

Deep Learning Modulation Recognition for RF Spectrum Monitoring

A. Emad¹, H. Mohamed¹, A. Farid¹, M. Hassan¹, R. Sayed¹, H. Aboushady², H. Mostafa^{1,3}

¹Cairo University, Electronics and Communications Department, Giza, Egypt.

²Sorbonne University, LIP6 Laboratory, CNRS UMR 7606, Paris, France.

³Zewail University of Science and technology, Nanotechnology Department, Giza, Egypt.

Abstract—This paper presents a classification Convolutional Neural Network model for modulation recognition. The model is capable of classifying 11 different modulation techniques based on their In-phase and Quadrature components at baseband. The classification accuracy is higher than 80% for signals with a Signal-to-Noise Ratio higher than 2 dB. The model performance is evaluated using the same In-phase and Quadrature component data-sets used in the state of the art. Compared to previous work, the number of parameters and multiplications/additions is reduced by several orders of magnitude. The proposed Convolutional Neural Network is implemented on FPGA and achieves the same performance as the GPU model. Compared to other FPGA implementations of RF signal classifiers, the proposed implementation classifies twice as much modulation schemes while consuming only half the dynamic power.

Index Terms—Deep Learning, Convolutional Neural Networks, Modulation Recognition, Cognitive Radio, Spectrum Monitoring, Dynamic Spectrum Access, FPGA.

I. INTRODUCTION

Radio spectrum is witnessing a high increase in traffic especially with the deployment of the 5G standard in the sub-6 GHz range [1]. Furthermore, the number of wireless Internet-of-Things (IoT) devices will reach tens of billions in the next few years. All these devices are also expected to establish wireless communication in the sub-6 GHz frequency range [1]. Spectrum utilization is not identical on all frequency bands. Some bands are over-utilized especially the unlicensed bands whereas other bands are under-utilized. This leads to a congested wireless communication and a degradation of the Quality of Service (QoS). Cognitive radio is a technique that aims at optimizing the spectrum utilization in order to improve the Quality of Service. Before establishing a communication, the spectrum is analyzed to help the Cognitive Radio select the optimum frequency band to used [2]. Modulation recognition is an important step in cognitive radio and spectrum sensing as information about signals and channels is unknown. Moreover, in spectrum monitoring and security applications it is very important to identify nature of the signals which helps in identifying suspicious activities such as intrusion and jamming.

Being inspired by the remarkable success of deep learning in many applications such as image recognition [3] [4] and speech recognition [5], deep learning techniques have been proposed for signal identification in wireless communication

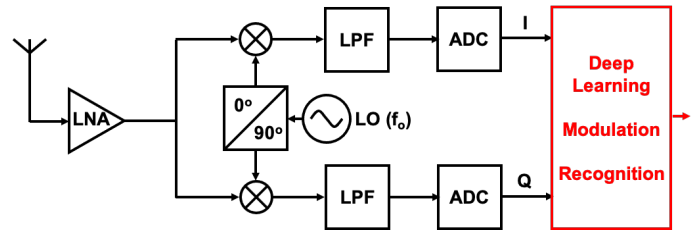


Fig. 1: The Deep-Learning modulation recognition model is based on the baseband In-phase (I) and Quadrature (Q) components of an RF receiver.

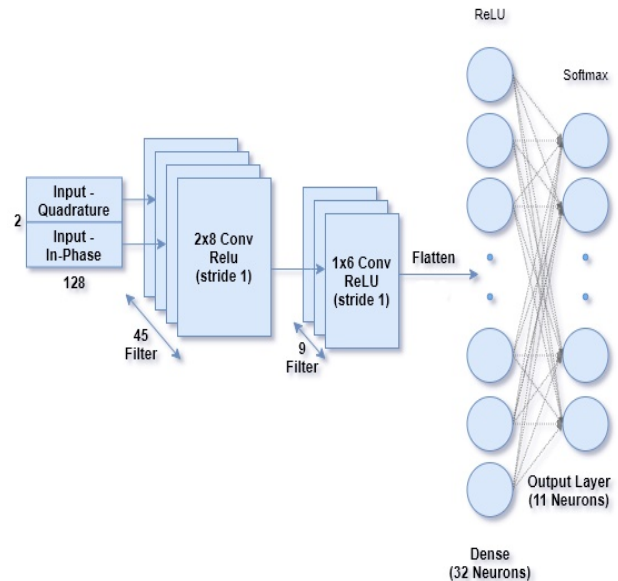


Fig. 2: The proposed architecture for the Modulation Recognition Deep-Learning I/Q Model.

networks. In [6], it is shown that a Convolutional Neural Network (CNN) trained with In-phase and Quadrature data outperform in accuracy the traditional approaches for automatic modulation recognition based on expert features such as cyclic-moment based features, and conventional classifiers such as decision trees, support vector machines (SVM), K-nearest neighbours (k-NNs), Artificial Neural Networks (ANNs), and Naïve Bayes [6] [7].

