Spectrum Sensing using CNN based Deep Learning

Deep Ashish Jariwala
University of Waterloo
Waterloo, Canada
djariwal@uwaterloo.ca

Sayonee Bimal Parikh
University of Waterloo
Waterloo, Canada
s26parikh@uwaterloo.ca

Vasu Jayeshkumar Patel
University of Waterloo
Waterloo, Canada
vj4patel@uwaterloo.ca

***Abstract*—Deep learning (DL) is a new paradigm of machine learning (ML) that has shown exceptional performance in computer vision, voice and natural language processing. However, researchers have not explored the use of DL to wireless communication to its full potential. The use of DL technology for wireless communication applications has recently gained popularity. This paper looks into the application of deep learning based approach for Spectrum Sensing. Spectrum Sensing has a wide range of applications ranging from civilian to military. A novel convolutional neural network (CNN) based on deep learning architecture for Spectrum Sensing is proposed in this paper. We demonstrate using experiments that the proposed architecture is better than the existing CNN based architectures for spectrum Sensing.**

***Index Terms*—Convolutional Neural Network, Spectrum sensing, Machine learning, Deep learning,**

I. INTRODUCTION

Deep Learning and Machine Learning are two subsets of Artificial Intelligence (AI). Machine Learning utilizes algorithms to parse data, then learn from that data and make informed decisions based on the learning. Deep Learning will structure algorithms into different layers to build an artificial neural network that can independently learn and create intelligent choices. While deep learning and machine learning fall under AI, deep learning is a subfield of machine learning; deep learning powers the most human-like AI.

The arrival of present-day technologies and applications such as Cyber-Physical Systems, the Internet of things (IoT), etc., has pushed forward the rise in demand for wireless spectrum. As a spectrum is a limited resource, this increase in demand cannot be achieved easily, and its expansion is also difficult due to technological limitations. Therefore, an increase in the utilization of the spectrum has become the main aim of researchers.

Mitola proposed cognitive radio technology for the improvement of spectrum utilization. It allows unlicensed devices to utilize the licensed spectrum, like TV broadcast bands given the opportunity. This approach has been a potential proposed method to address the shortage of spectrum [1-3]. It is a technology that allows secondary users (SUs) to access the licensed band of primary users (PU) when the primary user is not utilizing it. Few challenges could be encountered for the successful practical deployment of cognitive radio systems. The main challenge here is to provide ample protection to the users who are licensed. Therefore, detecting the presence of reliable primary signals is of utmost priority to overcome the challenge [4].

The most broadly used conventional detector is the energy detector because of its simple work. In practice, because its performance heavily relies upon noise density knowledge and existing noise uncertainties, when the signal to noise ratio drops below the threshold value, the energy detector also fails to work. According to the existing literature, the SNR-wall for practical noise uncertainty is about -6dB, far from the SNR limit of -15 dB as required by IEEE 802.22. The authors in [4] discussed three different approaches to get around the SNR wall, namely, exploiting the structure of the primary signal, using diversity, and reducing the noise uncertainty.

Quite often, the prior knowledge of the primary signals is not contained by the secondary users; in such cases, it is helpful to devise a method called the blind sensing method, which is capable of identifying the underlying structure of the primary signals. Deep learning (DL) recently demonstrated an extraordinary potential in extracting the hidden structure of various objects in complicated tasks was demonstrated by deep learning (DL) recently. Those tasks include wireless communication, computer vision, etc. [6]. Machine learning in the context of spectrum sensing has been discussed in the paper [9, 10]. Some detailed reviews of the application of deep learning in the physical layer can also be found in [7, 8].

Inspired by the concepts discussed and the results obtained in [10], here we develop a DL-based detector using convolutional based neural networks (CNN) [11], which is applicable for arbitrary types of primary signals. One key point about this DL-based detector is that it does not require extra information on the noise density or primary signal when deployed online. Additionally, improving the sensing performance further, a DL-based-based soft combination strategy is proposed for cooperative detection. This DL-based detection is better and outperforms other conventional methods upon the comparison of simulation results.

This work proposes novel CNN-based architecture with some significant regularization and pooling techniques, which are essential parts of the proposed architecture. The flow of this paper starts with a brief discussion about the problem statement, followed by a deep learning-based solution for the problem. The paper shows the network architecture used to train the model that allows spectrum detection based on the dataset preparation. Finally, the paper includes information about the training, validation, and testing of the created model, followed by its simulation results and conclusion.

