the following subsections. The web-based spectrum management software, Mosaico (based in Spectrum-E), has been provided through the courtesy of Anatel for the calculations.

A. Database Analysis

Determining the number of operating stations in Brazil is challenging. National policies were established in the past to allow broadcasters to start transmissions even before getting a formal license by Anatel [12]. It has promoted a fast process for installing new DTT channels because broadcasters were able to provide services by only paying radio frequency usage fees and just starting the required application for obtaining the license. However, many broadcasters had started licensing process and did not install transmission sites or, on the opposite - they had installed transmission sites and did not finalize the licensing process. The result is that only 43% of authorized digital stations are currently licensed [11]. However, this number does not reflect the real number of operating stations in Brazil.

To achieve better accuracy of coverage simulations, it is primordial that the database of DTT stations reflects as precisely as possible the currently operating ones. Considering that a database of operational DTT stations is unavailable, as mentioned before, we have made some assumptions to retrieve data from Anatel's database source. We consider that all registers containing at least a valid DTT authorization act issued by the Ministry of Communications, and an assigned radio frequency act published by Anatel, configure an operational station. These documents were considered for selecting stations due that without both of them Brazilian broadcasters are not able to start transmissions. The obtained result is that 12,711 DTT channels (about 64,45% of the total number of planned channels) are estimated to be operational in Brazil. Figure 2 and Figure 3 show the distribution of the estimated operative DTT Channels in Brazil and comparative percentage, respectively:

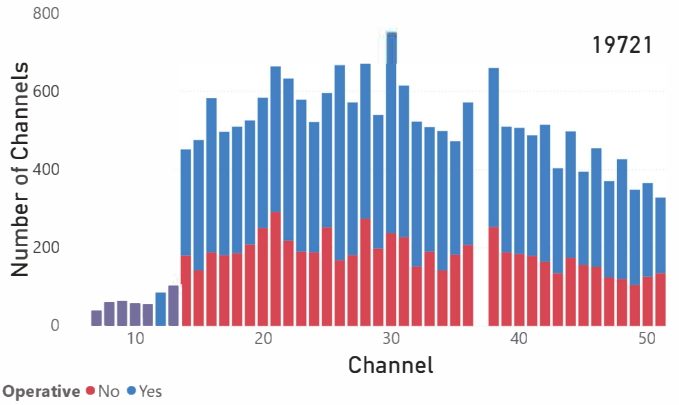


Fig. 2: Distribution of Operative Digital TV Channels in Brazil.

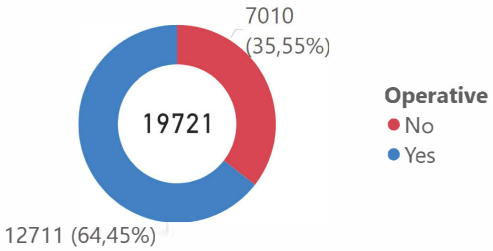


Fig. 3: Percentage of Operative Digital TV Channels in Brazil.

Secondly, as few of the total DTT Channels are currently licensed, the availability of technical data containing the necessary inputs for simulating the estimated coverage is scarce. It occurs because only by completing the registration of the station's parameters (antenna height, radiated diagram, ERP, etc) technical data can be retrieved from Anatel's database.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

So, once again, some approximations were considered. For simulating all predicted coverage, an algorithm from Mosaico software that simulates an ideal station was applied. This algorithm calculates the best antenna diagram and ERP for a station in a determined location. It takes the maximum protected contour, determined by Anatel Technical Regulation [13], and terrain data to perform calculations. The result is that for each channel it is created an ideal station that could reach the maximum protected contour in each direction, from 0 to 360 degrees with a 1-degree step. Figure 4 and Figure 5 show the calculated antenna pattern in different types of terrain profiles. The protected contours are plotted in those figures using Recommendation ITU-R P.1546-5 [14], which is the reference prediction method for determining the maximum protected area of a DTT station, as defined by Anatel [13].

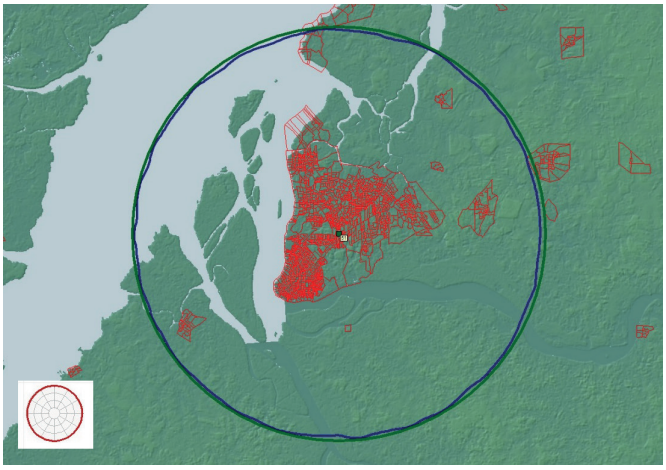


Fig. 4: Example of a computed antenna pattern in flat terrain.

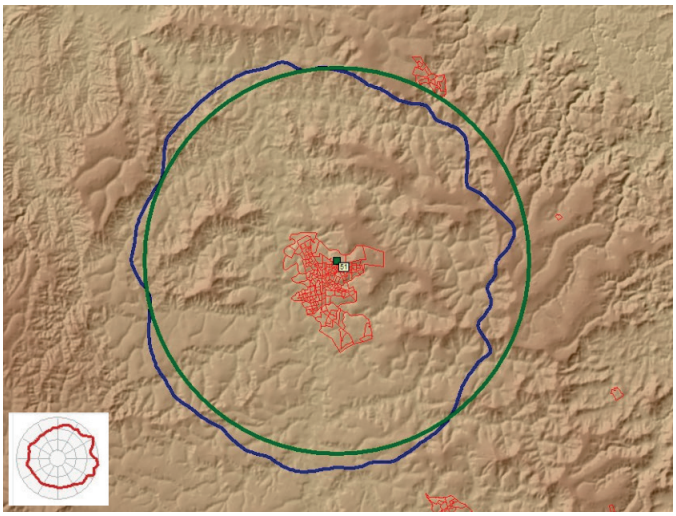


Fig. 5: Example of a computed antenna pattern in rough terrain.

B. Simulation Parameters

The estimation of the coverage of DTT stations involves frequency management aspects and databases with equipment-related information; accurate knowledge of terrain data where the system is to be deployed; and detailed information on the distribution of the population inside the service area [15]. In addition, the nature of the propagation model in use will be of paramount importance for realistic predictions and efficient and precise network dimensioning [15].

TABLE I: ITU-R P.1812 simulation parameters

Parameter	Value
Receiving Height	10 m
Calculated Distance	100 km
Percentage of time	50 %
Percentage of locations	50 %
Subpath	Delta Bullington
Clutter Resolution	Default
Terminal Clutter Losses	Not considered
Profile Sampling	1,000 points

As mentioned before, the web-based spectrum management software, Mosaico, was used to estimate the coverage of all 12,711 operating DTT channels in Brazil. The software makes available various propagation models for DTT coverage estimation. For the purpose of the present article, the Recommendation ITU-R P.1812-5 was chosen. This ITU-R Recommendation provides a deeper consideration of potential propagation phenomena and it will provide more accurate path loss results in some specific links. In consequence, ITU-R states that P.1812 should be used for the detailed evaluation of point-to-area signal levels[16] [15].

Furthermore, ITU-R P.1812 is a modern recommendation (last updated in August 2019) and the ease of reproducibility of results and implementation transparency of the ITU-R P.1812 model can allow for a better degree of standardization of a propagation prediction method for point-to-area terrestrial services in the VHF and UHF frequency bands [17]. Parameters of Table I were used to achieve a balance between speed and accuracy of the simulations.

Finally, to determine the service area it was considered the minimum field strength defined by Anatel on its technical rule: $51 dB\mu V/m$ for channels in the UHF band and $43 dB\mu V/m$ for channels in the VHF Band (see Table 1 in [13]).

C. Results

After establishing all the required parameters, simulations were made to predict the service area of all considered DTT stations. Figure 6 illustrates the predicted coverage of channel 20 in a part of Brazil.

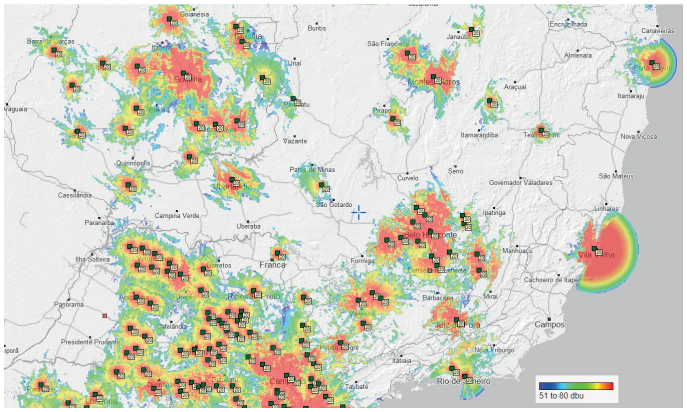


Fig. 6: Example of predicted coverage for DTT Channel 20.

The population distribution inside the service area was calculated considering the percentage of coverage on each

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

sector of the census geographic boundary shapefiles provided by the Brazilian Institute of Geography and Statistics [2]. A municipality was considered covered by a DTT station when at least 90% of its urban population is inside the predicted coverage area. A high coverage percentage was chosen because the predictions are overestimated by the assumptions made to compute ideal antenna patterns and also because the used software automatically reduces terrain resolution on networks with a large number of entries.

Simulations were made considering the current DTT channel distribution. However, until December 2023, analog television will remain operative, as mentioned before. Applying the same estimation methodology, it is estimated that 9,230 analog TV stations are still operative. Most of these stations are allocated in the VHF band as shown in Figure 7.

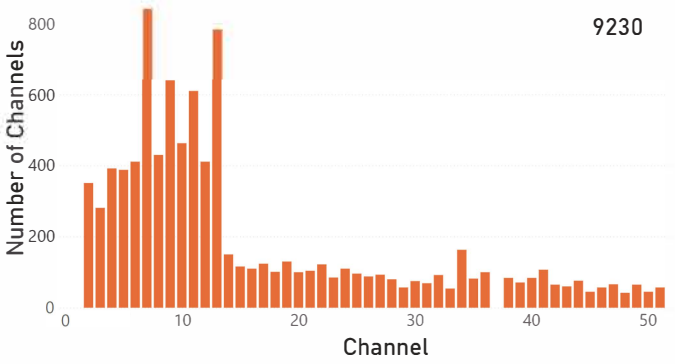


Fig. 7: Distribution of Analog TV Channels in Brazil.

The same methodology for predicting DTT coverage was also applied to simulate the coverage of those estimated operating analog TV stations. The considered minimum field strength, however, differs from digital to analog TV, as stated in Anatel's technical rules. It was used $58dB\mu V/m$ for channels 2 to 6, $64dB\mu V/m$ for channels 7 to 13, and $70dB\mu V/m$ for channels in the UHF Band (see Table 1 in [13]). Figure 8 shows the number of municipalities covered by Analog TV or Digital TV Channels.

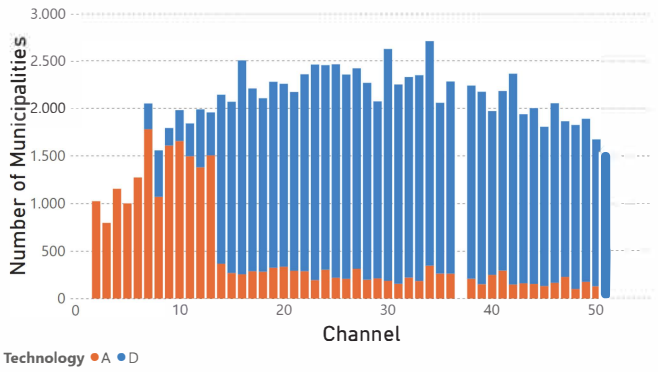


Fig. 8: Number of municipalities covered by Analog TV (A) and Digital TV (D) per channel in Brazil.

The numerical results of the coverage simulations of digital and analog operating stations are summarized in Tables II and III. One table is the opposite of the other: Table II contains the number of municipalities covered by at least one TV channel (digital, analog, or any of them) and Table III has the number of municipalities with no TV coverage (digital, analog, or none of them). Results were also classified by frequency band: sea

- VHF A: Channels 2 - 6 (55 - 72 MHz ; 76 - 88 MHz);
- VHF B: Channels 7 - 13 (174 - 216 MHz);
- UHF A: Channels 14 - 36 (470 - 608 MHz);
- UHF B: Channels 38 - 51 (614 - 698 MHz).

TABLE II: Number of municipalities covered per frequency band

Frequency Band	Municipalities with at least one digital channel	Municipalities with at least one analog channel	Municipalities with at least one channel
All Bands	5312 (96.27%)	4210 (75.57%)	5418 (97.32%)
VHF A Channels 2 - 6	0 (0.00%)	2691 (48.31%)	2691 (48.31%)
VHF B Channels 7 - 13	946 (20.36%)	3800 (67.99%)	4042 (72.73%)
UHF A Channels 14 - 36	5179 (94.47%)	2510 (44.45%)	5232 (95.01%)
UHF B Channels 38 - 51	4417 (81.69%)	1508 (26.61%)	4565 (83.70%)

TABLE III: Number of municipalities with no reception per frequency band

Frequency Band	Municipalities with no digital channels	Municipalities with no analog channels	Municipalities with no channels
All Bands	258 (4.63%)	1360 (24.42%)	152 (2.73%)
VHF A Channels 2 - 6	5570 (100%)	2879 (51.69%)	2879 (51.69%)
VHF B Channels 7 - 13	4624 (83.02%)	1770 (31.78%)	1528 (27.43%)
UHF A Channels 14 - 36	391 (7.02%)	3060 (54.94%)	338 (6.07%)
UHF B Channels 38 - 51	1153 (20.70%)	4062 (72.93%)	1005 (18.04%)

A straightforward finding that can be extracted from data analysis is that digital television has huge penetration in Brazil. As shown in Table II nearly 96% of the Brazilian municipalities can receive at least one DTT channel. Until the end of the analog switch-off, penetration may reach almost 100%. On the other hand, analog TV is still covering about 75% of the municipalities, which shows the importance of well-defined policies for switching-off analog television.

Besides DTT penetration is undoubtedly remarkable in Brazil, the diversity of channels is not so high. Approximately half of the Brazilian municipalities have less than 11 DTT channels, and about 13% have no more than 2, as shown in Figure 9.

Geographically, the Brazilian States from the North, Mid-west, and Northwest regions are the ones with a fewer average of received DTT channels. Figure 10 shows the average of received DTT channels per Brazilian State and Figure 11 contains a map view of the results.

Table II and Figure 8 also show that analog and digital TV occupy spectrum sub-bands distinctively. Analog TV is concentrated on the VHF Band and DTT on the UHF Band. Historically, analog television was first implemented in Brazil in the VHF Band, which partially explains why UHF Band was

[2]Based in 2010 census, available at https://www.ibge.gov.br/geociencias/organizacao-do-territorio/malhas-territoriais/26565-malhas-de-setores-censitarios-divisoes-intramunicipais.html?=&t=downloads

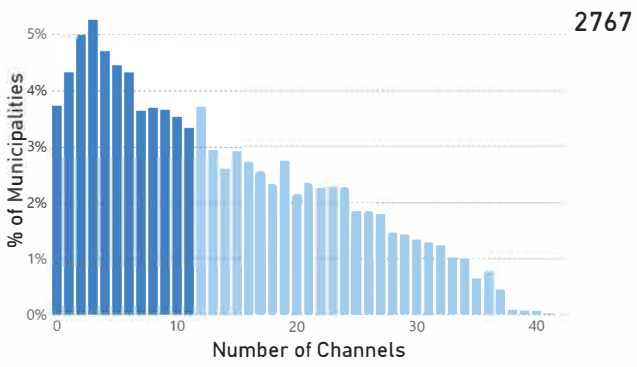


Fig. 9: Percentage of municipalities covered per number of channels in each municipality in Brazil.

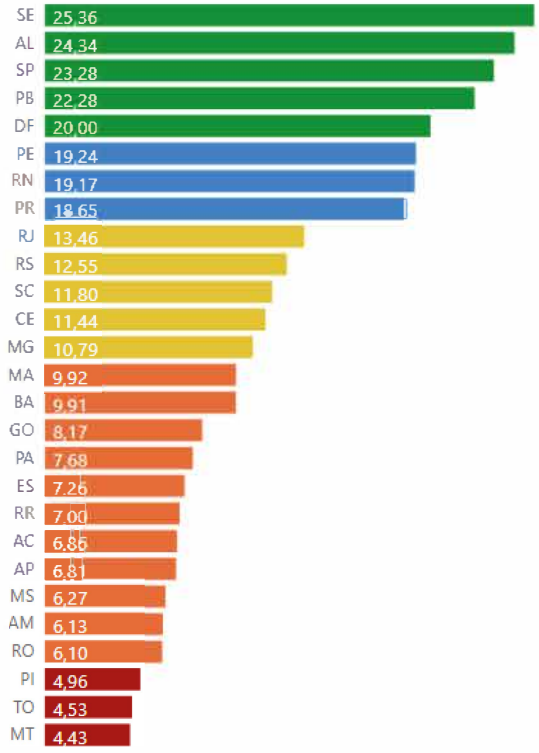


Fig. 10: Average of received DTT channels per Brazilian State: red (less than 5), orange (from 5 to 10), yellow (from 10 to 15), blue (from 15 to 20), and green (more than 20 channels).

preferable for DTT deployment. Furthermore, the susceptibility of some configurations of DTT systems to impulsive noise, more relevant in the lower frequencies, was found by the tests carried out in Brazil in 2001 for defining the Brazilian first-generation DTT system [18].

DTT coverage in Brazil, as mentioned, is concentrated in the UHF band. In particular, the UHF A Band (Channels 14 - 36, 470 - 608 MHz) has a high density of received channels. Figure 12 contains the distribution of the number of received channels in the UHF A Band. It shows that considering the number of received channels per municipality in the UHF A Band, about 50% have no more than 7 DTT channels. In other words, in half of the municipalities in Brazil, at least about 70% of the total amount of spectrum availability of the UHF A band (16 from 23 channels) is currently not being used.

Almost the same behavior occurs in the UHF B band

Fig. 11: Map view - Average of received DTT channels per Brazilian States: red (less than 5), orange (from 5 to 10), yellow (from 10 to 15), blue (from 15 to 20), and green (more than 20 channels).

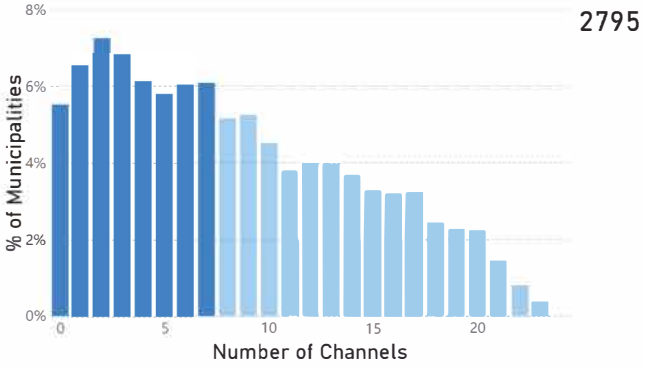


Fig. 12: Percentage of municipalities covered per number of channels in the UHF A Band.

(Channels 38 - 51, 614 - 698 MHz). Despite in about 82% of the municipalities this band being used by at least one DTT channel, about 48% of the municipalities have no more than 3 channels. So, at least 78% of the total amount of spectrum availability of the UHF B band (11 from 14 channels) is not being used in more than half of the municipalities in Brazil. Figure 13 shows the distribution of the number of received channels in the UHF B Band.

Availability of spectrum, however, depends also on the evaluation of the denied spectrum that a TV station produces. The fact that a determined channel is not being used in a municipality does not imply that it is viable to be used. Protection ratios between co-channel and adjacent channels are defined by Anatel [13] (Table 4). So, for a complete evaluation of spectrum availability, it is necessary to develop sharing studies between the operating TV channels and possible new radiocommunications services, considering the protection criteria and minimal field strength for the operation of the services.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

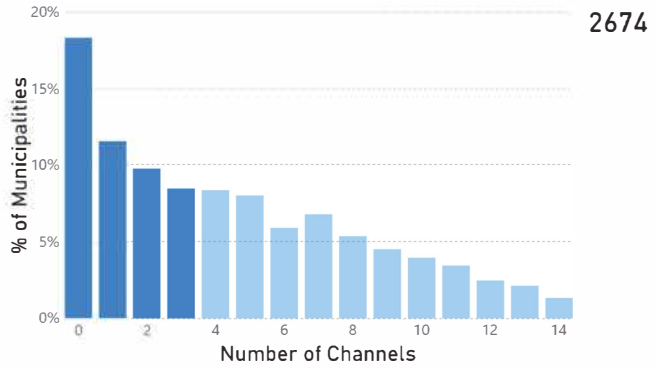


Fig. 13: Percentage of municipalities covered per number of channels in the UHF B Band.

III. Future Spectrum Usage

In Section II we showed how spectrum is currently being used by television services in Brazil, based on coverage simulations of all TV stations that are estimated to be in operation. However, spectrum usage is dynamic and changes practically everyday. In Brazil, there are clearly defined spectrum policies that directly impact television services for the next years. Furthermore, new technologies will certainly drive new possibilities for spectrum usage. The next subsections describe envisaged spectrum policies that can affect the provision of DTT in Brazil.

A. Spectrum Usage After the Analog TV Switch-off

As mentioned in Section II, television analog switch-off is expected to be completed in Brazil by the end of 2023. So, it is estimated that all 9,230 analog stations (see Figure 7) will soon no longer be operative, impacting 4,210 municipalities (75.58%) that receive at least one analog channel, as shown in Table II. A direct impact of the analog switch-off is that spectrum usage of the VHF Band will suddenly decrease, considering that about 65% of analog TV channels are operating in this band, as shown in Figure 14. The impact o UHF Band usage will also be relevant - 3,225 analog channels in this band will also cease operations.

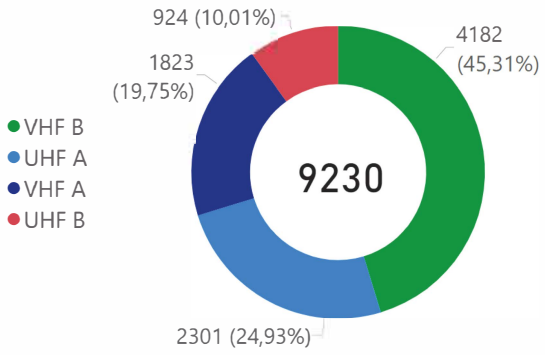


Fig. 14: Number (Percentage) of operative analog TV channels per frequency band.

On the other hand, as shown in Figures 2 and 3, there is a huge number of planned DTT channels (7,010) that are currently not operative. From this total amount, analysis of the data obtained from the Ministry of Communications by means of the Brazilian "Right to Information Law" [19] indicates that 3,229 channels (46%) will not begin operation in the short or mid-term. These channels were reserved for public policies that were not initiated or concluded (for educational and public broadcasting), or were planned for transitioning analog TV channels for digital operation, but were not used. The other 3,781 channels (54%) are mainly located in regions where simulcasting is still ongoing and clear public policies have already been defined.

Specifically for these regions, the Brazilian Ministry of Communications established in 2021 a government program to facilitate the transition for DTT [20]. The main objective of the program is to install complete shared DTT transmission sites in 1,638 small municipalities where only analog TV stations are operative. Besides transitioning current analog TV channels, the program also provides rules for expanding the variety of TV content for the population by adding two new DTT channels (up to 8 different TV programs) on each transmission site for public broadcasters. Funding for the program is provided by resources from the 700 MHz band auctioning process for 4th generation IMT Advanced Systems, which account for approximately R$ 850 million (US$ 160 million) specifically reserved as counterparts for the digital dividend [21].

To accomplish the task, a huge spectrum planning process for including new DTT channels has been carried out by Anatel. Analyzing Anatel's database from April 2021 to May 2022, it was planned 4,661 channels in the 1,638 municipalities of the public program. The effective operation of the channels, however, depends on qualifying analog TV broadcasters and the municipalities, considering the requirements established in Chapter II [20].

So, channels located in those 1,638 municipalities have a higher probability of starting operating, as public policies for funding transmission sites have already been established. Data analysis also indicates that some channels were recently included by request of broadcasters that wish to expand their existing DTT coverage. These channels were also considered to have a high probability to be licensed sooner and starting operating. Figure 15 summarizes the obtained results:

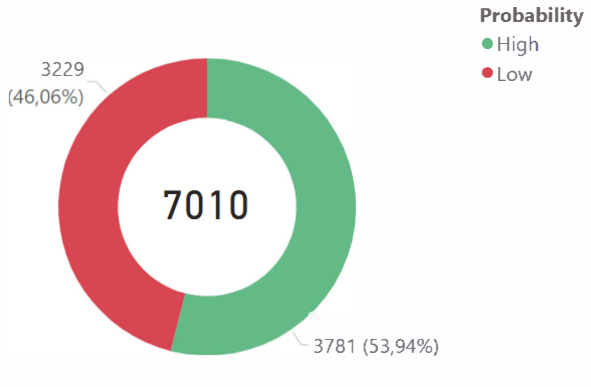


Fig. 15: Number of DTT Channels that are not operatively classified by probability to start operation in the short-term.

Lastly, the conclusion is that besides 9,230 analog channels will soon cease operations with the analog switch-off in Brazil,

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

about 4 thousand new DTT channels are expected to start transmission in the short-term. Hence, DTT channel distribution in Brazil after the analog switch-off is expected to contain about 16,492 operative channels as illustrated in Figure 16.

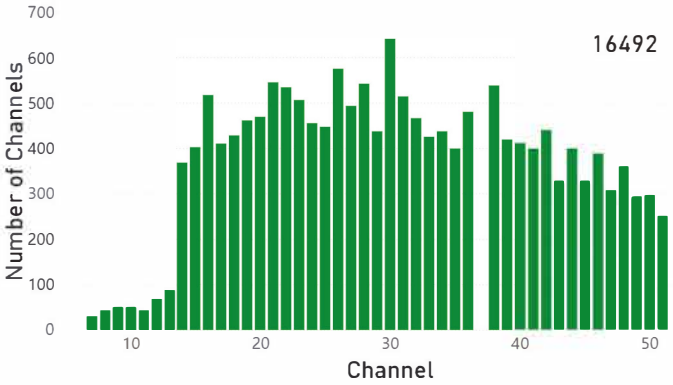


Fig. 16: Expected distribution of Digital TV Channels in Brazil after the analog switch-off.

The next subsection contains the impact of these channels on the transition to Next-Generation DTT Systems in Brazil.

B. Spectrum availability for Next-Generation DTT Systems

The analog switch-off will release huge portions of spectrum, specially in the VHF band, as mentioned in the last subsection. In Brazil, VHF A band is not used for DTT as defined in the technical regulation normative [13]. So, VHF channels 2 to 6 (54 - 72 MHz; 76 - 88 MHz) will be completely released from television services after 2023.

In addition, currently there are no defined public policies for broadcasting services in the band of channels 2 to 4 (54 - 72 MHz). This band is being considered for the usage of TV White Spaces (TVWS) [22]. Public policies for the usage of channels 5 and 6 (79 - 88 MHz), however, have already been established in Brazil for FM broadcasting services by means of a presidential decree [23]. Further, technical parameters for channel planning on this band were established by Anatel in [24] and [25].

