BACALHAUNET: A TINY CNN FOR LIGHTNING-FAST MODULATION CLASSIFICATION

José Rosa[1], Daniel Granhão[2], Guilherme Carvalho[2], Tiago Gonçalves[2], Mónica Figueiredo[1,3], Luís Conde Bento[1,4], Nuno Paulino[2], Luís M. Pessoa[2]

[1]School of Technology and Management, Polytechnic of Leiria, Leiria, Portugal, [2]INESC TEC and Faculty of Engineering - University of Porto, Porto, Portugal, [3]Instituto de Telecomunicações, Portugal, [4]Institute of Systems and Robotics, Coimbra, Portugal

NOTE: Corresponding author: José Rosa, 2190383@my.ipleiria.pt

Abstract – *Deep learning methods have been shown to be competitive solutions for modulation classification tasks, but suffer from being computationally expensive, limiting their use on embedded devices. We propose a new deep neural network architecture which employs known structures, depth-wise separable convolution and residual connections, as well as a compression methodology, which combined lead to a tiny and fast algorithm for modulation classification. Our compressed model won the first place in ITU's AI/ML in 5G Challenge 2021, achieving 61.73× compression over the challenge baseline and being over 2.6× better than the second best submission. The source code of this work is publicly available at github.com/ITU-AI-ML-in-5G-Challenge/ITU-ML5G-PS-007-BacalhauNet.*

Keywords – 5G, deep neural networks, machine learning, modulation classification, model compression

1. INTRODUCTION

The growing demand for wireless data is driving a need for improved radio efficiency. Being able to rapidly understand the Radio Frequency (RF) spectrum in an automatic manner will be of utmost importance to address several open problems such as spectrum interference monitoring, radio fault detection, dynamic spectrum access and opportunistic mesh networking. A task that is required in all of these challenges is Automatic Modulation Classification (AMC), where the main goal is to monitor the radio frequency spectrum and determine the modulations in use [1, 2]. The first approaches to AMC consisted of handcrafted feature extractors for specific signal types and properties [1]. Later, the growing success of Deep Learning (DL) started to play a role in this field [3, 4, 5].

DL comprises a group of Machine Learning (ML) algorithms that uses multilayered Artificial Neural Network (ANN) architectures. These models can automatically extract the features needed to optimise a given task which allows these deep neural networks to be fed with raw data and to extract discriminating features with minimal domain knowledge and human effort [6, 7]. With the increase in the availability of computational power and the democratised access to huge quantities of data, these algorithms were shown to achieve high predictive performance in several domains of knowledge [8, 9]. However, despite their high effectiveness, these complex ANN architectures usually have high computational and power requirements (e.g., many models require several Graphics Processing Unit (GPU) devices to train and evaluate) [10]. To leverage the potential of DL for AMC in a real-world context, one must be able to implement these algorithms with low latency and high throughput thus enabling real-time spectrum analysis. Additionally, many use cases require at least the evaluation part to be performed on the edge, thus directly on site, using devices with limited computational capability and power resources.

Despite the efforts of research and industry communities towards the migration of ML models from the cloud to the edge, this is not an easy task due to the already mentioned complexity of these models. Two approaches have been typically followed. The first has been to implement simpler, more efficient ANN models [11, 12]. The second has resorted to power efficient heterogeneous computing platforms. One type of computing device that can be particularly efficient is the Field-Programmable Gate Array (FPGA), which allows very high power efficiency when using custom hardware acceleration engines [13]. This work focuses on both approaches, that is, designing simpler and more efficient DL models for AMC, while targeting FPGA hardware platforms. Specifically, this paper presents the methodologies and results achieved by our team "BacalhauNet" on Problem Statement 7 (PS-007), "Lightning-Fast Modulation Classification with Hardware-Efficient Neural Networks", of the ITU AI/ML in 5G Challenge 2021 [14].

In this challenge, the participants were encouraged to design an ANN that is computationally efficient while retaining a minimum required accuracy of 56% on the RadioML 2018.01A dataset from DeepSig [1, 15]. The overall dataset structure is illustrated in Fig. 1. This dataset comprises 24 types of digital and analog radio modulations that were synthetically generated and over-the-air captured. For each modulation type, there are samples recorded over 26 Signal to Noise Ratio (SNR) levels. For each SNR level, there are 4096 frames (captured signals). Each frame has 1024 samples of both in-phase (I) and quadrature (Q) components resulting in a frame with a shape of (1024, 2).