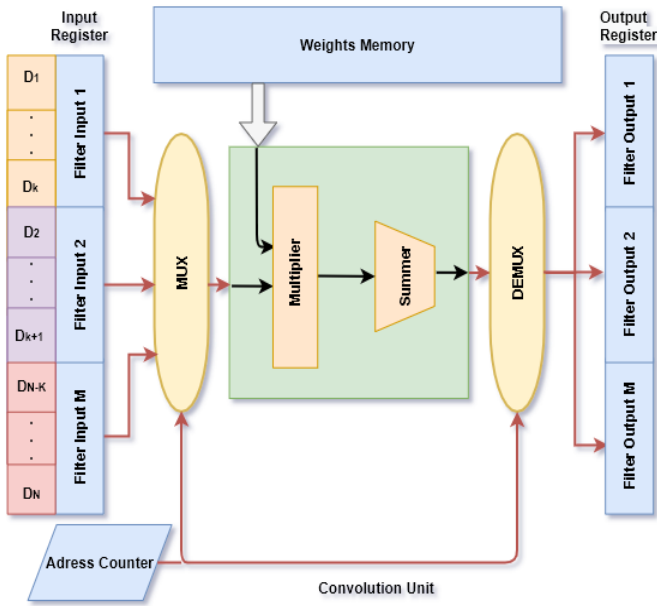


Fig. 3: Convolution operation implementation.

In this paper, we present a deep-learning modulation recognition model based on the In-phase (I) and the Quadrature (Q) components, as illustrated in Fig.1. The modulation recognition is based on the classification CNN shown in Fig.2. The paper also presents the implementation of the CNN model on FPGA. The accuracy, size, speed, and power consumption of the proposed model implementation is compared the state of the art.

II. THE CNN MODEL DESIGN AND IMPLEMENTATION

A. Dataset

RADIOML 2016.10A dataset [8] has been used to train and test the proposed model, this data is generated at sampling rate of 1 MSample/sec, it consists of eleven modulation schemes equally distributed over a range of Signal-To-Noise Ratios (SNR) from -20 dB to +18 dB where eight of them are digital modulation schemes: **BPSK** (Binary Phase Shift Keying), **QPSK** (Quadrature Phase Shift Keying), **8-PSK** (8-Phase Shift Keying), **16-QAM** (16-Quadrature Amplitude Modulation), **64-QAM** (64-Quadrature Amplitude Modulation), **CPFSK** (Continuous Phase Frequency Shift Keying), **GFSK** (Gaussian Frequency Shift Keying), **4-PAM** (4-Pulse Amplitude Modulation) and the other three are analog modulation schemes: **WBFM** (Wide Band Frequency Modulation), **AM-DSB** (Amplitude Modulation Double Side Band), **AM-SSB** (Amplitude Modulation Single Side Band). In this work, 70% of the data have been used for training and 30% have been used for test.

B. Model Architecture

Since the model is developed to be implemented on FPGA, the size and the number of calculations of the model are important design factors. At the same time, it is very important to maintain accuracy as high as possible. As shown in Fig.2,

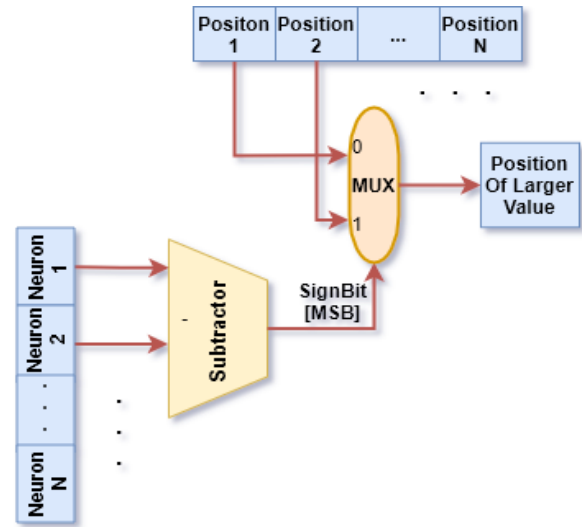


Fig. 4: Softmax algorithm implementation.

this has been achieved by decreasing the number of filters while increasing the size of each filter in the CNN model. The proposed model consists of 2 convolution layers and 2-dense layers. The first convolution layer has 45 filters each having 2x8 parameters while the second convolution layer has 9 filters each having 1x6 parameters. The second convolution layer is followed by two dense layers, the first layer consists of 32 neurons and second layer consists of 11 neurons classifying 11 modulation schemes.