SBTVD Forum studies are also not considering VHF A Band for the deployment or Next-Generation TV Systems in Brazil. TV 3.0 over-the-air physical layer is to consider that it should, in principle, be deployed in the bands currently allocated for DTT in Brazil (High-Band VHF and UHF), using the 6 MHz channel raster and it should co-exist with adjacent ISDB-Tb channels for a long time without mutual interference [6]. Hence, only VHF B and UHF Band will be available for DTT services.

Considering the expected DTT channel distribution in VHF B and UHF Bands, as shown in Figure 16, coverage simulations were computed to estimate the spectrum usage after the analog switch-off in Brazil. Figure 17 shows the number of municipalities covered by at least one DTT Channel. Numerically, the results are presented in Table IV.

To better evaluate the expected spectrum availability in a determined region, a spectrum index (I) was computed by normalizing the total number of received channels in the ith municipality (R_i) with the total number of allocated channels for DTT services (A), as showed in Equation 1.

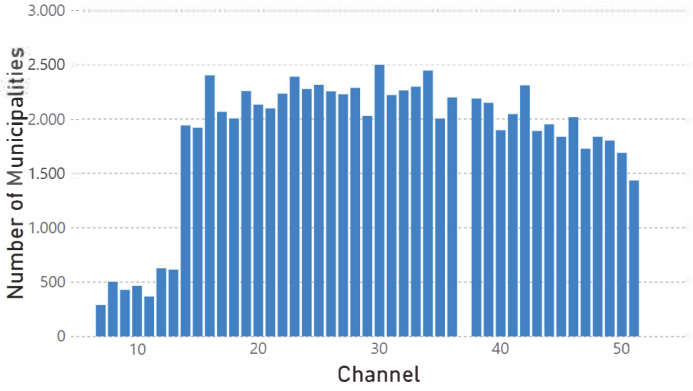


Fig. 17: Number of municipalities covered by at least one DTT Channel.

TABLE IV: Number of municipalities with at least one digital channel per frequency band.

Frequency Band	Municipalities with at least one digital channel (percentage)
All Bands	5398 (96.91%)
VHF B (Channels 7 - 13)	1502 (26.97%)
UHF A (Channels 14 - 36)	5334 (95.76%)
UHF B (Channels 38 - 51)	4724 (84.81%)

$$I = \sum_{i=1}^{N} \frac{R_i}{A} \quad (1)$$

in which:

A = total number of allocated channels for DTT services
R_i = number of received channels in the ith municipality
N = number of municipalities

Considering the available channels between channels 7 to 51, $A = 44$. So, applying the methodology of Equation 1 it is possible to obtain the spectrum index I in all Brazilian municipalities. Results are geographically illustrated in Figure 18.

An important conclusion that can be drawn from the results shown in Figure 18 is that besides there is a huge number of municipalities where the spectrum occupancy in VHF B and UHF Bands is not high (green areas), the main metropolitan regions (yellow and red areas) are densely used by DTT services. These regions concentrate about 73% of the Brazilian population. Hence, new approaches for transitioning to Next Generation Television Systems should be considered. The next subsection contains some proposed transition policies based on the above-mentioned results.

C. Transition Policies for TV 3.0

As could be found in the coverage analysis, spectrum occupancy of DTT services is not uniformly distributed. Some regions in Brazil have a high number of received DTT channels and others lack of it. Figure 18 was classified into three regions: $I \leq 0.3$ (green), $0.3 < I < 0.6$ (yellow) and $I \geq 0.6$ (red).

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

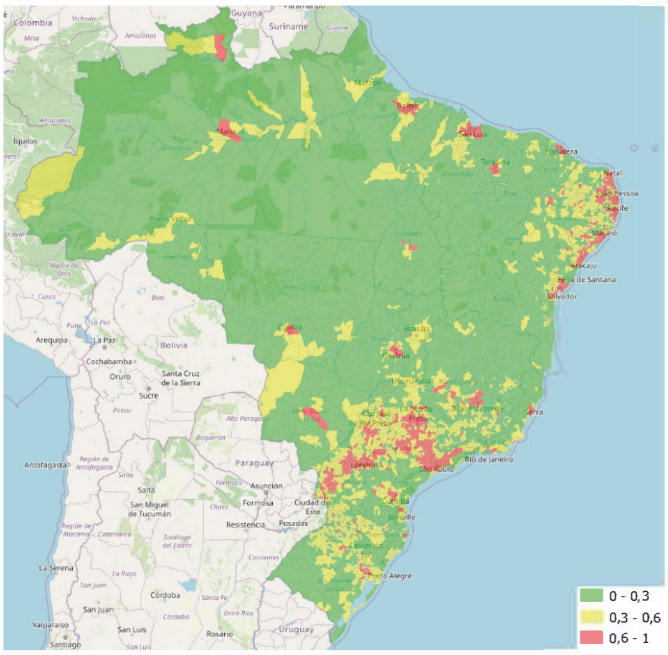


Fig. 18: Spectrum index I per Brazilian municipality.

Ideally, a smooth transition to TV 3.0 would require an additional 6 MHz channel for the transmission of new digital carriers, as there is no general requirement for backward compatibility with first-generation systems, as mentioned in Section I. So, regions where $I \leq 0.5$ in principle would allow simulcasting of first and second generation of DTT systems. However, to avoid interference in zones where spectrum occupancy is higher and facilitate channel planning, establishing a margin would be advisable. So, the $I \leq 0.3$ (green areas of Figure 18) was considered to identify regions where there is sufficient spectrum availability and, consequently, transition policies would be effortless.

For $0.3 < I < 0.6$ (yellow areas of 18) transition for TV 3.0 requires specific policies as it would not be possible to reserve additional 6 MHz channels for the transition of all operative DTT channels. This is a scenario that is currently occurring in the transition for ATSC 3.0 in the United States, as UHF spectrum availability for DTT services has declined after the conclusion of the incentive auction made by the Federal Communications Commission (FCC) to repurpose the 600 MHz Band for both licensed use and unlicensed use [3]. So, because a TV station cannot, as a technical matter, simultaneously broadcast in both ATSC 1.0 and ATSC 3.0 format from the same facility on the same physical channel, FCC has established that local simulcasting must be effectuated through voluntary partnerships that broadcasters seeking to provide Next Gen TV services enter into with other broadcasters in their local markets [26].

Hence, American TV broadcasters are being encouraged to share their facilities in order to implement ATSC 3.0 and also keep ATSC 1.0 transmissions from other partners to minimize the impact on viewers that still do not have ATSC 3.0 receivers. In Brazil, a similar approach should be adopted. TV 3.0 requirements do not include backward compatibility with ISDB-Tb, so, especially in regions where $I > 0.3$, shared transmissions will be required to enable the transition of all broadcasters.

A more challenging situation is to enable transition for TV 3.0 in regions where I is greater than 0.6. In these regions, even sharing facilities may not be sufficient to keep simultaneous transmissions between ISDB-Tb and TV 3.0. Thus, it is required to establish policies to promote direct transition to TV 3.0 in the same 6 MHz channel by switching-off ISDB-Tb transmissions of some broadcasters. To exemplify the transition complexity according to the classified municipalities shown in Figure 18, three cities with different spectrum indexes are included in Figure 19. White slots represent possible channels that can be used for the transition to TV 3.0.

Fig. 19: Example of cities with different spectrum indexes I. White slots represent possible channels that can be used for the transition to TV 3.0. The graphs are colored according to the subtitle of Figure 19.

São Paulo city is a clear example where few empty channels are available for the transition to TV 3.0. It is expected that only 9 slots are suitable for the transition of the expected operating DTT Channels. So, switching-off ISDB-Tb transmissions of some broadcasters will be needed to promote TV 3.0 in the region. Brasilia is an example of a lower complexity city, but sharing infrastructure may be required. Porto Seguro, on the other hand, is an example of a green region where the spectrum availability is sufficient to enable simulcasting ISDB-Tb and TV 3.0.

It is also important to highlight that new DTT channels are expected to begin operations in the mid and long-term (see Figure 15 and consequently increase the spectrum index in some municipalities. Two main policies are imminent: repressed demands for new relay stations and the usage of idle capacity of the facilities installed under the Ministry of Communications Program created by the act [20]. So, the total number of operative DTT channels may vary during a transition to TV 3.0.

So, the spectrum index analysis shows that hybrid approaches should be implemented to introduce new digital television systems in Brazil. Some proposed policies to facilitate the transition to TV 3.0 are listed below:

- ***Reserve High-VHF Band (Channels 7 to 13) for the transition to TV 3.0***. After the analog switch-off, the High-VHF Band will be released in most municipalities. Furthermore, the CfP for TV 3.0 in Brazil includes the

[3] See https://www.fcc.gov/wireless/bureau-divisions/broadband-division/600-mhz-band

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

requirement that the Over-the-air Physical Layer should consider the deployment in this band. Hence, updating the regulation to reserve the band would bring the benefit of having a specific spectrum portion to deploy next-generation DTT networks.

- ***Re-plan current DTT channels in some areas to free continuous spectrum portions***. Current DTT channels were planned in a simulcast scenario where analog TV channels had to be protected to guarantee a smooth transition. So, the planning process was not optimized. Channeling optimization would promote spectrum efficiency and release parts of the UHF spectrum for the deployment of next-generation DTT networks.
- ***Update regulation to allow multi-programming***. The current Brazilian regulation just allows public broadcasters to transmit more than one program in a single 6 MHz channel. However, due to the high utilization of the UHF band by 1st generation DTT transmissions, it will not be possible to allocate a second 6 MHz channel for all broadcasters for the transition to next-generation DTT Systems, mainly in Brazilian state capitals regions and high dense metropolitan areas. So, multi-programming will be necessary to optimize spectrum usage and facilitate the transition.
- ***Promote installation of shared infrastructure***. DTT was implemented in Brazil in a selfish way. Broadcasters have taken advantage of their current analog TV infrastructure to install DTT transmission sites on their own, doing the transition channel by channel. So, there are a few unbidden initiatives in Brazil where DTT transmission sites were shared by more than one broadcaster. Recent public policies have been established to install complete shared DTT transmission sites in small municipalities to facilitate the transition from analog to digital television, but more incisive policies are required for the deployment of shared next-generation DTT transmission sites.
- ***Promote the production of TV sets with ISDB-Tb and TV 3.0 receiving capability as soon as the system technology has been defined***. The spectrum availability analysis has indicated that in many regions there will be no sufficient channels for simulcasting both generations of DTT systems. Hence, a strong policy for the production of TV sets with TV 3.0 receiving capability will be required to accelerate the replacement of the current digital television receivers and minimize the impact on viewers.

IV. Conclusions

This paper presents important contributions to the transition to TV 3.0 in Brazil. It presents coverage simulations results using Recommendation ITU-R P. 1812 that quantifies the spectrum availability in the 174-216 MHz VHF band and 470-698 MHz UHF band. Simulations were made considering also the expected deployment of new DTT channels, based on the assessment of Anatel's database and public policies that have been established by the Ministry of Communications.

Three categories of regions were defined based on their spectrum availability, which was calculated considering the number of receiving DTT channels that were predicted by coverage simulations. Results indicate that in the main metropolitan regions, where concentrates most of the Brazilian population (about 73%), there will be no spectrum availability after the analog TV switch-off for simulcast transmissions between the current ISDB-Tb System and the future TV 3.0. So, hybrid approaches should be implemented to introduce a new digital television system in Brazil.

Furthermore, some policies are proposed to facilitate the transition: reserve High-VHF Band (Channels 7 to 13) for TV 3.0, re-plan current DTT channels in some areas to free continuous spectrum portions, update regulation to allow multi-programming, promote the installation of shared infrastructure and promote strong policies for the production of TV sets with TV 3.0 receiving capability as soon as the system technology has been defined.

As future works, we envisage simulating the estimated denied spectrum generated by DTT channels considering co-channel and adjacent channel interference field strength to optimize DTT frequency planning and better identify areas where spectrum can be used for facilitating the introduction of TV 3.0. Furthermore, we identify that simulations contain conservative values due to software restrictions for the prediction of a high number of channels. So, we consider repeating simulations to have better coverage resolution. Finally, we intend to develop methods to gather better information from Anatel's database about the technical parameters of DTT stations, as database assumptions were made in the present paper to estimate the number of operative DTT channels.

References

[1] International Telecommunications Union (United Nations), *ITU-R Recommendation BT.1306-8, Error-correction, data framing, modulation and emission methods for digital terrestrial television broadcasting*, last updated in February 2015. [Online]. Available: https://www.itu.int/rec/R-REC-BT.1306-8-202004-I/en.

[2] ——, *Status of the transition to digital terrestrial television*. [Online]. Available: https://www.itu.int/en/ITU-D/Spectrum-Broadcasting/DSO/Pages/default.aspx.

[3] ——, *ITU-R Recommendation BT.1877-3, Error-correction, data framing, modulation and emission methods and selection guidance for second generation digital terrestrial television broadcasting systems*, last updated in April 2021. [Online]. Available: https://www.itu.int/rec/R-REC-BT.1877-3-202012-I/en.

[4] M. Nakamura, A. Sato, H. Miyasaka, S. Aasakura, T. Shitomi, N. Shirai, T. Takeuchi, M. Okano, K. Tsuchida, Y. Narikiyo, and K. Murayama, 'A study on the transmission system of an Advanced ISDB-T,' in *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2019, pp. 1–5. DOI: 10.1109/BMSB47279.2019.8971915.

[5] European Telecommunications Standards Institute, *5g broadcast system for linear tv and radio services; lte-based 5g terrestrial broadcast system, Etsi ts 103 720 v1.1.1 (2020-12)*.

[6] Brazilian Digital Terrestrial Television System Forum), *Call for proposals: Tv 3.0 project*. [Online]. Available: https://forumsbtvd.org.br/tv3_0/ (visited on 08/27/2021).

[7] Brazil (Federative Republic of), *Contribution 6/210 submitted to itu-r study group 6 (broadcasting) meeting, Brazilian next generation digital terrestrial television (information document)*. [Online]. Available: https://www.itu.int/md/R19-SG06-C-0210/en.

[8] International Telecommunications Union (United Nations), *Resolution 811 (WRC-19), Agenda item 1.5 for the 2023 World Radiocommunication Conference*. [Online]. Available: https://www.itu.int/dms_pub/itu-r/oth/0c/0a/R0C0A00000D0041PDFE.pdf.

[9] ——, *ITU-R Report BT.2140-13, Transition from analogue to digital terrestrial broadcasting*, last updated in March 2021. [Online]. Available: https://www.itu.int/pub/R-REP-BT.2140-13-2021.

[10] Ministério das Comunicações (Brasil), *Portaria nº 2.992, de 26 de maio de 2017, Estabelece o cronograma de transição datransmissão analógica dos serviços de radiodifusão de sons e imagens e de retransmissãode televisão para o SBTVD-T*, publicada no Diário Oficial da União em 28 de maio de 2017.

[11] National Agency of Telecommunications (Brazil), *Plano Básico de Distribuição de Canais de Televisão Digital*. [Online]. Available: http://sistemas.anatel.gov.br/se/public/view/b/srd.php (visited on 05/13/2022).

[12] Ministério das Comunicações (Brasil), *Portaria nº 159, de 11 de junho de 2013, Autoriza o funcionamento em caráter provisório das entidades prestadoras de Serviços de Radiodifusão e seus ancilares*. publicada no Diário Oficial da União em 12 de junho de 2013.

[13] National Agency of Telecommunications (Brazil), *Requisitos Técnicos de Condições de Uso de Radiofrequências para os Serviços de Radiodifusão de Sons e Imagens e de Retransmissão de Televisão, aprovado pelo Ato nº 9.751, de 06 de julho de 2022*, *publicado no Diário Oficial da União em 18 de julho de 2022*. [Online]. Available: https://informacoes.anatel.gov.br/legislacao/atos-de-requisitos-tecnicos-de-gestao-do-espectro/2022/1688-ato-9751 (visited on 08/15/2022).

[14] International Telecommunications Union (United Nations), *ITU-R Recommendation BT.1546-6, Method for point-to-area predictions for terrestrial services in the frequency range 30 MHz to 4 000 MHz*, last updated in August 2019. [Online]. Available: https://www.itu.int/rec/R-REC-P.1546/en.

[15] A. Aragón-Zavala, P. Angueira, J. Montalban, and C. Vargas-Rosales, 'Radio propagation in terrestrial broadcasting television systems: A comprehensive survey,' *IEEE Access*, vol. 9, pp. 34 789–34 817, 2021. DOI: 10.1109/ACCESS.2021.3061034.

[16] International Telecommunications Union (United Nations), *ITU-R Recommendation BT.1812-5, A path-specific propagation prediction method for point-to-area terrestrial services in the frequency range 30 MHz to 6 000 MHz*, last updated in August 2019. [Online]. Available: https://www.itu.int/rec/R-REC-P.1812/en.

[17] Spectrum Center, *ITU-R P.1812-5, Terrestrial point-to-area service propagation prediction method in the vhf and uhf bands, March 31st 2020*. [Online]. Available: https://public.spectrum.center/public/wp-content/uploads/2020/04/ITU-R-P.1812-5-Report_vfinal.pdf (visited on 10/18/2021).

[18] Research and Development Center in Telecommunications - CPqD, *Planejamento de Canais de Televisão Digital,PD.33.10.53A.0007A/RT-01-AB*, 2003.

[19] Brasil, *Resposta à Manifestação Fala.BR 53125.000975/2022-10*.

[20] Ministério das Comunicações (Brasil), *Portaria nº 2.524, de 04 de maio de 2021, Institui o Programa Digitaliza Brasil, que estabelece as diretrizes para a conclusão do processo de digitalização dos sinais da televisão analógica terrestre no Brasil e dá outras providências*. publicada no Diário Oficial da União em 05 de maio de 2021. [Online]. Available: https://www.in.gov.br/en/web/dou/-/portaria-mcom-n-2.524-de-4-de-maio-de-2021-317928590.

[21] National Agency of Telecommunications (Brazil), *Acórdão nº 242, de 28 de junho de 2021*, *publicado no Boletim de Serviço Eletrônico da Anatel em 29 de junho de 2021*.

[22] ——, *Resolução nº 747, de 05 de outubro de 2021*, *publicado no Diário Oficial da União em 06 de outubro de 2021*. [Online]. Available: https://informacoes.anatel.gov.br/legislacao/index.php/component/content/article?id=1593 (visited on 06/26/2022).

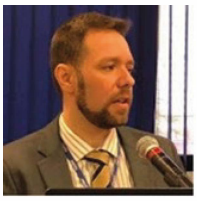
[23] Brasil, *Decreto nº 8,139, de 07 de novembro de 2013, Dispõe sobre as condições para extinção do serviço de radiodifusão sonora em ondas médias de caráter local, sobre a adaptação das outorgas vigentes para execução deste serviço e dá outras providências*. publicado no Diário Oficial da União em 08 de novembro de 2013. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/decreto/d8139.htm (visited on 05/27/2022).

[24] National Agency of Telecommunications (Brazil), *Resolução nº 721, de 11 de fevereiro de 2020*, *publicado no Diário Oficial da União em 12 de fevereiro de 2020*. [Online]. Available: https://informacoes.anatel.gov.br/legislacao/resolucoes/2020/1383-resolucao-721 (visited on 05/27/2022).

[25] ——, *Requisitos Técnicos de Condições de Uso de Radiofrequências para os Serviços de Radiodifusão Sonora em Frequência Modulada, Retransmissão de Rádio na Amazônia Legal e Radiodifusão Comunitária, aprovado pelo Ato nº 8.104, de 10 de junho de 2022*, *publicado no Diário Oficial da União em 18 de junho de 2022*. [Online]. Available: https://informacoes.anatel.gov.br/legislacao/atos-de-requisitos-tecnicos-de-gestao-do-espectro/2022/1687-ato-8104 (visited on 08/15/2022).

[26] Federal Communications Commission (United States of America), *Second further notice of proposed rulemaking, Authorizing Permissive use of the "Next Generation" Broadcast Television Standard*, 2021. [Online]. Available: https://www.fcc.gov/document/fcc-seeks-comment-atsc-30-multicast-requirements (visited on 08/04/2022).

Thiago Aguiar Soares graduated in electrical engineering from the University of Brasília (UnB), Brazil, and post-graduated in telecommunications regulation from the National Institute of Telecommunications (INATEL). He worked for 12 years with the National Agency of Telecommunications (Anatel), coordinating projects on digital television, digital radio, spectrum policies, broadcasting technical regulation, IT systems implementation, among others. Since 2020, he works as Head of Innovation & Regulation of Broadcasting Services in the Brazilian Ministry of Communications. He is Vice-Chairman of the Study Group 6 (Broadcasting) of the Radiocommunication Sector of the International Telecommunications Union (ITU-R). His area of interest is digital broadcasting and spectrum policies.



Paulo E. R Cardoso holds a PhD from DECOM-FEEC-Unicamp (2018) in Digital TV Regulation; Master's degree in Electrical Engineering (Electronics) from DEMIC-FEEC-Unicamp (2005); and degree in Electrical Engineering from FEEC-Unicamp (2002). He is currently Regulation Expert of the National Telecommunications Agency - Anatel, where he works as Coordinator of Systems and Models of Broadcasting Management, in the Spectrum, Orbit and Broadcasting Management, being responsible for studying, improving and elaborating the Technical Regulation of Broadcasting, including its technical and operational requirements acts; for accompanying the development of new broadcasting technologies and prospecting the future of broadcasting, including the definitions of its needs; in addition to accompanying the development of the broadcasting modules of the Mosaic System. He is the leader of the Reporting Group of GRR6: Broadcasting, of the Brazilian Communication Commission - CBC2: Radiocommunications, coordinating the group's international activities, including the coordination of broadcasting service stations in border areas; working mainly in Study Group 6 - SG6 of the International Telecommunication Union - ITU. He participated as an observer of the Federal Government in the Digital Sound Broadcasting tests, both in the tests of the American standard - HD Radio, in 2008 and 2012, as in the tests of the European standard - DRM, in 2010. He worked as a Telecommunications Researcher at Fundação Centro de Pesquisa and Telecommunications Development - CPqD.

Ugo Silva Dias (S'04-M'08-SM'18) was born in Belém, Pará, Brazil, in 1981. He received the B.Sc. degree in Electrical Engineering from The Federal University of Pará, Brazil, in 2004, and the M.Sc. and Ph.D. degrees in Electrical Engineering, from The State University of Campinas, Brazil, in 2006 and 2010, respectively. Since March 2010, Dr. Ugo Dias is an Assistant Professor at University of Brasilia (UnB), Brazil. He is a faculty member of the Department of Electrical Engineering. His main research interests include system development for government, decision-making processes of public agents, AI applied to government systems, and government technologies in general. Prof. Dias has been involved on the Organizing Committee of several conferences. He acts as a Scientific Consultant of the National Council of Scientific and Technological Development (CNPq), Brazil, and he is a Productivity Research Fellow of CNPq. Besides the academic experiences, he also worked in several companies in the ICT industry. Currently, Prof. Dias serves as Vice-President of the Brazilian Telecommunications Society (SBrT), chair of the IEEE ComSoc CN Brazil Chapter, and advisor of the IEEE ComSoc UnB Student Branch Chapter. He is a Senior Member of both IEEE and SBrT, and Member of IEEE Communications Society and Brazilian Communications Committee.

Received in 2022-06-06 | Approved in 2022-07-25

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.2 Web Link: http://dx.doi.org/10.18580/setijbe.2022.2

Advances in video compression: a glimpse of the long-awaited disruption

Thomas Guionnet
Marwa Tarchouli
Sébastien Pelurson
Mickaël Raulet

CITE THIS ARTICLE

Guionnet, Thomas; Tarchouli, Marwa; Pelurson,Sébastien; Raulet, Mickaël ; 2022. Advances in video compression: a glimpse of the long-awaited disruption. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.3. Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

Advances in video compression: a glimpse of the long-awaited disruption

Thomas Guionnet, Marwa Tarchouli, Sébastien Pelurson and Mickaël Raulet

Ateme, {t.guionnet, m.tarchouli, s.pelurson, m.raulet}@ateme.com

***Abstract*— The consumption of video content on the internet is increasing at a constant pace, along with an increase of video quality. As an answer to the ever-growing demand for high quality video, compression technology improves steadily. About every decade, a new major video compression standard is issued, providing a decrease of bitrate by a factor two. Interestingly, the technology does not change radically between codecs generations. Instead, the same block-based hybrid video coding scheme principles and ideas are re-used and pushed further. All along the video compression history, there were several attempts to depart from this model, but none achieved to be competitive. Following the latest codec generation, VVC, the research community has started focusing on deep learning-based strategies. Could it be the new contender to the classical hybrid approach? This paper analyzes the benefits and limitations of deep learning-based video compression methods, and investigates practical aspects such as rate control, delay, memory consumption and power consumption. Overlapping patch-based end-to-end video compression strategy is proposed to overcome memory consumption limitations.**

***Index Terms*—Video Compression, Video codec, MPEG-2, H.264, AVC, HEVC, VVC, artificial intelligence, machine learning, deep learning, end-to-end video encoding.**

I. Introduction

THE consumption of video content on the internet is increasing at a constant pace, along with an increase of video quality. Cisco [1] estimates that by 2023, two-thirds of the installed flat-panel TV sets will be UHD, up from 33 percent in 2018. The bitrate for 4K video is more than double the HD video bitrate, and about nine times more than SD bitrate. As an answer to the ever-growing demand for high quality video, compression technology improves steadily. Video compression is a highly competitive and successful field of research and industrial applications. Billions of people are impacted, from TV viewers and streaming addicts to professionals, from gamers to families. Video compression is used for contribution, broadcasting, streaming, cinema, gaming, video-surveillance, social networks, videoconferencing, military, you name it.

The video compression field stems from the early 80's. Since then, it has grown continuous improvements, and strong attention from the business side - the video encoder market size is planned to exceed USD 2.2 Billion by 2025 [2]. About every decade, a new major video compression standard allows halving the required bitrate to achieve a given quality. The latest milestone is the Versatile Video Coding (VVC) standard, issued in 2020. From generation to generation, until VVC, coding efficiency has been improved by relying on the same principle, that is, the block-based hybrid video coding scheme [4]. For more than 30 years, the video compression field has known no revolution or disruption. Instead, the same principles and ideas have been re-used and pushed further. At each generation, existing tools are enhanced, new local coding tools are added, but the overall structure remains the same. In other words, each generation is a complexified version of the previous one. The algorithmic complexity increase is directly reflected by the implementation complexity. For instance, the VVC verification software model (VTM) is about 10 times slower than its predecessor, the High Efficiency Video Coding (HEVC) verification model (HM). Many attempts have been made to depart from the block-based hybrid scheme, none of them have been successful so far.

As of today, the tremendous progression of video compression technology is not compensating for the increase in the demand for always more and higher quality video services. Therefore, the research effort is still ongoing, seeking improvements over VVC, as it was over each previous codec generation. Indeed, The Joint Video Expert Team (JVET), a working group managed by both ISO/IEC MPEG and ITU-T VCEG international standardization bodies, responsible for the development and support of VVC, is currently conducting explorations beyond VVC. There is a new situation arising though: this exploration is following two distinct tracks. One is "classical", consisting in adding or enhancing coding tools to VVC, while the other is dedicated to the exploration of the usage of machine learning (ML). The field of ML, and more particularly deep learning (DL), has made dramatic advances during the last decade, especially in the computer vision domain. There are several ways of applying ML to video compression. One can consider creating elementary coding tools, replacing, or complementing the existing tools in the hybrid block-based scheme. At the other extremity of the spectrum, one can completely replace the hybrid block-based scheme by a deep learning model. The latter solution is highly disruptive with respect to the current video compression history. Hence the question: to what extent is ML becoming essential to video compression?

