An Effective Convolution Neural Network for Automatic Recognition of Analog and Digital Signal Modulations for Cognitive SDR Applications

Marte Valerio Falcone
University of Roma Tre
(graduated MS student)
Rome, Italy
marte.vfalcone@gmail.com

Gaetano Giunta
University of Roma Tre
Rome, Italy
gaetano.giunta@uniroma3.it

Luca Pallotta
University of Roma Tre
Rome, Italy
luca.pallotta@uniroma3.it

***Abstract*—This paper proposes a convolution neural network (CNN) architecture for automatic recognition of signals on the basis of their modulations for Cognitive software defined radio (SDR) applications. It is developed starting from two CNNs specifically designed for this problem and is characterized by having a low number of convolution and fully connected layers sharing also a very low number of filters/units. Moreover, Batch normalization is used to increase learning rate and reduce training time. The reduced complexity together with its low operating time make it compliant with real-time SDR applications. The proposed architecture is validated on the RadioML2016.10a dataset showing interesting results in discriminating both analog and digital modulations under different signal to noise ratio (SNR) regimes.**

***Index Terms*—Signal modulation recognition, Software Defined Radio, deep-learning, cognitive radio.**

I. Introduction

Cognitive radio (CR) is one among the most emerging technologies to cope with the problem of dynamic access to the e.m. spectrum. In fact, with the use of cognition, an unlicensed user can have the possibility to opportunistically access to licensed frequency bands that are not-used in the specific time. Since CR systems do not have any prior information about the behavior of licensed users as well as about the instantaneous occupancy of the spectrum, they must resort to the exploitation of procedures for getting knowledge about the current e.m. environment. A possible way to obtain knowledge about the kind of signals that are present in the environment at a specific time could consists in identifying the modulation of the acquired signals by the CR module. This, together with parameter estimation (power, central frequency, and bandwidth), represents a fundamental task to extract knowledge from the received signal through the demodulation process.

Modulation recognition techniques are typically divided in three competing groups: 1) likelihood-based [1], 2) feature-based [2]–[7], and 3) deep-learning-based methods [8]–[10]. The first category makes use of inferences on the received signals for the identification of its modulation. However, it requires a-priori knowledge about the signal and/or noise contribution as well as a high computational cost. The second categories comprises methods that extract features like signal statistics [1], cumulants [2], [7], wavelet [4], [5], and STFT [6]. Finally, approaches based on deep-learning have recently became predominant because of their capabilities of better discriminating among different classes if a sufficiently large amount of data is used for training [8]–[10].

978-1-6654-5298-4/22/$31.00 ©2022 IEEE

The scope of this paper is to design a convolution neural network (CNN) to automatically discriminate incoming signals on the basis of the type of modulation they use. More specifically, we propose a modification of the CNNs proposed in [9] and [11], using a low number of convolution layers with a reduced number of filters to contain the complexity of the network that is compliant with SDR applications. Moreover, we introduce some Batch normalization to increase learning rate and reduce training time. Results, shown in terms of overall accuracy and of confusion matrices, confirm the effectiveness of the devised architecture.

II. Background

This section describes two CNNs (referred as Baseline and CNN2) specifically designed to solve the problem of modulation classification of the incoming signals. Then, these two CNNs are used as starting point for the development of the proposed architecture.

A. Baseline CNN

The Baseline is a CNN architecture ad-hoc designed in [9] for the RadioML2018.01a dataset, whose block scheme is shown in Fig. 1.

The Baseline neural network architecture accepts as input a matrix of size 2 × 128 representing the complex (real and imaginary part) baseband signals (more details about the signals structure are given in Section IV). Moreover, it returns as output a one-dimensional vector of C elements whose size is equal to to the number of labels that appear in the dataset. The number of elements of the output vector is determined by the fact that, in the training phase, there is a conversion of the labels in the dataset using a one-hot encoding. Specifically, the

