Radio Modulation Classification Using Deep Residual Neural Networks

Adeeb Abbas*, Vasil Pano*, Geoffrey Mainland†, and Kapil Dandekar*

*Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA
E-mail: adeeb.abbas@drexel.edu, vasilpano@gmail.com, dandekar@drexel.edu
†College of Computing and Informatics, Drexel University, Philadelphia, PA 19104, USA
E-mail: mainland@drexel.edu

***Abstract*—We propose a new deep residual network for Automatic Modulation Classification, OPResNet-18. It achieves state-of-the-art accuracy on the RadioML 2016.10a data set. We train the proposed model and other state-of-the-art networks with augmented data by adding a Carrier Frequency Offset (CFO). We find that the previously proposed IQNet-3 is robust to CFO. We demonstrate that this robustness allows the performance of IQNet-3 to be further improved through data augmentation in contrast to existing neural networks that cannot handle CFO. Finally, we provide evidence that standard data pre-processing techniques for time-domain data that reportedly perform well in many domains do not perform as well as a simple alternative, the outer product, in the IQ domain.**

***Index Terms*—machine learning, convolution networks, deep learning, modulation recognition, radio frequency**

I. Introduction

Radio frequency (RF) modulation research has garnered a great deal of interest within the machine learning community. In radio communication, modulation can be defined as the process of varying properties of the carrier signal to convey information. This work is primarily focused on machine learning to classify analog and digital modulation techniques. Analog modulation techniques are used for transmitting an analog baseband signal such as an audio signal, while digital modulation is used to transmit a digital bit stream over an analog communication channel.

Modulation recognition has applications in cognitive radio, link-adaptive communications, and electronic warfare. This field has been spurred on by recent advances in machine learning and flexible software defined radio implementations. This work explores the performance of Deep Residual Networks with complex-valued time series data (raw IQ data) for modulation recognition. We explore the effects of data augmentation on these deep residual networks. In order to validate the performance improvement of these networks, the proposed method is compared to the state-of-the-art deep neural networks developed for Automatic Modulation Classification (AMC) [1, 2]. The main contributions of this paper are as follows:

- We propose two deep residual networks, IQNet-5 and OPResNet-18. OPResnet-18 outperforms state-of-the-art networks proposed in the literature, with up to 10% improvement from the current networks at high SNRs (> -5 dB).
- We apply the time series data transformations described by Wang and Oates [3] for non-radio applications to the radio IQ domain and find that they do no better at modulation recognition than a simpler transformation, the outer product.
- We augment training data by applying a Carrier Frequency Offset and show that training on the augmented data improves the performance of IQNet-3. The performance of all the other neural networks, including the proposed OPResNet-18, deteriorates when trained on this augmented data, indicating that they are not robust to CFO.

Background on current techniques and related works is presented in Section II. The proposed methodology is detailed in Section III. Results and performance comparison are shown in Section IV. Finally, the conclusions of this work are presented in Section V.

II. Related Work

In recent years, AMC has been an area of interest among machine learning researchers [4]. There are two types of AMC: Likelihood Based (LB), and Feature Based (FB). LB classification suffers from high computation requirements, while FB classification works through extracted features and therefore is considerably more efficient [5]. AMC involves three steps: i) Preprocessing of the received modulated signal, ii) feature extraction, and iii) classification of extracted features. Another method of performing these steps is a wavelet transform and using a Support Vector Machine (SVM) or decision tree to predict the modulation from those features [6].

Many researchers have utilized the aptly-named *RadioML* family of datasets to obtain results [2]. Within this family of GNURadio-generated datasets, the RadioML 2016.10a dataset is the most popular in the literature. Convolutional neural network (CNN) models such as AlexNet, GoogLeNet, and different versions of Resnet have been used previously to classify various modulations [7, 8]. CNNs are excellent feature extractors, and have previously exhibited remarkable performance on image classification problems [9, 10]. In addition to the convolutional models, Long Short-term Memory (LSTM) architectures have been considered for AMC [11]. Therein, the authors exploit the ability of LSTMs to learn arbitrary-length sequences [12]. Specifically, their most notable results are gathered with a two-layer LSTM. They convert I/Q data of each RadioML 2016.10a signal to its respective amplitude-phase representation, where the amplitude vectors are L2-normalized and the phase vectors are normalized between -1 and $+1$, and report a classification accuracy of approximately 90% on data with a signal-to-noise ratio (SNR) of 0 dB and above. Also, in an effort to exploit the best qualities from each type of aforementioned architecture, the Convolutional Long Short-term Deep Neural Network (CLDNN) architecture was developed [13]. In AMC literature, it has also achieved competitive results [1, 14, 15].