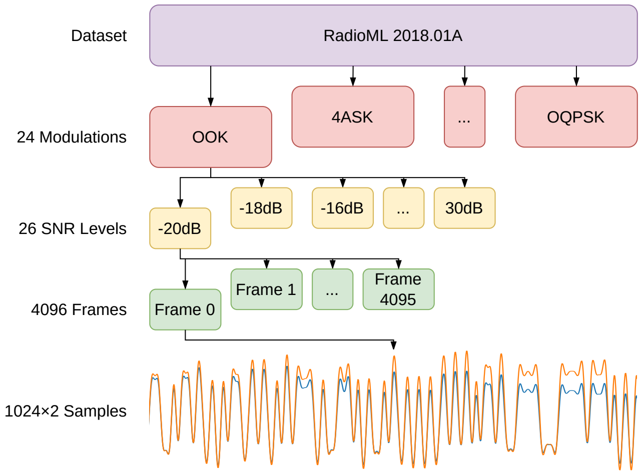

Fig. 1 – DeepSig Inc. RadioML 2018.01A dataset structure

A metric to evaluate the network's computational efficiency was defined by the challenge organizers and is given by an inference cost score (η) as expressed in Equation *(1)*. Two factors are equally considered: the number of binary operations (b_{ops}) normalized against a provided baseline implementation ($*b_{ops}$) and the number of bits used to represent the weights (b_{mem}), also normalized against the baseline implementation($*b_{mem}$). It is important to note that bias is not included in the these metrics but only the parameters for the convolutional and fully-connected layers themselves. The exact definition of the inference cost can be found in [16].

$$\eta = \frac{1}{2} \times \left(\frac{b_{ops}}{*b_{ops}} + \frac{b_{mem}}{*b_{mem}} \right) \qquad (1)$$

The provided baseline implementation is an 8 bit quantized version of a Convolutional Neural Network (CNN) based on the VGG10 topology described in [1]. The baseline achieves 59.82% accuracy while consuming ≈ 807.7 million binary operations and its parameters amount to ≈ 1.245 Mbits. The submissions were ranked according to the inference cost score, where lower is better, as long as they stay above the 56% accuracy threshold and are end-to-end reproducible.

The remainder of this paper is organized as follows: Section 2 presents methodologies used to build and compress the proposed CNN model; Section 3 describes BacalhauNet architecture as well as the employed training and compression methods; Section 4 presents the obtained results for both the uncompressed and compressed model; Section 5 concludes the article and provides possible lines of future work.

2. METHODS

High performance ANNs for AMC have been highly influenced by recent advances in CNN-based architectures, yet they still lack the computational efficiency required to implement them in edge devices. Novel approaches leading to solutions that can effectively be deployed in resource-constrained environments must be pursued. This section describes some well-known CNN structures and popular compression techniques that have proven to be effective when targeting computational efficient models.

Depth-wise Separable Convolutions (DSCs) were introduced in MobileNetV1 [17] as an efficient alternative to standard convolutions. The depth-wise separable convolutions factorizes a standard convolution into a depth-wise convolution followed by a pointwise convolution. The depth-wise convolution handles the filtering of the Input Feature Map (IFM). The pointwise convolution then produces the new features by combining all channels. In an $\hat{n}$-dimensional depth-wise convolution each $\hat{n}$-dimensional IFM channel C is convolved with an $\hat{n}$-dimensional kernel K. In the pointwise convolution each channel of the $\hat{n}$-dimensional feature map produced by the depth-wise convolution is convolved by D filters with shape $(1, 1)$. Fig. 2 illustrates a 1-dimensional depth-wise convolution. The IFM with shape ($\mathrm{IFM_h}$,1) and C channels is firstly convolved with a kernel shaped ($\mathrm{K_h}$,1) with the same amount of channels. Each C channel of the resulting Intermediate Feature Map is then convolved with D unitary kernels which results in an Output Feature Map (OFM) with a shape of ($\mathrm{OFM_h}$,1) and D channels.