C. Training Model

The Model has been trained on Kaggle [9] using the Keras framework [10] with an Adam optimizer and a learning rate of 0.001. Each batch size was 512 and 200 epochs. Biases have been turned off and average validation accuracy of the overall SNRs has reached 54.38%.

D. Floating-Point versus Fixed-Point Representation

Extracted weights from Keras are represented in floating-point which significantly increase the number of calculations and processing time in a hardware implementation. A floating point representation would occupy a significant amount of resources in an FPGA implementation. It will also lead to higher power consumption. It is then better to avoid a floating-point representation and to use a fixed-point representation. A model using integer weights and a fixed-point representation has been implemented. Simulation results have shown that a 16-bit representation for the weights minimizes quantization error and maintains accuracy unchanged compared to the floating-point representation.

III. HARDWARE IMPLEMENTATION

A. Convolution

The implementation of the convolution operation is shown in Fig.3. The input data to the filter is being multiplexed at each clock cycle with an address counter which functions

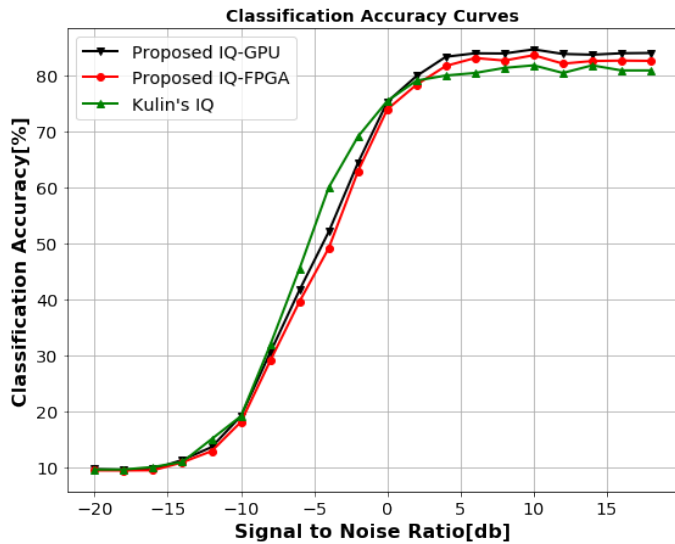


Fig. 5: The classification accuracy of Kulin's IQ model [13], the proposed IQ GPU-model and the measurement results from the proposed IQ model implemented on FPGA.

as a selection line. Each clock cycle, the input data is a shifted by one stride. The selected data is then multiplied with the corresponding weights from memory. Afterwards, all multiplier outputs are summed and demultiplexed to the corresponding output line.

B. Neuron

Operations in neuron can be described as a weighted sum. Weights are read from memory then multiplied with their corresponding inputs then all outputs are being summed. The number of multiplications in a neuron of the first fully connected layer is 1044 which is very large number and processing all these multiplications at one time will consume almost all FPGA resources. It has been found that neuron operations should be reduced by 12 (87 Multiplications/Clock) to get a reasonable utilization and at the same time keep processing time as small as possible.

C. Activation Functions

a) *Rectified Linear Unit*: As shown in Fig.2, the activation function of the dense layer is a Relu. It is used due to its interesting performance in terms of accuracy and simplicity [11]. The ReLU function can be easily implemented without any approximations or complex calculations. The idea is to filter out negative numbers to zero and to allow only positive numbers. The ReLU is implemented using a multiplexer with the sign-bit (Most Significant Bit) as its selection line. If the sign-bit is zero (positive number) then the output is identical to the input otherwise the output is zero.

b) *Softmax*: As shown in Fig.2, The activation function of output layer is Softmax. It consists of 11 neurons each one of them corresponds to a class of Modulation scheme. The