II. PROBLEM STATEMENT

The secondary users' signal detection, depending on whether the primary user is in a busy state or idle can be modelled as the following binary hypothesis testing problem, EQUATION where s(n) is the signal generated by primary user, y(n) is the nth received sample, h is the channel gain which is under the assumption that it remains unchanged during the sensing period and w(n) is the additive noise which follows the zero mean circularly symmetric complex Gaussian (CSCG) with variance. The two hypotheses which represents the absence and presence of primary signal in a specific band is denoted by H0 and H1 respectively. In a conventional energy detector, the energy of the received signal normalized with respect to the sample number N and noise variance is its test statistic and is given by, EQUATION EQUATION

III. DEEP LEARNING BASED MODEL FOR SPECTRUM SENSING

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections VII–**??** below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

IV. NETWORK ARCHITECTURE

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

V. DATASET DESCRIPTION

In this section, we give details of the publicly available RadioML dataset used in this work for evaluating the performance of the proposed model.

A. SOURCE

The dataset used for building the model is labelled as "DEEPSIG DATASET: RADIOML 2016.04.C" that can be downloaded from https://www.deepsig.ai/datasets in pickle python format. It is a dataset with variable-SNR included with moderate local oscillator (LO) drift, light fading, and numerous different labelled SNR increments for use in measuring performance across different signal and noise power scenarios.

B. GENERATION

The dataset was generated by combing the logical modules together in a GNU Radio as mentioned in [**?**]. The process of generation starts with selecting a voice signal for analog signals and ASCII symbols for digital modulation. The second step involves the normalization of steps per symbol to form a constant rate of symbols before modulation and applying modulation to the signal. The final step is channel simulation, which involves using the GNU Radio dynamic channel model hierarchical block. It includes several desired effects such as random processes for centre frequency offset, sample rate offset, additive white Gaussian noise, multi-path, and fading. This process generates a dataset which different classes of modulation at different SNR.

C. DETAILS

The dataset consists of 1,61,800 rows categorized in three columns: SNR, modulation type, and data of the signal. The dataset includes a comprehensive range of SNR values starting from -20dB to +18dB, progressing at an interval of 2dB. Furthermore, the dataset is classified into 11 different types of modulation, of which eight are digital, and three are analog. The modulation type parameters are AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM64, QAM16, QPSK, WBFM, and 8PSK. The signal data (complex-valued inputs) for each SNR and modulation type consists of an array of dimensions [2, 128] where the orthogonal synchronously sampled In-phase, and Quadrature (I & Q) samples make up this 2-wide dimension. The table 1 describes the details of the dataset

D. PREPROCESSING

The preprocessing for the dataset is carried to filter out the digital modulation signals. 8PSK, BPSK, GFSK, QAM64, PAM4, QAM16 modulation signals are included in the dataset. CSCG noise is generated based on the I and Q value's variance to create an equal amount (number of signal data) of random data with a zero mean. The noise data is added to the signal data and appended with the noise data to double the number of samples. The samples with (noise + signal) are labeled as 1, and those with only noise are labeled as 0.

TABLE I: RADIOML 2016.04.C DATASET PARAMETERS

Modulation type	8PSK, BPSK, GFSK QAM64, PAM4, QAM16
Sample length	128
SNR Range	-20dB to +18dB
Total Data	161800 samples 80900 (signal + noise) data 80900 noise data
Training samples	113260 vectors
Validation samples	33978 vectors
Test samples	48540 vectors

E. VISUALIZATION

The visualization of this dataset is employed by the time-domain plot of the I and Q values of signal data. The fundamental differences among all the classes can be easily spotted, but the channel effects are not readily visible. The Fig. 2 shows all different samples.

VI. SIMULATION RESULTS

This section will discuss the results obtained in training, validation, and testing the final model. It provides details regarding the training, testing, and validation performance by providing plots for accuracy and loss vs epochs. The confusion matrix is used as a performance metric to show all SNR values' true and predicted labels.

A. TRAINING DETAILS

The dataset is split into two parts 70% for training and 30% for testing. The training is conducted using the ADAM optimization technique that uses binary cross-entropy as a loss measure. Implementing the model for training and testing takes place on an NVIDIA Cuda enabled GeForce 1080 Ti GPU. The implementation uses an early stopping call-back method (patience of 4) that measures the validation set's progress for a number of epochs to reduce overfitting and roll back to the best model. The model was trained over 113260 data samples that achieved an accuracy of 86.11%, with a loss of 0.2463.

B. VALIDATION DETAILS

Validation details: The validation split for the best model was selected as 0.3. The training results gave the best validation accuracy of 86.61% and a least loss of 0.2432. All the results were recreated using our dataset for the model proposed by [**?**] [**?**] [**?**] [**?**] provided a validation loss of more than 0.84, which is a vital improvement by our model as it reaches a new minimum value of before early stopping.