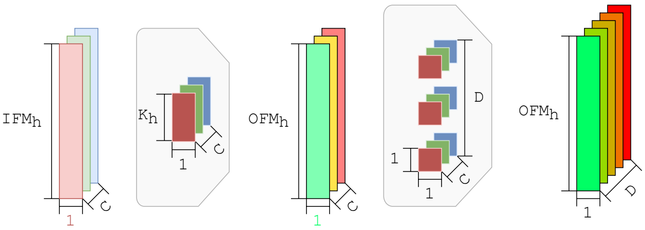

Fig. 2 – Depth-wise separable convolution

The replacement of standard convolutions by DSCs allows a reduction of operations and parameters as expressed by Equation *(2)*, while having little impact on accuracy [17]. According to Fig. 2, the replacement of a standard convolutions by the DSC enables an operation and parameter reduction of $1/\mathrm{D} + 1/\mathrm{K_h}$.

$$\frac{1}{D} + \frac{1}{\prod K_{\hat{n}}} \qquad (2)$$

Residual connections were introduced in the ResNet architecture [18] to improve the learning ability of a network. Intuitively, the accuracy of a Deep Neural Network

(DNN) can be enhanced by using more layers, i.e., making the network deeper, however there is a point where the accuracy gets saturated and can even degrade. This degradation of accuracy in deep networks can be a consequence of the vanishing or exploding gradient problem, which can be tamed by the use of residual connections. Fig. 3 shows a residual connection example where the input skips one or more layers and is added to the output of the skipped layers. We were then motivated to experiment with residual connections for two main reasons. First and foremost, it overcomes the problem of the vanishing/exploding gradient. Secondly, we believe that information carried downstream by residual functions allows the extraction of features with different temporal resolutions, potentially improving learning [18].

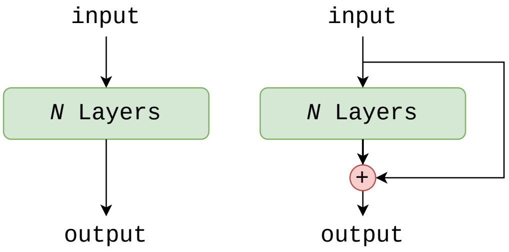

Fig. 3 – Residual connection

Quantization In order to provide a higher gradient definition in the back-propagation step, ANNs are usually trained with a higher precision than required at the inference stage. As a result, weight and activation quantization has become a common practice to reduce storage requirements in the forward pass [19]. Quantization can be either applied after or during training (e.g., Quantization Aware Training (QAT)), with the latter being capable of achieving consistently better results [20]. Quantization to every bit-width is possible, even to binary or ternary weights, however, the more aggressive the quantization is, the higher the chance of significant accuracy degradation [21].

Pruning is a compression method that removes the least important parameters of a neural network. Pruning methods can be classified as unstructured pruning or structured pruning [22]. Unstructured pruning removes the least important individual weights or biases from a model. However, the potential speed-up offered by skipping zeroes in unstructured sparsity is challenging to achieve due to the irregular memory accesses [22]. Structured pruning aims to remove structures from the network (e.g., layers or kernels) that have low impact in the model accuracy. Structured pruning is usually preferred for its higher potential speed-up. Unstructured and structured pruning methodologies can be used to compress a CNN model potentially enabling its deployment in resource-constrained devices.

3. BACALHAUNET

BacalhauNet is a tiny CNN built for the classification of ra- dio modulations, employing depth-wise separable convo- lutions and residual connections. It was designed through design space exploration, i.e., different architectures were explored in order to minimize the inference cost score while achieving the required accuracy threshold. The mo- tivation to develop BacalhauNet arises from the far higher inference costs score displayed by other efficient CNNs. One of such examples is MobileNetV3-Small [23], a more compact version of MobileNetV3, which, even when quan- tized to 8 bits, only achieves $\approx$ 7.25 inference cost score, significantly higher than our proposed architecture.

3.1 Architecture exploration

As we previously mentioned, BacalhauNet was developed using a design space exploration methodology. We started our experiments using a model with a single DSC layer for feature extraction and progressively stacked more DSC layers until we reached an accuracy of at least 59%. This minimum was set to allow for some accuracy degradation in later steps, namely quantization and pruning. For each stacked DSC layer several training runs using different parameters were made (i.e., kernel size, number of output channels, stride length). The chosen parameters for each layer were the ones that led to a good balance between accuracy and the inference cost score. We ended up with a model composed of four DSC layers that are preceded by a *hardtanh* layer and followed by a global max pool and finally a Fully Connected (FC) layer. We also tuned the *hardtanh* layer minimum and maximum clipping values to achieve higher test accuracy. Later, the quantization and pruning steps had a larger than anticipated negative effect on accuracy. Thus we ended up adding an additional DSC layer to which we did not apply the same optimization procedure.

3.2 Building blocks

The proposed model consists of a *hardtanh* activation function followed by five depth-wise separable convolutional blocks ending with a global *max pool* and a fully connected layer. The overall architecture and its parameters are described in Table 1.