The goal of this paper is to analyze the benefits and limitations of deep learning-based video compression methods, and to investigate practical aspects such as rate control, delay, memory consumption and power

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

consumption. In a first part, the evolution of video compression is recounted, with a few words on previous attempts to depart from the hybrid block-based model. In a second part, the deep-learning strategies are described, with a focus on tool-based, end-to-end, and super-resolution-based strategies. In a third part, the practical limitations for industrial applications are analyzed. Finally, a technology is proposed, namely overlapping patch-based end-to-end video compression, to overcome memory consumption limitations. Experimental results are provided and discussed.

II. A SHORT HISTORY OF VIDEO COMPRESSION

A. *CODECs and applications*

The idea of temporal prediction for video compression can be tracked back to 1929, with a patent advocating the coding of successive image differences [3], but the modern history of video compression really starts in the 80's. Two organizations are essentially responsible for video coder/decoder (codec) standardization [5][6]: the International Telecommunications Union – Telecommunication Standardization Sector (ITU-T) Video Coding Expert Group (VCEG), a United Nations Organization (formerly CCITT) [7], and the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC) Moving Picture Expert Group (MPEG). ISO is an independent, non-governmental international organization with a membership of 167 national standards bodies [8]. Aside from standardization, many proprietary or independent codecs exists. Nonetheless, the most successful and well-known line of codecs stems from standardization and constitutes the focus of this paper.

The first standardized video codec, ITU-T H.120 [63], has been issued in 1984, then updated in 1988. It already includes a form of intra prediction (Digital Pulse Coded Modulation, DPCM), scalar quantization, entropy coding in the form of variable length coding (VLC) and motion compensation.

ITU-T H.261 [64] was first issued in 1988. It is dedicated to video telephony and introduces the most important block-based motion compensation and Discrete Cosine Transform (DCT). It is the first practically successful video codec. It was later replaced by the dramatically improved ITU-T H.263 [65] in 1995.

Meanwhile ISO/IEC developed MPEG-1 [66], issued in 1993. It was designed to compress VHS-quality raw video, thus enabling first digital TV applications (videos CD, Cable, satellite). One may note that the best-known part of MPEG-1 is the MP3 audio format it introduced. MPEG-1 has been followed by the non-obviously numbered MPEG-4 part 2 [67], in 1998, also called MPEG-4 visual because of its object-oriented approach.

Interestingly, in the 90's, two lines of standards were coexisting. The ITU-T H.26X line was designed for video telephony, while the ISO/IEC MPEG was meant for digital TV broadcasting. However, both were sharing many technological aspects. There is even a certain degree of compatibility between MPEG-4 visual and H.263. Quite logically, ISO/IEC MPEG and ITU-T VCEG have been joining their effort in the development and publication of common video compression standard, thus starting a particularly successful line of video codecs.

MPEG-2/H.262 [68] has been a tremendous success in the 90's, and the enabler of widespread digital TV. MPEG-2 has been present on cable TV, satellite TV, DVD, and is still running nowadays. In the early 2000's, AVC/H.264 [69] has been a key component of the HD TV development, on traditional networks as well as on internet and mobile networks. AVC/H.264 is also used in HD Blu-Ray discs. Ten years later, in the 2010's, HEVC (H.265) [70] has been the enabler of 4k/UHD, HDR and WCG. Finally, VVC (H.266) [71] has been issued in 2020. Although it is a young codec, not yet widely deployed, it is perceived as an enabler for 8k [9] and as a strong support for the ever-growing demand for high quality video over the internet.

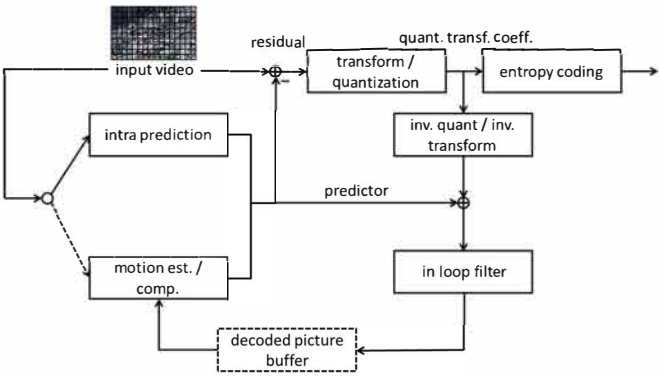


Fig. 1. The block-based hybrid video coding scheme.

B. *The block-based hybrid video coding scheme*

The block-based hybrid video coding model is depicted on Fig. *1*. It constitutes the basis of all current video compression standards. The main elements are

- Intra prediction, for coding intra frames, i.e., frames without temporal dependency, or intra blocks inside inter frames, for managing local areas that cannot be temporally predicted, such as uncovering areas.
- Inter prediction combines the capabilities of keeping a buffer of previously encoded frames and addressing these previous frames with motion compensation for efficient prediction.
- Transform and quantization are applied on residual blocks of pixels output by the prediction step. The transform tends to compact the information on a few coefficients, while the quantization adjusts the trade-off between quality and bitrate. Quantization is a lossy process.
- Entropy coding is a fundamental information theory concept. Its goal is to determine the statistically shortest representation of the data. It is a lossless process.
- In-loop filtering is applied on the frames which are stored for future temporal prediction, to improve their quality, hence the quality of the upcoming predictions. The most advanced codec, VVC, implements four successive loop filters (LF), luma mapping with chroma scaling

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

(LMCS), deblocking filter, sample adaptive offset (SAO), and adaptive loop filter (ALF).

Interestingly, the technology does not change radically between codec generations. Instead, the same principles and ideas are re-used and pushed further. Of course, there are new coding tools, but the overall structure remains the same.

Compared to MPEG-2, AVC/H.264 brought notably reduced complexity integer discrete cosine transform, multiple reference inter-frame prediction, in-loop deblocking filter, variable block sizes and flexible handling of interlaced video, all contributing to its excellent coding efficiency. The profiles definition allows adapting to multiple use-cases, making it suitable for any application. At the same period, the video compression research community has also been focusing on the concept of 3D wavelet filtering [10]. The wavelet transform has been used successfully in the JPEG2000 image compression standard [11]. The wavelet transform is a signal decomposition and analysis tool. Applied on an image, it replaces usual transforms such as DCT for energy compaction and provides a resolution scalable representation. That is, a wavelet compressed image can be reconstructed progressively, from lowest to highest frequencies, without coding efficiency loss. When applied to video, the same principle is extended on a 3D pixel volume [12]. The MC-EZBC codec is a good example of state-of-the-art 3D wavelet-based video coding [13]. This kind of technology was promising, but never reached the AVC/H.264 performance [14].

Jumping to the next generation, HEVC brought many improvements over AVC/H.264, including a much more flexible partitioning scheme, with up to 64x64 pixels partition sizes instead of 16x16, multiple transforms, improved motion compensation filtering and a new loop filtering restoration tool called Sample Adaptive Offset (SAO).

In parallel, a strong research focus was set on sparse modeling for image and video representation and analysis. As explained in [15], sparse coding consists in representing data with linear combinations of a few dictionary elements. Generally, the dictionary must be learned to be best adapted to the data. Considering images, the underlying idea is that only a tiny subset of the huge set of all possible pixel values combinations actually represents viewable images. Therefore, images can be represented by a smaller set of variables, as few as possible if compact representation is desired. Although the idea seems quite simple, building a dictionary is a non-trivial task. In [16], a method is proposed to learn basic texture elements representations and is used for intra coding. Video compression is tackled in [17], where dictionary learning is performed in the DCT domain of a block motion compensated structure. These methods can outperform the state-of-the-art codecs, well, if one considers AVC/H.264 as such. None of these methods reached the general performance and flexibility of HEVC. One may note though that sparse coding shines in specialized applications, such as very low bit rate human face coding [15]. Also the dictionary learning strategy anticipates the upcoming machine learning.

Finally, VVC outperforms HEVC thanks to further enhanced coding tools, such as an even more flexible partitioning scheme or a new in-loop restoration filter. Moreover, VVC includes from start several features that makes it “versatile”, including 360° video coding, screen content coding, gradual decoder refresh for low delay applications, and scalability, based on reference picture resampling (RPR) the ability to perform temporal prediction on reference images of different resolutions.

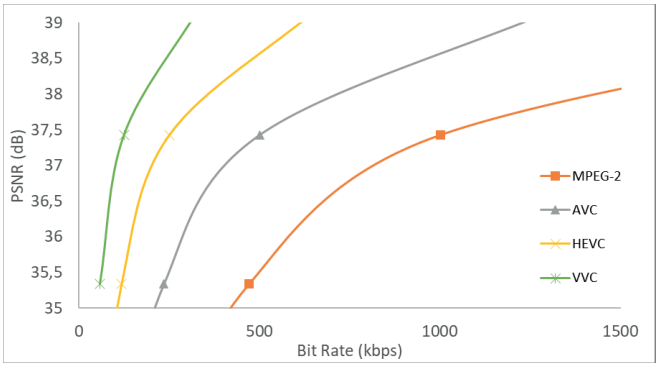


Fig. 2. Video codecs rate distortion performance progression example.

C. *Video CODECs performance, limits, and discussion*

Each codec generation allows decreasing the bitrate approximately by a factor two (Fig. *2*). This comes however at the cost of increased complexity. For instance, the reference VVC encoder is about 10 times more complex than the reference HEVC encoder. Let us illustrate this process with a simple example: Intra prediction mode, illustrated on *Fig. 3,* which consists in encoding a block of a frame independently from previous frames. In MPEG-2, intra block coding is performed without prediction from neighboring blocks. In AVC/H.264, intra blocks are predicted from neighboring blocks, with 9 possible modes. In HEVC, the prediction principle is reconducted, with 35 possible modes, while VVC is pushing further to 67 possible prediction modes. Having more prediction modes allows better predictions, hence better compression (even though mode signaling cost increases), at the cost of more complexity.

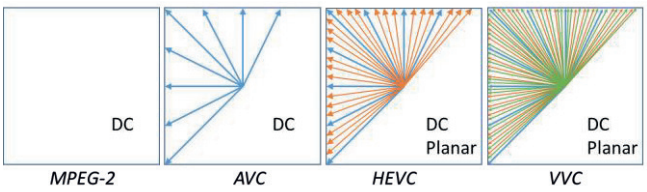


Fig. 3. Evolution of intra prediction in video codecs from MPEG-2 to VVC.

One natural question which arises is how far we can push this model. In other words, can we improve steadily the compression performance of this model decades after decades, by pushing the parameters and adding more local coding tools, or are we converging to a limit? At each codec generation, the question has been raised, and answered by the next generation. None of the proposed competing models have ever succeeded in outperforming the hybrid block-based model.

Nowadays, the recognized industry benchmark in terms of video compression performance is VVC. Can we go beyond the VVC performance? Well, the answer is already known,

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

and it is yes. Indeed, the JVET standardization group is currently conducting explorations. The Ad-Hoc Group 12 (AHG12) is dedicated to the enhancement of VVC. Around 15% coding efficiency gains are already achieved, only two years after VVC finalization [18]. So, we may continue the process for at least another decade.

However, there is a new contender arising: artificial intelligence; or more precisely, machine learning, or deep learning. In another Ad-Hoc Group, AHG11, JVET is exploring how machine learning can be the basis of new coding tools. This also brings coding efficiency gains of about 12% [19]. Hence the question: will the future of video compression include machine learning? At this stage, we would like to point-out two new facts.

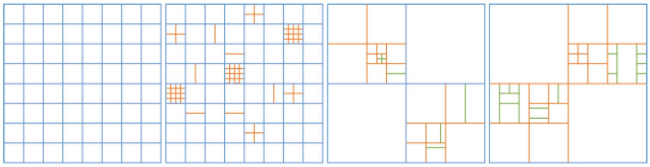


Fig. 4. Evolution of frame partitioning in video codecs from MPEG-2 to VVC.

First, considering the "traditional" methods explored by AHG12, there is a coding tool which seems to stop bringing gains: frame partitioning. The partitioning is a fundamental tool for video compression. It defines how precise can be the adaptation of the encoder to local content characteristics. The more flexible it is, the better the coding efficiency. All the subsequent coding tools depend on the ability to partition the frame efficiently. As illustrated on *Fig. 4,* AVC/H.264 has 16x16 pixels blocks, with some limited sub-partitioning. HEVC implements a much more flexible quadtree based partitioning from 64x64 pixels blocks. VVC combines quadtree partitioning with binary and ternary tree partitioning, from 128x128 pixels blocks for even more flexibility. During the exploration following HEVC standardization, the single fact of enhancing the partitioning brought up to 15% coding efficiency gains. Similarly, in the AHG12 context, people came with new extended partitioning strategies. However, only marginal gains were reported [20]. Does that mean we are finally approaching a limit?

The second fact is the development of end-to-end deep learning video compression. This strategy is highly disruptive. In short, the whole block-based hybrid coding scheme is replaced by a set of deep learning networks, such as auto-encoders. These types of schemes are competing with state-of-the-art fixed image coders [21]. For video applications, they are matching HEVC performance [22][23]. This level of performance has been reached in only five years. That's an unprecedently fast progression. One may easily extrapolate, even if the progression slows down, that the state-of-the-art video compression performance will soon be the end-to-end strategy prerogative. Therefore, we may very well be at a turning point of the video codecs history.

III. THE ADVENT OF ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) refers to technologies that allow computers to perform tasks that have so far required human intelligence. The AI term is born in the 1950s with among others the work of Alan Turing and its famous Turing test [34]. Already at this time, researchers tried to create artificial neurons to mimic the human brain. The first neural machine, the Stochastic Neural Analog Reinforcement Calculator (SNARC) has been built in 1951 by Marvin Minsky and has been the beginning of larger research in this field. This leads to the creation of the well-known Perceptron in 1957 which is the basis of modern deep learning. But researchers were too optimistic creating an intelligent machine. After several failures, funding and interest in the field dropped off, leading to the first AI winter. Researchers were mainly constrained by the limited computing power. Some of them persisted in the idea of creating a machine capable of carrying out complex human tasks, and in 1997, IBM's Deep Blue became the first computer to beat a chess champion. A lot of modern deep learning architectures such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) were designed during the 1980s et 1990s, but they required too many data and power to train, and they were forgot during several years. Researchers then focused on more practical and more humble problems. This was the emergence of machine learning (ML).

ML is a technology that allows algorithms to realize tasks without having been explicitly programmed to do them. It is a subfield of artificial intelligence that let the algorithms discover patterns in data to improve their performances on a specific task. A dataset needs to be prepared in order to train the model for a given task. This is the training step. After several iterations, the algorithm, or model, learns to extract more useful features from the dataset which makes him do better predictions. The model is then evaluated on a validation set to check that it generalizes well, i.e., that it performs as well on data that it has never seen before. ML have been used in different fields such as speech or character recognition for example, and many of the ML algorithms widely used today were invented before the 2000s (nearest neighbor, boosting, multilayer perceptron, …)

In the 2000s, computing power had improved, and the era of big data started to make a lot of, well, data available. AI started to succeed in many industrial use cases such as robotic. In the 2010s, computing power had improved further, specifically with the progress of Graphical Processing Units (GPUs). Moreover, public datasets were built with annotated samples, such as ImageNet [35], created for image classification task. Few years after the dataset creation, ImageNet launched the ImageNet Large Scale Visual Recognition Challenge (ILSRVC), an annual AI object recognition challenge. This led to the first deep learning-based solution in 2012 [36], outperforming all other solutions based on traditional machine learning.

This work triggered an explosion of applications using deep learning technologies, allowing to achieved performances never reached before on various tasks related to natural language processing and computer vision. Deep learning allows to automatically create a hierarchical representation of data, highlighting features and their relations that are hard, if possible, to describe manually. Today, DL allows cars to drive themselves, robots to

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

communicate with humans, but it can also generate data, with automatic picture colorization or realistic/artistic creations such as the recent Meta's make-a-scene tool [32]. This field is growing very quickly since a decade, and models improve every year.

IV. Machine learning based video compression

A. *Tool based ML-based video compression*

As AI and ML is gaining attention in all possible fields of research, video compression is no exception. The JVET standardization group has started an exploration activity dedicated to the introduction of ML in the VVC framework [38]. The idea here is to keep the hybrid block-based model and to replace or complement elementary coding tools by ML-based tools.

Intra prediction is addressed in [39] and [40]. Both approaches consider the prediction of a block of pixels from a causal pixel neighborhood. The prediction is performed by a neural network (NN) replacing the traditional directional or planar intra prediction (*Fig. 3*). The NN is expected to be able to predict complex shapes and textures. 2 to 3% coding efficiency gains are reported.

Inter prediction is considered using several strategies. In [41], an enhanced bi-prediction mode is proposed. Instead of predicting a block with an average of two motion compensated reference blocks, the two predictors are fed to a NN which outputs the final prediction. Up to 1% coding efficiency gains are reported. A similar approach is considered in [42], but with a single predictor. In [43], a whole reference frame is generated by a NN. This new frame is added to the reference list for temporal prediction. Therefore, each image block can be predicted either from a past encoded frame or from a NN generated frame without changing fundamentally the encoding/decoding process. Up to 2% coding efficiency gains are reported.

A strong focus has also been set on loop filtering [38]. The general idea of loop filtering is to restore already encoded frames. It serves both as a post processing, improving the visual quality of the compressed video, and as a coding efficiency improvement, as it allows better temporal predictions. As ML has been explored for many image processing tasks, it is a natural candidate for loop filtering. First attempts considered replacing all the loop filters by a ML process. The idea has then been refined with adaptive methods trying to take advantage of the best of two worlds, signal processing and ML. The CNNLF [44] is proposed as an alternative to the deblocking and SAO filters of VVC. It is up to the encoder to decide locally to activate CNNLF or not. The filter inputs many data, including quantization parameter (QP), a prediction image and a partition image. The filter also includes a scaling as a post processing after the NN step. Up to 12% coding efficiency gains are reported, illustrating the huge impact of the single loop filter coding tool. The filter proposed in [45], replaces all VVC LF, though it can be turned off at block level. It similarly relies on rich input and post-scaling, while making use of attention models. Similar performance is reported.

Overall, the coding efficiency gains obtained with these approaches are largely significant. However, they come at an unprecedented cost in complexity, with figures going up to 400 times slow-down of the decoder.

B. *Super-resolution-based video compression*

The idea of using super-resolution stems from the well-known over the top (OTT) streaming concept. Depending on available bitrate there exists an optimal combination of resolution and compression tuning. In other words, when the bitrate decreases, it becomes more efficient to decrease the video resolution rather than getting more compression artifacts. By factoring in the fact that ML has been studied as a mean of recovering fine details when increasing image resolution, one arrives naturally to the idea that encoding videos at a lower resolution may lead to better trade-off than the current approaches, thanks to the capability of NN to up-sample content without generating the traditional blurring and aliasing phenomenon.

In [46], the video sequence is first decomposed before being compressed using a traditional codec. It is then synthesized to retrieve the original resolution. The decomposition consists in down-sampling only the inter coded frames. Thus, the intra frames are carrying texture information, while inter frames are carrying the temporal information. Synthesis, or up-sampling, is assisted by a motion compensated NN. Up to 9% coding efficiency gains are reported.

In [47], the whole video is encoded at a lower resolution. NN-based super-resolution is applied as a post-processing to recover the original resolution. However, in order to better adapt to frame characteristics, the last layer of the NN is specialized for each sequence. The corresponding parameters are transmitted along with the video stream. About 6.5% coding gains are reported.

Depending on the complexity of the chosen NN approach, complexity gains can be observed. Indeed, thanks to the lower video resolution, the coding/decoding step is much less complex, potentially compensating for the NN complexity [48].

Finally, [49] proposes a different approach. No NN based super-resolution is involved. Rather, linear down/up-sampling is performed using the VVC RPR feature. The point is that resolution is chosen on encoder side by classical rate-distortion optimization. Therefore, depending on the sequence, one obtains either identical or better results than VVC. It shows that ML is not necessary to obtain gains by playing with resolution.

C. *End -to-end learned video compression*

Nowadays, learned image and video compression has been the target theme for both the Machine Learning and image/video compression communities. In this context, the Challenge on Learned Image Compression (CLIC) [37] aims to encourage both communities to advance the field of image and video compression using machine learning algorithms by either designing new codec architectures or by introducing new perceptual metrics. Therefore, each year, publications are gathered, evaluated, and compared against the traditional codecs. Then, the winners are presented in the CVPR workshop.

Recently, learned image compression has achieved

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

significant progress in coding performance. The state-of-the-art of such methods are currently competitive with the latest traditional coding system VVC in intra mode. Inspired by this success, deep learning methods were extended to learned video coding.

Learned video compression approaches can be divided into two main categories. The first one keeps the traditional coding pipeline, that deals with the inter frame redundancies, unchanged (motion estimation, motion compensation, residual coding). Then, for each step, deep learning architectures such as auto-encoders and optical flow architectures are used. For instance, [51] introduce the first low latency compression framework called DVC using auto-encoders to code motion vectors and residuals, a pretrained optical flow model for motion estimation and a bilinear warping for motion compensation. [53] improves the DVC performance by using multiples frames as references. This new coding system is called MLVC, it added four neural modules to the DVC framework. The first explored a buffer of multiple previous motion vectors to achieve motion estimation of the current frame. The second does the same for motion compensation. The two remaining modules aim to refine the motion vectors and the residuals, respectively. In the same context of low latency coding, [55] introduces a recurrent learned video codec (RLVC) using a recurrent autoencoder and a recurrent probability model to compress the motion and the residual features. The goal is to thoroughly explore the temporal correlation between frames and latent representations. In fact, this work enables using all the previous decoded frames as reference for compressing the current frame. In addition, the recurrent probability model tends to achieve lower bitrate since the latent representation of the current frame is conditioned with the previous ones. While DVC manages to outperform the low-delay P frames (LDP) configuration of x264 and compete with the same configuration of x265, MLVC and RLVC outperform DVC and x265 in terms of coding efficiency.

[52] presents a framework to code a GOP structure, which includes P and B frames, with different level of quality (HLVC). P and B frames are coded using two architectures of networks which achieve motion estimation, motion compensation and residual coding with hierarchical quality levels. Then, a Recurrent Neural Network (RNN) module is used to enhance quality of the decoded frames. This work's proposed framework depends on GOP structure, which is set manually before proceeding to the training stage. Although this method codes a GOP structure with B and P frames, it was evaluated against the LDP mode of x265 and the low latency model DVC. Compared with x265, it manages to obtain gains in BDBR: -6% for PSNR models and −35.94% for MS-SSIM models. [54] presents a method to achieve perceptual learned video compression (PLVC) using a recurrent conditional GAN. This framework consists of a compression network based on RLVC [55], that serves as generator, in addition to a recurrent discriminator that take as input spatial and temporal conditions as well as the current and previous reconstructed frames. The training process minimize a combination of an adversarial loss function with the rate distortion one. This work manages to get the best results in terms perceptual metrics such as LPIPS [60] and FID [61] compared to the leaned codecs: RLVC, HLVC, MLVC and the traditional codec HEVC (HM16.20). However, in terms of objective metrics like PSNR and MS-SSIM, it is on-par with DVC, and it performs worse than the previously mentioned learned video codecs as well as HEVC (HM 16.20).

While in the previous works, I frames are compressed using either the BPG codec for [51][53][55][52] or a learned image codec for[54], [56] proposes a neural coding framework for I and P frames, and [22] introduced a neural architecture, consistent with all type of frame I, P and B frames. The system contains two networks: MOFNET deals whith motion estimation and compensation and CodecNet achieves conditional coding which replace residual coding. This approach achieves performance competitive with the state-of-the-art video codec HEVC (HM 16.20).

The second category of end-to-end learned video compression focuses on reducing temporal redundancy using algorithms that are different from the traditional pipeline. For example, [57] proposes a video compression framework based on an 3D auto-encoder combined with temporally conditioned entropy model. The performance of this method is competitive with x265 in terms of MS-SSIM. Other works used frame interpolation for video coding. [58] explores Generative Adversarial Networks (GAN) as a decoder to reconstruct separate frames, then used linear interpolation to reconstruct the missing frames. This work is evaluated on low resolution gray sequences in low bitrate. Unfortunately, the results of this approach are only comparable with MPEG4. [59] uses a learned image codec to compress key frames and then uses an interpolation model to predict the missing frames. This approach is compared with handcrafted codecs such as HEVC, AVC and MPEG on the VTL dataset [62]. It outperforms MPEG4 and is matching H264.

All in all, although learned video coding in intra mode (learned image coding) performance is on-par with the latest handcrafted codec VVC, extracting spatiotemporal features is more challenging which makes learned inter coding more difficult. Therefore, the state-of-the-art of learned video compression currently matches the coding efficiency of HEVC. However, one can predict that the progress in this field will be significant in a short period of time.

V. Practical Application of Machine Learning Based Video Compression

A. Delay, rate-control and content adaptation

There is a huge difference between a codec, as defined by standards, and a ready to production live video encoder. The codec is only a part of a video encoder. A video encoder must manage various inputs or capture, decoding, encoding, muxing and output, all along with system functions and user interface. Even when focusing on the encoding part, there is more than the codec. Live encoding requires optimization of the complexity/quality trade-off, which generally translates into added delay. This delay must of course stay under control. Delay is caused among other things by Pre-processing and analysis in a look-ahead buffer, frame reordering for efficient group of pictures (GOP) structure coding, pipelining, and rate-control.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

Content adaptation is desirable for optimal quality. GOP structure is generally adapted to the nature of the content. Scene-cuts are detected, and temporal prediction is avoided across them. Considering end-to-end video coding, the same ideas may apply. However, depending on the end-to-end implementation, it may be simpler. The idea of GOP structure may be managed in a transparent manner by the ML model. The notion of successive GOP may be easily conserved, allowing easy chunking for OTT and short zapping time.

In short, content adaptation does not seem to be an obstacle to the end-to-end video encoders development. Rate-control, on the other hand, may be more difficult. Indeed, in traditional video coding, there is an understandable, though non-trivial, relationship between the QP and the bitrate. In an end-to-end video encoder, there exist a parameter tuning the bitrate. However, the effect of this parameter is generally not easy to model. Some encoders are actually trained for a single value of this parameter. It implies that if one needs 64 rate levels, like the 64 QP values of VVC, 64 models must be trained and stored. In an attempt to answer to this issue, [50] proposes a new loss function, where the λ parameter, responsible for rate tuning, is non constant. It allows to design a training procedure where several values of λ are fed randomly to the system, thus making a model that can react appropriately to any value of λ at inference time. Literature on this topic is limited as of today, and there is no doubt that further research is needed, but this example is encouraging.

Finally, the main difficulty to handle may very much be the huge operational complexity of NN.

B. Computing resources

During several years after the deep learning emergence, researchers did not really care about models' complexity. The solutions proposed for various public challenges were more complex every year, while their performances continued to grow exponentially. In their analysis, [24] have shown that the largest model training runs have doubled the computational power used every 3.4 months since 2012. As an example, Danish researchers used the "Carbontracker" tool [25] to show that the energy required to train a GPT-3 model (one of state-of-the-art model for natural language tasks) could have the carbon footprint of driving 700,000km. The training step of machine learning models is highly resource-intensive, but the inference one consumes far more power. Indeed, while the model is trained once, it can be used for billions of inferences. It is estimated that inference accounts for up to 90% of the computing cost [26].

