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] [2] [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

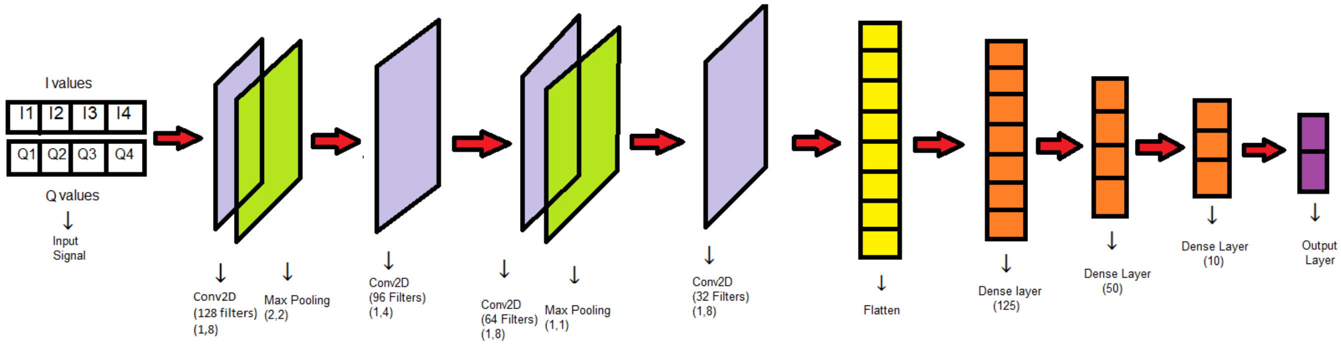


Fig. 1: Network Architecture of the model

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

Consider X(n) = [x1(n),x2(n),x3(n),x4(n),...xm(n)] , where M is the sample length of the signal. There are n signal data, n = 0,1,2,3....N-1. 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,

$$H1 : X(n) = h * R(n) + U(n) \tag{1}$$

$$H0 : X(n) = U(n) \tag{2}$$

where R(n) is the signal generated by primary user, X(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 U(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.

The aim is to build a classifier that can detect the status of the spectrum in terms of its occupancy. In mathematical terms the classifier gives out the probability $P\left[x\left(t\right) \in N_i | y\left(t\right)\right]$ where N_i represents the i^{th} class. Due to the flexibility and simplicity of mathematical operations, the received signal is commonly represented in IQ format. Let A and ϕ represent the instantaneous amplitude and phase of the received signal $x\left(t\right)$, respectively. Using this we can represent in-phase and quadrature components as $I = A\cos\left(\phi\right)$ and $Q = A\sin\left(\phi\right)$, respectively.

III. DEEP LEARNING BASED MODEL FOR SPECTRUM SENSING

The solution for the problem discussed in the paper can be obtained by building a deep learning based binary classifier that can seperate out the signal and noise data in order to detect the occupancy of the channel or spectrum. It is a binary classification problem that uses sigmoid activation to find the probability of the presence of signal.

$$\boldsymbol{D_{dl}}(\boldsymbol{y}) = \mathrm{argmax}(f^L(f^{L-1}(f^{L-2}(\cdots f^1(\boldsymbol{y}))))).$$

where the input y is the vector of received samples, which is processed through a customized neural network consisting of L layers.f power i, i = 1, · · · , L − 1, represents for the computation with weights and activation function of the i-th layer. f power L is the SIGMOID function which gives the probabilities of two hypotheses, and argmax is an operator returning the index of the largest number in a list.

