International Journal of Gender, Science and Technology
..ISSN: 2040-0748

UGC Care Group I Journal
Vol-10 Issue-02 Oct 2021

Real-Time Signal Classification through the Implementation of Convolutional Neural Networks.

Prasant Sethi,
Department of Electronics and Telecommunication, College of Engineering Bhubaneswar

***Abstract*— Discriminating signals in the wireless spectrum is an essential skill for many applications, including mesh networks, defense, and the military. We suggest a signal categorization system used in embedded software-defined radio, inspired by these applications. The design of the system for over-the-air communications and consideration of real-world circumstances, such as hardware limitations and fluctuating channel congestions, set apart the proposed work. A lightweight CNN is integrated with the modified Bartlett-based approach on a stand-alone embedded device. Over-the-air measurements from a WLAN modem that replicate the sparse and dense channel conditions yield datasets. It is tracked how the dataset's bit resolution and dimension count affect classification. We demonstrate the intended system's functionality at various traffic rates.**

***Index Terms*—Bartlett method, convolutional neural network (CNN), deep learning, signal classification, software defined radio.**

–

I. Introduction

THE INCREASING demand for connecting more and more wireless devices requires smarter radios to find a convenient transmission opportunity when primary users do not exploit the available spectrum [1]. The cognitive radio (CR) meets the efficient utilization requirements with spectrum sensing to acquire information about the active users in the available spectrum by taking advantage of different methods, such as energy detection or cyclo-stationary feature detection [2], [3]. They have some drawbacks, such as human intervention, computationally complex receiver designs, or long observation times. Real-time detection and classification in CR systems is an important requirement to construe the signals in wireless communication applications, such as spectrum interference monitoring, dynamic spectrum access, and jammer detection. By inspiring these problems, the proposed signal classification scheme is designed according to the conditions of real-time systems.

The availability of immense data and the enormous growth in the computers' computation power changed the trend in the signal classification area from traditional methods to machine learning (ML)-based methods in the last years [4]. Some ML-based methods learn signal patterns implicitly, result- ing in better adaptability and operation capability without prior knowledge [5]. In [6], AlexNet and GoogLeNet, well- known convolutional neural networks (CNNs) are employed by using the constellation diagrams to train networks.The results show the superior performance of CNN-based methods compared to the traditional methods and SVM. O'Shea *et al.* [7] used prominent CNNs: ResNet, VGG, and trained the networks with 24 different modulated syntheticand real-time signals generated via GNU Radio. They achieve satisfactory results on synthetic *(*94%*)* and real-time (87%) signals.

The above-mentioned studies use the datasets gathering via imitative designs on laboratory or simulation-based systems, namely, the systems are presented conceptually. Moreover, the results are obtained when the target signals are contin- ually available in the communication channel, though they have achieved high accuracy. This makes the system unsta-ble in real-life scenarios due to the various time gaps between adjacent messages. Additionally, they have implemented their models on larger-sized high-performance computers that lead to hassles on-field applications. Some other works utilize the CNNs using hardware accelerators by focusing on time and space costs besides accuracy [8], [9]. Conversely, we centered our main point on field applicability by unifying the field-programmable gate array (FPGA) and central processing unit (CPU) for signal processing and CNN.

In this letter, the flexible and directly applicable system is proposed to employ in signal classification problems. Our major contributions are listed as follows.

1) A CNN and modified Bartlett-based preprocessing method are utilized. The results indicate that the proposed CNN achieves comparable classification accuracy with the state-of-the-art (SOTA) architectures. By considering the real-life challenges, the results are extended in terms of model size complexity and floating-point operations per second (FLOPS). Even though our model demonstrates faintly less accuracy *(*0.7% – 2.1%*)*, it is reasonably practicable due to its lightweight structure. Our model has at least 135 times reduced parameters and 7.43 times fewer FLOPS than the close competitor.

