Pacific Northwest
NATIONAL LABORATORY

PNNL-38559

Passband Signal Detection at the Edge

September 2025

Gavin Parpart
Peter Martin
Stephen Jones
Noah Elmore
Jeremiah Rounds

U.S. DEPARTMENT of ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-38559

Passband Signal Detection at the Edge

September 2025

Gavin Parpart
Peter Martin
Stephen Jones
Noah Elmore
Jeremiah Rounds

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Traditional algorithms for Radio Frequency detection rely on static lookup tables and current machine learning models rely on large computation platforms and processor-intensive baseband processing. Future algorithms for radio frequency (RF) spectrum awareness need to be compatible with edge hardware to be practical for many portable and rapid characterization applications. As part of the RF Spectrum at the Edge effort at PNNL, we developed a signal detection and classification model for the ZCU111 RF System-on-a-Chip (RFSoC) that operates on the fast Fourier transform of passband RF data. The system can detect and classify multiple signals of interest and display the predictions in real-time. The model consists of a modified ConvNeXt backbone and YOLOv3 head to operate on the Deep Learning Processing Unit on the RFSoC. We gathered datasets for training and testing by using software defined radio to transmit example signals of Wi-Fi 802.11 b/g, Wi-Fi 802.11 n, FM Radio, LTE and LTE-M. By leveraging multiple inputs on the RFSoC frontend, the datasets span up to 4 GHz of bandwidth. The models showed high performance in classification accuracy, center frequency error, bandwidth error, and detection accuracy for both single and multi-signal datasets.

Summary

Under funding of the laboratory directed research and development program of the RF Spectrum at the Edge effort, PNNL developed methods for low size, weight and power (SWaP) RF Spectrum Awareness and processing using deployable hardware and commercial software tools. A large portion of this effort was focused in developing both a framework to deploy models on the Deep Learning Processing Unit (DPU) of the Xilinx ZCU111 development board and developing models that could be reduced in size to fit on the edge hardware and accurately classify RF signals in real-time. The effort culminated in a demonstration using the ZCU111 to detect and classify RF signals in 4 GHz of spectrum and display those results to users in real time. This work demonstrates a new ability to classify many RF signals simultaneously over a wide bandwidth, whereas current methods convert to baseband and analyze small sub-bands sequentially. This will enable faster classification of RF signals demonstrated on a small, portable platform. This report details the methods, challenges and results encountered with efforts to achieve this new method of machine learning-based classification on nascent edge-deployed hardware.

Acknowledgments

This research was supported by the Radio Frequency Spectrum at the Edge project under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Acronyms and Abbreviations

CNN	Convolutional Neural Network
DETR	Detector Transformer (Model)
DPU	Deep Learning Processing Unit
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GELU	Gaussian Error Linear Unit (Activation Function)
ReLU	Rectified Linear Unit (Activation Function)
RF	Radio Frequency
ROC AUC	Area Under the Receiver Operating Characteristic Curve
SDR	Software Defined Radio
SoC	System-on-a-Chip
SWaP	Size, Weight and Power
YOLO	You Only Look Once (Model)

Contents

Figures

Tables

1.0 Introduction

This work introduces effective use of a ZCU111 RF System-on-a-Chip (RFSoC) for detection and classification of RF signals of interest at the edge. We found that a passband signal processor, on a integrated field programmable gate array (FPGA), combined with a convolutional neural network (CNN) deployed on the DPUCZDX8G Deep Learning Processing Unit (DPU) can effectively perform real-time RF detection of select signals of interest within a 4-GHz passband. However, it is necessary to constrain feature engineering, model design, and overall scope to be within the capabilities of a ZCU111 RFSoC.

1.1 Background

Most prior applications of deep learning to RF signal detection and classification have involved data containing one in-band signal [1]–[5], which is often referred to as automatic modulation classification. Other more recent works that studied detecting and classifying multiple in-band signals typically focused on spectrogram representations of data, e.g., using U-Net segmentation [6], You Only Look Once (YOLO) object detection [7]–[9], and transformer-based object detection with Detector Transformer (DETR) [9]. The work outlined in this study focuses on an object detection approach applied to the entire frequency domain available at a given sampling frequency (Fourier transform representation), which contains many in-band signals, both in- and out-of-distribution with respect to the training set signal types.