Techniques for imaging time series have been used in a variety of fields, such as Human Activity Recognition (HAR) classification [16] and electroencephalography (EEG) classification [17]. Furthermore, the use of spectrogram images for AMC is explored in [18]. Using MATLAB, the authors transform all I/Q signals in the RadioML 2016.10a dataset to their respective spectrogram representations and export them as $100 \times 100 \times 3$ PNG images. The CNN architecture was trained separately with Gaussian-filtered and non-Gaussian-filtered images. The classifier achieves approximately 90% accuracy on data with SNRs of 6 dB and above. Additionally, the performance of the Gaussian-filtered-spectrogram-image classifier is compared to that of two other classifiers: one that takes the spectral correlation function (SCF) representation of each signal as input, and another that takes the ambiguity function (AF) representation as input. The spectrogram image classifier outperforms the other two significantly. Recognizing that data representation seems to be equally as important as the deep architecture used (as evidenced in [18, 19]), this paper explores the performance of time series data transformation techniques on raw I/Q data. Performance of residual networks and other architectures with data augmentation is also analyzed.

III. Methodology

In this section, an overview of the model and data augmentation process is provided.

A. The Deep Residual Network

Deep Residual Networks are popular in the world of image classification due to their ability to overcome the accuracy degradation problem. We use a ResNet-18 which has 5 residual stacks. We also compare two additional residual networks: a network proposed by Ramjee et al. [1] with 3 residual stacks, which we term IQNet-3, and a version with 5 residual stacks (IQNet-5). The architecture of IQNet-3 and IQNet-5 is shown in Table I and Table II respectively.

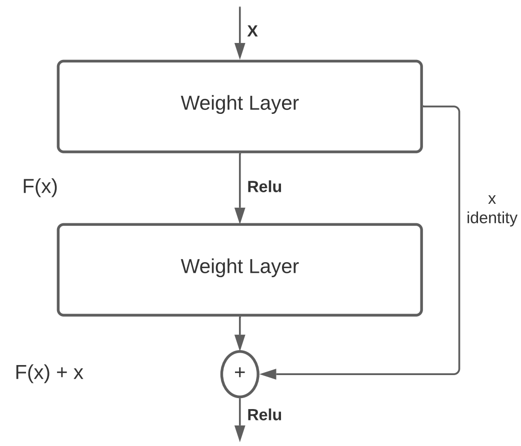

Figure 1: A residual connection block as described in [20] that is used in the proposed networks.

Table I: IQNet-3

Layer	Output Dimensions
Input Layer	(2, 128)
Upsampling Layer	(256, 256, 3)
Residual Stack 1	(32, 64)
Residual Stack 2	(32, 32)
Residual Stack 3	(32, 16)
FC/Relu	256
FC/Softmax	11

Residual Network as proposed by Ramjee et al. [1]

A pre-trained ResNet-18 model [20] is implemented. In both Residual Networks (IQNet-5 and ResNet-18), a dropout of 50% is used. Dropout, as explained in [21], randomly selects neurons in a neural network and removing them during training. Random removals of neurons (i.e. "dropout") help excessive fine-tuning and prevent overfitting. Random removal of neurons means that at any point in the training phase only a portion of the neural network is trained. Dropout regularization effectively produces an ensemble learning effect where training of these sub-networks is comparable to using multiple weaker/ineffective algorithms combining them to produce a more robust and powerful algorithm which is better than the individual sub-network/algorithm.