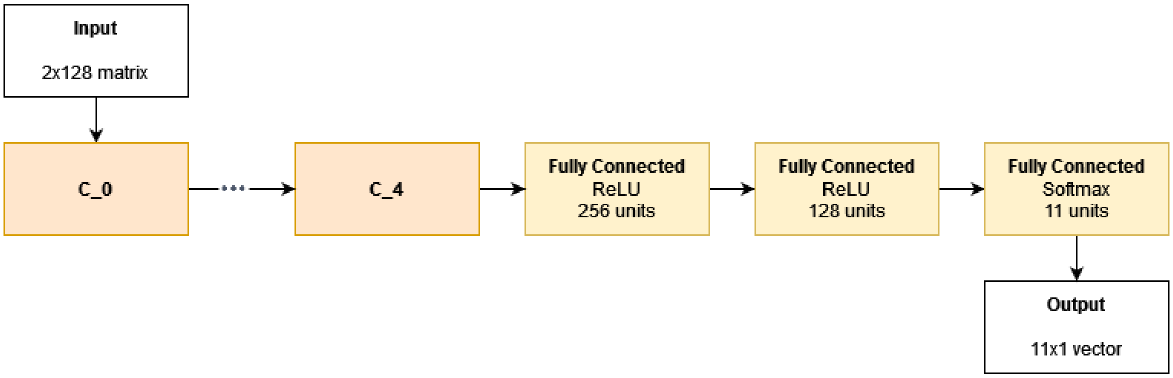


Figure 1. Baseline neural network architecture [9].

output vector $\hat{\boldsymbol{y}}$ of a neural network trained with a re-labeled dataset one-hot is represented in the equation

$$\hat{\boldsymbol{y}} = [\hat{y}_1, \dots, \hat{y}_C] \,, \tag{1}$$

where C is the number of input classes. From the vector $\hat{\boldsymbol{y}}$ it is extracted the position index of the highest valued element as

$$i_{\hat{\boldsymbol{y}}} = \arg\max\left(\boldsymbol{y}\right). \tag{2}$$

Then, a vector $\hat{\boldsymbol{y}}'$ is constructed with all null-elements except the one at position $i_{\hat{\boldsymbol{y}}}$. As a consequence, the class assigned by the neural network to its input is the one associated with the vector $\hat{\boldsymbol{y}}'$, which coincides with one of the one-hot encoded vectors. Baseline neural network architecture is composed of five layer blocks and three Fully Connected layers. The five layer blocks are equal each other and are composed of two one-dimensional convolution layer with the ReLU activation function and are followed by a max pooling layer. The convolution layers are composed of 128 filters of size 7×1 and 5×1, respectively. The output of these five blocks of sequential layers is given as input to the three Fully Connected blocks of 256, 128, and 11 units, respectively. The activation functions of the Fully Connected are again ReLU, except for the last one which uses a softmax function instead. Details on the architecture are given in Fig. 2 with respect to one of the five layers repeated in a cascade before the three Fully Connected layers.

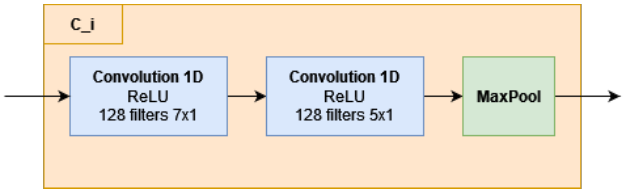


Figure 2. Group of layers in cascade of Baseline neural network architecture that repeats five times [9].

B. CNN2

The CNN2 architecture designed in [11] for the RadioML2016.10a dataset is illustrated in Fig. 3.

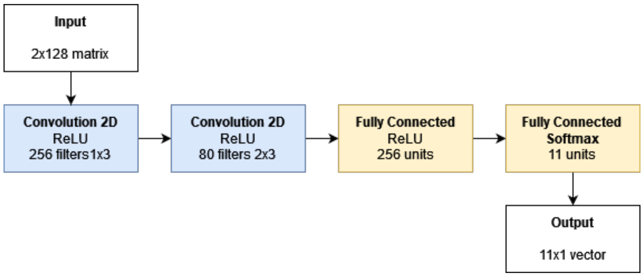


Figure 3. CNN2 architecture [11].

This architecture also starts from the complex baseband signals in input of size 2×128 and uses two two-dimensional (2-D) convolution layers with a high number of filters (viz., 256 filters of size 1×3 and 80 filters of size 2×3). Finally, the convolution layers are followed by two Fully connected layers with 256 and 11 units, respectively.

III. Proposed FreeHand CNN

In this section, the proposed CNN (indicated in the following as FreeHand) for automatic recognition of signal modulations is described in details. This CNN evolves from both the Baseline and CNN2 described in Section II, and is characterized by having a reduced number of convolution layers sharing a low number of filters. Therefore, it is capable of maintaining a reduced degree of complexity that is compliant with the requirements of SDR devices. Its overall architecture is graphically represented in Fig. 4.