1.2 Deep Learning Processor Unit (DPU)

We utilized the DPUCZDX8G DPU to perform model inference. The DPUCZDX8G DPU is a co-processor designed for running CNNs with 8-bit integer weights and activations. Models were compiled and loaded onto the DPU using Vitis-AI. This DPU does not support matrix multiplication between two arbitrary inputs, and so it cannot perform multihead attention. As a result, this work focused on a CNN architecture for our model.

2.0 METHODOLOGY

2.1 Dataset

Our training and validation dataset consisted of five signal classes: Wi-Fi 802.11 b/g, Wi-Fi 802.11n, FM radio, LTE, and LTE-M. The MATLAB WLAN and LTE toolbox were used to generate random baseband signals for each class. A custom implementation in MATLAB was used to generate the FM radio signals.

To generate a large corpus of training data, each of these signals were transmitted 20,000 times from an Ettus X310 software defined radio (SDR) to the RFSoC by coaxial cable for a total of 100,000 signals. To allow for realistic signal identification and classification training and model development, a wide-band antenna was also coupled to the coaxial cable to collect and add background noise from the environment. There was a delay of 1 second between each random transmission. Following coaxially cabled data collection and training, the models were validated using actual signals using over-the-air transmission at short ranges.

For each random transmission, several factors were varied. A random carrier frequency was uniformly sampled between 0.7-3.7 GHz. The SDR transmission power varied uniformly between 0-20 dB of gain. Signals are captured at a sample rate of 3.93216 Giga samples per second (GSPS) and each capture is 65,536 samples or 16.7 µs. The final datasets were split 70% training, 20% validation, 10% test.

2.2 Preprocessing

An FPGA hardware fast Fourier transform (FFT) was applied to the captured complex time-domain passband samples, $X[t]$, of length $n = 65536$, such that $X[k] = \sum_{t=0}^{n-1} X[t]e^{-i2\pi kt/n}$, where $k \in \{0, \ldots, n-1\}$ and $X[k]$ is the FFT output for the kth frequency bin. The FPGA FFT outputs, $X[k]$, are 16-bit integers. To produce 4 GHz FFTs, two separate inputs on the RFSoC frontend were utilized. One had a filter for 0-2 GHz and the other had a filter for 2-4 GHz which aliases when captured. The FFT of each capture was then taken and the two FFTs are concatenated together. Training records of FFT data, X[k], were cast to 32-bit floating point and then preprocessed using the following transformations in sequence before each batch:

- Random Phase Rotation: $X[k] = X[k]\, exp(-i\phi)$, where $\phi \sim$ Uniform$(0, 2\pi)$.
- Random DC Offset: $X[k] = X[k] + x0 + x1i$ where $x0, x1 \sim$ Normal$(\mu = 0, \sigma = 19)$.
- Rescale to $[-1,1]$: $X[k] = g(\boldsymbol{Re}(X[k])) + g(\boldsymbol{Im}(X[k]))i$ where $g(x) = 2(x - m0)/(m1 - m0) - 1, m0$ and $m1$ are the minimum and maximum component values for the sample $X[k]$, respectively.

For testing and validation data, only the rescale to $[-1,1]$ operation took place prior to the 32-bit floating point model activation. For application on the RFSoC, the DPU model used 8-bit integers as input. The DPU model input rescaled the $[-1, 1]$ FFT to 8-bit integer range. Handling that transform appropriately is an included feature of the Vitis-AI framework and quantization strategy.