The increase in the model's complexity has been made possible thank to the hardware evolution. For deep learning technologies, Graphical Processing Units (GPUs) are often the default choice, because of their ability to perform a lot of low-level mathematical operations in parallel. Initially designed for games and graphically intensive applications, researchers thought their capabilities were suited to run deep learning models. This market is dominated by Nvidia, and since the deep learning development, they have built new GPU architectures that make their hardware more effective for models training and inference. But this kind of hardware still is a general-purpose solution. Some manufacturers decided to build specific chips designed to run deep learning models even more effectively. One can think about Google Tensor Processing Units (TPUs), or Microsoft Catapult project. They are based respectively on Application-Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA) and allow power consumption reduction related to GPUs. These solutions are available in cloud infrastructures, so they can be used for models training and online inference. These use cases are rarely constrained by consumption resources. If more power is needed to speed up training or inference, it is simple to scale by adding GPUs for example. But what about edge devices?

Edge devices are appliances on which data collection takes place. It can be desktop computers, smartphones, or connected devices. While GPUs or TPUs are still the default solutions for training models, a lot of works has been done for performing inference on edge devices. In contrary to cloud platforms, scaling is very hard due to limits in space, power, and connectivity. But this is a very important use case as it allows processing data locally, mitigating networks limitations, increasing security, and improving data privacy. Researchers and manufacturers have then put a lot of effort improving edge computing hardware for processing deep learning models. Hence new types of AI-optimized accelerators have been designed during the past few years, that can be regrouped under the name Neural Processing Units (NPUs). Main mobile manufacturers have designed their own solution. This includes chips such as the Apple Neural Engine, the Kirin 980 from Huawei, or the Exynos 9820 from Samsung. There also exists development boards such as the Nvidia Jetson Nano or the Google Coral Edge TPU. NPUs are based on specific architectures that make deep learning model execution faster while having limited consumption. A lot of accelerators exist today [27], and this is a very active research field. Few years ago, MLPerf benchmarks [28] have been released in order to make AI platforms performances comparison simpler. It allows to get training time, inference time, and more recently power consumption of a specific hardware configuration for different AI models. Despite these initiatives, AI accelerators comparison remains very hard as performances are related to too many factors, not only the accelerator itself. Performances are also impacted by the CPU, and the software library used to deploy the model.

In addition to work on specialized hardware, a lot of work has been done on the software part. Some of them are designed for CPUs (OpenBLAS, Intel MKL, …), and others for GPUs (cuBLAS, cuDNN, …). All of them optimize matrix operations in order to make AI model execution faster using only algorithmic optimizations. These are libraries allowing low-level mathematic operations, but they are mainly used through higher-level frameworks and tools. For example, Openvino [29] and TensorRT [30], respectively developed by Intel and Nvidia, are platforms offering runtimes with optimized operations implementation, but also some model optimization strategies. This includes weights quantification, network pruning, or operations fusion.

The combination of hardware and software evolution allows the execution of powerful AI models on edge devices in real time. But the AI field is evolving really fast. Even with this progress, models' complexity keeps growing every year,

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

and hardware and software providers must continue to improve their solutions to make model execution faster or less energy consuming. Recent trends such as neuromorphic computing [31] show there is room for improvement with completely different designs. Also, new hardware is challenging dominant existing solutions. For example, the Hailo 8 chip [33] presents performances up to 13x those of Google TPUs. All of this shows that the Moore's law continues and makes possible further improvements in AI.

VI. A CASE-STUDY: END-TO-END MEMORY CONSUMPTION

A. *Problem statement*

As models' sizes are growing continuously, memory consumption is also becoming an issue, along with computing power. The case of end-to-end learned encoding is considered here. The auto-encoder architecture, built with convolutional layers, enables processing different video resolutions, no matter the resolution used during the training step. However, with growing models' sizes and video resolutions (4K, 8K), these solutions are facing hardware memory saturation. One way to solve this issue is to use a patch-based coding approach. The video frames are divided into patches smaller than the frame size, that can be encoded independently. Then, the decoded patches are gathered to reconstruct the decoded frames.

This solution addresses the hardware limitation issues, but the reconstructed frames can have block artifacts at the patch boundaries, widely deteriorating the video quality.

B. *Patch-based end-to-end video encoding*

A solution to the memory saturation is proposed to perform patch encoding while removing block artifacts. The idea is to encode overlapping patches and then use a linear function to combine the reconstructed overlapped pixels. If b_m and b_{m+1} are two consecutive reconstructed patches overlapping horizontally on N pixels, the value of the i^{th} overlapped pixel $p_{rec}(i)$ for a given line in the reconstructed frame is determined by the following equation:

$$p_{rec}(i) = \left(1 - \frac{i}{N-1}\right) p_{b_m}(P + i) + \left(\frac{i}{N-1}\right) p_{b_{m+1}}(i), \quad (1)$$

where $i \in \{0, \dots, N-1\}$ is the index of the overlapped pixels, P is the size of the patch without overlapping, p_{b_m} and $p_{b_{m+1}}$ are pixels values, for a specific line, of two consecutive decoded patches b_m and b_{m+1} respectively. The same equation applies for vertically overlapping patches.

The proposed approach has been applied to encode I frames, using an end-to-end learned image codec which is an implementation of the model architecture introduced in [21]. This model was trained on CLIC 2020 dataset [37]. For training, 256×256 sized patches were randomly cropped from each image of the training set. The loss function to be minimized is:

$$J = D + \lambda R \quad (2)$$

where D refers to the distortion measured by the Mean Square Error (MSE) or the Multi-Scale Structural Similarity Index (MS-SSIM) metrics, and R refers to the rate used to transmit the bitstream, estimated using the Shannon entropy. λ is the Lagrangian multiplier, allowing to adapt the bit rate targeted by the learned image coding model.

The method is then evaluated on Class B, C, D, E and F of the JVET Common Test Conditions (CTC) sequences (8-bit sequences) [72]. For each sequence, one frame is extracted and compressed both entirely (referred to as the full image approach) and by the proposed patch-based approach, with and without overlapping, where $N \in \{0, 2, 4, 8, 16, 32\}$ overlapped pixels and $P = 256$, as the training resolution.

BD-rate gains of the patch-based learned image coding with and without overlapping were computed comparing to full image learned image coding, using an end-to-end model trained to minimize MSE as distortion metric.

For MSE models, patch-based image coding without overlapping presents a slight loss in BD-rate (Average BD-rate +0.013), comparing to full image coding, which mostly corresponds to the block artifacts issue caused by patch-based approaches.

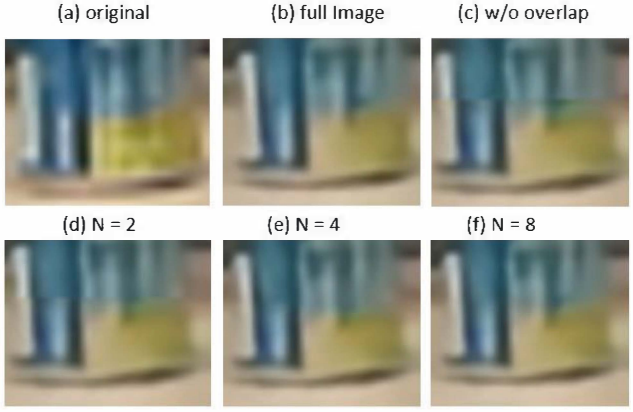


Fig. 5. Visual results of comparison for FourPeople. The model used optimizes the MSE metric with λ = 4096.

On the other hand, the proposed method achieves a gain in BD-rate, which increases as the number of overlapped pixels is increased. For N = 2, the average BD-rate gain among CTC sequences is: -0.025, which shows that two overlapped pixels seems to be sufficient to eliminate the borders artifacts. With N = 8 and N = 16, small gains are observed comparing to full image coding: -0.034 and -0.041 respectively. For N > 16, BD-rate gains saturation is observed. An example of decoded images is presented in Fig. 5. Overlapping with N = 2 and N = 4 reduce the block artifacts while overlapping with 8 pixels eliminates them entirely.

The experimental complexity and memory consumption are reported in Table I. Frames of different resolutions were extracted from the JVET CTC and were coded using two machines with powerful GPUs: GeForce RTX 2080ti and GeForce RTX 3090 with memory capacity of 11Go and 24Go respectively. Full resolution coding of an HD image is not possible on both GPUs due to "Out Of Memory" (OOM) error, while full coding an 1280x720 image, can only be run on the machine with the GPU RTX 3090. It is important to note that these resolutions are standard resolutions in practical applications of image compression. 4k is not even considered. Therefore, the fact that they cannot be run on one of the latest GPUs is inconvenient. In this case, the proposed

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

method provides a solution that enables coding high resolution images without deteriorating quality. While the method is necessary for resolution 720p and above, it is adding some complexity to the system for smaller resolutions. For instance, when running the resolution 832x480 on 2080ti GPU, patch-based coding increases the coding time by 3.63% compared with full resolution coding. This is expected since our method requires coding more pixels to overlap patches.

To conclude this section, the proposed approach addresses the hardware memory limitation problem since it allows coding resolutions such as HD and 720p, while maintaining same or better quality as the full resolution learned coding.

TABLE I: PERFORMANCE OF PATCH BASED END-TO-END ENCODING.

Resolution	Method	Coding Time GPU 2080 11Go	Coding Time GPU 3090 24Go
1920x1080	Full Resolution Coding	OOM	OOM
	Patch coding in parallel with overlapping	3.82s	2.05s
1280x720	Full Resolution Coding	OOM	0.93s
	Patch coding in parallel with overlapping	1.91s	1.012s
832x480	Full Resolution Coding	1.06s	0.52s
	Patch coding in parallel with overlapping	1.10s	0.55s

VII. WRAP-UP

From MPEG-2 in the 90's to VVC nowadays, four successive major generations of codecs have made video ubiquitous, from TV screen to smartphones, from over-the-air to internet. All these codecs are based on the same general structure, the hybrid block-based model. Previous attempts to overcome this model have all failed, despite of their numerous technical qualities and features. But how long will this model continue to dominate?

Today, one observes a small hint of a decline of the hybrid block-based model, along with the rise of machine learning. Machine learning is the state-of-the-art technology in many image and video processing fields, but still not in video compression. ML may not be ready yet for video compression, but it is progressing fast. We argue in this paper that current limitations can be addressed, either through plain technological progress, or through dedicated algorithmic progress.

As an example, a new method of memory management for machine learning based end-to-end image and video compression is described in this paper, namely patch encoding with overlapping.

All in all, for the upcoming video codec generation, two approaches are competing. Time will tell, but our guess is that there will be another generation of hybrid block-based model before the advent of machine learning based video compression. Researchers are just needing a few years to refine and make the technology practical. Model sizes and hardware capabilities will eventually converge.

REFERENCES

[1] Cisco Annual Internet Report (2018–2023) White Paper, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
[2] Video Encoder Market with COVID-19 Impact by Number of Channel (Single, Multichannel), Mounting Type (Standalone, Rack-mounted), Application (Broadcast, Surveillance (Commercial, Residential, Institutional)), and Geography - Global Forecast to 2025, https://www.marketsandmarkets.com/Market-Reports/video-encoder-market-109133493.html.
[3] R. D. Kell. Improvements relating to electric picture transmission systems. British Patent No. 341,811, 1929.
[4] Thomas Wiegand and Heiko Schwarz (2016), "Video Coding: Part II of Fundamentals of Source and Video Coding", Foundations and Trends® in Signal Processing: Vol. 10: No. 1–3, pp 1-346. http://dx.doi.org/10.1561/2000000078.
[5] Gary Sullivan, "Overview of International Video Coding Standards (preceding H.264/AVC)" ITU-T VICA Workshop, 22-23 July 2005, ITU Headquarter, Geneva.
[6] S.Vetrivel et. al., "An Overview Of H.26x Series And Its Applications", International Journal of Engineering Science and Technology, Vol. 2(9), 2010, 4622-4631
[7] https://www.itu.int/
[8] https://www.iso.org/about-us.html
[9] T. Biatek, M. Abdoli, T. Guionnet, A. Nasrallah and M. Raulet, "Future MPEG standards VVC and EVC: 8K broadcast enabler" IBC365, 14 September 2020.
[10] E. Moyano, F. J. Quiles, A. Garrido, T. Orozco-Barbosa and J. Duato, "Efficient 3D wavelet transform decomposition for video compression," Proceedings Second International Workshop on Digital and Computational Video, 2001, pp. 118-125, doi: 10.1109/DCV.2001.929950.
[11] Taubman, D., Marcellin, M. (2012). *JPEG2000 Image Compression Fundamentals, Standards and Practice: Image Compression Fundamentals, Standards and Practice* (Vol. 642). Springer Science & Business Media.
[12] V. Bottreau, M. Benetiere, B. Felts and B. Pesquet-Popescu, "A fully scalable 3D subband video codec," *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, 2001, pp. 1017-1020 vol.2, doi: 10.1109/ICIP.2001.958669.
[13] Peisong Chen and J. W. Woods, "Bidirectional MC-EZBC with lifting implementation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 10, pp. 1183-1194, Oct. 2004, doi: 10.1109/TCSVT.2004.833165.
[14] P. Lambert, W. De Neve, P. De Neve, I. Moerman, P. Demeester and R. Van de Walle, "Rate-distortion performance of H.264/AVC compared to state-of-the-art video codecs," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 134-140, Jan. 2006, doi: 10.1109/TCSVT.2005.857783.
[15] Julien Mairal; Francis Bach; Jean Ponce, *Sparse Modeling for Image and Vision Processing* , now, 2014.
[16] Y. Sun, M. Xu, X. Tao and J. Lu, "Online dictionary learning based intra-frame video coding via sparse representation," *The 15th International Symposium on Wireless Personal Multimedia Communications*, 2012, pp. 16-20.
[17] Irannejad, M., Mahdavi-Nasab, H. Block Matching Video Compression Based on Sparse Representation and Dictionary Learning. *Circuits Syst Signal Process* **37,** 3537–3557 (2018). https://doi.org/10.1007/s00034-017-0720-5.
[18] JVET-Z0012-v1 "JVET AHG report: Enhanced compression beyond VVC capability (AHG12)" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 26th Meeting, by teleconference, 20–29 April 2022.
[19] JVET-Z0023 "EE1: Summary of Exploration Experiments on Neural Network-based Video Coding" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 26th Meeting, by teleconference, 20–29 April 2022.
[20] JVET-Y0150-v2 "EE2-1.1: Tests on unsymmetric partitioning methods" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 25th Meeting, by teleconference, 12–21 January 2022.
[21] Z . Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 7936–7945, 2020

[22] Ladune, T., Philippe, P., Hamidouche, W., Zhang, L., & Déforges, O. (2021). Conditional Coding for Flexible Learned Video Compression. 1–18. http://arxiv.org/abs/2104.07930
[23] Hu, Zhihao and Lu, Guo and Xu, Dong, "FVC: A New Framework Towards Deep Video Compression in Feature Space", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021,pp 1502-1511.
[24] Amodei, D. & Hernandez, D. (2018). AI and Compute. https://openai.com/ blog/ai-and-compute/. 3, 12, 14
[25] Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. arXiv preprint arXiv:2007.03051.
[26] Thomas D.. Reducing machine learning inference cost for pytorch models - aws online tech talks. https://www.youtube.com/watch?v=ET2KVe2du3Y, 2020.
[27] Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., & Kepner, J. (2021, September). AI accelerator survey and trends. In 2021 IEEE High Performance Extreme Computing Conference (HPEC) (pp. 1-9). IEEE.
[28] Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C. J., ... & Zhou, Y. (2020, May). Mlperf inference benchmark. In 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA) (pp. 446-459). IEEE.
[29] https://docs.openvino.ai/latest/index.html#
[30] https://developer.nvidia.com/tensorrt
[31] Imam, N., & Cleland, T. A. (2020). Rapid online learning and robust recall in a neuromorphic olfactory circuit. Nature Machine Intelligence, 2(3), 181-191.
[32] Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., & Taigman, Y. (2022). Make-a-scene: Scene-based text-to-image generation with human priors. arXiv preprint arXiv:2203.13131.
[33] https://hailo.ai/products/hailo-8/
[34] Turing, A. M. (2009). Computing machinery and intelligence. In Parsing the turing test (pp. 23-65). Springer, Dordrecht.
[35] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
[36] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
[37] "Workshop and c. on learned image compression," Https://www.compression.cc/, 2020.
[38] Elena Alshina et al., "JVET AHG report: Neural network-based video coding", JVET-AA0011-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 27th Meeting, by teleconference, 13–22 July 2022.
[39] Maria Meyer and Christian Rohlfing, "AHG11-related: Investigation on CNN-based Intra Prediction", JVET-U0105-v3, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 21st Meeting, by teleconference, 6–15 Jan. 2021.
[40] T.Dumas et al.,"EE1 test 3.1: intra prediction using neural networks" ,JVET-Y0082-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022
[41] F. Galpin et al., "AHG11: Deep-learning based inter prediction blending", JVET-V0076-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 22nd Meeting, by teleconference, 20–28 Apr. 2021.
[42] Changyue Ma et al., "AHG11: Neural Network Based Motion Compensation Enhancement for Video Coding", JVET-Y0090-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.
[43] Zizheng Liu et al., "AHG11: NN-based Reference Frame Interpolation for VVC Hierarchical Coding Structure", JVET-Y0096-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 25th Meeting: by teleconference, 12–19 Jan. 2022.
[44] Liqiang Wang et al., "EE1-1.1: neural network based in-loop filter with constrained storage and low complexity", JVET-Y0078-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.
[45] Yue Li et al., "EE1-1.2: Test on Deep In-Loop Filter with Adaptive Parameter Selection and Residual Scaling", JVET-Y0143-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.
[46] Ming Lu et al., "EE1: Tests on Decomposition, Compression, Synthesis (DCS)-based Technology", JVET-V0149, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 22th Meeting, by teleconference, 20 – 28 Apr. 2021.
[47] Takeshi Chujoh et al., "EE1.2: Additional experimental results of NN-based super resolution (JVET-U0053)", JVET-V0073-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 22nd Meeting, by teleconference, 20–28 Apr. 2021.
[48] Chaoyi Lin1 et al., "EE1-2.3: CNN-based Super Resolution for Video Coding Using Decoded Information", JVET-Y0069-v21, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.
[49] Elena Alshina et al., "EE1-2.1: Super Resolution with existing VVC functionality", JVET-Y0061-v21, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.
[50] Chaoyi Lin et al., "AHG11: Variable rate end-to-end image compression", JVET-U0102, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 21st Meeting, by teleconference, 6–15 Jan. 2021.
[51] Lu, Guo and Ouyang, Wanli and Xu, Dong and Zhang, Xiaoyun and Cai, Chunlei and Gao, Zhiyong, "DVC: An End-To-End Deep Video Compression Framework", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June, 2019
[52] Yang, R., Mentzer, F., Van Gool, L., & Timofte, R. (2020). Learning for Video Compression With Hierarchical Quality and Recurrent Enhancement", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020
[53] Lin, Jianping and Liu, Dong and Li, Houqiang and Wu, Feng, "M-LVC: Multiple Frames Prediction for Learned Video Compression", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June, 2020
[54] Ren Yang, Luc Van Gool, and Radu Timofte.."Perceptual Learned Video Compression with Recurrent Conditional GAN", arXiv preprint arXiv:2109.03082 (2021).
[55] R. Yang, F. Mentzer, L. Van Gool and R. Timofte, "Learning for Video Compression With Recurrent Auto-Encoder and Recurrent Probability Model," in IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 388-401, Feb. 2021,
[56] H. Liu et al., "Neural Video Coding Using Multiscale Motion Compensation and Spatiotemporal Context Model," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 8, pp. 3182-3196, Aug. 2021
[57] Habibian, Amirhossein and Rozendaal, Ties van and Tomczak, Jakub M. and Cohen, Taco S., "Video Compression With Rate-Distortion Autoencoders", Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October, 2019
[58] S. Santurkar, D. Budden and N. Shavit, "Generative Compression," 2018 Picture Coding Symposium (PCS), 2018, pp. 258-262,
[59] Wu, Chao-Yuan and Singhal, Nayan and Krahenbuhl, Philipp, "Video Compression through Image Interpolation", Proceedings of the European Conference on Computer Vision (ECCV), September, 2018.
[60] Richard Zhang, Phillip Isola, Alexei AEfros, et al., "The unreasonable effectiveness of deep features as a perceptual metric." In CVPR, 2018.
[61] Martin Heusel, Hubert Ramsauer, et al. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In NeurIPS, 2017.
[62] Video trace library. http://trace.eas.asu.edu/yuv/index.html, 2001. Accessed: 2020-11-11. 5, 6
[63] ITU-T Recommendation H.120 (11/88).
[64] ITU-T Recommendation H.261 (03/93).
[65] ITU-T Recommendation H.263 (01/05).
[66] ISO/IEC JTC 1/SC 29 (2010-07-17). "MPEG-1 (Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s)".
[67] ISO. "ISO/IEC 14496-2:2004 - Information technology -- Coding of audio-visual objects -- Part 2: Visual"
[68] ITU-T Recommendation H.262 (02/12).
[69] ITU-T Recommendation H.264 (08/21).
[70] ITU-T Recommendation H.265 (08/21).
[71] ITU-T Recommendation H.266 (04/22).
[72] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "VTM Common Test Conditions and Software Reference Configurations for SDR Video," document JVET-T2010 of JVET, Oct 2020.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

Thomas Guionnet is a Fellow Research Engineer at Ateme, where he currently leads the Innovation team's research on artificial intelligence applied to video compression. Beyond his work for Ateme, he has also contributed to the ISO/MPEG - ITU-T/VCEG – VVC, HEVC and HEVC-3D standardization process; he teaches video compression at the ESIR engineering school in Rennes; and he has authored numerous publications including patents, international conference papers, and journal papers. Prior to joining Ateme, he spent 10 years at Envivio conducting research on real-time encoding, video-preprocessing, and video quality assessment. He holds a Ph.D. from Rennes 1 university.

Marwa Tarchouli received an engineering diploma in electronic engineering from Ecole Nationale Supérieure d'Electronique, Informatique, Télécommunications, Mathématique et Mécanique de Bordeaux (ENSEIRB MATMECA), Bordeaux, France, in 2020. Since 2021, she is a PhD student at Ateme and INSA Rennes. Her Phd focuses on improving video coding schemes using machine learning algorithms.

Sébastien Pelurson received a PhD degree in computer science from the University of Grenoble Alpes, Grenoble, France, in 2016. From 2016 to 2019, he worked in the field of augmented reality, and more specifically on the use of deep learning models to improve 3D tracking algorithms on mobile devices. He joined Ateme, Rennes, France, in 2020. His research interests include video coding optimization based on machine learning technology.

Mickaël Raulet is the chief technology officer at ATEME, where he drives research and innovation with various collaborative research and development projects. He represents ATEME in several standardization bodies: ATSC, DVB, 3GPP, ISO/IEC, ITU, MPEG, DASH-IF, CMAF-IF, SVA, and UHD Forum. He is the author of numerous patents and more than 100 conference papers and journal scientific articles. He previously worked for the research Institute of Electronics and Telecommunications of Rennes (IETR) where he was a researcher in rapid prototyping of video coding standards, and he was project leader of several French and European projects. He was also a member of the research institute IRT B-COM (http://b-com.org). His interests include dataflow programming, signal processing systems and video coding. Currently, his focus is directed towards ATSC 3.0, next-generation video codecs and artificial intelligence. He served as a member of the technical committee of the Design and Implementation of Signal Processing Systems (DISPS) of the IEEE Signal Processing Society and as member a "Circuits and Systems for Video Technology" editorial board. In 2006, he received a PhD from INSA in electronic and signal processing, in collaboration with Mitsubishi Electric ITE, Rennes, France.

Received in 2022-06-22 | Approved in 2022-07-08

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

Implementation of convolutional NEURAL network in FPGA for image recognition

Victor Mendonça Aguirre
Fadi Jerji

CITE THIS ARTICLE

Aguirre, Victor Mendonça; Jerji, Fadi; 2022. Implementation of convolutional NEURAL network in FPGA for image recognition. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.4. Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

Implementation of convolutional NEURAL network in FPGA for image recognition

1st Victor Mendonça Aguirre
Mackenzie Presbyterian University
São Paulo, Brazil
victor.1605@hotmail.com

2nd Fadi Jerji
Mackenzie Presbyterian University
São Paulo, Brazil
fadi.jerji@gmail.com

***Abstract*—Neural Networks (NN) are being researched and improved to a degree that machines can closely resemble the capacity to execute complex tasks that only an intelligent animal is capable of. Vision used in the interpretation and recognition of the environment is one of such tasks that is being researched so future technologies can simulate vision in autonomous vehicles to further improve self-driving capabilities, increase driver convenience, help avoid accidents, and even autonomous delivery. Convolutional neural networks, inspired by the mechanics of animal vision, are utilized for the complex task of image recognition. Field Programmable Gate-Arrays (FPGA) recent developments have given it more parallel processing and processing speed making it a prime candidate for the implementation of NNs efficiently, with more processing capabilities and low response times compared with the alternatives. The objective of this work is to evaluate the performance viability of FPGA implementation of an image classification NN with acceptable accuracy and low response time.**

***Index Terms*—FPGA; CNN; autonomous vehicles; MNIST.**

I. Introduction

Artificial intelligence has attracted attention from the most varied types of industries, with several studies being carried out on the subject and with recent advances in hardware, increasingly complex algorithms are being developed with adequate processing time, making it possible to derive useful and fast information from large amounts of information, but the question of efficient hardware implementation remains to execute these algorithms quickly and efficiently.

Along with the development of techniques and studies, the applications of convolutional neural networks (CNNs) have grown considerably, mainly in activities that require understanding at a level comparable to that of a human being, such as natural language processing and computer vision and it is possible to incorporate CNNs to assist in processing audio, image classification, scenario labeling, and facial recognition [1], [2]. Some networks achieve better results than human performance as evidenced in the work of [3]. The impressive performance of these networks comes at the cost of large memory bandwidth and intensive use of computational logical resources [4].

CNNs have excellent performance when it comes to image classification, but such networks require millions or even billions of operations per second to classify an image, which makes network implementation a challenge in terms of computational power and memory storage capacity. For example, in 2012 Alexnet [5], with a network architecture that required the storage of 60 million parameters to process an image, won the Imagenet contest [6]. In 2014 the VGGNET network [7] wins the same contest, but its design required loading about seven times more parameters, because of this, the network required dedicated hardware to run.

CNNs have enabled the increasing automation of tasks and machines such as autonomous cars, the idea of product delivery via drone is already being discussed [8], but in the same way, as autonomous cars require a large amount of information and an equally large processing power together with a robust algorithm to detect and recognize obstacles [9], drones or autonomous aircraft require an even faster processing and image recognition capacity. To be able to follow the movements and maneuvers performed by these vehicles. The choice of hardware for implementing CNNs is important because it will influence the needs and results of the network, for training the choice mostly adopted is a graphics processing unit (GPU), due to its great capacity for parallelism of calculations, reaching 11 trillion floating point operations per second (TFLOP/s) and due to the need to train the neural network only once, the GPU's energy consumption does not significantly impact the process, as for the implementation of the network in themselves, they can be implemented either on GPU, Field Programmable Gate-Arrays (FPGA) or Central Processing Unit (CPU), being more flexible FPGAs when compared to ASICs, counting on easy and fast implementation in the market and upgradeability even after implementation compared to CPU and it is also worth mentioning its potential for improving the architecture, energy savings compared to GPU and the possibility of using different formats and numerical representations. Microsoft has recently explored the possibility of CNNs on FPGAs as cost-effective network accelerators in a data center [10], [11]. Many studies are being carried out on accelerators for CNNs implemented in FPGAs [12], [13], as well as tools to generate such accelerators automatically [14], [15]. Studies have also been carried out on

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

CNNs networks with low accuracy, networks using weights and activation function with numbers in binary format [6], [16] and in some cases, the network has an accuracy comparable to networks using 32-bit floating point, these types of implementation are attractive in FPGAs because they take advantage of the efficiency of operations performed on Look Up Tables (LUTs). The implementation in FPGAs has seen more and more attention due to the constant development of tools that help and automate the development of implementations on the board, the Xilinx Vivado tool for high-level synthesis (HLS) allows the user to write code with a reasonable level of abstraction and the tool's algorithm compiles the code for register transfer level (RTL) between registers [17].