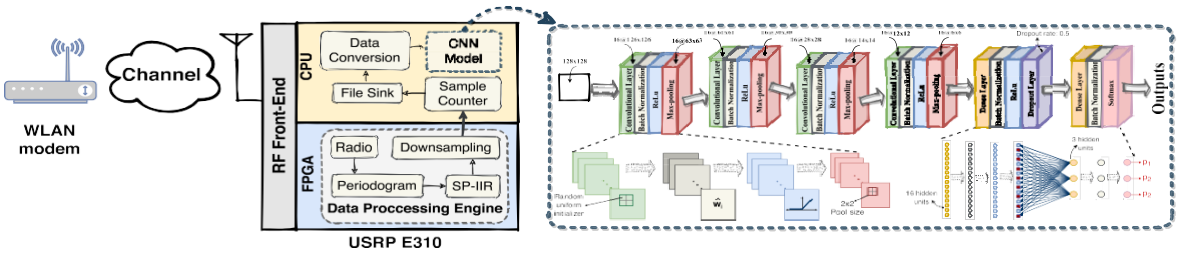

Fig. 1. System model. After the WiFi signals are received by the RF front-end, they are processed in FPGA to generate Bartlett vectors. The dataset samples are acquired from the vectors and passed through the lightweight CNN architecture to identify the belonging class.

2) The proposed system is implemented on a small-sized software-defined radio (SDR) to be utilized in real-life applications by handling low memory and computational capability challenges.
3) We generated a dataset using wireless local area network (WLAN) modem signals, including real-user activities, intractable, or impulsive circumstances.
4) The proposed CNN architecture offers a robust performance against the sparse and bursty channel conditions.

The designed system is capable of classifying other types of signals besides WLAN signals owing to the feature extraction ability of CNNs. WLAN signals are employed to measure the performance of the designed system due to the easy availability of those signals. The results show that the presented CNN-based method detects and identifies the WLAN signals with a high forecasting performance in real-time scenarios.

II. Background

A. Signal Model

The received signal on the SDR device is formulated as $y(t) = x(t) * h(t) + n(t)$, where $x(t)$ represents the transmitted WLAN signal from the modem, $*$ is the convolution operator, $h(t)$ and $n(t)$ are the impulse response of the time-invariant channel, and a sample function of noise in the indoor environment. Following the signal reception, $y(t)$ is sampled with 640 MS/s (megasample per second) on ADCs and represented as $y[n]$.

B. Modified Bartlett's Method

Target signals have distinguishable features in the frequency domain; thus, power spectral density (PSD) estimation techniques are convenient for preprocessing raw signals to solve the signal classification problem. A periodogram-based method is employed in this letter due to the strong performance of nonparametric PSD estimation techniques in real environments. The periodogram relies on performing a discrete Fourier transform (DFT) on a given signal. Let $y[n]$ denotes a real finite-length sequence where $n = 0, \ldots, 512$, and the DFT of this sequence is defined as $Y(f) = \sum_{n=0}^{N-1} y[n]e^{-j2\pi fn}$.

The periodogram is not a preferable estimator due to the high variance. Bartlett's method introduces the averaging operation of the L consecutive periodograms to reduce large fluctuations. The Bartlett spectral estimation method is given as

$B(f) = (1/L) \sum_{j=0}^{L-1} P_j(f)$, where $P_j(f)$ is the jth periodogram.

The spectral resolution of Bartlett's method is degraded by L, and this degradation brings the variance reduction. There is a tradeoff between two parameters; therefore, one can obtain the desired resolution or variance by tuning L's value. The single-pole infinite impulse response (SP-IIR) filter is used instead of the averaging technique to smooth periodograms. SP-IIR filter is a powerful filter; its simplicity satisfies the parallelization phenomenon of the FPGA design. SP-IIR filter can be defined as $r[n] = \eta p[n] + (1 - \eta)r[n - 1]$, where a single parameter η is called a smoothing factor between 0 and 1. $p[n]$ is the periodogram sample, and $r[n]$ is the output sample of the filter. As the η value increases, better-smoothed spectrograms are obtained. Effortlessly, parallelable equation $r + \eta(p - r)$ is derived from above equation.

III. Design and Implementation

A. Data Preparation

In this letter, the proposed system in Fig. 1 is implemented on USRP E310 that provides practical benefits for real-life scenarios due to its stand-alone nature and small size. The computationally intensive signal processing operations of the design are realized on FPGA by employing an RF network on chip (RFNoC), and the CPU is utilized for control tasks and low-speed operations. The blocks that are designed using hardware description language (HDL) to achieve specific signal processing tasks are called computation engines (CE). The RFNoC framework handles configuration and data flow between CEs in FPGA. GNU Radio is used as a high-level software above the HDL to connect CEs. The hardware implementation is shown in Fig. 1, and it contains three major parts: 1) the data processing engine (DPE); 2) data conversion; and 3) CNN-based classification.