The Fig. 6 shows the plot for loss versus epochs, and the Fig. 7 shows the plot for accuracy versus epochs for both training and validation data. The presence of regularization methods like adding Gaussian noise and dropout layers is responsible for early convergence, i.e., within 15 to 25 epochs with better results.

C. TEST DETAILS

The test accuracy obtained by the proposed system is 68.946%, with a loss of 0.8242. The parameters obtained by this architecture is better than the recreated results on [**?**] [**?**] [**?**] [**?**] that achieved the highest test accuracy of 65.39% with this dataset.

The effect of noise layers in deep learning has a significant impact on performance parameters. It mitigates overfitting and generalizes the data very efficiently to classify. One of the many observations obtained is that the noise layer eliminates the memorization effect and improves the algorithm's learning rate. Based on the paper [**?**], it was observed that Gaussian noise has the maximum impact of regularization, and the convergence rate is faster with a new minimum loss. All the observations are compared with the models proposed by [**?**] [**?**] [**?**] [**?**]. Another consideration from [**?**] is equally applicable to the proposed system because accuracy and convergence rate improves with a complex deep neural network.

D. CONFUSION MATRIX AND SNR PLOT DETAILS

The Fig. **??** shows the confusion matrix visualization of test data for all SNR values. The results highlight that WBFM is misclassified as AM-DSB up to some extent and vice-versa. Moreover, the test result depicts a misclassification among BPSK, CPFSK, and GFSK. The Fig. **??** shows the confusion matrix of test data split into different SNRs. The significant aspect observed is that the confusion matrix makes an almost perfect diagonal with an increase in SNR value. It can be noted from the figure that the algorithm is confused at -6dB SNR, but as the SNR increases, the classification is almost perfect. Furthermore, Fig. **??** shows the best classification result obtained at $18dB$ SNR. However, the algorithm is still a bit confused about the classification of WBFM and 8PSK.

The proper way to analyze the results is to break down the test model's accuracy across different SNR values and generate the efficiency of the algorithm for each SNR. The highest accuracy obtained by the proposed model is 96.57% at 18dB SNR. Moreover, the general trend observed in the results is that the accuracy value increases significantly from 20.59% at $-20dB$ to 96.57% at $18dB$.

The Gaussian noise and dropout layer acts as a regularization method to improve the model's robustness and efficiency. The system described in this paper was tested with different parameters (the standard deviation in the case of Gaussian noise and rate in case of a dropout layer). The variation in parameters generated different results, which were evaluated by checking the accuracy versus SNR plot.

The Fig. **??** shows accuracy versus SNR plot for the same architecture as proposed above but with different Gaussian noise parameters. The highest accuracy of 97.24% was observed at 0.0001 standard deviation value in the Gaussian noise layer and $18dB$ SNR. Furthermore, it could be seen that varying the regularization parameters can cause a significant change in the accuracy of the model.

VII. CONCLUSION

In this work, we propose a novel CNN based deep learning algorithm for automatic modulation classification. The main focus was to obtain high classification accuracy for the test dataset and minimize the validation loss. The architecture proposed in this paper has incorporated regularization layers (Dropout, Gaussian noise) and average pooling layer to form a unique architecture that out performs other algorithms. The algorithm provided promising performance in the classification of the test dataset and minimization the validation loss. Moreover, this work also highlights the benefit of using Gaussian noise after the flatten layer to improve the regularization performance and reduce the over-fitting. Another substantial gain from this work is that using the average pooling layer instead of the max-pooling layer is proven to be efficient.

Acknowledgment

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks . . .". Instead, try "R. B. G. thanks. . .". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

net_arch.png

Fig. 1: Network Architecture of the model

sample_signals.png

Fig. 2: Signal Graphs (Input)

n20snr.png

(a) -20dB SNR

n16snr.png

(b) -16dB SNR

n12snr.png

(c) -12dB SNR

n10snr.png

(d) -10dB SNR

n8snr.png

(e) -8dB SNR

n6snr.png

(f) -6dB SNR

n4snr.png

(g) -4dB SNR

n2snr.png

(h) -2dB SNR

0snr.png

(i) 0dB SNR

2snr.png

(j) 2dB SNR

4snr.png

(k) 4dB SNR

6snr.png

(l) 6dB SNR

8snr.png

(m) 8dB SNR

10snr.png

(n) 10dB SNR

12snr.png

(o) 12dB SNR

14snr.png

(p) 14dB SNR

16snr.png

(q) 16dB SNR

Fig. 3: Confusion Matrix for all SNR value

overall_confusion_matrix.png

Fig. 4: Confusion matrix for entire test set

snr_vs_acc.png

Fig. 5: Accuracy vs SNR plot

loss_epochs.png

Fig. 6: Loss plot for the CNN architecture

acc_epochs.png

Fig. 7: Accuracy plot for the CNN architecture