B. Data Augmentation

Training these deep networks requires a massive amount of varied data and a good representation of the environment the model will be tested in. Data augmentation is widely used in image classification to increase the robustness of the machine learning models. It helps increase the number of training samples, reduce overfitting, and improve generalization capability. Unlike for image classification, the effects of data augmentation on deep residual networks for AMC have not yet been studied.

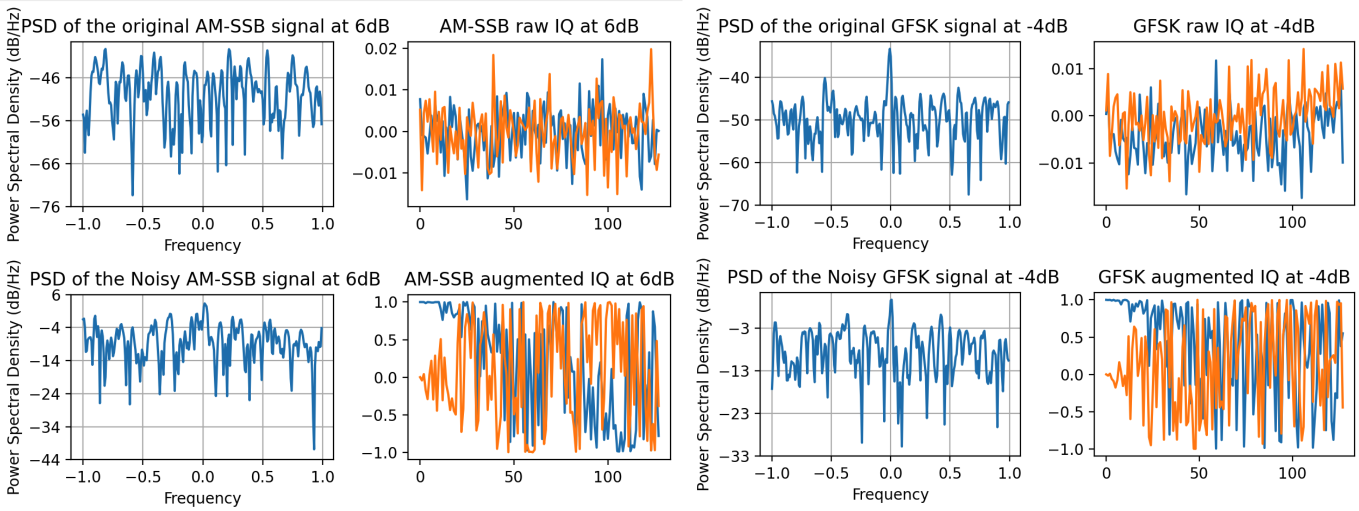

Figure 2: Before and after data augmentation

Table II: IQNet-5 Network Architecture

Layer	Output Dimensions
Input Layer	(2, 128)
Upsampling Layer	(256, 256, 3)
Residual Stack 1	(32, 64)
Residual Stack 2	(32, 32)
Residual Stack 3	(32, 16)
*Residual Stack 4	(32, 8)
*Residual Stack 5	(32, 4)
FC/Relu	256
FC/Relu	256
FC/Softmax	11

*residual stacks added

The raw I/Q samples of the data are augmented by adding a Carrier Frequency Offset (CFO). Adding a CFO represents one of the common impairments in a wireless channel. CFO often occurs because of two major factors: i) frequency mismatch between the transmitter and receiver oscillators, ii) any relative motion between the transmitter and the receiver (Doppler effect).

Since an I/Q data point in the 2016a dataset can be represented as a stack of two arrays, one for each I and Q, it can be expressed as

$$x_k^{IQ} = \begin{bmatrix} x_i^k \\ x_q^k \end{bmatrix}, \tag{1}$$

where x_i and x_q represent the I and Q arrays respectively. After applying Euler's formula, the real component after adding the CFO for an x_k, where $k \in C^N$ and represents a data vector in the dataset, is given as, as shown in [22],

$$x_i^{k'} = x_i^k cos(f) - x_q^k sin(f), \tag{2}$$

and the quadrature phase component as

$$x_i^{k'} = x_i^k sin(f) + x_q^k cos(f). \tag{3}$$

Therefore, the augmented data vector would be given as

$$x_{k'}^{IQ} = \begin{bmatrix} x_i^{k'} \\ x_q^{k'} \end{bmatrix}, \tag{4}$$

where $x^{k'}$ represents an augmented data vector and f is the normalized frequency that is taken from $\mathcal{N}(\mu = 0, \sigma^2 = 0.01)$. An example of before and after augmenting the raw IQ signal modulated with AM-SSB is shown in Figure 2.