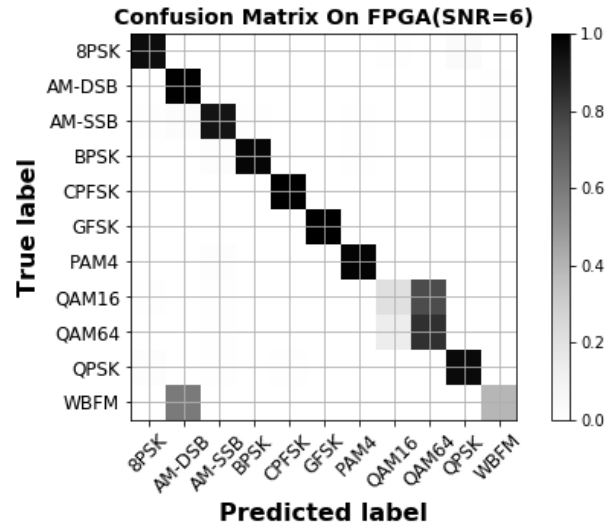


Fig. 6: The confusion matrix measurement results of the proposed IQ model implemented FPGA. The results are obtained using RF signals with a 6 dB SNR.

Softmax turns outputs of neurons to a probability distribution for each class as follow:

$$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (1)$$

for $i = 1, \dots, K$ and $x = (x_1, \dots, x_K)$. The predicted class is the one with the highest probability [12]. Implementing softmax requires complex calculations to calculate exponentials and dividers. In this work, softmax is implemented by simply finding the largest neuron and its position will correspond to the predicted class. The shape of output is transformed from one-hot encoding to ordinal encoding, as shown in Fig.4. The algorithm works by comparing each neuron with the adjacent neuron using a subtractor as a comparator. The sign bit of the subtractor output determines which operand is larger so if the first operand is larger then its position will be selected through the multiplexer. Otherwise, the position of the other operand will be selected. This operation is performed on all the neurons until figuring out the position of largest neuron.

IV. MEASUREMENT RESULTS

The proposed modulation classification IQ model has been implemented with a floating-point representation on a GPU and with a 16-bit fixed-point representation on an FPGA with the architecture described in section III.

It is worth noting that all the results presented in this section are obtained using the RadioML.2016.10a datasets [8].

A. Accuracy Curves

The classification accuracy curves have been measured for the GPU and the FPGA implementations. In Fig. 5, the classification accuracy for SNR values varying from -20 dB to +18 dB for both implementations. These results are also compared to classification accuracy reported for Kulin's IQ

model [13]. It can be seen that the classification accuracy of that the 3 models achieve very similar results and that there is very little degradation in the performance of the 16-bit fixed-point FPGA implementation compared to the floating-point GPU model.

B. Confusion Matrix

The confusion matrix is another metric that is used to understand and visualize the classification accuracy and to determine which modulation schemes might be wrongly classified. as shown in Fig.6 it is obvious that some miss-classification happens between WBFM (True) and AM-DSB (Predicted) also between QAM16 and QAM64. The same miss-classifications have also been reported in Kulin’s model [13]. However the miss-classification between QPSK and 8-PSK has been significantly improved in the proposed model compared to Kulin’s model [13].

C. Comparison with other Hardware Implementations

In this subsection, the FPGA implementation of the proposed IQ model is compared with other another similar FPGA implementation. In Soltani et al.’s work presented in [14] [15], the authors present an FPGA implementation of a Neural Network (NN) consists of 4 dense layers used to classify 6 modulation schemes. The FPGA implementations are compared in terms of the number of classified modulation schemes, the power consumption, the utilization and the processing time.

1) *Number Of Calculations and Model Size:* The number of calculations and the number of parameters are the most important advantages of the proposed IQ model. The number of operations has a direct impact on the number of multipliers and adders on the FPGA and the number of parameters affects the number flip-flops on the FPGA. Table I shows the number of parameters and calculations of the proposed IQ model compared Kulin’s [13] and Soltani’s [14] model. A significant reduction in the number of parameters and calculations has been achieved by increasing the filter size of the convolution layer and by decreasing the number of filters which has been very effective in preserving the high accuracy of the model. The dimensions of the second convolution layer have been decreased to reduce the number of parameters of the fully connected layer. The proposed architecture is then very suitable for an FPGA or an ASIC implementation.

2) *Power Consumption and Utilization:* The FPGA power consumption and utilization are shown in Table II. The proposed model relies mostly on DSP blocks as the convolution operations make heavy parallel use of multipliers. On the other hand, Soltani’s model presented in [14] is a Neural Network (NN) model. It utilizes LUTRAM as NN and has more weights needed to be stored. The dynamic power consumption of the proposed IQ model is almost half of the dynamic power consumption in Soltani’s model.

