AN OPEN-SOURCED TIME-FREQUENCY DOMAIN RF CLASSIFICATION FRAMEWORK

R. David Badger[1,2], Kristopher H. Jung[1], Minje Kim[1]
[1]Indiana University, Luddy School of Informatics, Computing, and Engineering, Bloomington, IN, USA 47408
[2]Naval Surface Warfare Center Crane, Crane, IN, USA, 47522
rdbadger@iu.edu, nhj940913@gmail.com, minje@indiana.edu

***Abstract*—In this paper we present a machine learning-based approach to solving the radio-frequency (RF) signal classification problem in a data-driven way. To this end, we propose an efficient and easy-to-use graphical user interface (GUI) for researchers to collect their own data to build a customized RF classification system. The GUI operates in the time-frequency (TF) domain, which is achieved by applying short-time Fourier transform to the in-phase and quadrature (IQ) time domain signals. Using the proposed GUI, a radio frequency (RF) dataset is collected from the ultra high frequency industrial, scientific, and medical (ISM) bands using commercial-off-the-shelf (COTS) transceivers, and COTS transceiver modules. We train three different variants of convolutional neural network models, such as VGG and ResNet, using the collected dataset and show that they can perform acceptable test-time classification (up to 95% accuracy) on unseen real-world RF signal recordings. Our experimental results also show that a carefully prepared TF domain without a loss of information can achieve better performance than a magnitude-only representation that loses phase information during the TF transformation. We open-source our project to provide the public with access to the labeled datasets, programming code, and the GUI software that can expedite the labeling process.**

***Index Terms*—Radio Frequency Machine Learning (RFML), Deep Neural Network (DNN), Software Defined Radio (SDR), electromagnetic spectrum (EMS)**

I. Introduction

Wireless technology delivers data seamlessly and securely via waveforms such as long-term evolution (LTE), and the fifth generation (5G) cellular networks. Further growth of the electromagnetic spectrum (EMS) is driven by new wireless devices, e.g., Internet of Things (IoT), wireless network protocols (Wi-Fi IEEE 802.11), and autonomous unmanned aerial vehicles (UAVs) that heavily leverage the spectrum to operate.

Worldwide growth in EMS access is leading to unprecedented spectrum congestion that most users are not aware of and do not directly control. Most transceivers are not cognitive of the spectrum in which they operate [1], [2], and co-located waveforms can degrade data throughput.

Therefore, efficient management of the EMS is important, and responsibility of governance varies by country. There are also global organizations, such as the international telecommunication union (ITU) within the United Nations (UN), that collectively promote international standards for spectrum access [3]. However, the majority of spectrum management consists of assigning spectrum users to specific bands of frequencies within the EMS or radio frequency (RF) spectrum.

For example, the unlicensed U.S. 2.4GHz industrial, scientific, and medical (ISM) band is only 83.5MHz wide, and it is shared with innumerable types of waveforms and commercial products such as Wi-Fi™, UAS controllers, Bluetooth®, and IoT devices, garage door openers, and even microwave ovens. This approach leads to many commercial products that deconflict with other signals by using methods such as frequency hopping and orthogonal signal coding that make transmitted data robust in the presence of interfering waveforms.

Cognitive transceivers operating in these traditional managed spectrum bands could classify RF traffic and dynamically re-tune to an optimal operating channel to deconflict. The transceivers themselves can aid in helping to ease spectrum conflict and promote band efficiency. The use of informatics and machine learning to help facilitate RF spectrum management is a relatively new field.

We propose a machine learning-based approach to enabling cognitive transceivers that dynamically adjust to RF channel conditions by recognizing the nature of the discovered signals in which they operate (e.g., waveform modulation). A convolutional neural network (CNN)-based RF signal classifier that operates in the time-frequency representations of the signals is presented. We will verify that some popular CNN architectures can achieve sensible classification performance; furthermore, we will also compare performance with methods that employ a slightly different approach.