Inspired by the baseline model the first layer of our CNN architecture implements a *hardtanh* activation function, a computational efficient alternative to the *tanh* that is characterized by Equation *(3)*. The *hardtanh* layer assumes the role of clipping the input minimum and maximum values to -2 and 3, decreasing inter-class variation of input amplitudes.

Table 1 – BacalhauNet architecture

Layer Type	Kernel	Residual	Input Size
HardTanh	-	-	1024×2
DSC 1D	27	-	1024×2
DSC 1D	21	⍰	512×24
DSC 1D	15	-	512×24
DSC 1D	9	⍰	256×48
DSC 1D	9	-	256×48
Global MaxPool 1D	-	-	128×48
Fully Connected	-	-	1×48

Particularly high input value amplitudes are exhibited by the *AM-SSB-SC* and *AM-SSB-WC* modulations, as shown in Fig. 4.

$$hardtanh = \begin{cases} min & x < min \\ x & min \leq x \leq max \\ max & x > max \end{cases} \qquad (3)$$

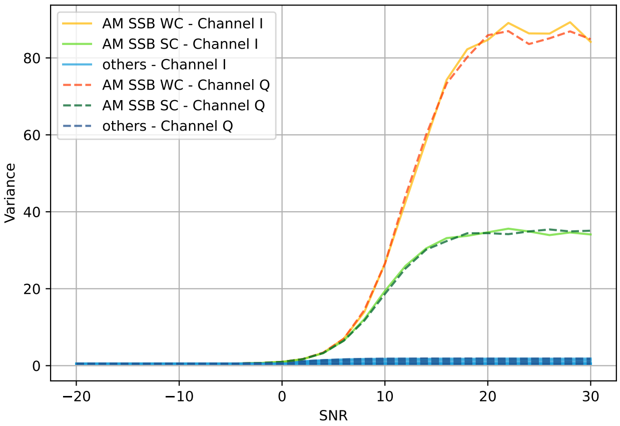

Fig. 4 – Variance for all dataset modulations and SNR pairs

The feature extraction is performed in depth-wise separable convolutional blocks. Fig. 5 represents a functional diagram of this structure. It implements a DSC, i.e., a depth-wise convolutional block followed by a pointwise convolutional block, which uses residual connections whenever stride is unitary and the number of output channels is the same as the number of input channels. Both depth-wise and pointwise convolutional blocks implement a depth-wise and a pointwise convolution, respectively, followed by a batch normalization and a Rectified Linear Unit (ReLU) activation function.

3.3 Training

Regarding the training procedure, only a subset of the total dataset was used. Specifically, only frames with an SNR equal or greater than -6dB were used. This allowed us to train the network faster with a negligible impact on the test accuracy since the low SNR frames contain little effective information for the network, i.e., the samples are mostly comprised of noise. For each modulation class and SNR level pair the train subset contains 90% of the frames while the remainder 10% of the frames belong to the test subset.

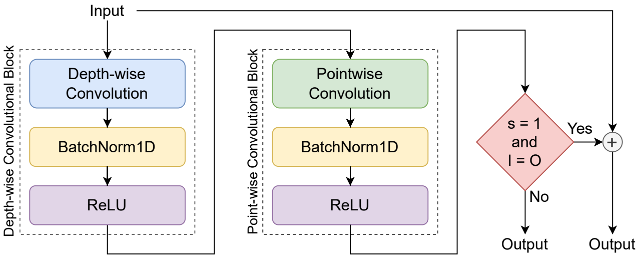

Fig. 5 – 1D depth-wise separable block. The residual connections are only used when stride (s) is unitary and the number of input channels (I) is equal to the number of output channels (O).

The model was trained using the Adaptive Moment Estimation (Adam) optimization algorithm [24] with a cosine annealing warm restarts [25] learning rate scheduler. The initial learning rate was set to 0.01, being reset each five epochs. The cross-entropy loss function was used to compute the loss between the model prediction and the target output. Each training procedure was terminated after a total of 20 epochs following the provided baseline implementation.

3.4 Compression

Quantization Despite BacalhauNet already being relatively small, Quantization Aware Training (QAT) was explored to further reduce its complexity using a PyTorch QAT library developed by Xilinx, called Brevitas [26]. The Brevitas library provides alternative PyTorch layer classes and its use can be as simple as defining the bit-width that is intended for each layer's parameters. An example of the definition of a Brevitas quantized convolutional layer and activation function can be found on Listing 1.