IV. NETWORK ARCHITECTURE

The Fig. 1 shows the detailed architecture of the proposed model. It is divided into six stages which are described as follows:

- Stage 1: The first stage consists of two layers involving zero-padding to the input array that progresses to the first convolution layer. The first convolution layer uses 128 filters along with a kernel window of (1,8) followed by a maxpooling2D layer with a stride of (2,2) which computes the maximum and reduces the parameters. This layer is instrumental in increasing the robustness and performance of the model. The activation used at this stage is ReLu.
- Stage 2: The second stage consists of a single convolution layer to extract the features after maxpooling2D in the 1st stage.The second convolution layer uses 96 filters along with a kernel window of (1,4).
- Stage 3: The third stage consists of two layers involving zero-padding to the input array that progresses to the first convolution layer. The third convolution layer uses 64 filters along with a kernel window of (1,8) followed by a maxpooling2D layer with a stride of (1,1).
- Stage 4: The fourth stage consists of a single convolution layer to extract the features after maxpooling2D in the 3rd stage.The second convolution layer uses 32 filters along with a kernel window of (1,8).
- Stage 5: The fifth stage consists of 2 layers. The first layer is used to flatten the output array from convolution

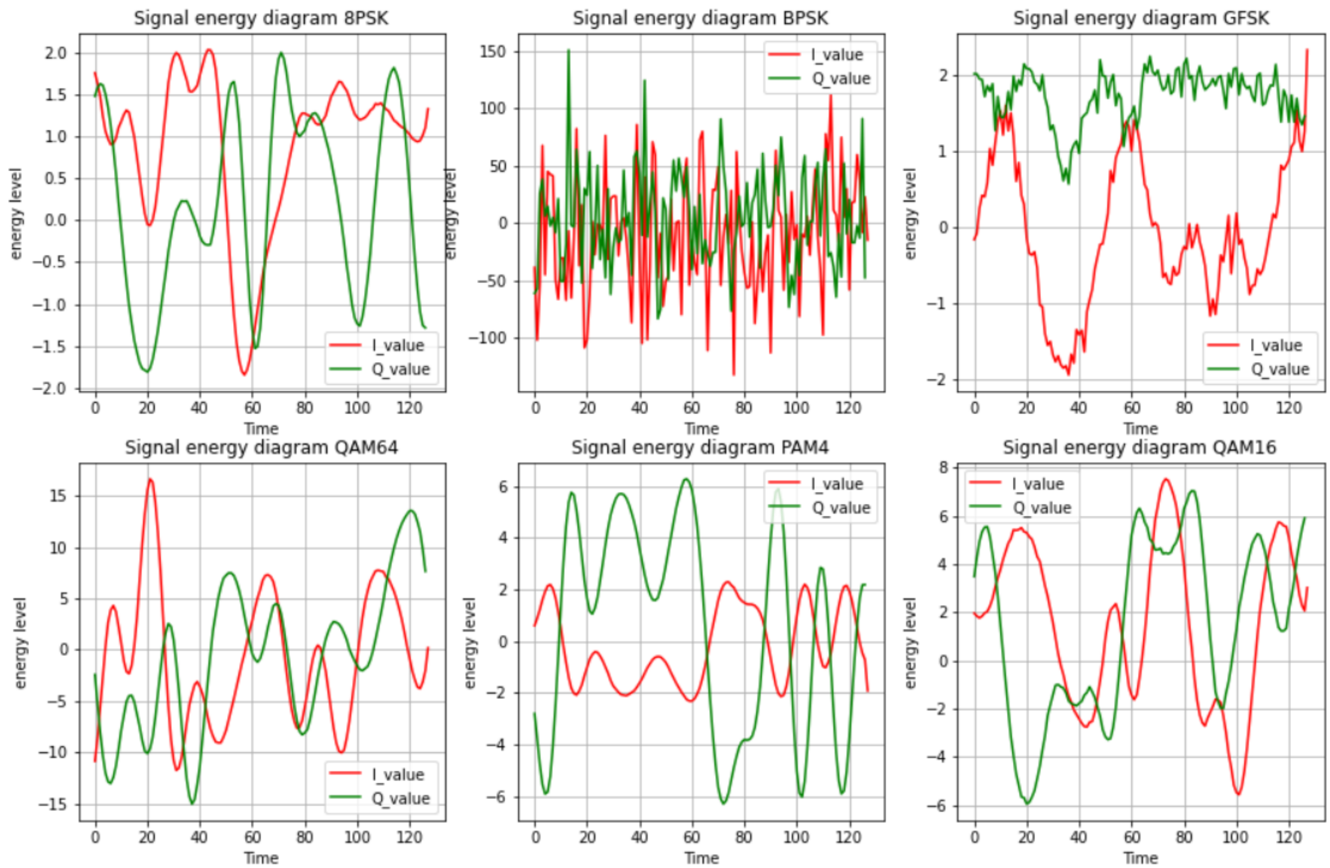


Fig. 2: Signal Graphs (Input)