There are primarily two sources of RF signal data for machine learning: spectrum sampling from commercial-off-the-shelf (COTS) transceivers via software defined radios (SDRs) [4] [5] [6] or software-generated synthetic datasets. The synthetically generated IQ data can either be augmented and used to train NN models or streamed to a SDR that transmits the data into the RF spectrum–either OTA or closed-loop–and then sampled by a receiver SDR into IQ data and stored. The latter approach can add additional augmentation to the IQ data that may not be easily artificially performed.

We open-source our project, which includes all the training and testing datasets and the source codes. Moreover, we also provide a graphical user interface (GUI) that researchers can easily utilize to build their own labeled datasets[1].

[1]https://saige.sice.indiana.edu/research-projects/rf-classification

II. Background

SDRs sample and quantize the analog RF spectrum data into in-phase and quadrature (IQ) time domain data. Similarly, synthetically derived data is generated and saved in IQ format or transmitted over-the-air (OTA), using SDRs, then saved at the receiver side via SDR. The IQ format is derived from the sinusoidal equation:

$$x(n) = \sin(2\pi f n + \phi(n)), \tag{1}$$

where f and n represent frequency and time, respectively, and $\phi(n)$ describes the time-varying phase. Trigonometry decomposes (1) into the orthogonal components:

$$\text{In-Phase} \quad i(n) = \sin(2\pi f n)\cos(\phi(n)) \tag{2}$$

$$\text{Quadrature} \quad q(n) = \sin\left(2\pi f n + \frac{\pi}{2}\right)\sin(\phi(n)) \tag{3}$$

Before the IQ data is used in a machine learning model, such as a CNN, the data domain should be considered. Classification of IQ data can be accomplished in multiple domains, and it is useful to consider the alternatives.

Time Domain: IQ data is natively time domain during capture, and no domain change is necessary. Time domain IQ data does not allow expert features to be easily extracted, which is a significant benefit when it comes to heuristic features. However, the sampled IQ training data must be clear of other signals, for which an end-to-end neural network can perform feature learning and classification holistically at the cost of learning features only from the data. Meanwhile, the sampled bandwidth will also fix the classification bandwidth. The center operating frequency of the SDR would be the only insight into what frequency the classified waveforms are occupying, without further processing. Motivating examples of time domain classification is demonstrated in [7]–[10].

Frequency Domain: To convert the IQ data to the frequency domain requires a discrete Fourier transform (DFT) or short-time Fourier transform (STFT) [11] to capture the temporal dynamics of the signal over time. The magnitude of the IQ spectrogram is often used in frequency domain classifiers for training [12], [13]. This method works well with many waveforms, but the magnitude of the STFT will forfeit any phase information in the waveform that may be useful for classification. We propose to use a 3D tensor representation to encompass all the details of the IQ data in the time-frequency space as an input to our CNN classifiers (Section IV).

Other Domains: Other domains are also used for RF classification. For example, image processing on the IQ data from plotting the I vs. Q data [14]. Waveforms that have clear constellation signatures [15] are good candidates for this domain (e.g., quadrature phase shift keying (QPSK), quadrature amplitude modulation (QAM)).

Additionally, The Wigner-Ville, Choi-Williams, Quadrature Mirror Filter Bank (QMFB), and cyclostationary domains are widely used to classify low probability of intercept (LPI), low probability of detect (LPD) radar waveforms [16].

STFT time-frequency (TF) domain: Out of the potential choices, we propose an STFT-based TF representation, or a *spectrogram*, that fully retains real and imaginary components of IQ signals after the transformation. Given that the TF representation will be in the shape of an image with four channels, it is natural for us to employ 2D CNN models for the classification problem. However, in order to deal with the issue of the large spectrogram size and the relative sparsity, we additionally propose a patching mechanism that processes smaller regions of the input data sequentially. To this end, we also propose a data collection and labeling method that expedites one's effort.