Pruning An unstructured pruning methodology was used to further compress BacalhauNet. Although unstructured pruning is prone to deployment issues the competition evaluation metrics didn't penalize unstructured sparsity which enable us to achieve a better inference cost score for the same accuracy. For that reason an unstructured pruning approach was implemented in the proposed network. In order to prune the model more effectively a retrain is done before each prune iteration with a weight decay (λ) which forces the weights to converge to zero. Following each retrain iteration, the weights below an minimum absolute threshold (ε) were set to zero which effectively prunes the model.

Listing 1 – Brevitas quantization example - Convolutional layer "conv" has its weights quantized to "w_bits" and activation layer "relu" has its outputs quantized to "a_bits".

```
from brevitas import nn as qnn

conv = qnn.QuantConv1d(
    in_channels=2,
    out_channels=2,
    kernel_size=1,
    weight_bit_width=w_bits
)

relu = qnn.QuantReLU(bit_width=a_bits)
```

4. RESULTS AND DISCUSSION

In this section, the architecture design and compression results are shown and discussed. All evaluations were made using the RadioML 2018.01A [1, 15] test subset. We start by presenting how the architectural design choices impact the accuracy and the train time. Afterwards, compression results are shown using the model with the lowest inference cost score that surpasses a 59% accuracy threshold.

4.1 Architecture exploration results

Table 2 contains the average best accuracy reached by 10 individual training runs, for the model trained with and without the *hardtanh* activation function as the first layer. The results display the average of 10 runs to account for any variance introduced by the random initialization of weights. Results show that this layer improves the test accuracy by $\approx$ 0.92%.

Table 2 – Average of maximum test accuracy achieved in 10 training runs with and without *hardtanh* activation function as first layer of the model.

Model	Accuracy
with *hardtanh*	58.99%
without *hardtanh*	58.07%

The exclusive usage of frames with higher SNR levels to train the model resulted in a considerable reduction of the training time with negligible impact on accuracy. Table 3 contains the average best accuracy and average training times per epoch using an NVIDIA GeForce RTX 3090 GPU. Once again, 10 training runs were performed to account for any variance introduced by the random initialization of weights. The results show a reduction of the training time of $\approx$ 1 min 28 secs per epoch. This is consistent with other results in literature that, in spite of their larger networks, fail to correctly classify most modulations below a certain SNR [27, 28].

Table 3 – Average of maximum test accuracy achieved in 10 training runs and training epoch time when using all training samples or only higher SNR ones.

Model	Accuracy	Time/Epoch
with SNR [-6, 30] dB	58.99%	2 min 06 sec
with SNR [-20, 30] dB	59.04%	3 min 34 sec

The best BacalhauNet model was chosen for further compression. The best model is the one that surpasses 59% accuracy with the lowest inference cost score. The 59% accuracy threshold is set in order to allow some accuracy degradation caused by compression techniques. The best model without any kind of optimization reached a test accuracy of $\approx$ 59.09% while having an inference cost score of $\approx$ 1.4155. To fairly compare it against the baseline model we quantized BacalhauNet's input, weights and activations to 8 bits, which improved the inference cost score to $\approx$ 0.1461, i.e., $\approx$ 188.718 million binary operations and 73088 parameters, with a negligible impact on model accuracy. More details such as the Compression Rate (CR), which reports the compression that the model achieved w.r.t. the baseline model, are shown in Table 4. These results demonstrate that our model is highly efficient for the modulation classification task.

Table 4 – Non-optimized BacalhauNet metrics. Column η reports the inference cost w.r.t. the baseline model and column CR reports the compression rate.

Data Type	Accuracy	η	CR
Float 32 bit	59.09%	1.4155	0.7065
Quantized 8 bit	59.06%	0.1461	6.8446

4.2 Compression techniques results

Quantization Table 5 displays the maximum obtained accuracy and the corresponding inference cost score for each fixed-point representation. The presented results of quantization are for a fixed bit-width on the weights and activations from 8 bit down to 5 bit, while quantizing input values to 8 bits using the initial *hardtanh* layer. As expected, quantization reveals a drastic reduction in the inference cost score with decreasing bit-width. However, as the representation goes below 6 bit, accuracy starts to be heavily affected and stops being above the required 56% threshold. In the end, 6 bit quantization was selected due to its results being a good compromise between accuracy and inference cost score. It is likely that if we had done more quantized training runs, using 6 bits would not provide such an advantage when compared to 7 bits. We nevertheless moved on to the next step and just enjoyed the luck we got in the initial weight initialization.