For complex tasks that require a large number of calculations, but which, at the same time, demand efficiency and low response time, CNN in FPGA focused on image recognition would be a tool that would drive the development and implementation of such technologies to advance sectors that would benefit from autonomous aerial vehicles and make the autonomous car industry even more robust.

II. NEURAL NETWORKS

A fundamental component of neural networks, in general, are artificial neurons, inspired by biological neurons, which are responsible for most of the processing that occurs in artificial neural networks (ANNs) and can be arranged within a network in various ways.

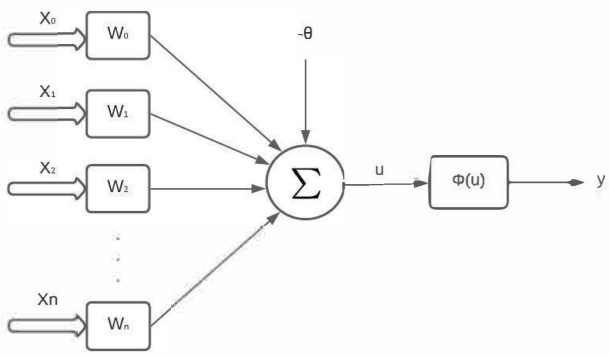


Fig. 1: Perceptron.

It's possible to observe the general structure of a neuron or perceptron in Fig. 1 where from X_0 to X_n representing input data or signals from neurons from another layer, the synaptic weights W_0 to W_n that determine how excitatory or inhibitory the signal is for the neuron. The adder block is responsible for summing the modified input signals with a predetermined value θ called bias, its function is to increase or decrease the net input, in order to translate the activation function on the axis, it can also be used so that, in the network training process, changes in synaptic weights result in less drastic changes in the network as a whole since the bias is independent of the input value in the system, which helps the network to converge on an ideal solution and can also be used to make the value needed to activate the activation function larger or smaller. NNs generally have a forward propagation architecture where signals entering the system propagate towards the output in a single direction. The model represents artificial neurons and can be represented by the following equations 1 and 2

$$u = (\sum_{i=1}^{n} X_n \times W_n) - \theta \tag{1}$$

$$y = \Phi(u) \tag{2}$$

The weight parameters W_n and bias θ are adjusted in the training of neurons in the network according to the final application.

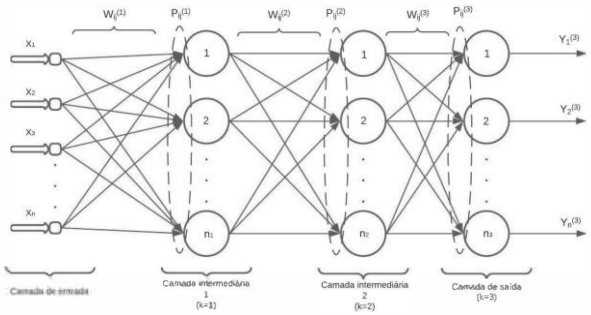


Fig. 2: MLP network.

An ANN can have a different number of layers which can be classified into three categories:

- Input layer: responsible for receiving data, signals, or characteristics from the external environment that are usually normalized in relation to the ranges of dynamic variations produced by the activation functions, which improves the accuracy of the network as a whole.
- Hidden layers: composed of neurons with the function of extracting characteristics associated with the system to be inferred.
- Output layer: a layer of neurons responsible for presenting the final results of the network, from the signals received by the layers that precede it.

The Multiple Layer Perceptron (MLP) network contains at least one intermediate layer, in contrast to the single-layer perceptron network, the MLP has one or more hidden or hidden layers between the input and output layers. MLP networks are more complex in their structure, which allows them to perform more complex work compared to the networks mentioned above and can solve problems that would go beyond binary classification. MLP networks have feed-forward regardless of the number of layers, the first layer captures the signals to be processed, then the intermediate layers extract information about the signals, process and encode through their respective synaptic weights, bias, and activation function and the output layer receives the resulting stimuli from the intermediate layers and produces the network response. Note

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

that in an MLP network it is possible to have multiple neurons in the output layer, resulting in the network having multiple output possibilities.

The adjustment of synaptic weights of the MLP network takes place through the backpropagation process, which consists of an algorithm that calculates the gradient of the error function, starting at the output layer and propagating towards the input layer, partially reusing the calculations of the gradient of the previous layer to carry out the weight adjustments of the next layer [18]. The specific configuration of an MLP network must be determined from a series of factors such as the class of problem to be treated by the network, arrangement of training samples, initial values, and attributes so that the network can be implemented efficiently.

A. Convolutional Neural Networks

CNNs commonly used for pattern recognition in images have an MLP network architecture, but they stand out for the presence of convolutional layers and often, pooling layers observed in Fig. 3 in addition to the concepts already present in MLP networks.

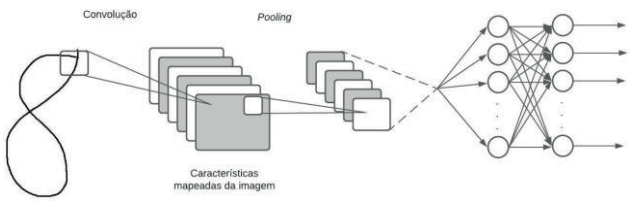


Fig. 3: Convolutional network representation.

Convolutional layers have the function of convoluting a weight matrix sectioned across an image to extract relevant information so that subsequent layers can make use of the extracted information to recognize patterns in specific parts of the image. Thus, a convolutional layer can identify intrinsic characteristics of different parts of the image such as horizontal and vertical lines, and specific angles, among other patterns to be processed by the next layers [19], [20]. Pooling layers accompany the convolution layers and their function is to reduce the dimensions of the data provided by the convolution layer, connecting the output of a group of neurons from the previous layer into a single neuron from the pooling layer. The pooling layer can have different aspects, neurons can extract the maximum value or they can average the values received from the group of the previous layer [19], [20].

III. FPGA

The FPGA has a structure with three main components, LUT is responsible for implementing the logic functions, the input and output blocks allow communication with peripherals and the interconnectors that carry out the communication between the blocks and some other components with more specific functions such as the blocks random access memory (BRAM) and digital signal processing blocks (DSP). All the blocks are configurable so that the user when programming the desired logic into the FPGA, the circuit forms the design structure using the blocks which are essentially a vector or array of combinational logic. Along with LUTs, other resources such as D-flipflops, multiplexers, and transport logic, carry, among others, to implement more complex functions such as boolean functions and multipliers, then LUTs are now called configurable logic blocks (CLB) illustrated in Fig. 4.

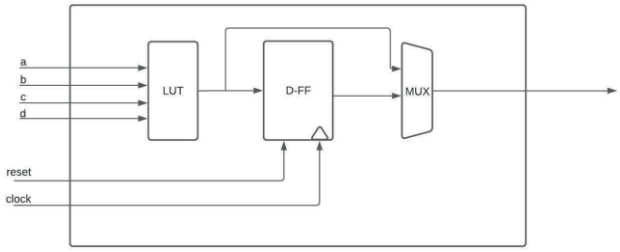


Fig. 4: CLB representation.

Fig. 5 represents a LUT with three inputs (A, B, and C), the possible values of the LUT are stored in a register, because of this, the LUT can be implemented as any function that has the same number of inputs. Once configured the output values are selected according to the inputs. Modern FPGAs have 6-input LUTs and 64-bit registers.

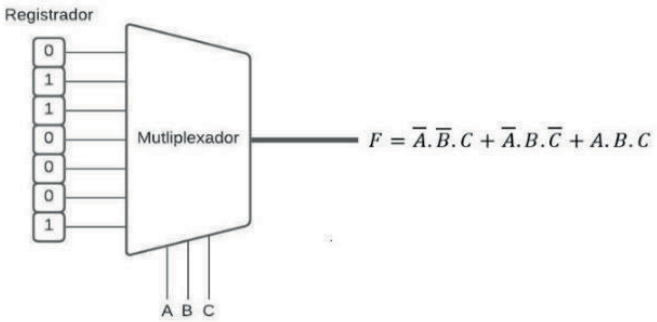


Fig. 5: 3 input LUT representation.

By itself, a single LUT or CLB is very simple and incapable of performing complex logic functions but connected in large amounts are capable of performing complex functions even if the individual power of each block is limited, there are also FPGAs that have a carry chain which connects the LUTs of CLBs with the LUTs of neighboring CLBs allowing the creation of arithmetic functions as adders, with low-level logic efficiently and quickly.

As the priority of FPGA circuits is efficiency in the use of resources present on the board, the interconnection of the most recent FPGAs has logic circuits to assist in the interconnection and routing of the other blocks as seen in Fig. 6, such as connection blocks (CB) that are responsible for connecting the logic blocks with the interconnection rails with the possibility of using any of the rails to assist in routing, and also the switching blocks, which are configurable blocks that connect

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

the rails themselves to provide more routing possibilities at the time of implementation [21].

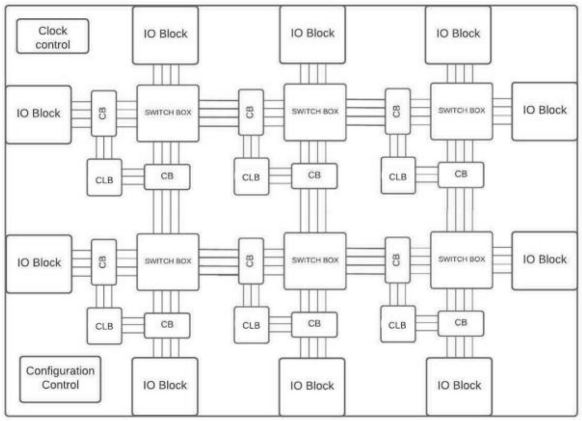


Fig. 6: FPGA structure representation.

BRAMs are memories that allow fast access to data and have the ability to read and write data and are much more efficient for data storage than LUT memory implementations, it is also possible to access two memories in the same block simultaneously, which allows the data preloading or data writing and reading at the same instant. BRAMs can be inferred by the synthesis tool or can be instantiated by the user during design elaboration.

DSP blocks are blocks that allow multiplication operations followed by result accumulation without the use of LUTs, DSPs have a series of configurable functional blocks, the DSP48E1 block present in FPGA Xilinx Series 7 features a 25-by-18-bit two's complement multiplier, a 48-bit accumulator, a pre-adder, a block that can perform the addition, subtraction or accumulating result of multiple data simultaneously, a unit of logical operation with bits like AND, OR, NOT, NAND, NOR, XOR, XNOR, overflow and underflow detectors and configurable pipeline [17].

IV. METODOLOGY

For the implementation of a neural network in an FPGA, it was necessary to generate and train the network externally, for this, R and Python programming languages and the Keras package were used for the elaboration of the neural network and the export of the weights and bias of the trained network with MNIST dataset [22], tests were performed with different network configurations, such as number of layers, number of neurons and activation function to test the implementation in FPGA.

A. Network Weights Preparation

Once the network was trained, the values of the weights and bias of each neuron were exported and manipulated using Excel, with the use of formulas elaborated in the spreadsheets, it was possible to easily process the data for later implementation in the FPGA. For an integer implementation, the weight and bias values were multiplied by a multiple of 10, depending on how accurate the decimal places would be, for example, for three decimal places the values were multiplied by 1000, then rounded using the formula "=round('Nº','Nº of decimal places')". The result was a number with no decimal places, for example, for a weight value of 0.6457, the value implemented in FPGA would be 646.

For training and testing the elaborate network, the MNIST dataset was used, which consists of handmade images of numbers from 0 to 9, widely used for training and testing image processing systems, containing 60,000 images for training and 10,000 for testing, all images are in grayscale with dimensions of 28x28 pixels, so the network input will be 784, a value for each pixel [22].

B. FPGA Implementation

Tests were carried out with different types of networks in the implementation, varying the number of layers, the number of neurons in each layer, except for the output layer, the type of activation function, the precision of decimal places, and the differences between implementation with number integer and with the library for numerical representation with fixed point. For the evaluation of the implementation results, the following parameters were observed: the response time, the network accuracy, the use of board resources, and the energy used for its operation.

1) Block structure: For the implementation in FPGA, using VHDL programming language, different blocks were elaborated that together compose the neural network elaborated previously.

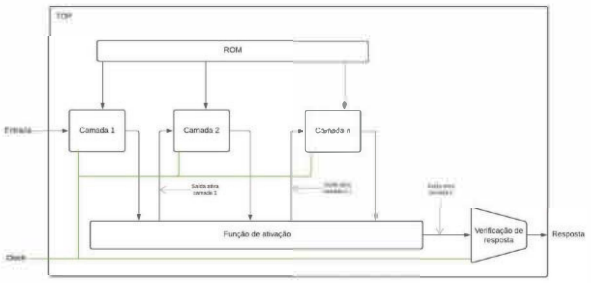


Fig. 7: Representation of the network structure implemented with data storage in LUT.

As illustrated in Fig. 7, the network has six basic blocks in its structure, the TOP block that receives the external data, in the case of this work, receives the values of the pixels of the image to be identified and provides the final response of the network according to the active output of the last layer, the neuron block, illustrated in Fig. 8, is responsible for weighting the inputs by synaptic weights and finally adding the bias, the neurons layer block is responsible for receiving the external input or data from the active output of the previous

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

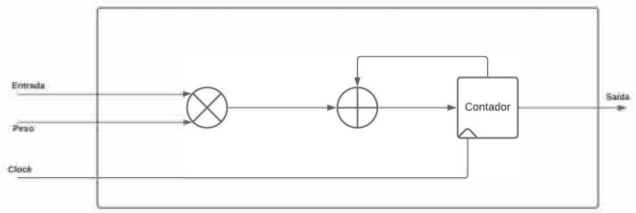


Fig. 8: Single neuron representation.

layer and distribute it to all neurons contained in it, after the completion of the calculations performed by the neurons, the layer sends the accumulated output to the block containing the activation function stored in LUTs or in BRAM, after activating the outputs of the last layer, the data is sent to the block responsible for comparing the activated outputs of the last one and verifying the value most likely to be the correct answer.

The VHDL code was designed in such a way that it is possible to modify the number of neurons and layers with minimal configuration of the code itself, which facilitates testing and adapting the code to different needs such as greater accuracy, fewer used resources, faster response time, etc. after any necessary modifications to the code, the Xilinx Vivado HLS tool is used to synthesize the code for RTL and implement it on the FPGA board.

2) Numerical Representation: As it is not possible to synthesize real numbers in VHDL, tests were carried out with two types of numerical representation. The first form of representation used as integers, with the arithmetic operations already implemented in VHDL through the numeric_std library, to represent decimal places the input and weight values were multiplied by multiples of ten, depending on the determined precision. For example, for the representation of two decimal places the values were multiplied by one hundred, and the remainder was rounded, the values were then transferred to the FPGA.

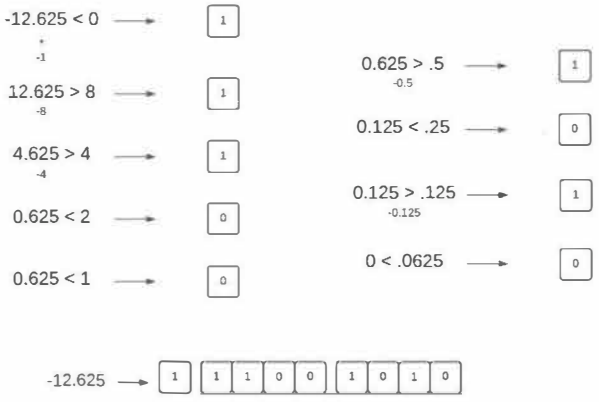


Fig. 9: Fixed point conversion.

The other form of numerical representation used for tests was binary with fixed point, for that, a package was elaborated and implemented that includes conversion from real number to binary with fixed point, addition and multiplication so that it was possible to implement the network using such numerical representation. with ten bits for the decimal part, 14 bits for the integer part, and one bit for the sign. The conversion was done by an algorithm elaborated in Octave and is done in software before the implementation, it consists of comparisons, between the number you want to convert and each of the numbers that are represented by the bits at a fixed point, as illustrated in Fig. 9.

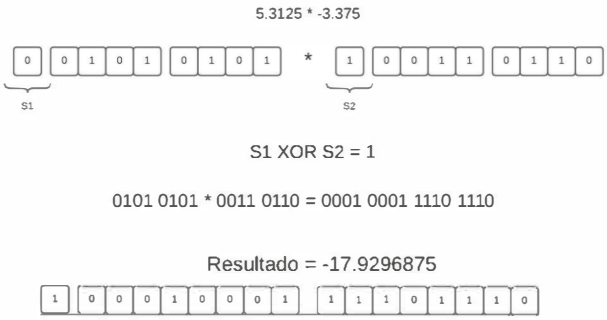


Fig. 10: Fixed point multiplication.

The implemented fixed-point number multiplication is represented in Fig. 10, the operation is relatively simple, performing a standard binary number multiplication and then dividing between the vector that represents the integer number and the vector that represents the decimal. In the example, it can be seen that for multiplication between two numbers with four bits in the decimal part, the result will have the eight least significant bits as the decimal part.

3) Activation Functions: Tests were performed with different activation functions to evaluate results, mainly observing the accuracy and resource requirements of the FPGA used to implement the activation function.

For better response times and simplicity of implementation, the activation functions were calculated in software with a programming language with a high level of Octave abstraction, used primarily for mathematical computation, the values were tabulated and stored in BRAM on the board.

Due to optimizations made by the Xilinx Vivado HLS tool during synthesis and implementation, the algorithm can conclude that the activation function can be implemented in a combination of LUTs, multiplexers, and registers for better use of the final design area.

The tests highlighted that implementing the activation function more efficiently is through LUTs elaborated outside the FPGA and stored in RAM for quick access, saving DSP blocks and processing time due to calculations of the implemented model. The test implementation of the activation function determined that the calculations necessary for the sigmoid activation function would be performed in at least four clock cycles for each value, however with the values in LUTs or RAM only one cycle is needed for each value, decreasing

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

response time without sacrificing accuracy, it is also possible to utilize more resources on the board to calculate the response of the activation function of all output values of a layer at the same time.

4) Weight Storage: Two network models were developed in terms of the way of storing the weights of the network, one of the ways was implementation directly in LUTs of the FPGA, once implemented the weights were stored as functions in LUTs, registers, and multiplexers while the other way is initial storage external value of the weights that are then stored in the FPGA's BRAM and during the calculations, the BRAM supplies the weights to the neurons.

The process of storing the values in RAM requires instantiating the blocks according to the number of neurons so that there is one block per neuron, so there will be no difference in the network response time, the process requires an initialization period for the data to be provided by an external source to RAM, however, it is only necessary to perform the process once.

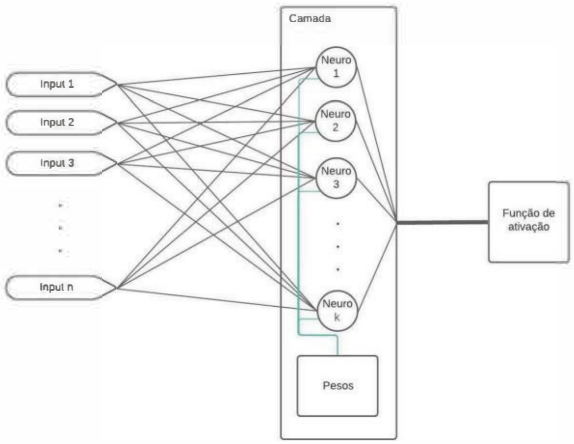


Fig. 11: Representation of a layer with data storage in BRAM.

Fig. 11 illustrates a layer when the method of implementing weights in BRAM is used, the layer block distributes the weights stored in RAM to the respective neurons according to the input, so the layer block has a vector of BRAMs, and each BRAM stores the weight value of each neuron in vector form, so that during the weighting calculation the neuron receives the weight corresponding to the input for each clock period.

5) FPGA Board Used in Tests: To perform the tests, the FPGA Basys 3 Artix-7 board was used, which has 33,280 logic cells in 5200 slices, each containing four LUTs of six inputs and eight flip-flops, 50 RAM blocks of 1,800 Kbits each, 90 DSP48E1 slices. For the implementation, 100MHz was used as clock time [17].

V. RESULTS

A. Parallelism

The initial implementation had total parallelism, with a completely asynchronous network it was possible to perform all the necessary calculations in parallel and obtain the result in the picoseconds scale, however for a small network of ten neurons in the hidden layer and ten in the layer of output were used around 103800 LUTs and more than 10000 DSP slices making the implementation unfeasible for the board available for testing.

Based on this experience, it was decided to perform the calculations synchronously, but many of the calculations are still performed in parallel. The network was then modified to perform the multiplication and accumulation calculations for each neuron synchronously, so that each multiplication and addition calculation for a neuron takes three clock cycles at 100MHz to perform, conserving board resources in exchange for a longer response time, but this allows all neurons to perform calculations in parallel so that the number of neurons in the hidden layer minimally changes the total response time of the layer, but increases the number of input values of the next layer which increases the response time on the order of approximately 20ns per neuron. With the reduction of parallelism, it was possible to reduce the resources used by the implemented network considerably.

B. Decimal Place Precision and Numeric Representation

The variation of the precision of decimal places allows both savings in the use of resources on the board with the use of less precision, while greater precision guarantees greater accuracy of the network in exchange for a greater number of resources used, both in the implementation with integers and in the implementation with fixed point. tests were carried out with a different number of decimal places to verify the use of resources. Table I and II demonstrates the test results for the implementation with two and three decimal points, respectively, while Table III test results for the fixed point implementation.

Varying decimal places and comparing with the final accuracy of the network, it was determined that even using more resources, the implementation with greater precision of decimal places, both in the case of the use of integers, as in the use of 25 bits at a fixed point, offered better results, since better network accuracy is expected due to higher numerical precision.

C. Network Design

The VHDL code was designed in such a way that it allows flexibility both in the number of neurons and in the number of intermediate layers and also in the number of input values in the network, to be able to adapt the implementation according to the need, without being limited for the MNIST dataset [22], to only one model or the board used for the tests performed in this work.

Tests were carried out with different numbers of layers and neurons, which varies the number of resources used, the

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

TABLE I: Integer implementation results with two decimal places of precision.

Integer – 2 decimal places	784x10x10	784x15x10	784x30x10	784x45x10	784x100x10	784x10x10x10
LUTs	6273	7704	11979	13960	33117	7811
DSP	10	25	40	55	100	30
BRAM(18KBits)	10	15	30	45	50	30
Registers	2023	2320	3240	4372	8377	2483
F7 MUX	323	546	973	2330	3229	682
F8 MUX	50	101	150	846	648	168
Original accuracy	92,75%	94,92%	96,70%	97,55%	97,88%	92,42%
Energy consumption	0,133W	0,184W	0,190W	0,197W	*	0,211W
Response time	23.77µs	23.97µs	24.57 µs	25.17µs	27.37µs	47.85µs

TABLE II: Integer implementation results with three decimal places of precision.

Integer - 3 decimal places	784x10x10	784x15x10	784x30x10	784x45x10	784x100x10	784x10x10x10
LUTs	9655	11300	13518	17559	27854	14132
DSP	20	25	40	55	110	30
BRAM(18KBits)	0	15	30	45	50	20
Registers	2955	3026	3423	4460	9652	3358
F7 MUX	1357	1187	1333	1689	3152	1493
F8 MUX	220	228	249	263	740	304
Original accuracy	92,75%	94,92%	96,70%	97,55%	97,88%	92,42%
Energy consumption	0,259W	0,274W	0,307W	0,312W	*	0,288W
Response time	23.77µs	23.97µs	24.57 µs	25.17µs	27.37µs	47.85µs

TABLE III: Implementation results with fixed-point binary.

Fixed point – 25 bits	784x10x10	784x15x10	784x30x10	784x45x10	784x100x10	784x10x10x10
LUTs	7863	8344	10573	12792	22570	9532
DSP	20	25	40	55	110	30
BRAM(18KBits)	10	15	40	45	50	20
Registers	2506	2729	3693	4888	8762	3058
F7 MUX	1146	917	1242	1799	3420	1493
F8 MUX	323	128	157	217	613	304
Original accuracy	0,9275	0,9492	0,967	0,9755	0,9788	0,9242
Energy consumption	.179W	.186W	.26W	.198W	*	.192W
Response time	23.77µs	23.97µs	24.57µs	25.17µs	27.37µs	47.85µs

accuracy of the model, and the response time of the network, after different tests it was concluded that networks with more than one layer did not only consumed more resources and more time, as they did not guarantee better results in the accuracy of the network in general, the network response time being independent of the number of neurons, since all neurons in a layer perform calculations in parallel, the network is more effective, both in response time and accuracy, by increasing the number of neurons in the hidden layer.

D. Activation Function

Tests were performed with different activation functions, taking into account the accuracy of the elaborated network and the consumption of resources on the board, as the output layer needs the Sigmoid or SoftMax activation function, tests were carried out with the two functions that presented the best accuracy during training was Sigmoid and as the number of resources consumed when implementing any of the functions is similar, around 2000 LUTs or five BRAMs for implementation with integers with three places of precision and binary with fixed point and around 1000 LUTs for precision with two decimal places the most used features in the implementations, the tests were carried out with Sigmoid, while the activation function of the middle layer were carried out tests with different activation functions as mentioned above and hyperbolic tangent and RELU, the tests carried out demonstrated that the hyperbolic tangent activation function has lower accuracy and would consume more board resources than the function RELU which also had better accuracy compared to Sigmoid.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

Observing the results of test implementations, it can be concluded that the number of neurons directly affects the resources used by the network and the expected accuracy, while the number of layers also affects the accuracy, consumes more energy and more resources, and increases the response time without a significant increase in.

Comparing the results of integers with representation using binary with a fixed point we can see that there is significant conservation of resources and less consumption of energy, the only obstacle is the conversion of the input data to binary with a fixed point since the algorithm runs externally, the data needs treatment before processing, which would make the network response time larger.

E. Comparison with CPU and GPU Acceleration

The response time in the different tests performed with the hardware implementation was measured and compared with the results of the software implementation using different tools. For testing with CPU, a test was carried out with Intel i7-6700K 4.00 GHz and 16 GB of RAM, it was also carried out in a Kaggle virtual environment, and the results showed an average response time of 20ms for each image, compared with the response time of one-layer network with 45 neurons implemented in 25.17µs FPGA, it is possible to notice that the response time is considerably lower. It was also compared to GPU acceleration on both an Nvidia GTX 980TI GPU with 6Gb of dedicated video RAM and a virtual environment so the time reduction was not very significant, reducing the average response time for individual images to 18ms.

VI. DISSCUSION

Regarding the network accuracy, it is possible to notice that the consumption of resources is directly related to the precision of decimal places of the network, which affects the final accuracy of the implementation, with the calculated error we can estimate the accuracy of the network according to the expected results.

Taking the results of tests carried out with the network implemented with a response time of 25.17us and considering the average speed of a commercial aircraft as 300 km/h, we can estimate that, with the network response time, the plane would travel 0.0020975 meters or 2.0975 millimeters approximately, between the input of the image and the decision making, considering that the response time depends directly on the number of inputs in the network, even if the number of inputs was ten times greater, the plane would still travel less than one meter between image input and decision making.