III. Data Collection and Labeling

Creating synthetic datasets is popular among researchers. Open-source software like GNU's-not-Unix (GNU) radio [17] enables researchers the ability to generate datasets of many types of waveforms for RF ML, and affords datasets to be shared with others in the community that may not have the ability to create their own datasets. Additionally, software augmentation of these synthetic waveforms generates additional datasets that simulate real-world RF environmental conditions that may be difficult to reproduce naturally, and varying the amount and type of augmentation can produce even more training data. Synthetic datasets serve an invaluable service for researchers in that the labeled datasets can help the research community focus on purely research versus dataset collection.

On the contrary, a dataset consisting of recordings of real-world RF signals can provide additional features that may not be created synthetically. However, those *labeled* real-world RF signals are not trivial to acquire due to the hardware and software requirements (e.g., attenuators, coaxial network, a variety of transceivers, etc.). Additionally, recording radio signals in the "wild" can introduce ethical considerations with respect to personally identifiable information (PII)—using personally controlled transceivers helps reduce PII concerns.

We use GNU radio in conjunction with out-of-tree (OOT) module SigMF [18]) to collect datasets of ten common types of waveforms in the ISM bands at 434MHz and 915MHz, as well as the 70cm (420-450MHz) push-to-talk (PTT) amateur radio bands. All RF transceivers used for dataset collection were under our control.

Data collection and format: The SigMF formatted data was collected OTA using an Ettus universal software radio peripheral (USRP) [19]. SigMF data consists of the IQ data file, and an accompanied JavaScript™ object notation (JSON) formatted labeled meta data file. Each waveform was recorded for ten seconds at 1 mega-sample-per-second (MSPS) integer 16-bit IQ and saved as a 32-bit float IQ. The OTA data collection was an indoor lab with many reflections that provided natural multi-path augmentation.

OTA Setup and SDR models: The OTA training, validation, and testing waveforms were recorded using an Ettus SDR N210 [20] with SBX daughter board [21], and B205-mini. In-line, and programmable attenuators were used to maintain a consistent signal-to-noise ratio (SNR) to the SDR during data collection. Waveforms were recorded with 30-40dB of SNR.

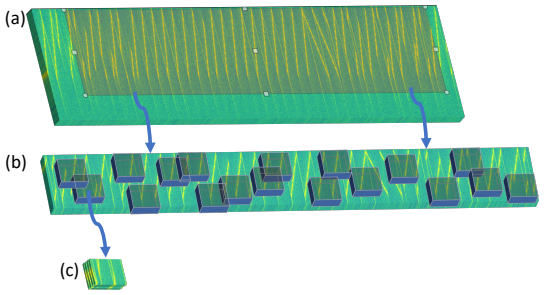


Fig. 1. The proposed GUI-based patch extraction process.

GUI software: We provide open-sourced tools with which researchers can collect and label their own real-world signals for RF classification instead of using synthetically generated datasets, in addition to those provided with this paper. Two custom GUIs, written in Python, allow the user to extract labeled feature patches from the IQ data to train machine learning models. In Fig. 1 (a), our first GUI allows the user to select a broad swath of spectrogram area, where a visible amount of signal activities can be found. Then using the second GUI, as illustrated in Fig. 1 (b), the selected area of the spectrogram randomly extracts a number of small patches from various locations. The user interacts with the second phase by defining the width, height, and total number of patches to produce. As a result, the 3D patches are stored for later use as the input "images" of the CNNs (Fig. 1 (c)).

Note that labeling is naturally done by assuming all patches extracted from a recording belong to the same class known to the researcher who operates the GUI system.