For the training split, 50% of the 2016a dataset is used which would amount to approximately 110,000 training samples. Later, for data augmentation, a CFO drawn from the distribution $\mathcal{N}(\mu = 0, \sigma^2 = 0.01)$ is added, to create a total of 220,000 samples.

C. Data Input Pipeline

For the IQNet-5, the unadulterated raw I/Q data is fed to the neural network in batches of (2, 128) tensors, as shown in Table II. In contrast, for the ResNet-18 model, three channeled combinations of Outer Product (OP), Gramian Angular Summation Field (GASF), Gramian Angular Difference Field (GADF), and Markov Transition Field (MTF) [3] are created using the Pyts library [23], and the resultant tensors are of shape (128, 128, 3). We refer to the resulting combination of pipelines and deep residual network in this paper as OPResNet-18, GASFResNet-18, GADFResNet-18, and MTFResNet-18, respectively.

The data input pipeline for Resnet-18, IQNet-3, and IQNet-5 is shown in Figure 3.

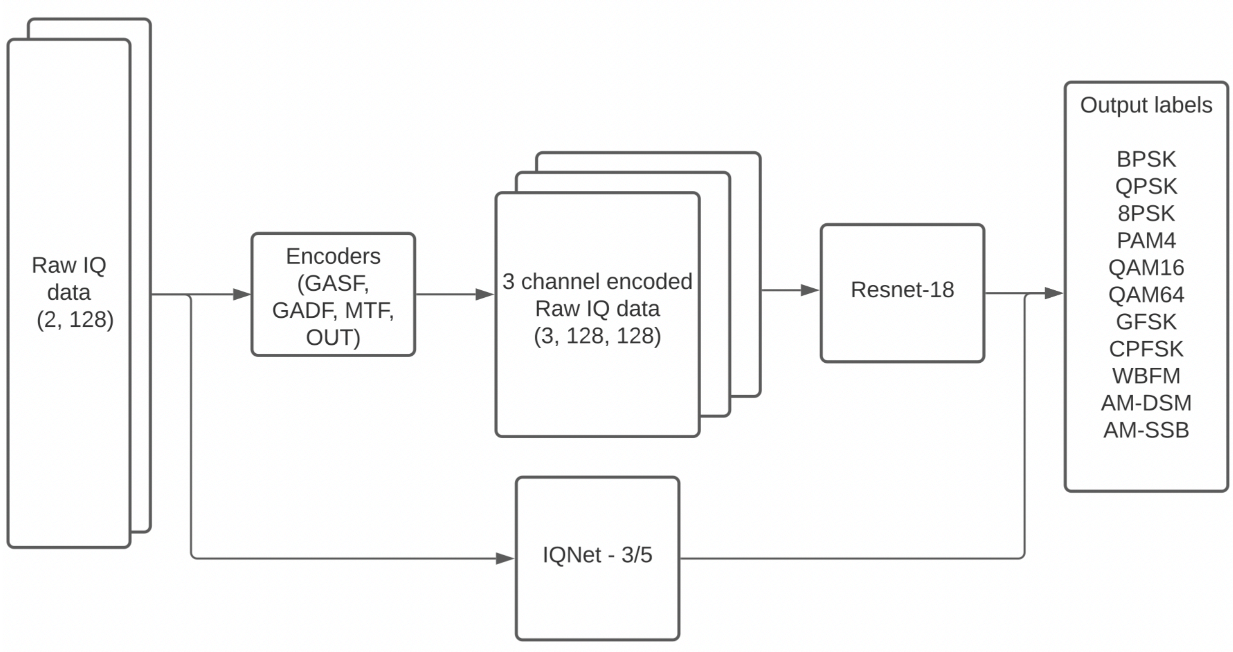

Figure 3: The data input pipeline for the Resnet-18 and IQNet-3/5

D. Setup

To evaluate the data transformation techniques and analyze other state of the art models the following setup is used. The main compute is driven by a 48 core Intel Xeon CPU E5-2650, an Nvidia RTX 2080ti, and 128 gigabytes of DDR4 memory. For software, Ubuntu 20.04, Python 3.8, and Tensorflow/Keras 2.4 are used. Code for the augemented dataset generation, training and inference is available in the Drexel Wireless RadioML Github Repository, https://github.com/drexelwireless/RadioML.

IV. Sensing Performance Analysis

In this section, the performance of the proposed networks with and without different data transformations are analyzed and compared with other state-of-the-art models on the RadioML 2016.10a dataset. Half of the data is used for training, and the other half is used for testing the performance of the model. The models are trained and tested three times with different seeds to ensure confidence in the results. Each of the following figures contain the average classification accuracy per SNR for each model. The shaded regions surrounding the curves represent the min/max of classification accuracy at each tested SNR value.