There is the possibility of using boards with a greater number of resources to implement networks with a greater number of inputs, neurons, and layers.

The greatest demand of convolutional networks is the memory to store parameters and processing power to perform arithmetic and logic operations the main components of FPGA for neural networks are LUTs, BRAM, and DSPs.

VII. FINAL CONSIDERATIONS

Considering the results acquired through the tests of the research carried out, it can be concluded that the implementation of neural networks for image recognition in FPGA has enormous potential for reducing response time, in addition to being economical in terms of energy consumed.

Initial comparisons with CPUs and GPUs show a significant reduction in response time with the potential for optimizations in the implementation to obtain even better results, in addition, the FPGA implementation allows flexibility in updating the network if necessary, complemented by the fact that the code is itself flexible.

A. OPTIMIZATION PROPOSALS

As previously mentioned, there are several ways to implement neural networks for image recognition in FPGAs, being important factors: the number of input values (pixels), the number of dense layers, the number of neurons in each layer, the representation number and its precision. All these factors influence the complexity of the calculations performed, the consumption of logical resources, memory and power on the board, and the total time required for image processing.

There are a series of optimizations that are possible to perform in the model elaborated, in future works, which could further reduce the response time and improve the consumption of resources by the model, some of these improvements are:

- Develop a controller unit for the arithmetic operations performed by the network, thus improving the use of DSPs, using fewer LUTs, and enabling greater parallelism of the calculations performed, limited only by the amount of DSPs on the FPGA board.
- Greater parallelization of the calculations performed, since the calculations consume more time in image processing, using techniques such as parallel reduction could considerably reduce the response time in exchange for greater consumption of resources and energy per image.

B. FUTURE WORKS

The focus of this work was to evaluate the response time for calculations performed by a neural network developed for image recognition, also considering the use of resources and the energy consumed during tests performed with different parameters, but due to the time required, in addition to limitations of resources on the board used for tests for studies carried out on neural networks, FPGA, VHDL, and implementation techniques. It was not possible to implement different types of layers such as the convolution layer and the max pooling layer to test convolutional neural networks completely implemented in FPGAs, considering that the code elaborated is flexible in terms of the number of input values, number of layers and amount of neurons per layer, it is possible to continue the work in the future with the development of convolution and max pooling layers and integrate them into the work already done. Due to the difficulties of testing large amounts of images to verify the accuracy of the network, it would be necessary

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.4 Web Link: http://dx.doi.org/10.18580/setijbe.2022.4

to devise a way to send images quickly to the board to verify the accuracy of the implementation.

REFERENCES

[1] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, "CNN architectures for large-scale audio classification," *CoRR*, vol. abs/1609.09430, 2016. [Online]. Available: http://arxiv.org/abs/1609.09430

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[4] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[6] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 525–542.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[8] R. Kellermann, T. Biehle, and L. Fischer, "Drones for parcel and passenger transportation: A literature review," *Transportation Research Interdisciplinary Perspectives*, vol. 4, p. 100088, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590198219300879

[9] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005621000059

[10] K. Ovtcharov, O. Ruwase, J.-Y. Kim, J. Fowers, K. Strauss, and E. S. Chung, "Accelerating deep convolutional neural networks using specialized hardware," 2015.

[11] A. Jahanshahi, M. K. Taram, and N. Eskandari, "Blokus duo game on fpga," in *The 17th CSI International Symposium on Computer Architecture Digital Systems (CADS 2013)*, 2013, pp. 149–152.

[12] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks," in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 161–170. [Online]. Available: https://doi.org/10.1145/2684746.2689060

[13] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang, "Going deeper with embedded fpga platform for convolutional neural network," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 26–35. [Online]. Available: https://doi.org/10.1145/2847263.2847265

[14] N. Suda, V. Chandra, G. Dasika, A. Mohanty, Y. Ma, S. Vrudhula, J.-s. Seo, and Y. Cao, "Throughput-optimized opencl-based fpga accelerator for large-scale convolutional neural networks," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 16–25. [Online]. Available: https://doi.org/10.1145/2847263.2847276

[15] S. Liang, C. Liu, Y. Wang, H. Li, and X. Li, "Deepburning-gl: An automated framework for generating graph neural network accelerators," in *Proceedings of the 39th International Conference on Computer-Aided Design*, ser. ICCAD '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3400302.3415645

[16] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *CoRR*, vol. abs/1602.02830, 2016. [Online]. Available: http://arxiv.org/abs/1602.02830

[17] Artix-7 fpgas data sheet: Dc and ac switching characteristics. Accessed: 2022-08-20. [Online]. Available: xilinx.com/content/dam/xilinx/support/documents/data_sheets/ds181_Artix_7_Data_Sheet.pdf

[18] I. N. d. Silva, D. H. Spatti, and R. A. Flauzino, *Redes neurais artificiais para engenharia e ciências aplicadas* . Artliber Editora, 2010.

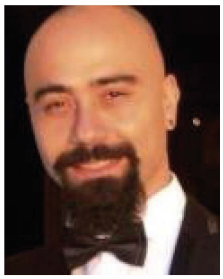
[19] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson, *Handwritten Digit Recognition with a Back-Propagation Network*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, p. 396–404.

[20] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 319–345. [Online]. Available: https://doi.org/10.1007/3-540-46805-6_19

[21] "Acknowledgments," in *The Design* Warrior's *Guide to FPGAs*, C. ldMaxrd Maxfield, Ed. Burlington: Newnes, 2004, pp. xv–xvi. [Online]. Available: www.sciencedirect.com/science/article/pii/B9780750676045500015

[22] "Mnist handwritten digit database," http://yann.lecun.com/exdb/mnist/, accessed: 2022-08-20.

VICTOR M. AGUIRRE received a B.S. degree in electrical and electronic engineering in 2021 from Mackenzie Presbyterian University, São Paulo, Brazil. His main research interests are Neural Networks, Field Programmable Gate Arrays (FPGA), Industrial Automation, and Programmable Logic Controllers (PLC)..



FADI JERJI (S'18) received a B.S. degree in computer engineering in 2010 and an M.S. degree in electrical and computation engineering from Mackenzie Presbyterian University, São Paulo, Brazil, in 2019. He is currently pursuing a Ph.D.degree in electrical and computation engineering at Mackenzie Presbyterian University. Since 2017 he has been a post-grad Researcher with the Digital TV Research Laboratory at Mackenzie Presbyterian University.

Received in 2022-07-15 | Approved in 2022-08-20

Audience and complexity aware live video encoders orchestration

Abdelmajid Moussaoui
Thomas Guionnet
Mickaël Raulet

CITE THIS ARTICLE
Moussaoui, Abdelmajid; Guionnet,Thomas; Raulet, Mickaël; 2022. Audience and complexity aware live video encoders orchestration. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.5. Web Link: http://dx.doi.org/10.18580/setijbe.2022.5

Audience and complexity aware live video encoders orchestration

Abdelmajid Moussaoui, Thomas Guionnet, and Mickaël Raulet.

Ateme {a.moussaoui, t.guionnet, m.raulet}@ateme.com

***Abstract*—Video encoding services are known to be computationally intensive. In a software environment, it is desirable to be able to adapt to the available computing resources. Therefore, modern live video encoders have the "elasticity" feature. That is, their algorithmic complexity adapts automatically to the number and capabilities of available CPU cores. In other words, the more CPU are allocated to a live video encoder, the higher the encoding performance. Until recently, the elasticity feature was used as an ad-hoc adaptation to uncontrollably varying conditions. In this paper, mechanisms allowing to take control of the computing resource are presented. Two real-time resource optimizations strategies are then proposed. The first one is based on video content complexity and manages the video head-end costs, while the second relates to audience measurements and targets network bandwidth usage optimization.**

***Index Terms*—Video compression, live encoding, Kubernetes, orchestration**

I. Introduction

In the field of video encoding, microservices architecture is becoming more and more beneficial over monolithic applications. The concept of microservices [1][2] allows a dramatic reduction of the design and implementation cycles durations and simplifies support and update of the applications. The virtualization concept on the other hand, allows being highly flexible and independent of the hardware. In the case of video compression, where performance is critical, the optimal granularity of the microservices must be optimized under constraints of real-time, low-latency, efficient data flow and availability. Practically, microservices must be stored in containers. The high number of containers requires orchestration. Among many available solutions [4][5][6], the work presented in this paper relies on Docker [7] for containerization and Kubernetes [5] for orchestration.

The video encoding solution considered in this paper is composed of several independent services which are thus managed by Kubernetes. However, the performance of a practical implementation of a video encoder is a trade-off between bitrate, perceived video quality, computing resource and architecture design. Kubernetes allows controlling the number of resources dedicated to each microservice. Thus, in the video compression context, one may consider allocating the resource non uniformly to different video services, depending on the desired trade-off for each video service. This must be carried out explicitly by the user though, since Kubernetes, as an orchestrator, is blind to the specifics of each application.

The proposed allocation solution will leverage previously introduced method [8] to seamlessly update the CPU for a service running on Kubernetes without service interruption. A full experimental system is demonstrated, applying the proposed dynamic resource allocation to a set of live encoders deployed in a Kubernetes environment. The rest of this paper is organized as follows: first, some elements of context and preliminary results are provided. Then two versions of the custom-orchestrator are detailed, complexity-based and audience-based. Finally experimental results are provided for each mode before conclusion.

II. Context, elasticity and CPU allocation

A given video encoder implementation can provide several trade-offs between resource consumption and video quality. This is the case, for example, with the High Efficiency Video Coding (HEVC) implementation x265 [9]. The tuning parameter (`-preset`) allows choosing a speed/coding performance trade-off in a range of predetermined settings. In this paper, the considered encoder adapts automatically to the available computing resources. That is, given the real-time constraint, the encoder chooses its parameters automatically depending on the platform capacity and current load. This tuning is updated dynamically. If the overall load of the platform changes, the tuning changes accordingly. The more computing resources available, the better the delivered coding efficiency. This concept is called video encoder elasticity [14].

As an illustration of the elasticity concept, example experiments have been conducted using the HEVC codec in its default configuration. All the considered video sequences have a 1080p (high definition, HD) resolution. Fig. 1 presents rate-distortion curves [12] for several encodings of the same 12 minutes movie extract. Each encoding is performed in real-time, with a fixed number of central processing unit (CPU) cores allocated to the corresponding microservice. In the video compression context, a rate-distortion curve illustrates the trade-off between bitrate and distortion (or quality) achieved by an encoder implementation or configuration. A configuration is found to be better than a reference configuration if its rate-distortion curve is above the reference rate-distortion curve. That is, for a given distortion, the bitrate is found to be lower, or conversely, for a given bitrate, the quality is found to be higher. The experimental observations confirm that the encoder adapts to the available

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.5 Web Link: http://dx.doi.org/10.18580/setijbe.2022.5

computing resource. Indeed, all the curves of Fig. 1 have been generated with strictly the same configuration, except for the number of CPU allocated. Thus, the rate-distortion performance improves as the CPU number increases.

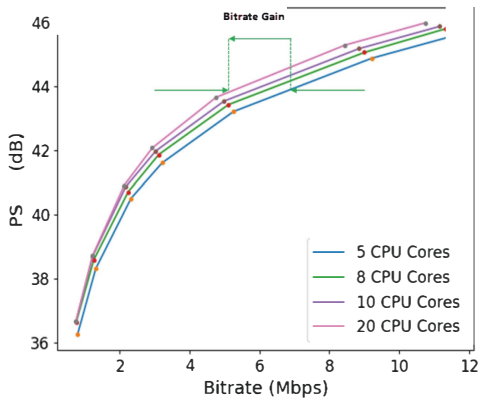


Fig. 1: Rate-Distortion curves for different CPU core allocations.

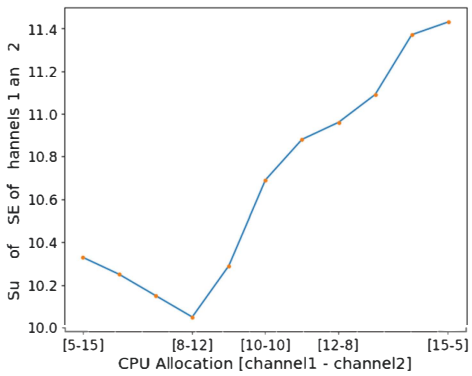


Fig. 2: Sum of Mean Squared Errors (MSE) for different CPU allocations among two video channels.

In a second experiment, the encodings of two different 12 minutes movie extracts are considered. The two contents have the same resolution and are both encoded using HEVC. An arbitrary fixed budget of 20 CPU cores is allocated to be shared between the two encoders. One must note that this fixed CPU budget is shared in a controlled manner between the two channels. A first part is allocated exclusively to the first channel, and the remaining part is allocated exclusively to the second channel. One may split it even and allocate 10 CPU cores to each channel or decide to allocate more CPU cores to one of the channels. The goal of this experiment was to find the optimal repartition of these 20 CPU cores between the two encoders, which minimizes the distortion for a given bitrate. The experiment showed that the allocation that maximizes the overall quality is not uniform, as illustrated on Fig. 2.

Both encoders have the same configuration, the difference is the encoded content itself. The channel 2 contains more complex content compared to channel 1. A video sequence is said to be more complex if it contains more information, like more motion or image texture, than the other sequence. The encoder must make more effort on a complex sequence to achieve the same coding efficiency as on a simple sequence.

III. Complexity Based Orchestration

A. *Dynamic CPU allocation*

The second experiment (Fig. 2) showed that for two channels with the same configuration, the allocation that minimized the distortion – thus maximizes the video quality – is not a uniform allocation, but rather a CPU cores distribution where the channel with high content complexity needs to be allocated more than the lower content complexity channel. Additionally, it is well known that the characteristics of contents are not constant in time. This is especially true for a 24/7 live channel. With a limited number of computational resources, dynamic resource allocation can improve the overall compression efficiency of a set of live channels.

The encoders run as part of a micro-services application in a Kubernetes cluster. All encoding services are running in Pods, the smallest Kubernetes manageable unit. A Pod contains one or several containers, and the hardware resources (CPU, memory, …) are managed at the container level. The native and supported way for Kubernetes to update the resources allocated to a container in a given Pod is to stop and restart the Pod with the desired resources allocation.

For a live video encoder, the reboot of the Pod even for milliseconds will lead to the loss of multiple video frames. However, service interruption of a live service is not acceptable. In a previous work [8] authors proposed a method for dynamic resource allocation for Kubernetes Pods with zero downtime.

The allocation system relies on an interaction between operating system features and Kubernetes device plugin feature [11]. It consists in updating the number of resources advertised to the Kubernetes scheduler and changing the current allocation using the Linux system tools in a way that is transparent to Kubernetes.

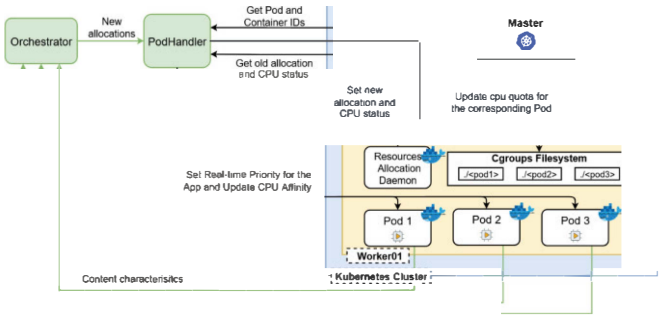


Fig. 3: Resources updating and orchestration process.

Fig. 3 present the interaction between Kubernetes cluster and the dynamic allocation service (PodHandler). The PodHandler gets the new allocation computed by the orchestrator, then interacts with the device plugin to update the number of custom resources advertised to Kubernetes Scheduler, the next step is to update the Pod's Cgroups [10] Completely Fair Scheduler Quota (CFS Quota) that controls the Pod's CPU usage limit. Linux tool taskset is used to change CPU affinity to meet the new allocation. Finally, the resource state is updated for every server in a database managed by the Resource Allocation Daemon service.

B. Complexity based orchestration

The orchestrator computes optimal CPU resource allocation and relies on the PodHandler to apply this allocation. The orchestration algorithm is organized in two steps:

- Predicting the bitrate gain with the help of a machine learning algorithm
- Computing the optimal allocation that minimizes the function (1), based on the bitrate gain predictions:

$$J = \sum_{i=1}^{N} b_i + \lambda * d_i , \quad (1)$$

where, N is the number of channels b_i is the bitrate of the channel i, d_i is its video distortion and λ is the Lagrange multiplier.

For the first step, the orchestrator uses a trained machine learning model that predicts for every channel the possible gain of a given CPU allocation with respect to a reference allocation, the model takes as input several parameters:

- Video Codec (HEVC, AVC, AV1…)
- Channel configuration (Frame rate, resolution, bit depth…)
- Video quality (PSNR)
- Number of CPU cores
- Channel's complexity estimation

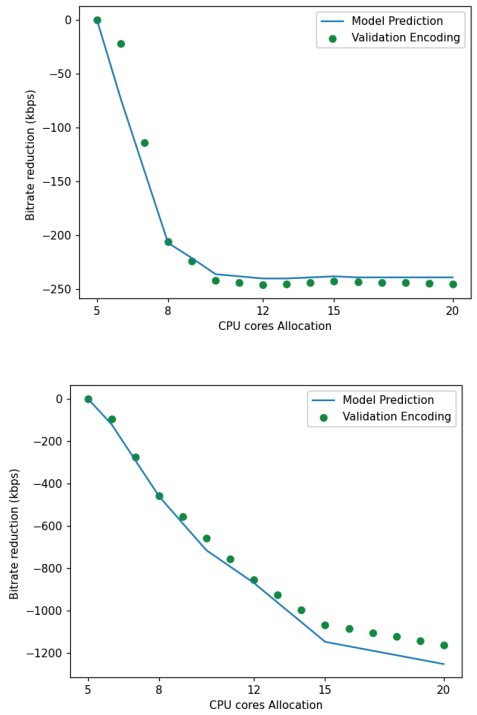


Fig. 4: Bitrate gain predicted by KNR model for, (a) low complexity channel, (b) high complexity channel.

A K-Neighbors Regressor (KNR) [18] algorithm is used for this task. The dataset is composed of various encodings with different configuration and content complexity. Fig. 4 illustrates the predictions made by the model for two different video sequences. The first sequence exhibits low complexity content and the second one high complexity. Both contents are encoded by an HEVC encoder and have the same configuration (25 fps, 1080p resolution, 8-bit depth…). The graphs in Fig. 4 show the KNR model estimation of the bitrate gain for a given CPU allocation with respect to the minimum allocation (5 CPU cores here), the video quality remaining constant. The complex channel takes better advantage of any additional CPU core. Additionally, from a CPU cores number threshold, additional CPU core will no longer provide bitrates reduction. This threshold is much higher for the high complexity channel than for the low complexity channel. Finally, the reliability of the prediction is assessed by comparing it to the actual encoder behavior.

In a second step, the result of gain estimation is used to compute the optimal allocation that will minimize the cost function. The algorithm proposed is a greedy algorithm, and since all curves predicted by the KNR model are strictly decreasing, it is guaranteed that it will return the optimal solution.

Let N be the number of channels in a given server, minAllocation returns the minimum allocation possible for a video channel to operate normally and M is the disputed CPU cores given in function (2):

$$M = totalCPUs - \sum_{i=1}^{N} minAllocation_i \quad (2)$$

Allocation Algorithm

Initialize with the minimum allocation
Output: allocation
For channel = 1 to N **do**:
  allocation[channel] <-- minAllocation[config]
End

For cpu <-- 1 **to** M **do**:
  For channel = 1 to N **do**:
    CurrentAlloc <-- allocation[channel]
    $gain[channel] <-- \boldsymbol{KNR}(currentAlloc, complexity, quality, config) + \lambda * d_i$
  End
ChosenChannel <-- argmax(gain)

allocation[ChosenChannel] = allocation[ChosenChannel] +1
End

Fig. 5: Allocation algorithm for complexity-based orchestration.

Since the model returns the bitrate gain for a constant PSNR, the distortion term in the algorithm is also constant, thus, the Lagrange multiplier can be put to 0.

The algorithm provided on Fig. 5 will return the allocation that minimizes the total bitrate of all channels while keeping the same video quality. Note that the number of allocated CPU cores to the channels is an integer number, in order to ensure optimal usage of threading and CPU cache memory. After receiving the number of cores, the PodHandler will take care of finding the best CPU affinity considering the NUMA architecture [13] of the physical processor.

C. Complexity-based orchestration experimental results

1) Bitrate minimization

Many tests have been conducted with various configurations and repeated to validate the stability of the

system running live. From a large set of varied sequences, several subsets have been selected to perform our experiments. Little variation in the results has been observed, as long as the subsets are heterogeneous. In a sense, the behavior of the system is comparable to a statistical multiplexer (statmux), as an allocation for a set of sequences having all the same characteristics brings little to no gain. The following example has been kept as a meaningful representative of these experiments.

Four 1080p channels of 5 minutes duration, encoded using an HEVC encoder, configured in constant quality mode and targeting the same video quality. This mode delivers variable bitrate (VBR) streams. Therefore, the performance at a given quality is measured by the bitrate. The better the allocation, the lower the bitrate. An arbitrary number of 28 CPU cores is available to be shared between the 4 channels. These channels have all different content complexity levels.

Two allocation scenarios are run and compared. The first is a uniform allocation, where every channel gets a fixed 7 CPU cores no matter its content. The second is dynamic allocation; in this mode, the orchestrator will compute the optimal allocation periodically based on the content complexity.

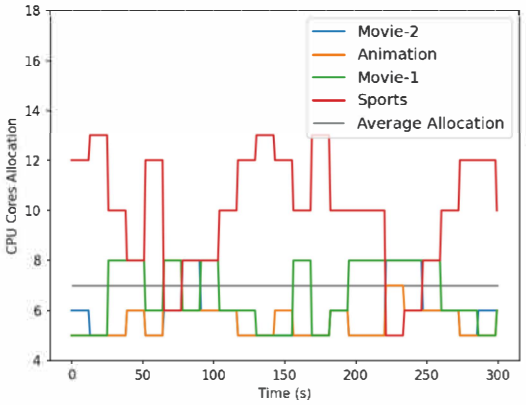


Fig. 6: Dynamic allocation of 28 CPU cores among 4 channels encodings.

Fig. 6 illustrates the changes of the allocation over time for the four channels depending on their respective complexities. As one may expect, the sports content is more complex than the others, hence a larger allocation has been granted to it almost all the time.

TABLE I: BDRATE GAINS COMPARED TO UNIFORM ALLOCATION.

	Movie-1	Movie-2	Sports	Animation	Mean
BDRate	2.93%	2.63%	-8.85%	1.23%	-0.51%

For the proposed combination of sequence and settings, rate distortion curves are derived from which Bjøntegaard Delta Rates (BDRates) [3] can be computed. Table I presents the BDRate gains relative to the uniform allocation. Negatives values indicate a gain (bitrate reduction), and positives values a loss. The first observation is that resource augmentation for one channel implies resource reduction for at least one other channel, leading to BDRate losses. Still, with the proposed dynamic allocation, an overall BDRate gain is achievable.

However, the BDRate is a relative performance metric especially when comparing sequences with different content types. The actual bitrates are provided in Table . The overall performance is measured by the sum of the bitrates for the 4 channels, with a lower total bitrate indicating better performance.

TABLE II: BITRATES IN MBPS FOR ALL RUNS AT THE SAME QUALITY.

	Uniform	Dynamic	Gain
Movie-1	0.686	0.709	3.35%
Movie-2	0.245	0.250	2.04%
Sports	4.015	3.469	- 13.6%
Animation	0.195	0.198	1.54%
Total	5.141	4.626	- 10%

Compared to uniform allocation, dynamic allocation reduces the bitrate by 10%. For the highest bitrate sequence, Sports, the required bitrate is reduced by 13.6% thanks to dynamic allocation, which represents more than 0.5 Mbps on a very demanding content. The absolute bitrate increase on the other channels is comparatively negligible. Gaining more than 0.5 Mbps on a channel is an opportunity to reach more users with the full resolution quality. For the content provider, it also translates into cost control. With uniform allocation, more CPU cores would be necessary to reach the same bitrate as the proposed solution, hence a higher cost. In a summary, this experiment showed 10% overall bitrate gain in dynamic allocation mode while using the same CPU budget and achieving the same video quality.

2) CPU Usage Optimization

In the previous experiment the goal was to allocate the available CPU cores in order to reduce the required bitrate at a given video quality. In a case where the aim is to minimize the encoding cost, i.e., to use less CPU cores (e.g., when using public cloud) or increase the channels density (have more channels in the same server), dynamic allocation allows reducing the total CPU cores required for a set of channels compared to the uniform allocation mode while achieving the same bitrate for the same video quality.

TABLE III: BITRATES (MBPS) FOR UNIFORM AND DYNAMIC ALLOCATION WITH DIFFERENT CPU BUDGET.

	Uniform 44 CPU Cores	Dynamic 28 CPU Cores	Gain
Movie-1	0.679	0.709	4.42%
Movie-2	0.246	0.250	1.63%
Sports	3.546	3.469	-2.17%
Animation	0.2	0.198	-1%
Total	4.671	4.626	-0.96%

The experiment setup is the same as the previous one, four live HD channels are encoded with an HEVC encoder in constant quality mode. For the uniform allocation, 44 CPU cores are allocated to the channels (11 cores for each). The Table III shows the bitrates achieved for a given video quality in the uniform and dynamic CPU allocation modes. For the dynamic allocation, a total budget of 28 CPU cores is allocated which is 36% less than the 40 CPU cores of the uniform allocation. Yet, a gain of 0,96% of required bitrate is achieved compared to the uniform allocation mode. In

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.5 Web Link: http://dx.doi.org/10.18580/setijbe.2022.5

summary, this experiment shows that one can save up to 36% of CPU cores when applying a content complexity aware dynamic allocation on a set of live channels in a public or private cloud.

IV. Audience and Complexity Based Orchestration

The complexity-based allocation method optimizes the video encoding performance in the video head-end. The result was a lower overall bitrate compared to a uniform allocation. However, some channels have seen their required bitrates increased because they are less complex. In a real use-case, some channels may be more popular than the others, so the bitrate gain, or loss, of a channel affects the video traffic on the network and is eventually multiplied by the number of viewers that are watching the channel.

In this chapter, a new method is introduced to take the channel audience into account in addition to the content complexity when computing the allocation. Just like complexity, the number of viewers of a live channel can change over time, therefore dynamic allocation is the more suitable allocation mode.

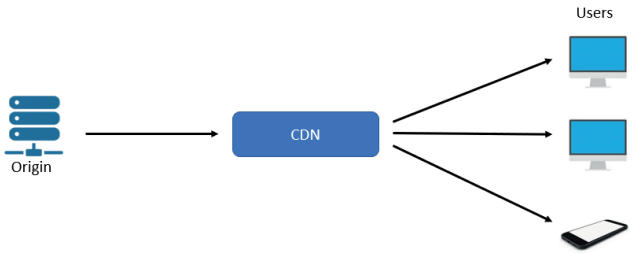


Fig. 7: Video streaming over CDN.

A. Video distribution network

Live Video streaming can be performed through different set-ups, either a digital video broadcast or an OTT (Over the Top) media streaming. A typical example is OTT streaming over a content delivery network (CDN) as presented in Fig. 7, which is one of the most used set-ups for live and VOD streaming (Video on Demand).