IV. The Time-Frequency Representation

We propose to convert the RF signals into a time-frequency domain using STFT, which is a process applying discrete Fourier transform (DFT) to a series of windowed short frames [11]. STFT on a time domain signal $x(n)$ results in a complex-valued matrix $\boldsymbol{X}$, having m and f as the time index and the center frequency,

$$\text{STFT}\big(x(n)\big) = X_{m,f} \equiv \sum_{n=-\infty}^{\infty} x(n)w(n-mR)e^{-j2\pi fn}, \quad (4)$$

where a windowing function w slides over the input signal x with a hop size R to define t-th short-frame input because w is typically defined with a bell-shaped function, zeros elsewhere.

Transforming the data to the TF domain has several benefits. With a proper choice of the windowing function and hop size, the data does not lose much information during the transformation. Meanwhile, the TF domain visualizes both the frequency and time variations contained in the data, which are otherwise difficult to represent. Hence, the TF representation is a suitable format as an input to the CNN models, where the input is defined as a stacked set of images, a 3D tensor with 4 channels.

While TF domain classification is often defined with the magnitude of the spectrogram, we propose to stack up both real and imaginary Fourier coefficients on top of each other. To this end, we apply STFT to each of the I and Q channels, respectively, and this garnered four real-valued arrays (e.g., spectrograms), $\mathbf{I}^{(\text{Real})}$, $\mathbf{I}^{(\text{Imag})}$, $\mathbf{Q}^{(\text{Real})}$, and $\mathbf{Q}^{(\text{Imag})}$:

$$\text{STFT}\big(i(n)\big) \equiv \mathbf{I}^{(\text{Real})} + j\mathbf{I}^{(\text{Imag})} \quad (5)$$

$$\text{STFT}\big(q(n)\big) \equiv \mathbf{Q}^{(\text{Real})} + j\mathbf{Q}^{(\text{Imag})} \quad (6)$$

Finally, the STFT IQ arrays were stacked to form a rank-3 (3D) tensor with 4 input channels. Similarly, we computed the I and Q channel magnitudes to produce a stacked 3D tensor with 2 channels to compare with each 4 channel network model. The GUI process described in Section III follows to extract small patches for model training.

V. The Proposed RF Classification Pipeline

A. Experimental Setup

Computing Environment: Model training was implemented via Pytorch [22] using GPU computing. RTX8000 (4608 NVIDIA® CUDA® cores) with 45GB of GPU memory.

Training Data and Patching: A total of six waveforms per class were recorded for each phase of the supervised learning, i.e., training, validation, and testing. 1,700 patches were collected per training waveform, totaling 10,200 patches per class, while we use 2,000 per class for validation from validation recordings. Each patch is saved as a 3D tensor of (time)×(frequency)×(channels), where the third one holds real and imaginary channels of both I and Q signals. Width and height dimensions of 224×224 were chosen for the GUI-based patching. This matched the VGG and ResNet native architectures designed for visual object classification [23], as well as provided enough spectral diversity for training the network on the selected waveforms.

Signal Classes: Ten waveforms were collected for classification. These classes were comprised of various analog PTT transceivers (NFM), three types of digital PTT transceivers (GD55, TYT, and YSF), two ISM RF doorbells (Vodeson and Sado), one RF light switch controller (light), two types of long-range (LoRa™) waveforms (lora125, and lora250), and a misc. key fob (click). The GD55, TYT and YSF use a 4FSK based modulation and are spectrally similar to one another. Two additional classes were used to account for spectrogram areas that contained noise and the DC artifact present at the center bin of the DFT (direct conversion receiver).

Data Augmentation: RF transmissions often have a variety of channel impairments. Multi-path wave propagation, noise (e.g., environmental, transceiver, etc.), oscillator frequency drift, IQ imbalance, and phase noise are common. For this work, the only augmentation was to SNR. We inject additive white Gaussian noise (AWGN) with a "loudness" randomly chosen from a range between 0.1 to 10.0dB for each patch.

CNN Models: Three primary network models were used to classify the waveforms: VGG16 [24] with 16 hidden layers, and ResNet [25] with 18 and 50 hidden layers, respectively. Each optimized model was run for 100 epochs.