3) *Processing Time:* Table III summarizes the processing time of the proposed IQ model compared to Soltani’s work [14]. Comparing processing time for the proposed model with

TABLE I: Number of Parameters and Mult-Add

Model	Parameters	Million Mult-Adds
Proposed IQ Model	36,910	0.443
Soltani’s Model [15]	1,804,067	3.6
Kulin’s Model [13]	2,667,615	20.59

TABLE II: Comparison with the state of the art.

Reference	This Work	Soltani’2019 [14]
Architecture	CNN	NN
Modulation Schemes	11	6
Clock Frequency	70 MHz	-
LUT	74680	158435
LUTRAM	14832	117380
FF	57726	16222
DSP	1116	210
Power	847 mW	1152 mW
Static Power	593 mW (70%)	651 mW (57%)
Dynamic Power	254 mW (30%)	501 mW (43%)
FPGA	Zynq Ultrascale+ ZCU104	Zynq UltraScale+ XCZU9EG

TABLE III: Processing Time: FPGA vs GPU.

Device	FPGA	GPU
This Work	26.78 μ s	36.6 μ s (Nvidia Tesla P100)
Soltani’2019 [14]	24 μ s	3.6 ms (Nvidia Jetson AGX)

Soltani’s work is difficult since the GPU is different and their paper do not mention the operating frequency of the FPGA. For the proposed model, the processing time has been simulated on Vivado [16] at an operating frequency of 70 MHz, FPGA results outperform GPU results simulated on Nvidia Telsa P100 which is able to perform 4.7 tera Flops for double precision floating point representation [17]. The processing time is 26.78 μ s on FPGA compared to 36.6 μ s on GPU.

V. CONCLUSION

In this paper a classification Convolutional Neural Network model for modulation recognition has been presented. The model is capable of classifying 11 different modulation techniques based on their In-phase and Quadrature components at baseband. The proposed model achieves the same classification accuracy reported in the state of the art with a much lower number of parameters and calculations resulting in an architecture suitable for FPGA and ASIC implementations. Compared to another FPGA implementation the proposed IQ model classifies more modulation schemes while consuming a much lower power.

ACKNOWLEDGMENTS

This work was partially funded by the French National Research Agency: TOLTECA Project ANR-16-CE04-001301.

REFERENCES

- [1] 5G PPP. [Online]. Available: <https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf>.
- [2] Y. Molina-Tenorio, A. Prieto-Guerrero, R. Aguilar-Gonzalez, and S. Ruiz-Boqué, “Machine Learning Techniques Applied to Multiband Spectrum Sensing in Cognitive Radios,” *Sensors*, vol. 19, no. 21, p. 4715, 2019.

- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [4] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Fei-Fei Li. "ImageNet: A large-scale hierarchical image database." 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009), pp. 248-255, 2009.
- [5] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," in *IEEE Access*, vol. 7, pp. 19143-19165, 2019.
- [6] T. J. OShea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, pp. 213–226, 2016.
- [7] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors", *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433-445, 2018.
- [8] RADIOML 2016.10A Dataset [online]. Available: <https://www.deepsig.ai/datasets>.
- [9] Kaggle, [Online]. Available: <https://www.kaggle.com/>.
- [10] F. Chollet et al. (2015). Keras. [Online]. Available: <https://github.com/fchollet/keras>.
- [11] C. Zhang, Joint Training Methods for Tandem and Hybrid Speech Recognition Systems using Deep Neural Networks, Ph.D. thesis, University of Cambridge, Cambridge,UK, 2017.
- [12] Venali Sonone. (2019). Notes on Deep Learning Softmax Classifier. [Online]. Available:<https://medium.com/datadriveninvestor/notes-on-deep-learning-softmax-classifier-971b3df27466>.
- [13] M. Kulin, T. Kazaz, I. Moerman and E. De Poorter, "End-to-End Learning From Spectrum Data: A Deep Learning Approach for Wireless Signal Identification in Spectrum Monitoring Applications," in *IEEE Access*, vol. 6, pp. 18484-18501, 2018.
- [14] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davaslioglu, H. Deng and T. Erpek, "Real-Time and Embedded Deep Learning on FPGA for RF Signal Classification," *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, Norfolk, VA, USA, 2019, pp. 1-6, 2019.
- [15] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davaslioglu, H. Deng and T. Erpek, "Real-Time Experimentation of Deep Learning-based RF Signal Classifier on FPGA," *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Newark, NJ, USA, pp. 1-2, 2019.
- [16] Xilinx Vivado Design Suite,[Online]. Available: <https://www.xilinx.com/products/design-tools/vivado.htm>.
- [17] Nivida Tesla P100,[Online]. Available: <https://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-PCIe-datasheet.pdf>.