Pruning Unstructured pruning was performed after quantization in order to reduce the computational complexity of the model. In total three pruning iterations were performed using multiple ε and the λ values. In each pruning iteration one model was selected as the base

Table 5 – Quantization results of BacalhauNet. Column η reports the inference cost w.r.t. the baseline model and column CR reports the compression rate.

Data Type	Accuracy	η	CR
Quant - 8 bits	59.06%	0.1461	6.8446
Quant - 7 bits	58.35%	0.1002	9.9800
Quant - 6 bits	**58.67%**	**0.0781**	**12.8041**
Quant - 5 bits	55.89%	0.0562	17.7936

model for the next pruning iteration. The selection criteria is as follows: (1) the model should always surpass a 57% accuracy threshold after intermediate pruning steps to allow for further pruning; (2) the model accuracy and inference cost score should be balanced, where the accuracy and inference cost score are favored in initial and last pruning iterations, respectively. After the last prune iteration a retrain phase is also performed so that the accuracy degradation is not so expressive. Fig. 6 presents the results from the design space exploration for each pruning iteration where each color represents an iteration and the triangles represent the selected model.

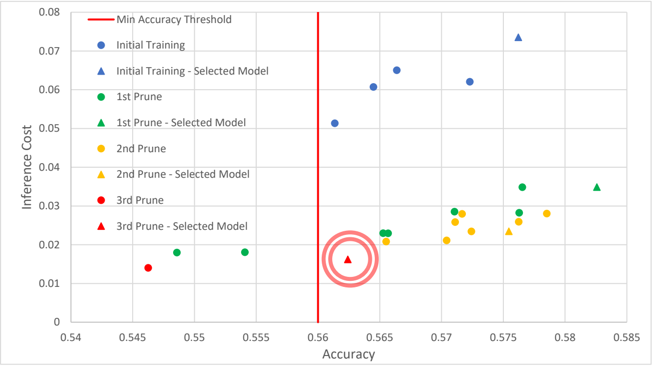

Fig. 6 – Results for the pruning design space exploration. Models that result from the initial training and from each pruning iteration are represented with a different color.

Table 6 contains details about the parameters used by the selected model in each prune iteration. The final compressed model submitted to PS-007, reached a test accuracy of $\approx$ 56.24% while consuming $\approx$ 19.243 million binary operations and using a total of 10704 parameters, which translates to $\approx$ 61.73$\times$ lower computational complexity when compared to the baseline model.

Table 6 – Pruning results of BacalhauNet. Column i is the pruning iteration, column λ reports the weight decay, column ε refers to the minimum absolute weight value, column η is the inference cost score and column CR reports the compression rate.

i	λ	ε	Accuracy	η	CR
0	1×10^{-4}	-	57.62%	0.0735	13.6054
1	5×10^{-5}	0.15	58.26%	0.0348	28.7356
2	1×10^{-4}	0.25	57.55%	0.0235	42.5532
3	$\mathbf{1 \times 10^{-5}}$	**0.25**	**56.24%**	**0.0162**	**61.7284**

5. CONCLUSIONS AND FUTURE WORK

In this paper, BacalhauNet, a computational efficient CNN for radio modulation classification was introduced. Building blocks and design decisions of the proposed architecture were described and results presented. A comparison between the proposed model and a baseline model, both quantized to 8 bits, reveals that BacalhauNet manages to reduce computational complexity by $\approx$ 6.84$\times$ while retaining an accuracy of $\approx$ 59.06%. Quantization and pruning methodologies were used to further compress the model. The model was quantized down to 6 bits which enabled us to reduce the computational complexity of the model by $\approx$ 12.80$\times$ w.r.t. the baseline while achieving $\approx$ 58.67% accuracy. The pruned model achieved an accuracy of $\approx$ 56.24% and an inference cost score of 0.0162, i.e., reduced the computational complexity by $\approx$ 61.73$\times$ w.r.t. the baseline and is over 2.6$\times$ better than the second best submission. The quantized and pruned model was the winning submission of the PS-007, “Lightning-Fast Modulation Classification with Hardware-Efficient Neural Networks”, of the ITU AI/ML in 5G Challenge 2021.

Further work should be devoted to: the optimization of the last depth-wise separable convolutional layer; the testing of different levels of quantization per layer, since it can increase even more the compression achieved; the exploration of different feature engineering approaches, to assess if the model's inference cost score can be further reduced while maintaining an acceptable accuracy.