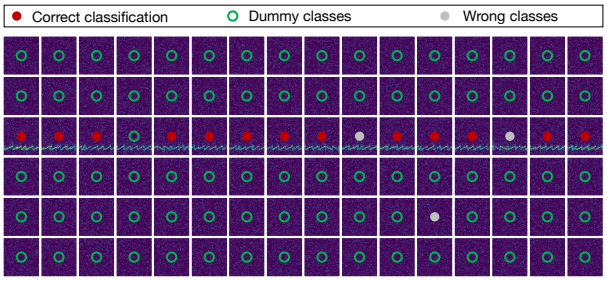
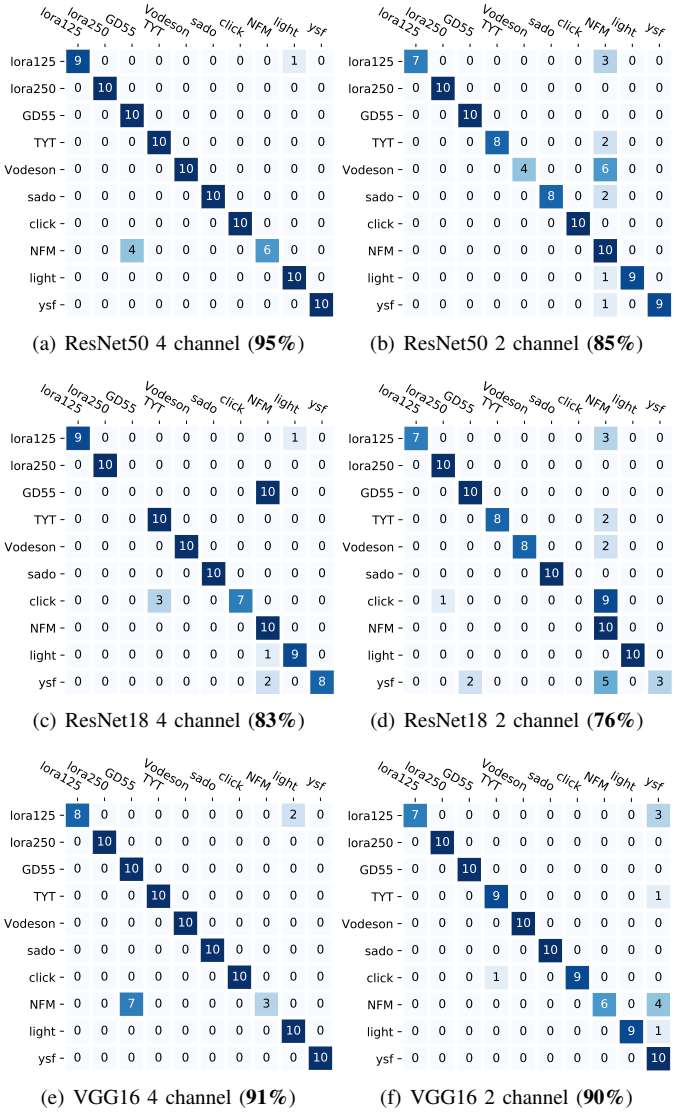


Fig. 2. Illustration of the proposed majority voting-based classification scheme on a LoRa™ 125kHz recording.