DPE performs Bartlett's method using the FPGA. Radio CE controls and configures the radio hardware parameters, such as sampling frequency, center frequency, and gain. It receives digital in-phase and quadrature (IQ) samples from the analog-to-digital converter (ADC) and passes through the periodogram CE. It receives samples as vectors with a predefined size to perform fast Fourier transform (FFT) and applies scaling and magnitude square operations on FFT results. The SP-IIR filter smooths the periodograms and generates Bartlett vectors. One of the challenges in implementing the design on an embedded system is to balance the processing time of the high-speed FPGA and the sluggish CPU to prevent the input buffer of CPU from overflowing; hence, a downsampling process is also introduced in FPGA design in addition to

TABLE I
TIMING AND SPACE COMPLEXITY OF FPGA DESIGN

	Max Frequency	Slice	LUT	FF	DSP	BRAM
RFNoC Infrastructure	-	6814	15930	20829	0	26
SP-IIR	70 02 MHz	1411	3217	5421	4	14
Periodogram	83 9 MHz	2439	6051	10739	52	18
Downsampling	90 2 MHz	1364	2772	5268	0	8
DPE (Total)	69 5 MHz	12028	27970	42257	56	66

the theoretical operations. Downsampling block simply selects one Bartlett vector from every N vectors and drops the others. The appropriate sampling ratio is obtained with empirical tests. Downsampled Bartlett vectors are stored in files to prepare datasets for CNN. Consecutive Bartlett vectors are unified to generate the dataset. Let $n, j\ 0, \ldots, 512$, $Y_j[n]$ denote a Bartlett vector where j is the column number of an image matrix. y-axis, x-axis, and pixel values of an image represent frequency, time, and signal power, respectively. Normalization is necessary to prevent outlier weights in the direction of the neural network. Each sample is normalized using the z-score normalization, denoted as $y\ [(x\ \mu_\varphi)/(\sigma_\varphi)]$, where μ_φ and σ_φ are the mean and standard deviation of the sample. The normalized sample is passed as an input layer to CNN. Further details of the proposed design are provided in [10].

B. Convolutional Neural Network

The proposed CNN architecture is given in Fig. 1. The network has six hidden layers combined with batch normalization, activation, and pooling functions. The dropout layer is employed to apply the regularization effect. The shape of the input and output layers are (128 × 128 × Channel Number) and 3, respectively. The three neurons at the output layer represent the input signals' types (802.11b, 802.11g, and 802.11n). The hidden convolution layers have 3 × 3 filter kernels with a count of 16, and the dense layer has 16 neurons. The adaptive moment estimation algorithm is selected as the optimizer with learning rate 0.01, β_1 0.9, and β_2 0.999. 66 000 samples are used to train the network. The hyperparameters of the training process: mini-batch and epoch sizes are 32 and 100. Each dataset is divided into three subsets: 1) training; 2) validation; and 3) testing with 65%, 16%, and 19% distributions, respectively. The training set is used to optimize the weights by updating model parameters. The validation set is employed for tracking the evolution of the training process to prevent overfitting and underfitting. The major target of parameter tuning is to obtain a better-optimized model. The test set is used for measuring the performance. The CNN is implemented in the CPU of the USRP E310 using basic Python libraries for testing the model in the target device.

IV. EXPERIMENTS AND RESULTS

The first step of the system analysis is to collect the dataset on a proper test-bed. A WLAN modem is employed as a transmitter with an analog gain of +20 dB and a center frequency of 2.437 GHz. USRP E310 is used as a receiver with 0 dB gain

TABLE II
TEST RESULTS OF CNN MODELS

		Predicted			
		802.11b	802.11g	802.11n	
True	802.11b	0.849	0.145	0.006	16-bit RGB
	802.11g	0.071	0.854	0.075	
	802.11n	0.013	0.056	0.932	
	802.11b	0.847	0.148	0.005	8-bit gray
	802.11g	0.055	0.936	0.009	
	802.11n	0.014	0.063	0.923	
	802.11b	0.841	0.154	0.006	16-bit gray
	802.11g	0.030	0.958	0.011	
	802.11n	0.008	0.070	0.922	

as 512, 0.986, and 1/18, respectively. The rate of IQ samples through FPGA should be at least 20 MHz to prevent the Nyquist rate violation. The appropriate sample rate is selected through experiments to provide an image of several WLAN channels simultaneously and not overload the processing capability of the CPU. The rate of IQ samples to FPGA is decided as 40 MHz, and they are decimated through CPU using the downsampling block, which results in approximately 2 MS/s. The timing and space complexity of FPGA design is given in Table I by indicating the maximum frequency and reserved resources for individual CEs and DPE. USRP E310 has an Artix-7 FPGA, which contains 13 300 slices, 220 DSP, and 140 BRAM blocks, where each slice includes 4 LUT and 8 FF. An initial space has to be allocated for RFNoC infrastructure to leverage RFNoC capabilities. DPE utilizes 39.2% of slices without RFNoC infrastructure and 90.4% with it. DPE can operate up to 69.5 MHz, which means the desired operating frequency of 40 MHz is achieved. The incoming samples to CPU (Bartlett vectors) are converted to three different types of images as 16-bit RGB, 16-bit grayscale (GS), and 8-bit GS to monitor the influence of precision and channel count on the accuracy.