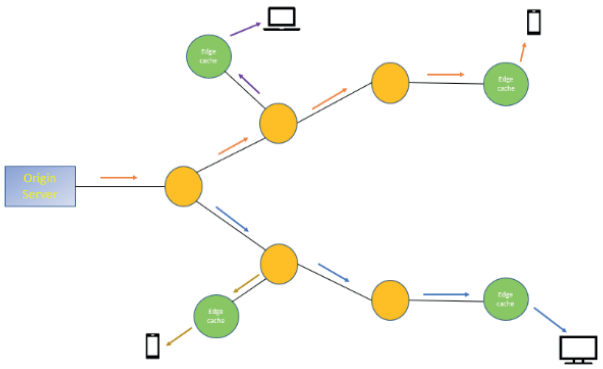


Fig. 8: Example of CDN architecture, with Origin server in blue, nodes in yellow and edge caches in green.

A CDN is a group of geographically distributed and interconnected servers, it provides cached content to the end users. In the field of video streaming, CDN is an essential component in the distribution scheme. Video channels require a high bandwidth to be transmitted, some channels can have thousands or possibly millions of viewers watching at the same time. The origin server usually has limited capabilities, thus cannot serve all the viewers, even if it can, the viewers could be distributed all over the world, the stream then should travel a long distance for every single viewer. Using a CDN, the stream is provided once by the origin and delivered to the edge cache servers in the viewers' regions, so all the viewers in the same geographic area can fetch the content from the nearest CDN edge cache (Fig. 8).

There are various solutions to get the audience measurements, from the CDN itself like Wowza Media System, which provides a near real-time API [16] to query the number of viewers for a given channel, or, directly from the players, as for example, Smart Sight API [17] by Media Melon, which gives real-time analytics collected from the players.

Allocation Algorithm

Initialize with the minimum allocation
Output: allocation
For channel = 1 to N **do**:
  allocation[channel] <-- minAllocation[config]
End

For cpu <-- 1 **to** M **do**:
  For channel = 1 to N **do**:
    CurrentAlloc <-- allocation[channel]
    $gain[channel] <-- (\mathbf{1} + v_i) * \boldsymbol{KNR}(currentAlloc, complexity, quality, config) + \lambda * v_i * d_i$
  End
ChosenChannel <-- argmax(gain)

allocation[ChosenChannel] = allocation[ChosenChannel] +1
End

Fig. 9: Allocation algorithm for audience and complexity-based orchestration.

B. Cost function optimization

The goal is to minimize the overall distributed data over the network by the streaming, going from the origin server where the channels are encoded, to the end users passing through the CDN edge cache servers, under the constraint of the video head end limited CPU resource. The cost function to minimize is given as (3):

$$\boldsymbol{J} = \sum_{i=1}^{N} \boldsymbol{b_i}\,(\mathbf{1} + \boldsymbol{v_i}) \; + \; \boldsymbol{\lambda} * \; \boldsymbol{v_i} * \boldsymbol{d_i} \qquad (3)$$

Where, $\boldsymbol{N}$ is the number of channels, $\boldsymbol{b_i}$ is the bitrate of the channel $\boldsymbol{i}$, $\boldsymbol{d_i}$ is its video distortion, $\boldsymbol{v_i}$ the number of viewers and $\boldsymbol{\lambda}$ is the Lagrange multiplier. The $\mathbf{1}$ in the term $(\mathbf{1} + \boldsymbol{v_i})$ corresponds to the stream distributed from the origin server to the CDN.

To optimize the function, the previously trained KNR model is still applicable. The greedy algorithm however

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.5 Web Link: http://dx.doi.org/10.18580/setijbe.2022.5

needs to consider the number of viewers of the channels. The updated algorithm is presented in Fig. 9.

TABLE IV: OVERALL BITRATES GENERATED BY TWO CHANNELS, FOR DIFFERENT VIEWERS DISTRIBUTIONS.

Sports-1 Viewers	Movie-3 viewers	Uniform (Gbps)	Dynamic (Gbps)	Gain (%)
0%	**100%**	602.51	577.02	-4.23
5%	**95%**	617.41	596.0	-3.47
25%	**75%**	676.99	651.5	-3.77
50%	**50%**	752.09	751.46	-0.08
75%	**25%**	825.94	790.15	-4.33
95%	**5%**	885.52	820.1	-7.39
100%	**0%**	900.41	829.24	-7.9

C. *Primary experiments*

Several experiments have been conducted to emphasize the importance of including the audience measurements along with the complexity to maximize the encoding performance in a cost-effective manner. In the first experiment, two HD channels that have a nearly equivalent complexity levels are encoded with an HEVC encoder in various scenarios. The first channel Sports-1 is a 5-minutes rugby match sequence, and Movie-3 is an action movie with the same duration. An arbitrary total number of viewers is taken as 100000 viewers for both channels. The considered scenario is that X% (percent) of the viewers are watching Sports-1 and (100 - X) % are watching Movie-3.

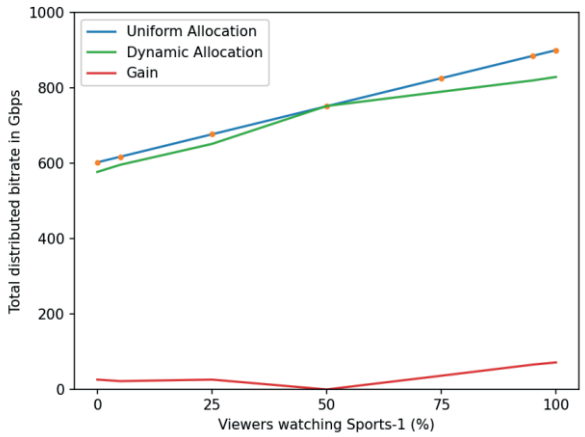


Fig. 10: Bitrate over the distribution network of two similar complexity channels with uniform and dynamic CPU allocation.

In all the different scenario presented in Table IV and Fig. 10, the proposed dynamic orchestrator managed to perform better than the uniform allocation, with a gain varying as a function of the viewers distribution. In the case where the two channels are equally popular, the number of viewers is the same and the complexity is equivalent, so the dynamic orchestrator will allocate approximately a uniform allocation and that explain why the gain is low. Also, the sequence Sports-1 is slightly more complex than Movie-3, that why the total bitrate and the gain are larger when the Sports-1 is the most viewed.

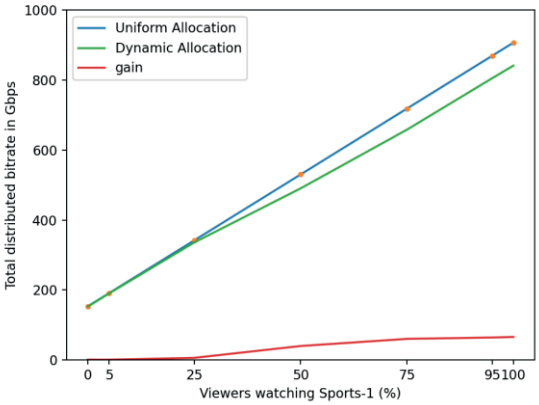


Fig. 11: Bitrate over the distribution network of two different complexity levels channels with uniform and dynamic CPU allocation.

The second experiment is conducted in the same configuration, two HD channels encoded with an HEVC encoder. However, the sequences in this experiment have different complexity levels. The first sequence is Sports-1, and the second is Movie-1, a historical movie. The results are shown in Fig. 11. In the case of equal popularity, the orchestrator still reduces the overall bitrate, because of the complexity difference. Moreover, when the less complex channel is the most viewed, there is always a gain with respect to uniform allocation even if it's smaller.

D. *Full-scale experimental results*

Here, a real use-case is simulated, where a content or service provider has a set of live channels distributed to its subscribers, each channel may be watched more or less than the others. In this experiment four HD channels are encoded with an HEVC encoder with the same configuration. The video sequences used have different intrinsic content complexity levels. The goal is to reduce the overall bitrate generated by the channels over the distribution network (CDN) when applying audience and complexity based dynamic allocation compared to a uniform static allocation.

Several scenarios are considered where the distribution of viewers is different. The total number of viewers is 100000, the different distributions of the viewers over the 4 channels are provided in Table V.

TABLE V: VIEWERS DISTRIBUTIONS SCENARIO DESCRIPTION.

	Sports	Movie-1	Movie-2	Animation
Scenario-1	70000	10000	10000	10000
Scenario-2	10000	70000	10000	10000
Scenario-3	10000	10000	70000	10000
Scenario-4	10000	10000	10000	70000

In the first scenario, the channel **Sports** is considered the most watched, with 70% of total number of viewers, while the other channels get each 10% of the total viewers. The results are presented in Table VI, corresponding to the total data generated by the four channels. The dynamic allocation mode reduced this number by 13.33%, it is 45.95 Gbps

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.5 Web Link: http://dx.doi.org/10.18580/setijbe.2022.5

(Gigabit per second), and more than 14% of the most popular channel spared just by changing the way that the available CPU cores are allocated to the channel.

TABLE VI: TOTAL DATA GENERATED BY THE CHANNELS IN (MBPS).

	Uniform	Dynamic	Gain
Movie-1	7812	8386	7.35
Movie-2	2920	2998	2.66
Sports	331813	285137	- 14.07
Animation	2163	2237	3.42
Total	344708	298758	- 13.33

The other scenarios have been tested as well; the results are summarized in Table VII. In all the presented use cases, the orchestrator succeeds in finding the optimal allocation that reduces the total bitrate. The most popular channel will have a larger weight (3), and consequently may be allocated more CPU cores. The gain is maximal when the complex channel is the most viewed, this is the expected behavior as illustrated in Fig. 2, where the potential bitrate gain increases with the content complexity.

TABLE VII: PERCENTAGE OF TOTAL BITRATE REDUCTION MADE IN DYNAMIC CPU ALLOCATION COMPARED TO UNIFORM CPU ALLOCATION.

	Scenario-1	Scenario-2	Scenario-3	Scenario-4
Movie-1	7.35	-8.12	0.06	0.63
Movie-2	2.66	12.37	-8.84	7.19
Sports	- 14.07	-6.73	-6.46	-6.56
Animation	3.42	1.84	-0.34	-1.06
Total	- 13.33	-5.42	-7.41	-2.01

V. CONCLUSION

In this paper, a new method is introduced for computing the CPU allocation for live video encoders. It is demonstrated that dynamic allocation is more suitable and give a significant bitrate reduction compared to a uniform static allocation. In a first mode, complexity aware dynamic orchestration showed a gain up to 13.6% on a very demanding content, and an average of 10% reduction of the overall bitrate required by four channels. The second proposed mode considers the number of viewers for each channel in an OTT streaming environment. This method offers further gains, with more than 14% reduction of the overall bitrate distributed over the network by a complex channel, and an average of 13.33% compared to a uniform allocation.

Finally, the proposed method needs 36% less CPU cores compared to a uniform allocation to achieve the same video quality at the same bitrate, which could reduce the operational costs significantly.

REFERENCES

[1] N. Dragoni et al, Microservices: yesterday, today, and tomorrow 2016. https://doi.org/10.48550/arXiv.1606.04036
[2] Francesco et al, Architecting with microservices: A systematic mapping study. https://doi.org/10.1016/j.jss.2019.01.001
[3] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, Technical Report VCEG-M33, ITU-T SG16/Q6, Austin, Texas, USA, 2001.
[4] Docker Inc., Docker Swarm, Container's orchestrator. https://docs.docker.com/engine/swarm/key-concepts/
[5] The Linux Foundation , Kubernetes Platform Container's Orchestrator. https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/
[6] The Apache Software Foundation , Mesos A distributed systems kernel. http://mesos.apache.org/
[7] Docker Inc , Docker Software containerization platform. https://docs.docker.com/get-started/overview/
[8] A. Moussaoui, M. Raulet and T. Guionnet, "Dynamic Seamless Resource Allocation for Live Video Compression on a Kubernetes Cluster," in SMPTE Motion Imaging Journal, vol. 131, no. 4, pp. 45-49, May 2022, doi: 10.5594/JMI.2022.3160832.
[9] MulticoreWare Inc, https://x265.readthedocs.io , x265 HEVC implementation.
[10] Serge Hallyn, Linux Control Groups File System https://man7.org/linux/man-pages/man7/cgroups.7.html
[11] The Linux Foundation , Kubernetes device plugin https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/device-plugins/
[12] Yochai Blau https://onlinelibrary.wiley.com/action/doSearch?ContribAuthorRaw=Berger%2C+Toby, Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff https://doi.org/10.1002/0471219282.eot142.
[13] Linux Kernel Organization, NUMA https://www.kernel.org/doc/html/v4.18/vm/numa.html
[14] Herbest et al, Elasticity in Cloud Computing: What It Is, and What It Is Not. https://www.usenix.org/conference/icac13/technical-sessions/presentation/herbst
[15] J. Vanne et al. Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs
[16] Wowza Inc. Viewers data API https://www.wowza.com/docs/how-to-get-viewer-data-for-a-wowza-cdn-stream-target-by-using-the-wowza-streaming-cloud-rest-api
[17] MediaMelon Inc. Players QoE analytics API https://www.mediamelon.com/product-smartsight-qoe
[18] Kramer, O. (2013). K-Nearest Neighbors. In: Dimensionality Reduction with Unsupervised Nearest Neighbors. Intelligent Systems Reference Library, vol 51. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38652-7_2

Abdelmajid Moussaoui received an engineering degree in multimedia networking from Telecom-Paris, Paris, France, in 2020. He is a research engineer in the Research and Innovation Department of ATEME. His current research areas are video encoding, codec orchestration in the cloud, and machine learning. He holds several pending patents related to the optimization of the orchestration of live video channels distribution in the cloud.

Thomas Guionnet is a fellow research engineer at ATEME, where he currently leads the innovation team's research on artificial intelligence applied to video compression. Beyond his work for ATEME, he has also contributed to the ISO/MPEG - ITU-T/VCEG - VVC, HEVC, and HEVC-3D standardization process; he teaches video compression at theESIR Engineering School, Rennes, France; and he has authored numerous publications including patents, international conference papers, and journal articles. Prior tojoining ATEME, he spent 10 years at

Envivio conducting research on real-time encoding, video-preprocessing, andvideo quality assessment. He holds a PhD from Rennes 1University, Rennes.

Mickaël Raulet is the chief technology officer at ATEME, where he drives research and innovation with various collaborative research and development projects. He represents ATEME in several standardization bodies: ATSC, DVB, 3GPP, ISO/IEC, ITU, MPEG, DASH-IF, CMAF-IF, SVA, and UHDForum. He is the author of numerous patents and more than 100 conference papers and journal scientific articles. In 2006, he received a PhD from INSA in electronic and signal processing, in collaboration with Mitsubishi Electric ITE, Rennes, France.

Received in 2022-07-08 | Approved in 2022-08-17

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.5 Web Link: http://dx.doi.org/10.18580/setijbe.2022.5

Contributions to TV 3.0 using 5G-MAG Reference Tools

Wesley Henrique Silva de Souza
Cristiano Akamine

CITE THIS ARTICLE

de Souza, Wesley Henrique Silva; Akamine, Cristiano; 2022. Contributions to TV 3.0 using 5G-MAG Reference Tools. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.6. Web Link: http://dx.doi.org/10.18580/setijbe.2022.6

Contributions to TV 3.0 using 5G-MAG Reference Tools

Wesley Henrique Silva de Souza and Cristiano Akamine

Electrical Engineering and Computing Program
Mackenzie Presbyterian University
Sao Paulo, Brazil

wesley.souza@mackenzista.com.br, cristiano.akamine@mackenzie.br

***Abstract*— The evolution of the evolved Multimedia Broadcast Multicast Service (eMBMS) to the Further evolved Multimedia Broadcast Multicast Service (FeMBMS) in Release 14 of 3GPP enabled broadcast transmission in a format 100% dedicated to user devices. As a result, the 5G standard for cellular networks expanded and began to be quoted in the broadcasting sector as 5G Broadcast. This standard is one of the quoted ones to integrate the TV 3.0 architecture in Brazil, being responsible for the physical layer. To be possible, this technology must meet some requirements, such as negative noise carrier ratio, MIMO antennas and channel bonding. To complement the tests conducted by the SBTVD Forum, this paper aims to evaluate and discuss the SNR and minimum signal level tests using an Open-Source receiver called 5G-MAG that is managed by the group of the same name. The tests were carried out using a Universal Software Radio Peripheral (USRP) Software Defined Radio (SDR) reproducing I/Q file with 5G Broadcast data with a bandwidth of 6MHz, the width of interest of the Forum, to transmit the signal via GNU Radio software and another USRP SDR as a receiver, having as interface the 5G-MAG.**

***Index Terms*—5G-Broadcast, TV 3.0**

I. INTRODUCTION

Linear TV has a prominent role in Brazilian society. According to a study called "Inside Video 2022" conducted by Kantar IBOPE Media, a global leader in media intelligence, the reach of linear TV arrived 93% of the Brazilian population in 2021, with a dedication of 79% of the time in open TV stations. and 21% on video platforms [1]. These data are important, as they encourage interest in modern technologies around broadcasting in the country. Digital TV in Brazil is undergoing a process of evolution, which has been called TV 3.0. A disruptive evolution with improvements in image definition and processing, in addition to audio. 5G Broadcast was one of the standards evaluated at the physical layer and will be analyzed in this article. As the name suggests, the standard uses the fifth evolution of the mobile network for linear TV broadcasting. This feat is possible using a feature called Further evolved Multimedia Broadcast Multicast Service (FeMBMS) released in Release 14 of LTE and which allows up to 100% dedication to broadcast and multicast services. The proponents responsible for presenting this standard to the SBTVD Forum's Call for Proposals were Qualcomm, Rohde & Schwarz, and Katherin [2]. An open-source solution is managed by the 5G Media Action Group (5G-MAG) [3]. This group is an independent, non-profit association whose objective is to be a bridge between the media sectors and the ICT (information and communication technology) industries. They manage a range of tools called 5G-MAG Reference Tools that aim to receive 5G broadcast transmission in dedicated mode, as specified in Release 14 and 16 of 3GPP; combine transmission with mobile broadband, with the possibility of switching modes and developing interactive applications. The receiving system has three interfaces, the MBMS modem, middleware, and a web interface. The last two are optional unless the media content to be received is in HLS format. The system can be configured using two SDRs, one to transmit the content played by GNU Radio and the other to receive the content (see Fig. *1*).

Fig. 1. Representation of the test setup used in the laboratory.

This paper aims to discuss the results of SNR and minimum signal level tests and to evaluate the performance of the open-source receiver.

Based on the above, this paper will be organized as follows: Section II will present the theoretical framework that provides the basis for the investigation; Section III will describe the materials and methods involved in the tests; Section IV will present the performance evaluation and comparison with the SBTVD Forum test results for TV 3.0 and conclusions are drawn in Section V.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.6 Web Link: http://dx.doi.org/10.18580/setijbe.2022.6

II. TECHNOLOGY OVERVIEW

A. TV 3.0

TV 3.0 is the name given to the new open TV project, already in its final testing phase, which aims to replace the current model, SBTVD, which is currently in transition to TV 2.5. Its development is being led by the SBTVD Forum together with the Brazilian Ministry of Communications. TV 3.0 will bring new high-tech experiences to Brazilian homes. The SET's TV 3.0 Working Group [4] cites interactive services such as multi-angle camera service, which allows the viewer to watch a certain scene with a specific camera view; TV social service, which provides interactivity between people in a chat room; Emergency Warning Broadcast System (EWBS), as the name suggests, is an emergency information service that consists in making the viewer aware of the importance of the message and how to proceed; multi-language hidden language service, this service offers the possibility of inserting subtitles in another language via the internet; and sign language animation service that enables the insertion of other services for hearing and visually impaired people. The TV 3.0 architecture proposed by the SBTVD Forum is seen in Fig. *1*Fig. 2.

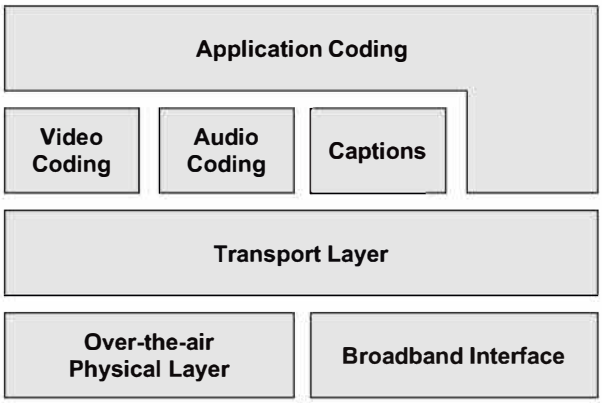


Fig. 2. TV 3.0 Architecture [2].

The system considers two complementary delivery methods: over-the-air and Internet. The physical Internet access interface (broadband interface) is outside the scope of the TV 3.0 system definitions and is considered any IP-based, bidirectional broadband access interface. From the architecture shown in Fig. 2, one can say:

The over-the-air physical layer comprises the unidirectional transmission modulation/ demodulation scheme and error correction. The transport layer comprises the multiplexing and transport of video, audio, subtitles, and applications, as well as all necessary metadata. The application coding in TV 3.0 includes, in addition to the interactivity and broadcast-band integration functions, handles the presentation of all audiovisual content. The audio, video, and subtitle encoding are self-evident and are responsible for the treatment of video, audio, and subtitles in transmission and reception. The intention of the third generation of Brazilian terrestrial television is to bring together the best of the current broadcasting technologies. Each block shown in the architecture of Fig. 2 was subject to proposals for evaluation by the SBTVD Forum. The over-the-air physical layer is the flagship of this work. For TV 3.0 this layer should, in principle, be deployed in the bands currently allocated to digital terrestrial TV in Brazil (VHF and UHF), using the 6 MHz channel bandwidth and should coexist with adjacent ISDB-T channels for a long time without mutual interference. Another specification is frequency-1 reuse, i.e., the use of the same RF channel by independent stations covering adjacent service areas. Such a feat would provide great flexibility for the transmission network, which could be freely expanded and subdivided using the same channel. It would also increase the resilience and robustness of the network. Transmission of the reuse frequency requires C/N $\leq$ 0 dB. Taking advantage of this robustness, the new physical layer is also intended to target external mobile and fixed internal mobile reception with the same signal with a single modulation, coding and quality while maintaining the current network topology so as not to increase the cost of signal distribution. Another specification is the use of MIMO antennas, such antennas and Channel Bounding increase the channel capacity to compensate for C/N $\leq$ 0 dB in a Rayleigh channel which implies a very limited channel capacity. Another feature needed for the TV 3.0 physical layer is to carry a "wake-up" signal (to turn receivers on stand-by) in case of an emergency warning. Finally, it should be noted that the physical layer must allow for future extensions.

B. 5G Broadcast

5G Broadcast is a candidate technology for the TV 3.0 physical layer. It enables broadcast, i.e., point-to-multipoint transmission, and can operate in receive-only mode without the need for SIM cards and network subscriptions. This system can operate in the same band used for terrestrial broadcasting services, it is important to note that 5G broadcast is different from 5G mobile, 5G defines two modes of broadcast communication, standalone and multicast in mixed mode. 5G broadcast is in standalone mode, while the other mode contemplates 5G New Radio (NR). The transmission of 5G Broadcast is conducted by broadcasters, the advantage is that cell phones with 5G can receive the 5G broadcast signal. With the 5G spectrum auction taking place in Brazil in 2021, 5G Broadcast would add value to traditional TV broadcasting, offering additional functionalities and applications and contributing to an efficient distribution of the TV signal to mobile users in the country.

5G Broadcast has been incorporated into Release-14 of 3GPP. The study of EnTV (Enhanced TV) has opened up the opportunity for broadcasters to offer all their services - linear and non-linear to 3GPP devices, enabling broadcast capabilities within the 3GPP system itself.

According to Barquero [5] the European Broadcasting Union (EBU) has coordinated all activities regarding broadcasters' involvement in 3GPP. In Release-15 [6] the emphasis was on the possibility to dynamically switch between unicast, multicast and broadcast modes in order to respond to varying demands, for example as a consequence of varying user distributions and simultaneous service requests. Transmitter site-to-site distances, even beyond, must be considered to cater for very large coverage areas, as well as dedicated modes to support mobile MBMS services up to 250 km/h.

For [7] and [8] the main improvements to the eMBMS system architecture introduced in EnTV are:

- - A receive-only device mode to allow the transmission of free-to-air content that can be received by all devices, including devices without uplink capabilities, SIM cards or 3GPP network subscriptions. A specific application of this mode is to allow the transmission of free-to-air content on eMBMS.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.6 Web Link: http://dx.doi.org/10.18580/setijbe.2022.6

- A transparent (transport/pass-only) delivery mode for using the eMBMS network as a content delivery platform that allows reuse of broadcast services without decoding, ensuring backward compatibility and minimal effort to migrate from legacy systems. It allows, for example, the use of MPEG-2 Transport Stream (TS) over IP. This mode has been added so as not to limit the supported TV formats to the standardized 3GPP media layer services.

The main enhancements to the eMBMS Radio Access Network introduced in EnTV are:

- Dedicated carriers with up to 100% allocation of broadcast content without any resources allocated for unicast and autonomous system information and synchronization signals.
- New subframe type without unicast control region to reduce signaling overhead in downlink-only eMBMS transmissions.
- Shared broadcast networks where different operators can aggregate their radio access networks and MBMS to create a common distribution platform, avoiding the transmission of the same broadcast content on multiple networks; and
- - Support for longer inter-site distances in SFN with a new OFDM numerology.

[5] discusses this topic in depth.
[9] presents two cases of open source 5G receivers using SDRs.

III. MATERIALS AND METHODS

A basic telecommunications system consists of something that transmits the message and something that receives it, and the medium that is the communication channel. The equipment used is shown in TABLE 1

Table 1. List of the equipment used.

1	PC Core i7, ninth generation, 16GB RAM (RX)
1	PC Core i7, 12th generation and 16GB RAM (TX)
1	USRP UHD Ettus B200 (TX)
1	USRP UHD Ettus B210 (RX)
1	Variable Attenuator Model 50DR-001
1	NOD 5200 Noise Generator - Micronetics
1	Anritsu MS8901A Spectrum Analyzer

The I/Q files for transmission are available on the 5G-MAG Github [10]. All files share the same characteristics of cyclic prefix (1.25 kHz), MCS16 (16QAM) and differ in bandwidth, with 3, 5, 6 MHz. From the tests performed by the SBTVD Forum [11] these configurations did not obtain negative C/N, which is one of the requirements for the new TV system in Brazil. This requirement was partially achieved with MCS3 and MCS2 values. Although TV 3.0 requires MIMO, 5G-MAG supports only SISO.

To transmit the RF signal on the SDR the open-source software GNU Radio that provides signal processing blocks for implementing radio and digital processing systems was used. Channel 37 which occupies the UHF band from 608 MHz to 614 MHz was used to transmit the data to the receiver. The 5G-MAG receiver consists of up to three interfaces depending on the payload to be received. If the payload contains DASH (Dynamic Adaptive Streaming over HTTP) or HLS (HTTP Live Streaming) content, then there is a need for MBMS Middleware, otherwise there is no need and the content can be viewed by another application, such as FFMPEG or VLC. The MBMS Modem is the head and can operate either automatically or manually via commands. The Webinterface is optional, but it shows the constellations and other data, such as the type of bandwidth used and the gain of the receiver. The transmit and receive setup has been validated and no errors have occurred. Fig. 6 shows the receiver interface, it displays the current SDR, synchronization and constellations settings of PDSCH, PMCH and services.

1) SDR: This field provides information about the interface device settings.

I. Frequency: is the system's operating frequency.
II. Gain: indicates the receiver gain.
III. Antenna: indicates which input of the SDR is being used.
IV. Sample rate: indicates the sample rate.
V. Filter BW: indicates the size of the bandpass filter; and
VI. Buffer Level: indicates the System buffer level.