B. Test-Time Inference

While the models are trained on the patches, our goal during the test time is to identify which class the signal belongs to, regardless how many patches we can extract from it. Hence, during the test time, we scan the entire spectrogram and sequentially extract non-overlapping patches. All patches are fed to the model, whose classification results are consolidated to make the final decision via a majority voting scheme. For example, we choose 1,024 as the frame size for the 1 MSPS signals, which also defines the frequency resolution, $1MHz/1024 = 976.56Hz$ as well as the number of frequency bins per spectrum, i.e., $F = 1024$. Note that we discard about a half of the frequency bins due to the complex conjugacy. With this setup and with a hop size of $R = F/2 = 512$, a 10 sec-long test signal turns into 19,532 spectra, each is with $1024/2 + 1 = 513$ frequency subbands. Eventually, given our patch size of 224×224, each spectrogram breaks down into about 174 patches. The test-time signal classification is based on the assumption that for a given time period there is only one dominant signal class. Indeed, we recorded the real-world test signals in a controlled environment to meet this constraint. However, a naïve approach that simply selects the majority class out of the patch-by-patch classification results is not a solution because signal sparsity in the time-frequency space will classify most of the patches as the `noise` class.

Fig. 2 depicts our winner-take-all majority vote scheme. We first count all the *meaningful* predictions that belong to the ten critical classes (red or grey filled circles in the figure) rather than the two dummy classes, `noise` and `center frequency` (hollow green circles). In the figure, it will correspond to the sum of red and grey dots, $13 + 3 = 16$. Out of 16 we see 13 patches belong to a single class (which is actually correct classification to LoRa™ 125kHz), while the the grey dots mean that those patches are misclassified into some other non-dummy classes. Consequently, the whole region of these 6×16 patches is classified into the red dot class, i.e., LoRa™ 125kHz. The failure mode of this scheme will be twofold. First, if the dominant signal patches are misclassified into the dummy classes, e.g., the (3,4) patch in the figure, the system will predict that no activity is happening. Second, if there are any confusing class pairs, the misclassification will result in a wrong prediction to the confusing class.



(a) ResNet50 4 channel (**95%**) (b) ResNet50 2 channel (**85%**)

(c) ResNet18 4 channel (**83%**) (d) ResNet18 2 channel (**76%**)

(e) VGG16 4 channel (**91%**) (f) VGG16 2 channel (**90%**)

Fig. 3. The confusion matrices of different systems in comparison. The total classification accuracy is presented in the parenthesis.

Since there is no efficient way to label each patch, we record a 10 sec-long signal containing only one dominant class's activity and performed inference on its patches. Likewise, the classification results are for the entire 10 sec-long recording.

C. Discussion

We used three popular 2D CNN network architectures: VGG16, ResNet50, and ResNet18, where the number indicates the number of hidden layers. Each network model was further distinguished by 2 and 4 channel 3D tensor input, where the former uses magnitudes of the DFT coefficients from I and Q channels, while the latter uses both real and imaginary coefficients of the IQ data. These network combinations produced six different test configurations. The models were trained using the Adam optimizer [26], and initial learning rates were found using validation: 1×10^{-4} for the ResNet models and 1×10^{-6} for the VGG models. With an early stopping strategy, we

stored the models that gave the best validation performance and used them for testing. All models achieved more than 99% validation accuracy, although the test-time performance varied due to the discrepancy between validation and testing, as well as the models' different characteristics. Each class consists of 10 signals, whose length is fixed to 10 seconds. Hence, the classification accuracy is defined by the number of correctly classified divided by 100. In Fig. 3 we present the confusion matrices for more analysis, where y-axis is the ground-truth class labels and x-axis is the prediction.

The first observation we make is the overall acceptable performance of all systems. Except for the ResNet18 2 channel input case, all the systems were able to achieve more than 80% accuracy, showcasing the robustness of the proposed classification scheme. We also point out that the data augmentation process that injects Gaussian random noise to the input patches helped stabilize the test-time performance. The comparison between the ResNet models leads us to the conclusion that the deeper ResNet50 models ($\sim$23.5M trainable parameters) outperform the shallower ResNet18 models ($\sim$11M), and imply that deeper models are more favorable.

On the other hand, VGG16 models showed an interesting behavior: their overall performance is good (91% with 4 channel input and 90% with 2 channels), but given that they have many more parameters (134M), the performance is not impressive. This behavior was expected because ResNet's more advanced features, such as skip connections, are known to outperform VGG. For example, considering the test-time inference complexity of these models, as well as the performance, ResNet50 on the 4 channels should be the choice rather than VGG16. In an extreme environment where minimal resource usage is required, ResNet18 should be the choice.