REFERENCES

[1] Timothy James O'Shea, Tamoghna Roy, and T. Charles Clancy. “Over-the-Air Deep Learning Based Radio Signal Classification”. In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (Feb. 2018), pp. 168–179. ISSN: 1941-0484.

[2] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. “Convolutional radio modulation recognition networks”. In: *International conference on engineering applications of neural networks*. Springer. 2016, pp. 213–226.

[3] Charles Clancy, Joe Hecker, Erich Stuntebeck, and Tim O'Shea. “Applications of machine learning to cognitive radio networks”. In: *IEEE Wireless Communications* 14.4 (2007), pp. 47–52.

[4] Kyouwoong Kim, Ihsan A Akbar, Kyung K Bae, Jung-Sun Um, Chad M Spooner, and Jeffrey H Reed. “Cyclostationary approaches to signal detection and classification in cognitive radio”. In: *2007 2nd ieee international symposium on new frontiers in dynamic spectrum access networks*. IEEE. 2007, pp. 212–215.

[5] Thomas Warren Rondeau. “Application of artificial intelligence to wireless communications”. PhD thesis. Virginia Tech, 2007.

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. en. In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 0028-0836, 1476-4687.

[7] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. en. In: *Neural Networks* 61 (Jan. 2015), pp. 85–117. ISSN: 08936080.

[8] Sina Mohseni, Niloofar Zarei, and Eric D. Ragan. “A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems”. en. In: *arXiv:1811.11839 [cs]* (Dec. 2019). arXiv: 1811.11839.

[9] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C. Corrado, Ara Darzi, Mozziyar Etemadi, Florencia Garcia-Vicente, Fiona J. Gilbert, Mark Halling-Brown, Demis Hassabis, Sunny Jansen, Alan Karthikesalingam, Christopher J. Kelly, Dominic King, Joseph R. Ledsam, David Melnick, Hormuz Mostofi, Lily Peng, Joshua Jay Reicher, Bernardino Romera-Paredes, Richard Sidebottom, Mustafa Suleyman, Daniel Tse, Kenneth C. Young, Jeffrey De Fauw, and Shravya Shetty. “International evaluation of an AI system for breast cancer screening”. In: *Nature* 577.7788 (Jan. 2020), pp. 89–94.

[10] Antonio Polino, Razvan Pascanu, and Dan Alistarh. “Model compression via distillation and quantization”. In: *arXiv preprint arXiv:1802.05668* (2018).

[11] Angel Martinez-Gonzalez, Michael Villamizar, Olivier Canevet, and Jean-Marc Odobez. “Efficient convolutional neural networks for depth-based multi-person pose estimation”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.11 (2019), pp. 4207–4221.

[12] Mario P Vestias. “A survey of convolutional neural networks on edge with reconfigurable computing”. In: *Algorithms* 12.8 (2019), p. 154.

[13] Murad Qasaimeh, Kristof Denolf, Jack Lo, Kees Vissers, Joseph Zambreno, and Phillip H. Jones. “Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels”. In: *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*. 2019, pp. 1–8. DOI: 10.1109/ICESS.2019.8782524.

[14] ITU. *PS-007 of ITU AI/ML in 5G Challenge*. URL: https://challenge.aiforgood.itu.int/match/matchitem/34.

[15] DeepSig Inc. *RF Datasets For Machine Learning*. URL: https://www.deepsig.ai/datasets.

[16] Nhan Tran, Benjamin Hawks, Javier M Duarte, Nicholas J Fraser, Alessandro Pappalardo, and Yaman Umuroglu. “Ps and Qs: Quantization-aware pruning for efficient low latency neural network inference”. In: *Frontiers in Artificial Intelligence* 4 (2021), p. 94.

[17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: *CoRR* abs/1704.04861 (2017).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[19] Song Han, Huizi Mao, and William J. Dally. “Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding”. In: *arXiv: Computer Vision and Pattern Recognition* (2016).

[20] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 2704–2713.

[21] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. “A survey of quantization methods for efficient neural network inference”. In: *arXiv preprint arXiv:2103.13630* (2021).

[22] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David S. Doermann. “Towards Optimal Structured CNN Pruning via Generative Adversarial Learning”. In: *CoRR* abs/1903.09291 (2019).

[23] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. “Searching for MobileNetV3”. In: *CoRR* abs/1905.02244 (2019).