to a dimension suitable for the fully stacked perceptron layer. The second layer is the Dropout layer applied as an input to the perceptron. Moreover, the second layer is a crucial part of this entire model, which improves the model's robustness and acts as a regularization layer.

- Stage 6: The sixth stage consists of four fully stacked hidden layers with 125, 50, 10, followed by a sigmoid activation for binary classification.

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 [15]. 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 standard deviation 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.

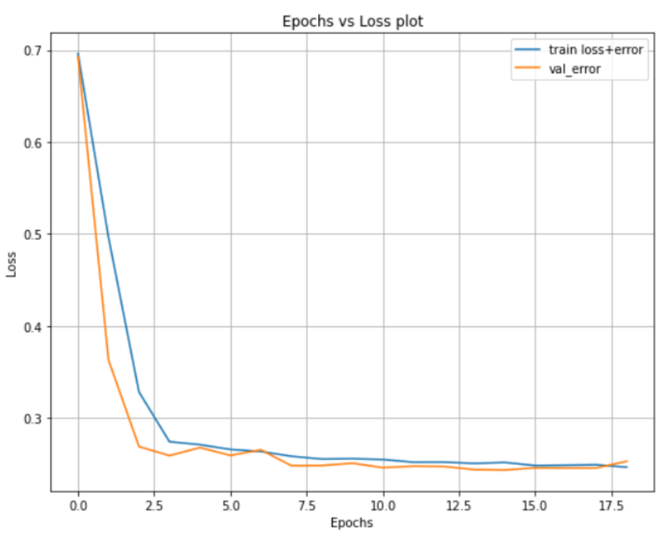


Fig. 3: Loss plot for the CNN architecture

VI. SIMULATION RESULTS

This section will discuss the results obtained in training, validation, and testing the final model. It provides details

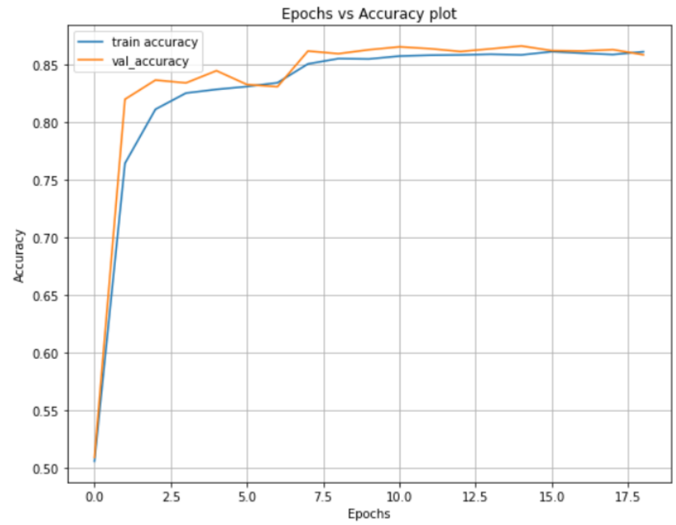


Fig. 4: Accuracy plot for the CNN architecture

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.