It is also noticeable that the proposed 4 channel input tensors greatly outperform the 2 channel inputs. It is because the 4 channels retain all the details about the phase information, which the 2 channel data are missing. The difference is more salient in the smaller ResNet models than the VGG16 models.

VI. Conclusion

This paper explored RF classification by defining it as an image classification problem on multi-channel input. We were able to show that popular 2D CNN models, such as ResNet and VGG, can classify waveforms in the time-frequency representation. We observed higher accuracy when using all the magnitude and phase information of the TF representation of the IQ signals than the magnitude-only cases that are more popular in the literature. We open-sourced our project not only to improve the reproducibility, but to help the researchers create their own dataset via our proposed GUI-based annotation system.

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[3] ITU-R. International Telecommunication Union. [Online]. Available: https://www.itu.int/en/ITU-R/Pages/default.aspx

[4] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26–38, 1995.

[5] C. Moy and J. Palicot, "Software radio: a catalyst for wireless innovation," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 24–30, 2015.

[6] C. Belisle, V. Kovarik, L. Pucker, and M. Turner, "The software communications architecture: two decades of software radio technology innovation," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 31–37, 2015.

[7] 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, no. 1, pp. 168–179, 2018.

[8] T. O'Shea, T. Roy, and T. C. Clancy, "Learning robust general radio signal detection using computer vision methods," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 829–832.

[9] C. Gravelle and R. Zhou, "SDR Demonstration of Signal Classification in Real-Time Using Deep Learning," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–5.

[10] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep Learning for Wireless Communications," 2020.

[11] J. B. Allen and L. R. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, Nov 1977.

[12] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, "Deep Learning Classification of 3.5-GHz Band Spectrograms With Applications to Spectrum Sensing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 224–236, 2019.

[13] S. Behura, S. Kedia, S. M. Hiremath, and S. K. Patra, "WiST ID—Deep Learning-Based Large Scale Wireless Standard Technology Identification," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1365–1377, 2020.

[14] H. Tamura, K. Yanagisawa, A. Shirane, and K. Okada, "Wireless Devices Identification with Light-Weight Convolutional Neural Network Operating on Quadrant IQ Transition Image," in *2020 18th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2020, pp. 106–109.

[15] G. Jajoo, Y. Kumar, A. Kumar, and S. Yadav, "Blind Signal Modulation Recognition through Density Spread of Constellation Signature," *Wireless Personal Communications*, vol. 114, 10 2020.

[16] P. Pace, *Detecting and Classifying Low Probability of Intercept Radar, Second Edition*. Artech House, 2008.

[17] A. M. Wyglinski, D. P. Orofino, M. N. Ettus, and T. W. Rondeau, "Revolutionizing software defined radio: case studies in hardware, software, and education," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68–75, 2016.

[18] B. Hilburn, N. West, T. O'Shea, and T. Roy, "SigMF: The Signal Metadata Format," *Proceedings of the GNU Radio Conference*, vol. 3, no. 1, 2018. [Online]. Available: https://pubs.gnuradio.org/index.php/grcon/article/view/52

[19] A. M. Wyglinski, D. P. Orofino, M. N. Ettus, and T. W. Rondeau, "Revolutionizing software defined radio: case studies in hardware, software, and education," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68–75, 2016.

[20] Ettus N210 SDR. Ettus Research. [Online]. Available: https://files.ettus.com/manual/page_usrp2.html

[21] SBX daughterboard. Ettus. [Online]. Available: https://files.ettus.com/manual/page_dboards.html\protect\@normalcr\relax#dboards_sbx

[22] A. Paszke et. al, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Ieee, 2009, pp. 248–255.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.