The effect of the SP-IIR filter on the classification accuracy of the CNN is experimented. According to the results, SP-IIR smoothing increases the classification accuracy by 6% rather than raw images. The test results of the models are given in Table II. The RGB image has a larger size than the others; therefore, it requires more computation and storage resources. The RGB dataset's size introduces a drawback in terms of the training time of the CNN. One epoch duration of an 8-bit GS dataset is 20% faster than the RGB dataset despite the 66% size difference. The reason for such a less speed gain is that the size difference of the input matrix influences only the first convolution layer. Overall accuracies are 90%, 90.2%, and 91.4% for RGB, 8-bit and 16-bit GS datasets, respectively. Regarding these datasets, image sizes are 48, 16, and 32 KB; training durations of one epoch are 30, 24, and 25 s. RGB encoded data does not contribute additional information for CNN compared to GS data despite its three channels. Although 8-bit and 16-bit GS datasets have their accuracy advantages, the 8-bit dataset has sample size and training time advantages compared to the 16-bit dataset.

One sample's processing time on FPGA, data conversion,

and 2.44 GHz center frequency. Target signals are dissemi- nated with the bandwidth of 20 MHz using two different trafficrates but the same modulation parameters. FFT size, η coef- ficient of the SP-IIR filter, and downsampling rate are chosen and CNN classification is 0.0065 (0.0037 if operated at
69.5 MHz), 2.6, and 46.3 s, respectively. The CNN models' robustness is also examined for different data rates ranging from 0.5 kb/s to 4 Mb/s. Data rate versus accuracy graphs are plotted with a 95% confidence interval in Fig. 2. According to the results, 802.11b is hard to predict for all models, but the 16-bit GS model is more rewarding than the others. Behind this, 802.11b signal having a DSSS modulation scheme resultsin a less distinguishable frequency fingerprint in the spectrum that confuses the classifier. It can be asserted that our designis satisfactory at data rates higher than 100 kb/s for 802.11b and 2000 kb/s for 802.11g & 802.11n. The input image frame captures fewer signal samples in low data rates; therefore, it results in a performance degradation.

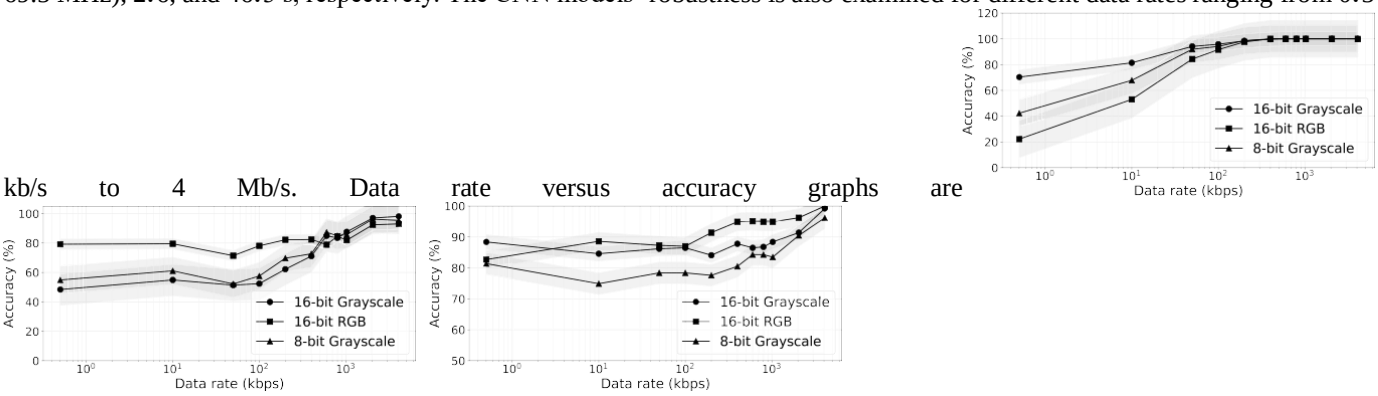

Fig. 2. Performance of (a) 802.11b, (b) 802.11g, and (c) 802.11n models in terms of accuracy with respect to data rate.