2.3 Model Architecture

2.3.1 Backbone

For the backbone of our detector model, we modified the ConvNeXt [10] architecture to better operate on one-dimensional, two-channel (I/Q) FFT data and to provide compatibility with the DPU. The full block is depicted in Figure 1. We utilized a one-dimensional kernel of size 7 for the depth-wise convolution. Additionally, the LayerNorm layers were replaced with BatchNorm [11] and Gaussian Error Linear Unit (GELU) activations were replaced with Rectified Linaer Unit (ReLU) activations. Our final backbone had 4 stages, the parameters of which are depicted in Table 1 we utilized one-dimensional convolution for the spatial down-sampling layers between stages, with a kernel size of 4 and a stride of 4. In initial experiments we also explored a ResNet-18 [12] backbone with one-dimensional kernels but later dropped it due to poor performance.

Table 1. Model and Training Parameters

Parameter	Value
Blocks per Stage	(3, 3, 9, 3)
Channels per Stage	(96,192, 384, 768)
Epochs	300
Batch Size	256
Optimizer	AdamW [13]
Weight Decay	0.05
$\beta 1, \beta 2$	0.9, 0.999
Peak Learning Rate (LR)	4e-4
LR Scheduler	Cosine Annealing
LR Warmup Epochs	20

2.3.2 Signal Detection Heads

We designed our detection heads based on YOLOv3 [14]. The 4 GHz bandwidth was split amongst 128 heads, giving each head a ≈30 MHz band to detect. We designed two variants of the heads: one for single signal detection (per frequency bin) and one for multiple signal detection. The output shapes for each model type are indicated in Table II.

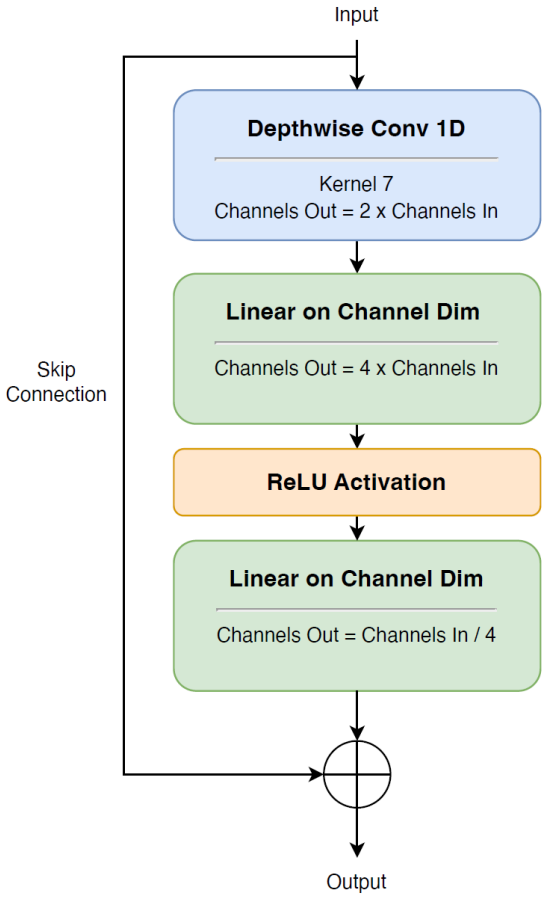

Figure 1. Modified ConvNeXt Block

The single signal detection model had four outputs: a score indicating whether a signal was present, a set of logits indicating the signal type, the center frequency of the signal, and the bandwidth of the signal. Each output from the head was a linear projection of the backbone output. The signal presence score was trained so that a positive score indicated a signal was present in the band, and a negative score indicated no signal was present. The signal type logits had one component for each signal type, and the index with largest magnitude component corresponded to the signal that was predicted. The center frequency was represented as an offset from the center of the frequency band corresponding to the frequency bin of that output head and was computed using a sigmoid activation function. The bandwidth was computed by using an exponential function and scaling it by a constant of 20 MHz. During training, we used binary cross entropy loss for the presence scores, cross entropy loss for the signal type logits, mean absolute error loss for the center frequency, and mean squared error loss for the bandwidth.