A. Classification Performance by Transform

We first compare the accuracy of the 4 data transformations on the RadioML 2016.10a dataset. The accuracy at each SNR on the test split of the data is shown in Figure 4. For a baseline comparison, the classification accuracy of the IQNet-5 is also included.

It can be seen that OPResNet-18 performs better (up to 10% better at SNR ≥ -10 dB) than the other transformations, and all of the ResNet-18 models perform up to 20% better than the IQNet-5 (especially at SNR $\geq$ 15dB) despite both models having the same number of residual stacks. The superior accuracy can be attributed to the increased degrees of freedom, which enables the deeper residual network (ResNet-18) to learn patterns that would be harder to recognize with a smaller number of trainable parameters. It appears that OPResNet-18 outperforms other ResNet-18 networks because the outer product utilizes the phase information in the IQ data as well. All the other data transformations (GASF, GADF, and MTF) do not use the phase information, which is disposed of during the transformation described in [3].

B. Comparison of the Classification Performance with other Models

The performance of the OPResNet-18 network is compared with the CLDNN, LSTM, IQNet-3, IQNet-5, and the VTCNN2 models [1, 24] (see Figure 5). It can be seen that the OPResNet-18 performs up to 10% better at higher SNR (>0dB) and up to 3-7% better on lower SNR (≥ -8 and $\leq$ 0dB). Figure 5 shows the performance comparison. As seen in Table 5, the OPResNet-18, relative to the other networks in this paper, has a lot more trainable parameters giving it many degrees of freedom. Consequently, the architecture can model highly complex functions, and thus is better equipped to differentiate

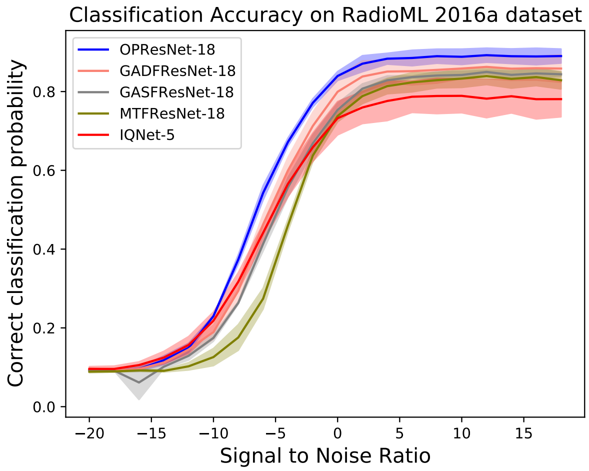

Figure 4: Comparing the different transformations of ResNet-18 and Modified ResNet - Accuracy vs SNR on the RadioML 2016.10a dataset