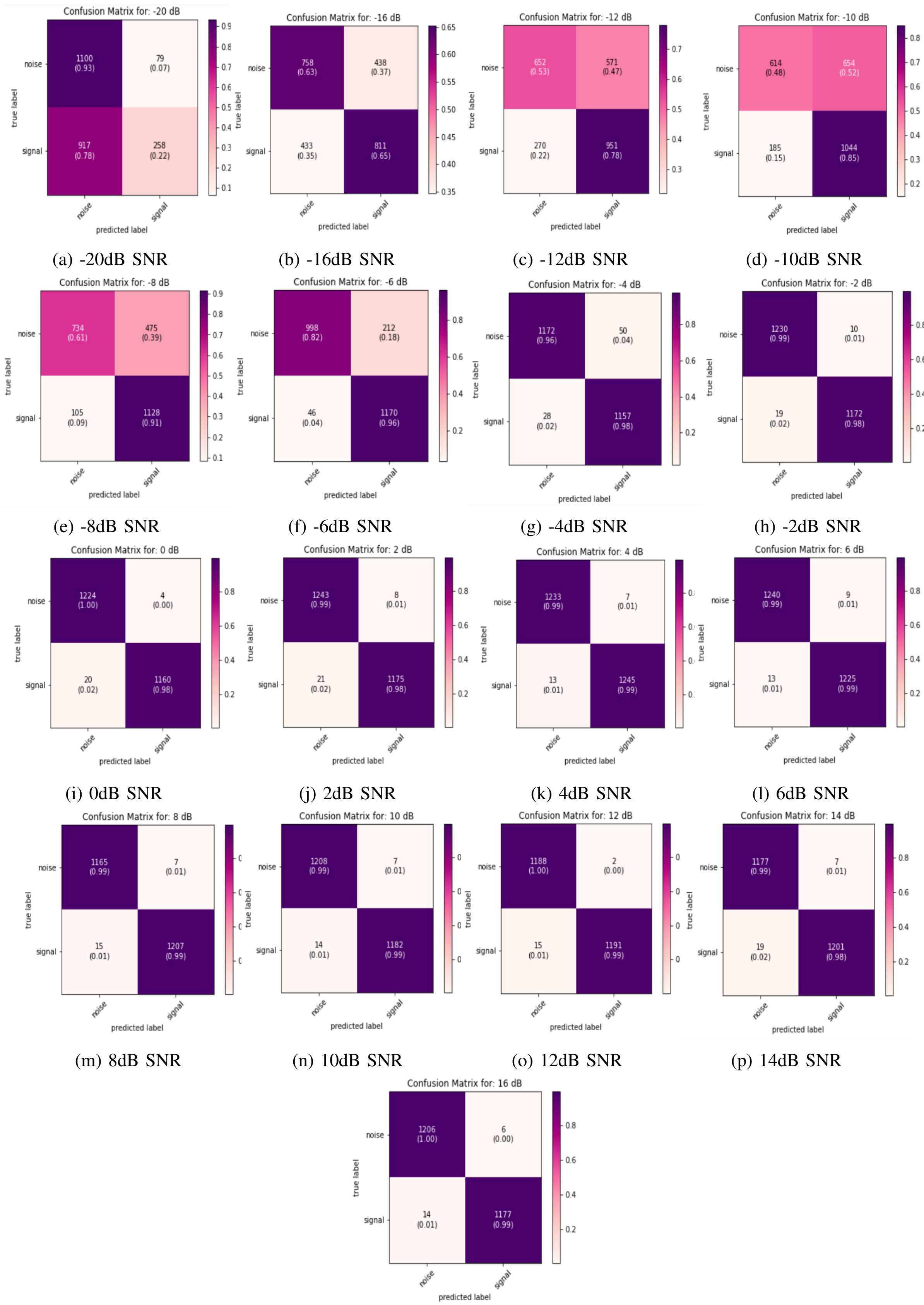
TABLE II: TRAINING PARAMETERS

Hyperparameters	**Values**
Filters per CONV2D layer	128/96/64/32
Input shape	[N, 1, 2, 128]
Filter size	(1,8), (1,4)
Neurons per FC layer	125, 50, 10, 1
Optimizer	Adam
Initial learning rate	0.01
Batch size	500
Dropout rate	0.25
Number of epochs	17 to 25
Early stopping patience	4
Activation Function	Relu/Sigmoid

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.