TABLE III
COMPARATIVE PERFORMANCE RESULTS WITH SOTA CNNS

Model	Accuracy	Parameters		FLOPS	
		Count	Ratio	Count	Ratio
Proposed CNN	87.1%	16,735	1×	0.874G	1×
EfficientNetB0	87.8%	4.052.831	242×	8.17G	9.34×
EfficientNetB1	89.0%	6,578,499	393×	12.1G	13.84×
MobileNet	88.4%	3.231.363	193×	11.8G	13.5×
MobileNetV2	89.2%	2.261.251	135×	6.25G	7.43×
LeNet-5	80.2%	1,741,111	104×	0.831G	0.95×

The proposed architecture is compared with five different SOTA [11]–[13] architectures toward more complex ones. All SOTA CNNs and the proposed CNN are trained with thesame amount of dataset. Table III presents the performanceof all architectures in terms of classification accuracy, model size complexity, and FLOPS. LeNet-5 has shown weaker performance than other architectures due to fewer convo- lutional layers that decrease the quantity of extracted fea-tures. The proposed architecture is the lightest, even though other CNNs' (except LeNet-5) classification performances are slightly better. The reason why similar accuracies are obtained with proposed CNN despite the complexity of SOTAs is feed- ing networks with a relatively small-sized data set and solving a moderate-sized problem. The proposed CNN employs an order of magnitude fewer parameters and FLOPS compared to other models, except LeNet-5. The complexity advantage of our model results in a faster reaction, which is the objective of most real-life applications.

V. CONCLUSION

For use in practical applications, we suggested a WLAN signal categorization technique. The signal processing processes in the FPGA transform WLAN signals into more unique image samples. Three datasets with varying bit resolution and color are used to train three models with the low-complexity CNN architecture. The results of the experiments showed that the GS outperforms the RGB model, indicating that bit resolution does affect performance but that channel count is not a reliable predictor of classification accuracy.The models provided a promising performance for differ- ent wireless throughputs. Additionally, the reasonableness is seen in the comparative results with SOTA methods that the proposed CNN has a remarkable performance besides ensur- ing lightweight architecture. Our model has at least 135 times smaller amount of parameters and 7.43 times fewer FLOPS compared to the rivals. FPGA or VLSI implementation of CNN will be considered as the future work to improve energy consumption [14] and the processing speed of the system.

REFERENCES

[1] S. Haykin, "Cognitive radio: Brain-empowered wireless communica- tions," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[2] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary signatures in practical cognitive radio applications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 13–24, Jan. 2008.

[3] M. V. Lipski, S. Kompella, and R. M. Narayanan, "Practical imple- mentation of adaptive threshold energy detection using software defined radio," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Nov. 24, 2020, doi: 10.1109/TAES.2020.3040059.

[4] O. A. Topal, S. Gecgel, E. M. Eksioglu, and G. K. Kurt, "Identification of smart jammers: Learning-based approaches using wavelet preprocess- ing," *Phys. Commun.*, vol. 39, Apr. 2020, Art. no. 101029.

[5] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2209–2221, Nov. 2013.

[6] S. Peng *et al.*, "Modulation classification based on signal constellation

diagrams and deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019.

[7] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[8] L. Song, X. Qian, H. Li, and Y. Chen, "PipeLayer: A pipelined ReRAM-based accelerator for deep learning," in *Proc. Symp. Comput. Archit. High Perform. Comput.*, Feb. 2017, pp. 541–552.

[9] A. Shafiee *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Comput. Archit. News*, vol. 44, no. 3, pp. 14–26, Jun. 2016.

[10] R. Cetin. (Dec. 2020). *CNN-Based-Signal-Classification-in-Real-Time*. Accessed: Dec. 30, 2020. [Online]. Available: https://github.com/rcetin/CNN-based-Signal-Classification-in-Real-Time

[11] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019. [Online]. Available: arXiv:1905.11946.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[13] Y. LeCunn, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[14] G. Krishnan, S. K. Mandal, C. Chakrabarti, J. S. Seo, U. Y. Ogras, and Y. Cao, "Interconnect-aware area and energy optimization for in-memoryacceleration of DNNs," *IEEE Design Test*, vol. 37, no. 6, pp. 79–87, Dec. 2020.