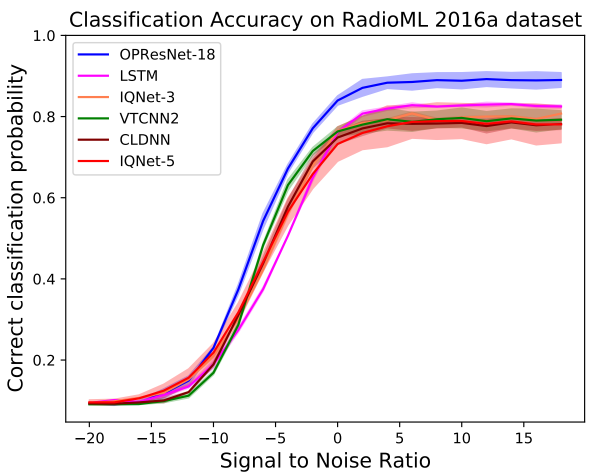

Figure 5: Accuracy vs SNR on the RadioML 2016.10a dataset - OPResNet-18, LSTM, CLDNN, IQNet-3, IQNet-5, and VTCNN2)

between highly complex patterns.

C. Classification Performance with Data Augmentation

When trained with augmented data, OPResNet-18 and all the other networks we compare against (i.e., CLDNN, LSTM, and VTCNN2) except for IQNet-3 and IQNet-5, deteriorate in classification accuracy (See Figures 6 and 7). The performance deterioration could be because it has far more free trainable parameters as compared to the other networks that we compare against (see Table III). The performance of IQNet-3 and IQNet-5 get better with data augmentation. IQNet-5 performance variance decreases and the network performs better than what it did without data augmentation by up to 5% at higher SNRs.

Table III: Model Complexity

Model Name	Trainable parameters
ResNet-18	11,313,614
IQNet-5	144,283
IQNet-3	149,867
VTCNN2	2,830,427
CLDNN	2,741,379
LSTM	378,891

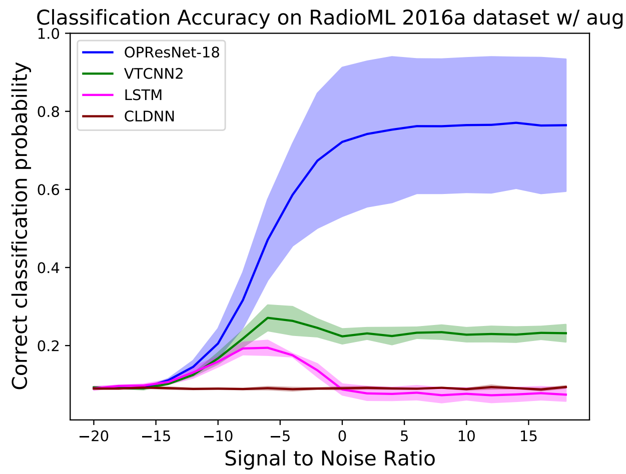

Figure 6: OPResNet-18, LSTM, CLDNN, VTCNN2 w/ data augmentation - Accuracy vs SNR on the RadioML 2016.10a dataset.

Both the IQNets perform similarly, which shows that adding more layers is not always helpful as shown by Ramjee et al. [1] except for non-augmented data.

V. Conclusion

The performance of deep residual networks trained with raw I/Q data is compared with that of the same networks trained with both transformed and augmented I/Q data. IQNet-5, which has 5 residual stacks, is created from [1] for comparison with the proposed ResNet-18 (see Section III). This is done in order to show that it is not just having more residual stacks that are responsible for the performance gains, but also the increased degrees of freedom. Two deep residual networks (ResNet-18 and IQNet-5) of the same depth are evaluated with and without the data transformations to see the effectiveness of such transformations. It is shown that OPResNet-18 performs better than any of the traditional time-series transformations. This could be because the Outer Product utilizes the phase information as well while all the other transformations end up disposing this vital information about the IQ signal data. OPResNet-18 performs up to 10% better at higher SNRs (>0dB) and up to 3-7% better on lower SNRs (≥ -8 and $\leq$ 0dBs) as compared to other state of the art networks in Figure 5.