The Fig. 3 shows the plot for loss versus epochs, and the Fig. 4 shows the plot for accuracy versus epochs for both training



(a) -20dB SNR (b) -16dB SNR (c) -12dB SNR (d) -10dB SNR

(e) -8dB SNR (f) -6dB SNR (g) -4dB SNR (h) -2dB SNR

(i) 0dB SNR (j) 2dB SNR (k) 4dB SNR (l) 6dB SNR

(m) 8dB SNR (n) 10dB SNR (o) 12dB SNR (p) 14dB SNR

(q) 16dB SNR

Fig. 5: Confusion Matrix for all SNR value

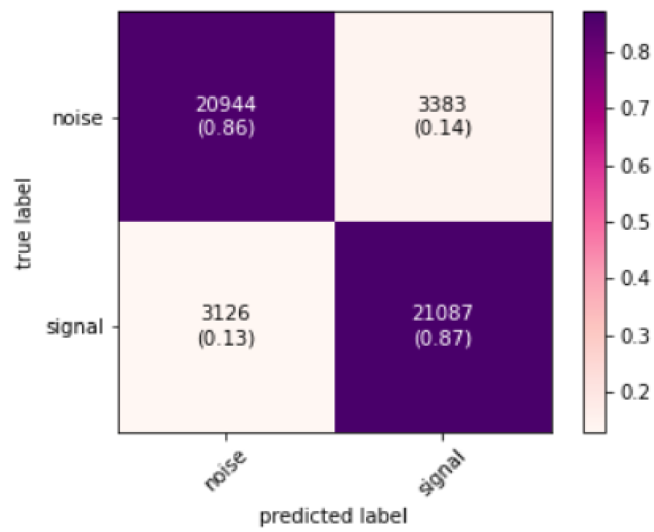


Fig. 6: Confusion matrix for entire test set

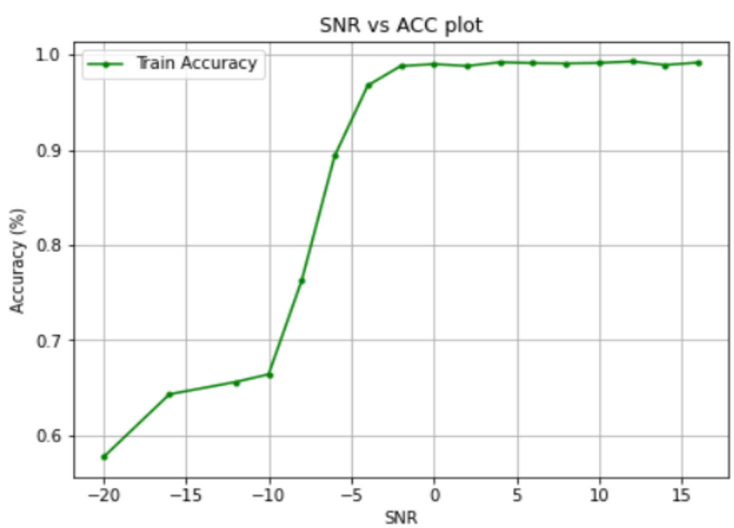


Fig. 7: Accuracy vs SNR plot

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 86.59%, with a loss of 0.2478. The parameters obtained by this architecture is better than the recreated results on [13].

D. CONFUSION MATRIX AND SNR PLOT DETAILS

The Fig. 5 shows the confusion matrix visualization of test data for all SNR values. It can be seen that the values of SNR below -4dB have a large misclassification rate compared to the values after 0. The best perfomance is seen in the case of 12dB SNR value while the most poor performance is observed in case of -20 dB SNR value. 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.

The Fig. 6 shows the overall confusion matrix for the test set for all SNR values. It can be seen that noise vectors are more misclassified compared the signal vectors. Moreover, BPSK and GFSK observes the most misclassification among other modulation type at low SNR values.

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 lower values of SNR obsereves a poor classification rate for signals and noise. The plot. 79 shows the variation of test accuracy with different values of SNRs. It can be observed that majority of the SNR values after -4dB has an accuracy over 90% while for the SNR values lower than -4dB observes a poor performance.