As before, the starting input is a matrix of size 2×128 representing the complex baseband signals and the output is still a vector of C elements. Therefore, the considered architecture adapts to a two-dimensional input, thus being a two-dimensional max pooling layer. The FreeHand network employs a normalization of the values between some layers. The considered normalization is called Batch Normalization, that is introduced to considerably reduce the training time (since it increases the learning rate) [12] and could have a positive impact also on performance. Then, the FreeHand neural network architecture employs two-dimensional convolution layers that are separated by a Fully Connected layer, which

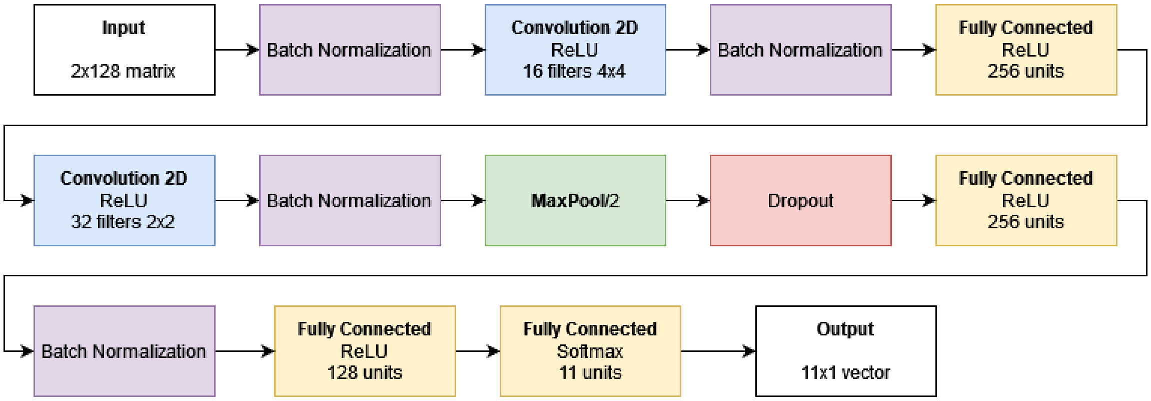


Figure 4. Architecture of the FreeHand neural network for signal modulations classification.

aims to increase the number of trainable parameters between the two layers. For these layers, the ReLU (Rectified Linear Unit), defined as

$$f(x) = \max(0, x), \tag{3}$$

is used as activation function for its properties, like the absence of the vanishing gradient problem.

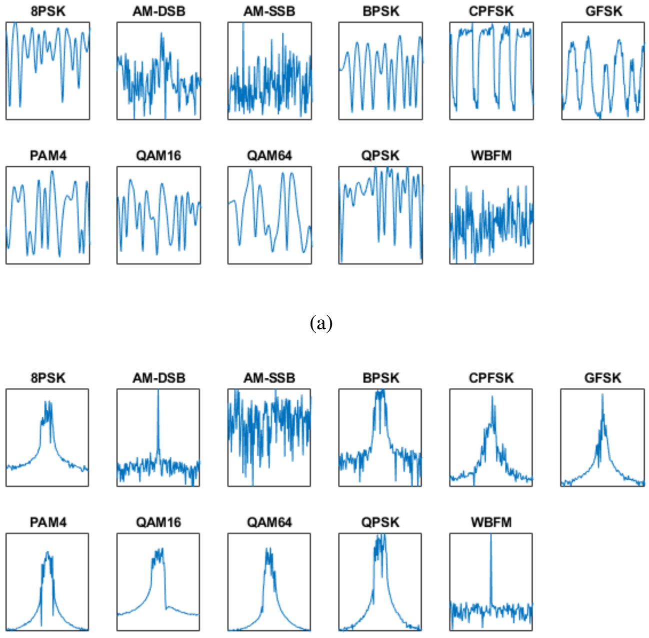
After this chain, in the architecture of the FreeHand neural network, a two-dimensional max pooling layer is used. At the output of the two-dimensional max pooling, the dropout technique is applied; it consists in randomly deactivate a prefixed number (50%) of units belonging to the layer at each epoch of training. Then, A chain of three Fully Connected layers relates the output of the max pooling layer with the output of the neural network. For these layers, the activation function is a ReLU for the first two and a softmax for the last one. Softmax is used to contain the values of the output elements of the network in the range $[0, 1]$, with their sum equal to 1, in order to making them compliant with the values of the probability to belong to a class. More formally, the softmax function, say S, is defined as

$$S_i = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}}, \; i = 1, \ldots, C \tag{4}$$

where x_i is the i-th value of the vector $\boldsymbol{x}$ of size C.