2) SYNC: Displays data regarding system synchronization. The features are:

I. Status: indicates the synchronization status.
II. CFO (Carrier Frequency Offset): indicates the size of the frequency offset to achieve synchronism.
III. Cell ID: Cell ID.
IV. PRB (Physical Resource Block): is the smallest provisioning unit in an LTE frame and differs with bandwidths.
V. Width: indicates the bandwidth used.
VI. Subcarrier Spacing: indicates the spacing between the carriers; and
VII. CINR (Carrier to Interference Noise Ratio): is a measure of signal effectiveness, the higher the level, the better.

3) PDSCH (Physical Downlink Shared Channel): This is the downlink data channel for the users. The more concentrated the points are, the better the reception of the
4) PMCH (Physical Multicast Channel) (MCCH, Multicast Control Channel): Control channel, does not need high rates, so MCS2.
5) PMCH (MCH 0): Transport channel, MSC16, constellation 16QAM.
6) Services: shows the information regarding the transmitted services.
All constellations show bit error rates and block errors.

B. Least Signal Level test versus Gain of the 5G-MAG Receiver

The minimum signal level test is intended to evaluate the reception sensitivity of the 5G-MAG. The transmitter signal level was set at -30 dBm, and the receiver gain varied from 10 to 50 dB, with a 5 dB interval. The I/Q files used were 3, 5, 6 MHz. The TOV (Threshold of Visibility) was taken as the 1 dB level above the point where artifacts occur in the image, observed on the TV. The setup used is shown in Fig. 3. For the transmission setup a GNU Radio flow Graph was used, to which the channel, TX gain and I/Q file was set. The attenuator 50DR-001 has an accuracy of 1dB, the spectrum analyzer was used to check the signal level when the video

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.6 Web Link: http://dx.doi.org/10.18580/setijbe.2022.6

started to show artifacts on the screen. The video time to validate the configuration was 1 minute. The receiver was configured in manual mode, i.e., controlled via commands in the Ubuntu terminal.

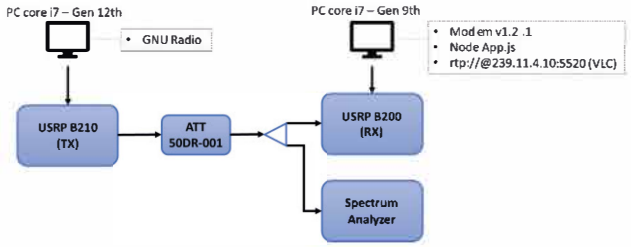


Fig. 3. Setup of the minimum signal level test.

C. C/N test

In addition to the minimum signal level test, the C/N test is important, because digital TV depends on other factors to work without interruptions, the signal level alone is not enough. The current SBTVD system uses a carrier/noise threshold of 19dB [12]. The purpose of this test is to find out this threshold for TV 3.0.

The transmitter signal level was set to -30 dBm, and the receiver gain to 20 dB. The I/Q files used were 3, 5, 6 MHz. The TOV was considered to be the level 0.1 dB above the point where artifacts occur in the image, observed on the TV. The setup used is shown in Fig. 4. In the transmission setup a GNU Radio flow Graph was used, to which the channel, TX gain and I/Q file was set. The NOD 5200 noise generator has an accuracy of 0.1dB, and has three levels of accuracy, tens, units, and hundredths. The control was made to change the values in the higher levels and taper off as the transmission image presented artifacts. The spectrum analyzer was used to fix the signal level. The video time to validate the configuration was 1 minute. The receiver was configured in manual mode, i.e., controlled via commands in the Ubuntu terminal.

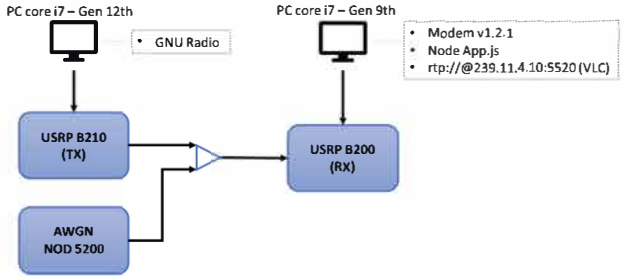


Fig. 4. Setup for the C/N test.

D. C/N test versus Gain of the 5G-MAG Receiver

The test setup is like the C/N test (see Fig. 4), the band chosen was 3MHz, as it was the configuration that was currently running. The transmitter signal level was set at -30 dBm, and the receiver gain varied from 10 to 50 dB, with a 5 dB interval. The TOV was considered to be the level 0.1 dB above the point where artifacts occur in the image, observed on the TV. The video time to validate the configuration was 1 minute. The receiver was configured in manual mode, that is, it was controlled via commands in the Ubuntu terminal.

IV. RESULTS AND COMMENTS

The C/N versus receiver gain test was performed first to have a basis for the other tests. Although Fig. 5 shows an increasing trend, there was inconsistency of values.

The best receiver setting was G = 20 dB. In this setting the signal synchronized quickly. The gain set at 50 dB did not get a good response, with the image freezing even without noise.

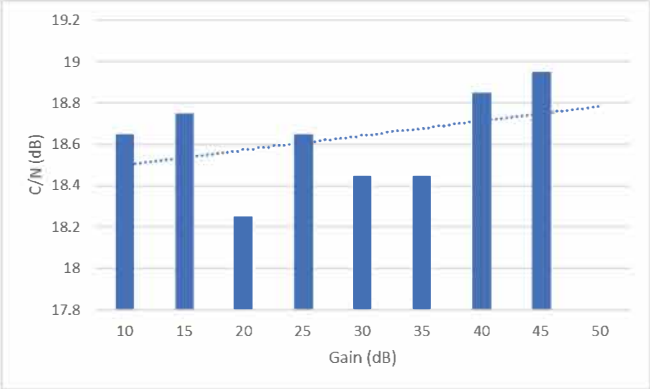


Fig. 5. C/N with different gain values in the 3MHz band receiver.

For the minimum signal level test (see Table 2) the average system margin was -80.5 dBm for the 3MHz band, -75.7 for the 5MHz band, and -65.8 for the 6MHz band. The best sensitivity occurred with the 45 dB gain.

Table 2. Result of the minimum signal level test.

	BW (MHz)		
Gain (dB)	3	5	6
10	-59,87 dBm	-61,9 dBm	-63,0 dBm
15	-67,17 dBm	-64,08 dBm	-63,2 dBm
20	-67,32 dBm	-64,40 dBm	-63,71 dBm
25	-67,25 dBm	-65,96 dBm	-65,81 dBm
30	-68,79 dBm	-67,06 dBm	-68,62 dBm
35	-77,58 dBm	-73,18 dBm	-72,19 dBm
40	-79,63 dBm	-76,50 dBm	-73,08 dBm
45	-80,5 dBm	-78,38 dBm	-78,08 dBm
50	-	-75,66 dBm	-75,70 dBm

The C/N test showed that the carrier-to-noise ratio was very close to the threshold in effect in the current transmission system which is 19dB. Figures Fig. 6 and Fig. 7 show the receiver interface without and with AWGN noise, respectively. The CINR value depicts what is seen in the constellations. Fig. 7 shows no BLER and BER errors in any of the constellations. However, it is important to note that the C/N≤0 condition was not observed in any of the tests, indicating that the system does not match the SBTVD Forum requirements for TV 3.0.

Table 3. Results of the C/N test.

BW (MHz)	C/N (dB)
3	18,25
5	19,24
6	18,60

Table 4 and Table 5 present the results obtained by the SBTVD Forum during the second phase of tests using the 5MHz band. Table 4 brings the results of the C/N test for VHF channels 7 and 10 and UHF channels 14, 33 and 51 with signal level of -28dBm. Analyzing the data in Table 4 with the tests performed in this paper, it is evident that the bit rate of the MCSs has an impacting factor on the C/N, the higher the modulation order, the more susceptible to errors the

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.6 Web Link: http://dx.doi.org/10.18580/setijbe.2022.6

system becomes, this demonstrates that 5G Broadcast is not a robust system. Table 5 presents the minimum signal level values for the 5MHz band. The tests done by the SBTVD Forum for the 5MHz band resulted in -83.9 dB for the field tests [13] and -98.5 dB [11] for the laboratory tests.
Compared to the test in this paper the receiver used by the Forum had a sensitivity of more than superior to the 5G-MAG used in this paper, this indicates that both receivers are sensitive to signal variations, with the receiver used by the Forum being more sensitive than the 5G-MAG. The sensitivity of the receiver is important, since the medium used by radio transmitters is air, and air is susceptible to various sources of noise, which degrade the signal.

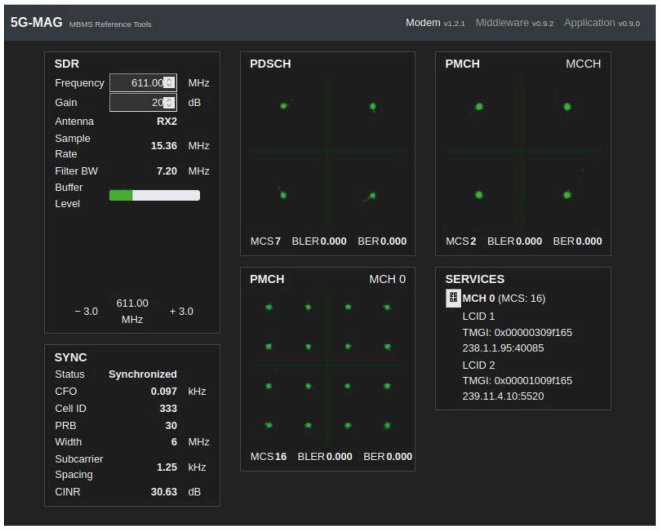


Fig. 6. Signal reception in the absence of noise for BW = 6MHz.

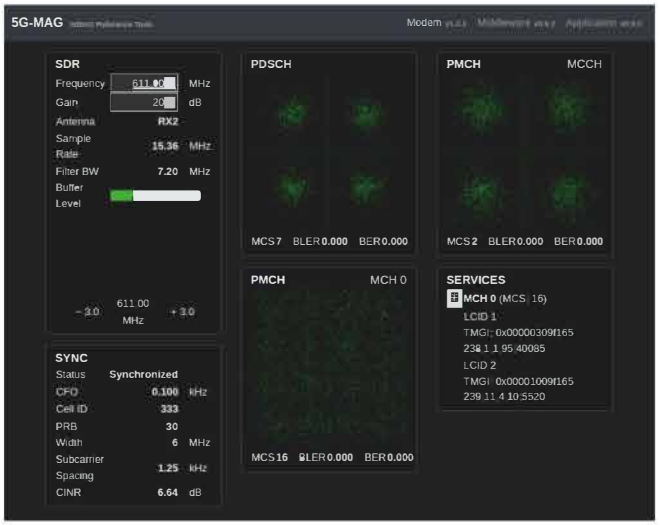


Fig. 7. C/N test at the reception threshold for BW = 6MHz.

Table 4. C/N test done by the SBTVD Forum using BW = 5MHz.

CH	C/N			
	MCS2	MCS3	MCS4	MCS9
7	0,1	0,00	0,10	4,50
10	0,2	-0,10	0,10	4,60
14	-0,1	0,00	0,00	4,70
33	0,2	-0,20	0,10	4,80
51	0,1	-0,10	0,30	4,60

Table 5. Receiver Minimum Level for SISO Configuration done by SBTVD Forum using BW = 5MHz.

CH	Min. Level Signal (dBm)
7	-98,7
13	-99,0
14	-98,2
33	-98,3
51	-98,6

V. CONCLUSIONS

The receiver does not have an automatic gain control, and although the webinterface presents a method for varying this characteristic, it does not work, because when changed the modem abruptly stops. This could be a point of improvement to the system. Another feature to be improved is the synchronization, sometimes the receiver cannot synchronize and locate the channel, being necessary to run the application several times for this to happen. Considering that TV 3.0 is a disruptive system, the fact that the 5G-MAG receiver is open-source contributes to the easy adhesion by the industry, which will have to change devices when this new system starts to be implemented in 2024. In 2023 phase 3 of the TV 3.0 trials will begin, and if 5G Broadcast can meet the requirements of TV 3.0 it is expected that this system will enter commercial operation in 2025. The fact that this receiver can operate in other bands is a positive point, and this feature is provided for in the Forum requirements, however, more importantly, the negative C/N and operability using MIMO have not been met, even for low MCS numbers. The use of MIMO is not contemplated by 5G-MAG/ 5G Broadcast. Further testing can be done, provided there is a configuration that meets the negative C/N.

ACKNOWLEDGEMENTS

The author would like to thank the PPGEEC program and the MACKPESQUISA.

REFERENCES

[1] Kantar IBOPE Media, "Inside Video Novos horizontes e descobertas 2022," 06 2022. [Online]. Available: https://www.kantaribopemedia.com/wp-content/uploads/2022/06/Inside-Video-2022-Kantar-IBOPE-Media_.pdf.

[2] Fórum SBTVD, "TV 3.0 Project," 2021. [Online]. Available: https://forumsbtvd.org.br/tv3_0/.

[3] 5G Media Group, "5G-MAG," 2022. [Online]. Available: https://www.5g-mag.com/.

[4] Grupo de Trabalho de TV 3.0 da SET, "GT de TV 3.0: Conceitos Fundamentais de TV Conectada," SET, São Paulo, 2020.

[5] D. G. Barquero, J. J. Gimenez and R. Beutler, "3GPP Enhancements for Television Services: LTE-Based 5G Terrestrial Broadcast," *Wiley Encyclopedia of Electrical and Electronics Engineering,* p. 16, 21 05 2020.

[6] ETSI TS 123 502, "5G, Procedures for the 5G System - (3GPP TS 23.502 version 15.2.0 Release 15)," ETSI, Sophia Antipolis, 2018.

[7] T. Stockhammer, I. Bouazizi, F. Gabin, J. Guyot, C. Lo, T. Lohmar and C. Thienot, "Enhanced TV service over 3GPP MBMS," *IBC 2017,* pp. 1-8, 2017.

[8] T. Stockhammer, G. Teniou and F. Gabin, "3GPP based TV service layer," *IBC 2016,* pp. 1-8, 2017.

[9] A. Ibanez, J. Sanchez, D. Gomez-Barquero, J. Mika, S. Babel and K. Kuehnhammer, "5G Broadcast SDR Open Source Platforms," *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB),* pp. 1-6, 25 07 2022.

[10] 5G-MAG, "Sample Files," 18 03 2022. [Online]. Available: https://github.com/5G-MAG/Documentation-and-Architecture/wiki/Sample-Files.

[11] SBTVD Forum, "Testing and Evaluation Report: TV 3.0 Project - Over-the-air Physical Layer - Laboratory Tests," 03 12 2021. [Online].

[12] ABNT NBR 15604, "Televisão digital terrestre — Receptores," ABNT, Rio de Janeiro, 2008.

[13] SBTVD Forum, "Testing and Evaluation Report: TV 3.0 Project - Over-the-air Physical Layer - Field Tests," 03 12 2021. [Online]. Available: https://forumsbtvd.org.br/wp-content/uploads/2021/12/SBTVD-TV_3_0-PL-Field-Report.pdf.

Wesley Henrique Silva de Souza - He is a Master's student in Electrical Engineering with research area in Telecommunications at Mackenzie Presbyterian University (MPU), holds a degree in Electrical Engineering with emphasis in Telecommunications, Electronics and Automation from the same institution (2020) and technical professional qualification in Electro-electronics from the Technical Institute of Barueri (TIB) since 2014.

CRISTIANO AKAMINE received a Ph.D. degree in electrical engineering from the State University of Campinas, Brazil, in 2011. He is a Professor at Mackenzie Presbyterian University, where he is a Coordinator of the Digital TV Research Laboratory. He is a member of the Board of the Brazilian Digital Terrestrial Television Forum and Society of Brazilian Broadcast Engineers (SET). He works with the ISDB-TB broadcasting standardization and holds several patents, intellectual property licenses. He also has published numerous articles and has a Brazilian scientific grant of Productivity and Technological Development and Innovative Extension—Level 2 from the National Council of Technological and Scientific Development. He has also served as a reviewer for several periodicals and conferences and has participated as a Guest Editor in the Special Issue Point-to-Multipoint Communications and Broadcasting in 5G of IEEE Communications Magazinestitute of Barueri (TIB) since 2014.

Received in 2022-07-10 | Approved in 2022-08-29

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.6 Web Link: http://dx.doi.org/10.18580/setijbe.2022.6

Comparison of the Physical-layer Performance between ATSC 3.0 and 5G Broadcast

Seok-Ki Ahn,
Sung-Ik Park

CITE THIS ARTICLE

Ahn, Seok-Ki ; Park, Sung-Ik; 2022. Comparison of the Physical-layer Performance between ATSC 3.0 and 5G Broadcast . SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.7. Web Link: http://dx.doi.org/10.18580/setijbe.2022.7

Comparison of the Physical-layer Performance between ATSC 3.0 and 5G Broadcast

Seok-Ki Ahn, *Member, IEEE* and Sung-Ik Park, *Fellow, IEEE*

***Abstract*—This paper compares the physical-layer performances of ATSC 3.0 and 5G broadcast in the scenario to provide mobile broadcasting services. The differences of physical-layer between ATSC 3.0 and 5G broadcast are discussed in terms of transmission efficiency, overheads, and BICM performance over mobile environments. Through the computer simulations, it is shown that ATSC 3.0 can provide more robust and enhanced physical-layer performance than 5G broadcast over mobile environments.**

***Index Terms*—ATSC 3.0, 5G broadcast, mobile broadcasting, physical-layer.**

I. Introduction

In 2016, the Advanced Television Systems Committee (ATSC) approved the physical layer standard of ATSC 3.0 systems, which support efficient and flexible delivery of broadcast services for both fixed and mobile receivers [1]. As a result, high quality broadcasting services such as three-dimensional and ultra-high definition (UHD) contents are possible over ATSC 3.0 thanks to its superior spectral efficiency, channel robustness, and flexible transmission. Since ATSC 1.0 was first standardized, many innovative physical-layer technologies have been evolved and applied to ATSC systems. For example, low-density parity-check (LDPC) codes and non-uniform constellation (NUC) are regarded as two representatives of the superiority of ATSC 3.0 physical layer.

On the other hand, in the 3rd generation partnership project (3GPP), a work item for LTE-based 5G terrestrial broadcast (referred to as 5G broadcast in this paper) is carried out during 3GPP release (Rel-) 16 and finished with some enhancements in 2019 [2], [3]. The major improvements of the 5G broadcast over the underlying technology further evolved multimedia broadcast multicast service (FeMBMS) is to support a large geographical area up to 100km inter-site distance (ISD) and high mobility up to 250km/h. Due to the above enhancements, 5G broadcast not only can meet the 5G terrestrial broadcast requirements but also becomes a competitive technology comparable to ATSC 3.0.

As the proportion of mobile terminals is increasing in terms of the receiving terminal for broadcasting services, the capability to provide services for mobile user equipment (UE) is becoming more and more important. For this reason, 5G broadcast has been drawing attention as a future terrestrial broadcasting solution. Recently, a Mark One smartphone has been released that enables ATSC 3.0 service in mobile terminals to facilitate the reception of ATSC 3.0 in a mobile environment. According to the trend of mobile broadcasting services, this paper comprehensively compares the capability of both technologies in terms of providing mobile broadcasting services.

TABLE I
PHYSICAL LAYER OVERHEAD OF ATSC 3.0 AND 5G BROADCAST

	ATSC 3.0	5G broadcast
Guard band overhead (%)	2.8	10.0
Cyclic prefix overhead (%)	20.0	20.0
Pilot pattern overhead (%)	16.7	16.7
Total overhead (%)	35.2	40.0

II. Physical Layer Overview

To provide stable terrestrial broadcasting services over mobile environments, superior and robust physical-layer performance is essential, which can be expressed as transmission efficiency as the following formula:

$$\text{Efficiency} = SE(SNR)\times(1-OH_{SP})\times(1-OH_{GB})\times(1-OH_{CP}) \quad (1)$$

where OH_{SP} is the scattered pilot overhead of channel estimation, OH_{GB} is the overhead of guard band, and OH_{CP} is the overhead of the CP of an orthogonal frequency-division multiplexing (OFDM) symbol. In addition, SE(SNR) represents bit-interleaved coded modulation (BICM) spectral efficiency, which means the number of data bits per channel use that can be received successfully, depending on the received signal-to-noise power ratio (SNR). The main overheads in the physical layer of both systems are compared in Table I. For fair comparison, CP length and pilot pattern of ATSC 3.0 are chosen as closely as possible to those of 5G broadcast, although other parameters are permitted in ATSC

TABLE II
BICM COMPONENTS OF ATSC 3.0 AND 5G BROADCAST

	Channel codes	Constellation	T/F interleaver
ATSC 3.0	LDPC codes	NUC/QAM	Used
5G Broadcast	Turbo codes/ Convolutional codes	QAM	Not

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-00846, Development of Convergence Transmission and Technology for 5G and ATSC 3.0 Networks).

Seok-ki Ahn and Sung-Ik Park are with the Media Research Division, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700 South Korea (e-mail: {seokki.ahn, psi76}@etri.re.kr)

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.7 Web Link: http://dx.doi.org/10.18580/setijbe.2022.7

TABLE III
EVALUATION ASSUMPTIONS

	ATSC 3.0	5G broadcast
FFT size	8192	12288
Guard interval	222.22us	200us
OFDM duration	888.83us	800us
Subcarrier spacing	1.125kHz	1.25kHz
Center frequency	500Hz	
Bandwidth	8MHz	
Channel Model	India-Urban [5]	
UE Mobility	60km/h	
Channel estimation	LS estimation + Linear interpolation	

3.0 standard. As shown in Table I, the total overhead of 5G broadcast is slightly larger than that of ATSC 3.0.

The BICM spectral efficiency and robustness is mainly determined by the component technologies such are channel codes, signal constellation, and time/frequency (T/F) interleavers [4]. Since 5G broadcast is based on the LTE physical layer, which is first standardized in 3GPP Rel-8, there are limitations in the viewpoint of physical-layer performance. Table II summarizes them between ATSC 3.0 and 5G broadcast.

III. PERFORMANCE RESULTS

In this section, physical-layer performances of ATSC 3.0 and 5G broadcast are evaluated by computer simulations in terms of block error rate (BLER). Note that the Log-MAP algorithm with 8 iterations is used for turbo codes and sum-product algorithm with 50 iterations is used for LDPC codes. The BLER performances of the data channel of two systems are evaluated over additive white Gaussian noise (AWGN) channel and India-Urban channel [5] in Fig. 1 and Fig. 2, respectively, whose evaluation assumptions are given in Table III.

For ease of comparison, data rate of 5Mbps, 10Mbps, and 15Mbps are chosen for performance comparison. Thanks to the superiority of the LDPC codes in ATSC 3.0 compared to the turbo codes in 5G broadcast, the decoding performance of ATSC 3.0 is better than that of 5G broadcast over AWGN channel. In addition, in the case of 15Mbps, ATSC 3.0 provides strictly better performance than 5G broadcast due to the shaping gain of NUC under high-order modulations (HOMs).

The performance gain of ATSC 3.0 under mobile environment is evaluated over India-Urban channel in Fig. 2. As shown in Fig. 2, the performance gain is larger than that over AWGN channel due to the time interleaver. For this reason, the performance gain increase as the target BLER decreases because the BLER curves of ATSC 3.0 is steeper than that of 5G broadcast. As a result, SNR gain for achieving BLER = 10^{-4} reaches up to about 8.5dB, 9.0dB, and 12.5dB for 5Mbps, 10Mbps, and 15Mbps, respectively.

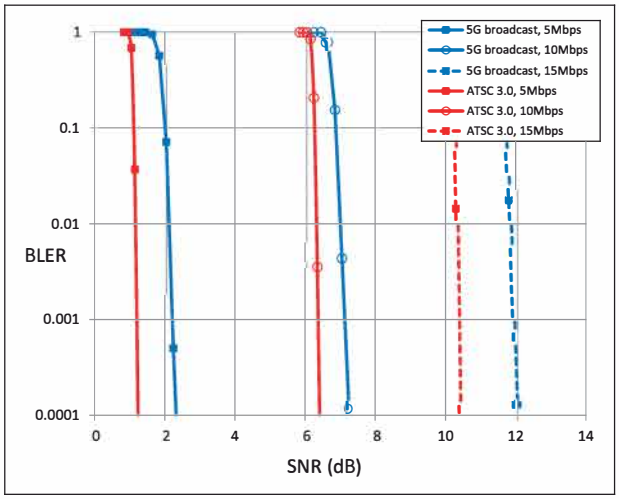


Fig. 1. BLER performance over AWGN channel

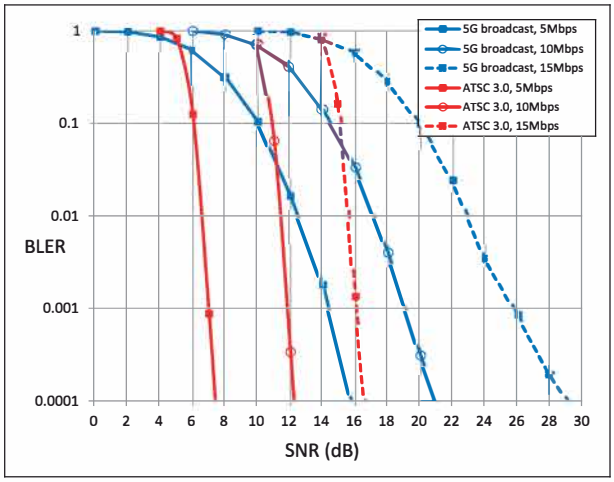


Fig. 2. BLER performance over India-Urban channel

REFERENCES

[1] Physical Layer Protocol, document A322, Adv. Television Syst. Committee, Washington, DC, USA, Sep. 2016.
[2] 3GPP TR 36.976 V1.0.0, "Overall description of LTE-based 5G broadcast (release 16)", December 2019.
[3] ETSI TS 103 720 V1.1.1, "5G Broadcast System for linear TV and radio services; LTE-based 5G terrestrial broadcast system", Dec. 2020.
[4] G. Caire, G. Taricci, E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927-946, May 1998
[5] S. Ahn *et al*., "Characterization and modeling of UHF wireless channel in terrestrial SFN environments: Urban fading profiles," to be published in *IEEE Trans. Broadcast*.

Dr. Sung-Ik Park (IEEE Fellow) joined the Broadcasting System Research Group of Electronics and Telecommunication Research Institute (ETRI) in 2002, and he is in charge of terrestrial broadcasting standardization, HW/SW implementations, and laboratory/field tests. Dr. Park is a principal researcher and leading several broadcasting research projects. Dr. Park has more than 300 peer-reviewed journal and conference publications and various best paper and contribution awards for his work on broadcasting technologies. Dr. Park currently serves as an associate editor of the IEEE Transactions on Broadcasting and ETRI Journal, and a distinguished lecturer at the IEEE Broadcasting Technology Society.

Sungjun Ahn is currently a Senior Research Engineer of Electronics and Telecommunications Research Institute (ETRI), where he has participated in Media Research Division since 2017. Mainly focused on media transmissions, his research covers the physical-layer design, system implementation, HW field experiements, and theoretic modeling/analysis for digital broadcasting and wireless system applications. He has authored more than 50 technical publications in peer-reviewed journals and conference proceedings. He is currently involved in research activities on the inter/intra-network cooperation of broadcasting and 5G.

Received in 2022-07-07 | Approved in 2022-08-16

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.7 Web Link: http://dx.doi.org/10.18580/setijbe.2022.7