The table below shows the accuracy for test set at each SNR value

TABLE III: PERFORMANCE MEASURE:ACCURACY V/S SNR

SNR	Accuracy
-20 dB	57.68 %
-16 dB	64.30 %
-12 dB	65.59 %
-10 dB	66.40 %
-8 dB	76.25 %
-6 dB	89.37 %
-4 dB	96.76 %
-2 dB	98.81 %
0 dB	99.00 %
2 dB	98.81 %
4 dB	99.20 %
6 dB	99.12 %
8 dB	99.08 %
10 dB	99.13 %
12 dB	99.29 %
14 dB	98.91 %
16 dB	99.16 %

VII. CONCLUSION

In this work, we propose a novel CNN based deep learning algorithm for Spectrum sensing. 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 max 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.

ACKNOWLEDGMENT

We would like to thank University of Waterloo and course instructor Prof. Otman Basir for providing us the opportunity to work on a research project as a part of the ECE 659 course (IoT and Intelligent sensor networks). Moreover, we would like to thanks all the teaching and non-teaching staff directly involoved in helping us to learn the course.

REFERENCES

[1] S. Zhang, et al, "Novel spectrum sensing and access in cognitive radio networks," Sci. China Inf. Sci., vol. 61, no. 8, 089302, Aug. 2018.

[2] J. Zhao, X. Guan, and X. Li, "Power allocation based on genetic simulated annealing algorithm in cognitive radio networks," Chinese journal of Electronics, vol. 22, no. 1, pp. 177-180, Jan. 2013.

[3] J. Zhao, T. Yang, Y. Gong, J. Wang, and L. Fu, "Power control algorithm of cognitive radio based on non-cooperative game theory," China Commun., vol. 10, no. 11, pp. 143-154, Nov. 2013.

[4] R. Tandra and A. Sahai, "SNR walls for signal detection," IEEE J. Sel. Topics Signal Process., vol. 2, no.1, pp. 4-17, Feb. 2018.

[5] A. Krizhevsky, I.Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097-1105.

[6] P. Jiang et al. "Artificial Intelligence-aided OFDM receiver: Design and experimental results," 2018. [Online]. Available: https://arxiv.org/abs/1812.06638

[7] X. You, et al, "AI for 5G: research directions and paradigms," Sci. China Inf. Sci., vol. 62, no. 2, 021301, Feb. 2019.

[8] T. Wang, et al, "Deep learning for wireless physical layer: Opportunities and challenges," China Commun., vol. 14, no. 11, pp. 92-111, Nov. 2017.

[9] F. Azmat, Y. Chen, and N. Stocks, "Analysis of spectrum occupancy using machine learning algorithms," IEEE Trans. Veh. Technol., vol. 65, no. 9, pp. 6853-6860, Sep. 2016.

[10] Q. Chen et al. "Deep learning network-based spectrum sensing methods for OFDM systems," 2019. [Online]. Available: https://arxiv.org/pdf/1807.09414.pdf

[11] T. N. Sainath, et al, "Convolutional, long short-term memory, fully connected deep neural networks," in Proc. 2015 IEEE ICASSP, 2015, pp. 4580-4584.

[12] Y. Liang, Y. Zeng, E. C. Y. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," IEEE Trans. Wireless Commun., vol. 7, no. 4, pp. 1326-1337, Apr. 2018.

[13] Surendra Solanki, Vasudev Dehalwar and Jaytrilok Choudhary, "Deep Learning for Spectrum Sensing in Cognitive Radio" 2021. [Online]. Available: https://www.mdpi.com/2073-8994/13/1/147/pdf

[14] Luo, Yixin and Fan Yang. "Deep Learning With Noise." (2014).

[15] O'Shea, Timothy, & Nathan West. " Radio Machine Learning Dataset Generation with GNU Radio." Proceedings of the GNU Radio Conference [Online], 1.1 (2016): n. pag. Web. 6 Sep. 2020.

[16] Luo, Yixin and Fan Yang. "Deep Learning With Noise." (2014).