IV. Dataset description and results

The aim of this section is to evaluate the performance of the proposed architecture described in Section III. The dataset used to train and test the proposed CNN is referred to as RadioML2016.10a dataset [13]. In particular, it consists of 8 digital and 3 analog widely used modulations, viz. BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, and PAM4 for digital modulations, and WB-FM, AM-SSB, and AM-DSB for analog modulations. Hence, a total of 11 classes is available. The signal composing the dataset are considered in the complex baseband representation, hence the data are stored as $2 \times N$-dimentional matrices, where the size 2 represents the in-phase (I) and quadrature (Q) components, whereas the size $N = 128$ is the time dimension (or samples). Before analyzing the results of the tests conducted on the RadioML2016.10a dataset as well as to better understand differences and similarities between classes, Fig. 5 shows an example for each class of the complex signals involved in the dataset. Specifically, in subplot a) the modulus of the signal can be observed in the time domain, whereas in subplot b) their corresponding power spectrum is depicted.



(a)

(b)

Figure 5. Examples of signals for each class involved in the dataset RadioML2016.10a dataset [13]. Subplots refer to a) signal modulus, and b) power spectrum.

A. Training and results

The analyses are conducted dividing the data into two sets, viz. training and validation. The considered division is done using the 75% of the dataset for the training, whose 10% is used for validation, and the remainder 25% is confined to test. By doing so, the training set results to be composed of 165000 datapoints (hence, 16500 for the validation set), while the test set comprises 55000 datapoints. Each of the above sets is further subdivided into batches whose size is set equal to 1024. Then, the neural network is trained for all the batches for a total number of 100 epoch and the number of step for each epoch is equal to 146 that is the number of data in the training set divided by the size of each batch. However, the training stops if the value of the loss function, calculated on the set of validation, does not improve within 5 epochs. As in [11], the considered loss function is the cross-entropy and the optimizer (i.e., the iterative algorithm used to find the parameters that minimize the loss function) is the Adam optimizer [14]. Finally, results are analyzed utilizing as performance metrics the overall accuracy expressed in percentage and the confusion matrices.

Fig. 6 shows the loss functions versus the number of epochs for both the training and validation phase of FreeHand. The curves clearly highlight the rapid convergence of the FreeHand architecture in both its training and validation phase.

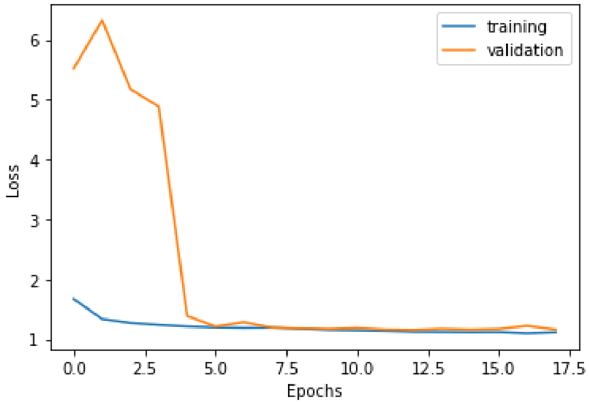


Figure 6. Loss and validation function versus epochs for the FreeHand architecture.

To have further insights about the effectiveness of the FreeHand, in Fig. 7 its confusion matrices are reported and compared with those of the CNN2 architectures. In particular, they are computed considering three different situations in terms of used data: a) entire dataset independently of the SNR, b) data at low SNR (i.e., 0 dB), and c) data at high SNR (i.e., 18 dB). Results highlight the effectiveness of the proposed FreeHand architecture with a confusion matrix having values mostly concentrated around its diagonal. However, the confusion matrices for both the FreeHand and CNN2 appear to be more dispersed around their diagonal when the entire dataset is used, that represent the most challenging situation. However, the FreeHand architecture is capable of ensuring satisfactorily performances both in the low and high SNR regimes, overcoming the CNN2 in all the considered scenarios.

Finally, to have a quantitative comparison about the two architectures, in Fig. 8 the overall accuracy is reported versus the SNR. As the curves show, FreeHand is capable of ensuring better performances than CNN2, reaching the 89% of overall accuracy.