To accommodate prediction of multiple signal types per frequency bin, the multi-signal detection model was modified such that the center frequency and bandwidth heads output values for each signal type, i.e. the same number of outputs as the logits head. Signal presence was predicted by the multi-signal detection model as if the signal prediction head output for a frequency bin was positive. The class(es) of the predicted signal(s) were determined by the logits indices corresponding to positive values. As each class can independently be predicted, the model can

identify multiple signals so long as the two signals are of different types or occur in the frequency bands of different model heads.

Table 2. Output Head Tensor Shapes for Each Model

Head Type	Single Signal	Multi-Signal
Signal Presence	(batch, freq_bins)	(batch, freq_bins)
Signal Type	(batch, freq_bins)	(batch, freq_bins, num_types)
Center Frequency	(batch, freq_bins)	(batch, freq_bins, num_types)
Bandwidth	(batch, freq_bins)	(batch, freq_bins, num_types)

3.0 Results

We utilized several measures to evaluate the performance of the different model outputs. Signal classification accuracy is reported for data examples containing signals, whereas background noise examples are excluded. To measure the performance of the center frequency estimation, we report the median absolute error between the predicted frequency and the true frequency. The median error is utilized as errors can be significant when a signal is predicted at the incorrect model head. Figure 2 shows this behavior: many points are very close to the line of the true frequency, while the scattered errors are from incorrect output heads. Similarly, we report the median absolute bandwidth error. Signal detection performance was evaluated by creating a receiver operating characteristic curve, which tracks the true positive rate vs. the false positive rate across all possible detection thresholds. The area under this curve (ROC AUC) indicates how high the true positive rate is across the range of false positive rates.

3.1.1 2 GHz Data

Our initial experiments were on FFTs spanning 2 GHz of bandwidth, with signals transmitted at frequencies between 0.7-1.7 GHz. This dataset also omitted the 802.11 n class. The confusion matrix of the model predictions is shown in Figure 3. Most signals were correctly classified, with the primary failure case being background noise being misclassified as LTE. A potential cause for this is that the LTE signals had gaps between transmitted frames, which may have caused the model to improperly associate the background noise. As shown in Table III, the median center frequency error of 35.5kHz is very close to the optimal 30 kHz, which is the FFT frequency resolution. The bandwidth error was marginally higher, potentially due to difficulty optimizing the continuous head output to produce discrete values for each bandwidth. The signal detection had a ROC AUC of 0.9974, indicating that the model has close to 100% signal detection even at very low false positive limits.