[24] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2015).

[25] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic Gradient Descent with Warm Restarts”. In: *arXiv: Learning* (2017).

[26] Alessandro Pappalardo. *Xilinx/brevitas*. 2021. URL: https://doi.org/10.5281/zenodo.3333552.

[27] Siyang Zhou, Zhendong Yin, Zhilu Wu, Yunfei Chen, Nan Zhao, and Zhutian Yang. "A robust modulation classification method using convolutional neural networks". In: *EURASIP Journal on Advances in Signal Processing* 2019.1 (2019), pp. 1–15.

[28] Ya Tu, Yun Lin, Changbo Hou, and Shiwen Mao. "Complex-valued networks for automatic modulation classification". In: *IEEE Transactions on Vehicular Technology* 69.9 (2020), pp. 10085–10089.

AUTHORS

José Rosa received his bachelor's degree in electrical and computer engineering - electronics and computers from the School of Technology and Management of Polytechnic of Leiria in 2019. Currently he is an MSc candidate in electrical and electronic engineering - electronics and telecommunications and assistant professor in the same institution. He is also a researcher at the Telecommunications Institute.

Daniel Granhão obtained his master's degree in electrical and computer engineering from the Faculdade de Engenharia da Universidade do Porto (FEUP) in 2019. Currently he is a PhD candidate and assistant lecturer researching efficient reconfigurable hardware architectures for recurrent neural networks. He is currently the beneficiary of FCT's PhD grant SFRH/BD/145481/2019.

Guilherme Carvalho received his MSc in electrical and computer engineering from the Faculdade de Engenharia da Universidade do Porto (FEUP) in 2017. Currently he is a PhD candidate within Programa Doutoral em Engenharia Electrotécnica e de Computadores (PDEEC) and is also an assistant lecturer at FEUP. His research interests include neuro-morphic and biologically plausible computing, in-memory computing and resistive switching devices.

Tiago Gonçalves received his MSc in bioengineering-biomedical engineering from the Faculdade de Engenharia da Universidade do Porto (FEUP) in 2019. Currently, he is a PhD candidate in electrical and computer engineering at FEUP and a research assistant at the Centre for Telecommunications and Multimedia (CTM) of INESC TEC with the Visual Computing & Machine Intelligence (VCMI) research group. His research interests include machine learning, explainable artificial intelligence (in-model approaches), computer vision, medical decision support systems, and machine learning deployment.

Mónica Figueiredo is an assistant professor at the Polytechnic of Leiria and a researcher at the Telecommunications Institute. She got her degree in electronic engineering from the University of Coimbra (1999), a master's in electronic engineering and telecommunications from the University of Aveiro (2003) and a PhD in electrical engineering from the University of Aveiro (2012). She has published more than 30 articles in peer reviewed journals and conferences, in the areas of high speed digital electronic systems, reconfigurable systems for telecommunications, synchronization systems and visible light communications. She has served as a reviewer for several international journals (IEEE/TVLSI, IEEE/T-CASII, IEEE/PTL, IEEE/CommMag, MDPI, among others). She is also CEO and co-founder of TWEVO Lda., a startup providing reliable wireless solutions for Industry 4.0.

Luís Conde Bento received a PhD degree in electrical engineering from the University of Coimbra, Portugal, in 2016. Since 2002, he has been a researcher with the Institute for Systems and Robotics, University of Coimbra. He is currently a professor with the Electrical Engineering Department, School of Technology and Management, Polytechnic Institute of Leiria, Portugal. He has also been enrolled in several funded research and development projects acting either as a co-manager or a supervisor, as well as a hardware and middleware developer for embedded systems. His research interests include automatic control, GNSS positioning systems, autonomous vehicles, and traffic management.

Nuno Paulino received an MSc degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, in 2011, and a PhD degree in electrical and computer engineering from the University of Porto, in 2015. He is currently an assistant professor at the University of Porto, and a researcher with INESC TEC. His research interests include reconfigurable and embedded systems in FPGAs, computing architectures, and tools for hardware/software co-design.

Luís M. Pessoa is a senior researcher at the Centre of Telecommunications and Multimedia (CTM) of INESC TEC, where he is the lead for the area of optical and electronic technologies. He received a "Licenciatura" degree in 2006 and PhD degree in 2011, in electrical and computer engineering from the Faculty of Engineering of the University of Porto. His research interests include coherent optical systems, radio-over-fibre, RF/microwave devices and antennas, and underwater wireless power/communications.