B. Data Augmentation and results

Applying data augmentation techniques described in [15], FreeHand neural network slightly increases the accuracy. With data augmentation, the dataset is enlarged creating altered copies of the datapoints. In particular, the rotation data augmentation is the best performing pre-processing in this work. It consists in creating four copies for each datapoint rotated by an angle $\theta = \{0, \pi/2, \pi, 3\pi/2\}$, respectively, by means of the following equation:

$$\begin{bmatrix} I' \\ Q' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} I \\ Q \end{bmatrix}. \tag{5}$$

Each datapoint copy is then added to the dataset before the train/test split. Applying the rotation data augmentation, FreeHand neural network reaches 92% accuracy on RadioML2016.10a dataset.

V. Concluding Remarks

In this paper, the problem of automatic recognition of signals modulations has been addressed. In particular, a new CNN architecture has been designed starting from two already existing architectures that were specifically designed to solve this problem. The main characteristics of the proposed CNN is that has a low number of convolution and fully connected layers, each sharing a very low number of filters/units. The architecture has been validated on the challenging RadioML2016.10a dataset characterized by having both analog and digital modulations. Results obtained in terms of confusion matrix and overall accuracy have shown the effectiveness of the proposed CNN also in comparison with its counterpart. The repository [16] contains the implementation of the content described in this paper.

Acknowledgment

The authors would like to acknowledge Deepsig for providing the RadioML2016.10a dataset. The dataset is available at https://www.deepsig.ai/datasets.

References

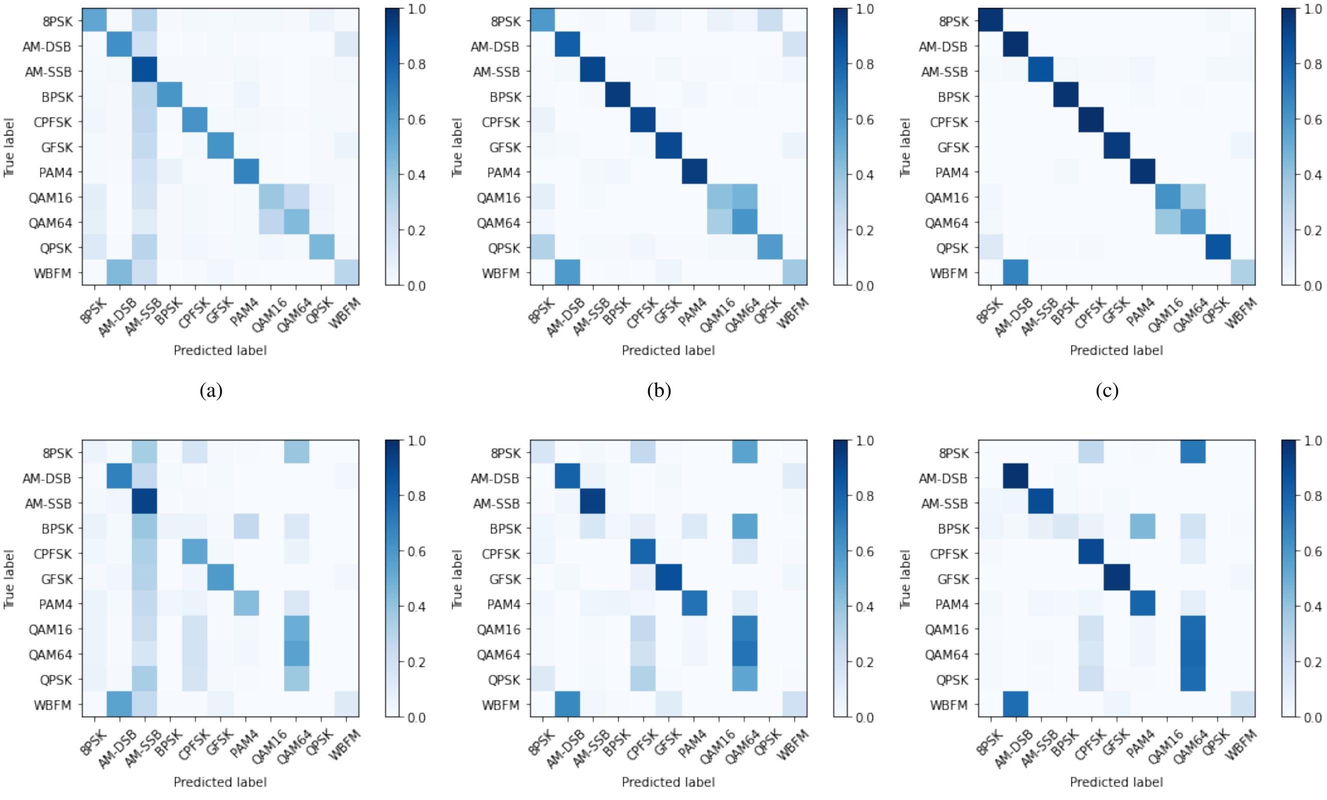
[1] O. A Dobre, M. Oner, S. Rajan, and R. Inkol, “Cyclostationarity-based Robust Algorithms for QAM Signal Identification,” *IEEE Communications Letters*, vol. 16, no. 1, pp. 12–15, 2011.