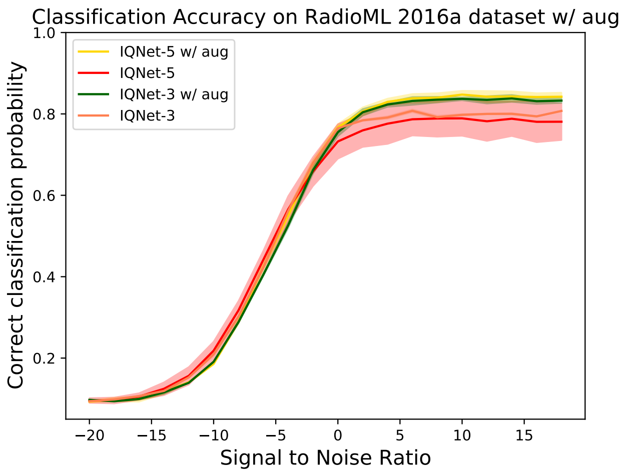

Figure 7: IQNets 3 and 5 w/ and w/o data augmentation - Accuracy vs SNR on the RadioML 2016.10a dataset.

We next augmented training data using a CFO from a normal distribution. This augmentation doubled the size of the training split and allowed us to observe the robustness of these different types of networks to CFO. It is found that only IQNet-3 maintained consistent performance with and without data augmentation. With 149,867 trainable parameters, it has just the right degrees of freedom to be able to generalize well.

For future work, deeper residual networks and different types of data transformation methods should be explored. Networks that are robust to CFO, and other techniques for dataset creation need to be studied further as well.

Acknowledgment

This research is supported by the National Science Foundation under grants CCF-1717088, CNS-1816387, and CNS-1730140. Any opinion, findings, and conclusion or recommendations expressed in this materials are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El Gamal, and Y. C. Eldar, "Fast Deep Learning for Automatic Modulation Classification," *arXiv e-prints*, p. arXiv:1901.05850, Jan. 2019.

[2] T. J. O'Shea and N. West, "Radio machine learning dataset generation with GNU Radio," *Proceedings of the GNU Radio Conference*, vol. 1, Sept. 2016.

[3] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, p. 3939–3945, AAAI Press, 2015.

[4] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, p. 67366–67376, 2020.

[5] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion Methods for CNN-Based Automatic Modulation Classification," *IEEE Access*, vol. 7, pp. 66496–66504, 2019.

[6] L. Qian and C. Zhu, "Modulation classification based on cyclic spectral features and neural network," in *2010 3rd International Congress on Image and Signal Processing*, vol. 8, Oct. 2010.

[7] S. Ujan, N. Navidi, and R. J. Landry, "An efficient radio frequency interference (RFI) recognition and characterization using end-to-end transfer learning," *Applied Sciences*, vol. 10, no. 19, p. 6885, 2020.

[8] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 168–179, Feb 2018.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," vol. 60, no. 6, pp. 84–90.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*.

[11] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," vol. 4, no. 3, pp. 433–445.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," vol. 9, no. 8, pp. 1735–1780.

[13] T. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," IEEE.

[14] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017.

[15] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 915–919, 2017.

[16] H. Xu, J. Li, H. Yuan, Q. Liu, S. Fan, T. Li, and X. Sun, "Human activity recognition based on gramian angular field and deep convolutional neural network," *IEEE Access*, vol. 8, pp. 199393–199405, 2020.

[17] M. M. Islam and M. M. H. Shuvo, "Densenet based speech imagery EEG signal classification using Gramian angular field," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 149–154, Sep. 2019.

[18] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," vol. 8, no. 3, pp. 929–932.

[19] 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," vol. 6, pp. 18484–18501.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.
[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
[22] M. Jacovic, K. Juretus, N. Kandasamy, I. Savidis, and K. R. Dandekar, "Physical layer encryption for wireless OFDM communication systems," *Journal of Hardware and Systems Security*, vol. 4, pp. 230–245, July 2020.
[23] J. Faouzi and H. Janati, "pyts: A Python package for time series classification," *Journal of Machine Learning Research*, vol. 21, no. 46, pp. 1–6, 2020.
[24] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks* (C. Jayne and L. Iliadis, eds.), (Cham), pp. 213–226, Springer International Publishing, 2016.