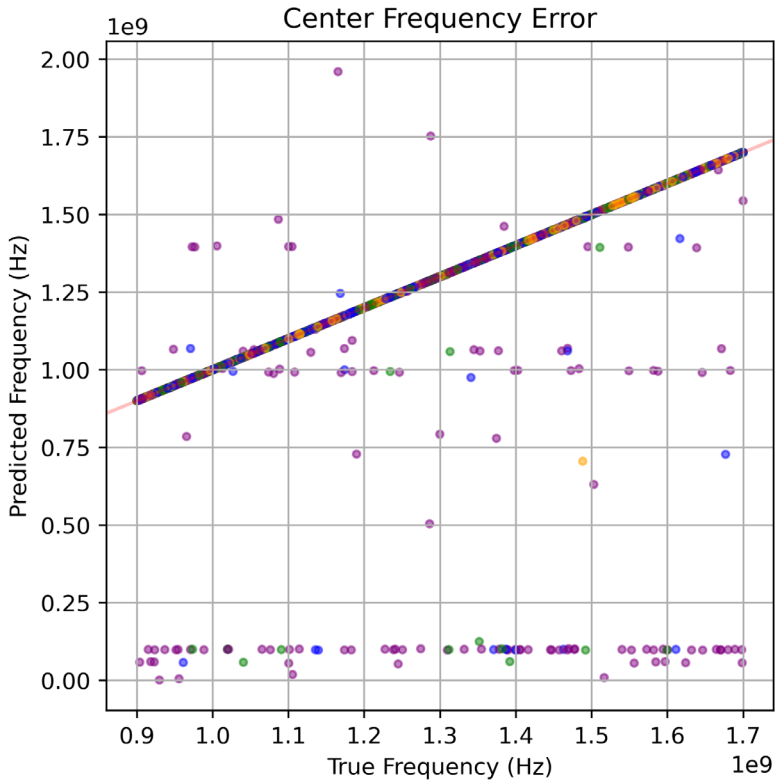

Figure 2. Center Frequency Prediction Error on 2 GHz Data

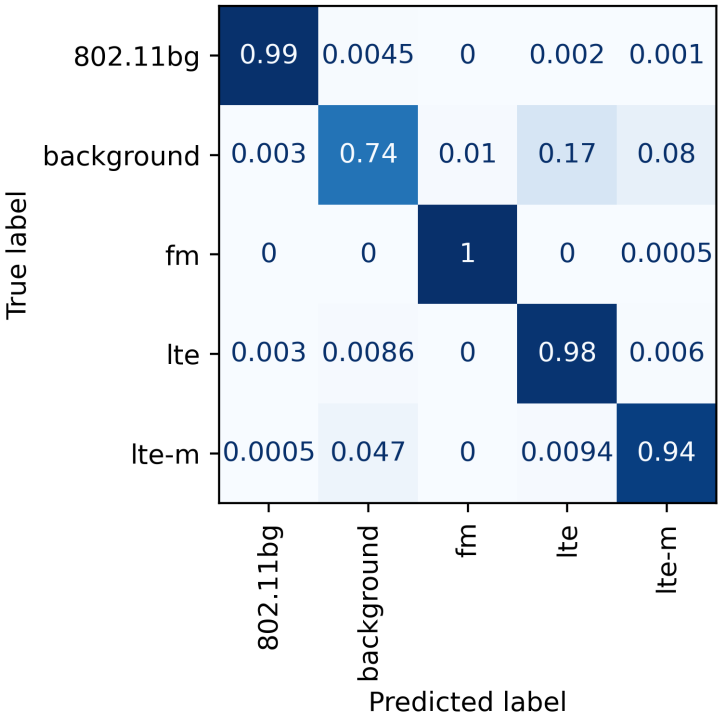

Figure 3. Confusion Matrix of Model on 2 GHz Data

3.1.2 4 GHz Data

For the 4 GHz data, the signals were transmitted at frequencies between 0.7-1.7 GHz and 2.2-3.7 GHz. The region from 1.7-2.2 GHz was excluded as the roll-off factor of our filters caused signals to appear as reflections in both the 0-2 GHz and 2-4 GHz FFTs. Additionally, we increased the scaling of the FFT during computation to improve the precision of the output. With the increased precision, the signal classification accuracy was similar to the 2 GHz model, and the signal detection performance (ROC AUC) was marginally better. The center frequency and bandwidth error were both notably higher in this setup. Comparing the FFT peak frequency to the expected carrier frequency of a large series of tones indicated that there were more significant differences between the FFT frequencies and SDR frequencies with this setup.

Table 3. Model Performance Across Test Scenarios

Dataset	Acc.	Freq. Error	Bandwidth Error	ROC AUC
2 GHz	98%	35.5 kHz	43.6 kHz	0.9974
4 GHz	96%	286.9 kHz	82.1 kHz	0.9985
Multi Signal	98%	62.2 kHz	4.7 kHz	0.9981