[2] O. A Dobre, Y. Bar-Ness, and W. Su, “Higher-Order Cyclic Cumulants for High Order Modulation Classification,” in *IEEE Military Communications Conference, 2003. MILCOM 2003.* IEEE, 2003, vol. 1, pp. 112–117.

[3] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, “Survey of Automatic Modulation Classification Techniques: Classical Approaches and new Trends,” *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[4] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, “Automatic Modulation Recognition of Digital Signals using Wavelet Features and SVM,” in *2008 10th International Conference on Advanced Communication Technology*. IEEE, 2008, vol. 1, pp. 387–390.

[5] K. Hassan, I. Dayoub, W. Hamouda, and M. Berbineau, “Automatic Modulation Recognition using Wavelet Transform and Neural Networks in Wireless Systems,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–13, 2010.



(a)

(b)

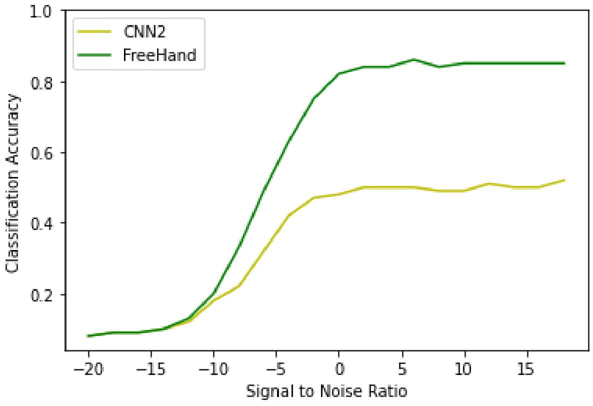
(c)

(d)

(e)

(f)

Figure 7. Confusion matrix on the RadioML2016.10a dataset. Figures on the top refer to the proposed FreeHand architecture, whereas figure on bottom to the CNN2. Subplots from left to right refer to the entire dataset, signals at SNR = 0 dB, and signals at SNR = 18 dB, respectively.



(a)

Figure 8. Accuracy versus SNR. Subplots refer to a) FreeHand, and b) CNN2.

[6] T. R. Kishore, D. S. Sidharth, and K. D. Rao, "Analysis of Linear and Non-Linear Frequency Modulated Signals using STFT and Hough Transform," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2015, pp. 490–494.

[7] F. Benedetto, A. Tedeschi, and G. Giunta, "Automatic Blind Modulation Recognition of Analog and Digital Signals in Cognitive Radios," in *2016 IEEE 84th vehicular technology conference (VTC-Fall)*. IEEE, 2016, pp. 1–5.

[8] Juan Zhang, Yong Li, and Junping Yin, "Modulation Classification Method for Frequency Modulation Signals based on the Time–Frequency Distribution and CNN," *IET Radar, Sonar & Navigation*, vol. 12, no. 2, pp. 244–249, 2018.

[9] N. Soltani, K. Sankhe, S. Ioannidis, D. Jaisinghani, and K. Chowdhury, "Spectrum Awareness at the Edge: Modulation Classification using Smartphones," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2019, pp. 1–10.

[10] Yuan Zeng, Meng Zhang, Fei Han, Yi Gong, and Jin Zhang, "Spectrum Analysis and Convolutional Neural Network for Automatic Modulation Recognition," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 929–932, 2019.

[11] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional Radio Modulation Recognition Networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2016, pp. 213–226.

[12] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[13] Timothy J. O'Shea and Nathan West, "Radio Machine Learning Dataset Generation with GNU Radio," in *Proceedings of the GNU Radio Conference*, 2016, vol. 1.

[14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2015.

[15] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data Augmentation for Deep Learning-based Radio Modulation Classification," 2019.

[16] "Project repository," https://github.com/thanh-guong/master-thesis.