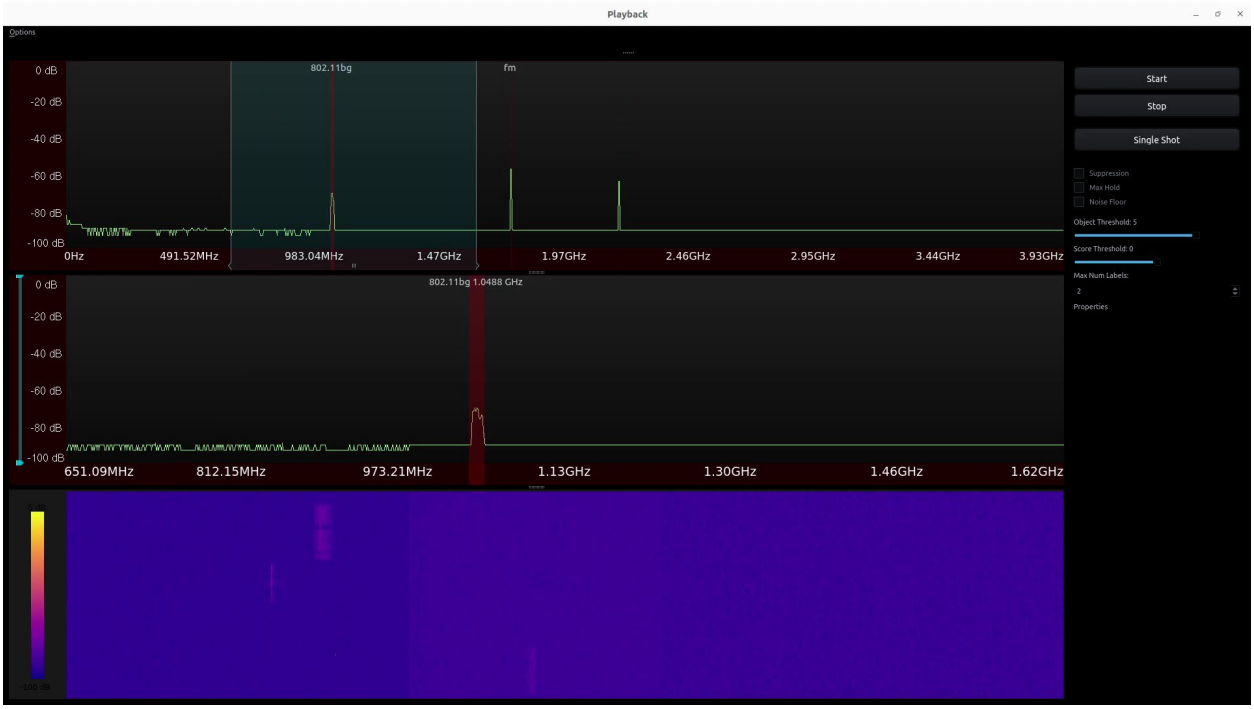

Figure 4. Multiple Signals Being Detected Over the Air

3.1.3 Multi-Signal Data

To produce examples with multiple signals, we took the 2 GHz FFT data and summed the FFTs together for between 1-20 signals per data-point. The model's accuracy did not significantly change from the single-signal model, and the signal detection ROC AUC was marginally better. The center frequency error was marginally higher, potentially due to the overlap between multiple signals. The signal bandwidth error was dramatically lower, as having a single bandwidth output for each signal type effectively makes the bandwidth prediction a lookup table.

3.1.4 Over-the-Air Tests

To demonstrate the full end-to-end system, we connected the SDR to an antenna and transmitted the various signal types over the air. The model was evaluated subjectively to confirm it operated as expected. We set up a user interface to display the model predictions in real-time, shown in Figure 4. Pre-processing the data took 18 ms and running the model took 15 ms, allowing for roughly 30 predictions per second. Further work may explore removing the re-scaling operation to reduce the pre-processing time

4.0 Conclusions

We collected multiple datasets of several signal types, broadcast using an SDR at a random carrier frequency between 0.7 to 3.7 GHz. We adapted a ConvNeXt backbone and YOLOv3 head to operate on one-dimensional signals and for compatibility with the RFSoC DPU. In the process we found that manually splitting the head outputs leads to better quantization performance. We trained this model on three different datasets, one with 2 GHz bandwidth, and another with 4 GHz bandwidth and 2 GHz bandwidth with multiple signals present. Across all datasets, the model is able to detect and classify signals with high accuracy. On the 2 GHz data, the model's center frequency estimation approached the optimal error of 30 kHz, and the increased error in the 4 GHz data was likely due to data collection or pre-processing error rather than an inherent limitation of the model. Finally, we set up a complete over-the-air demonstration of the system with an interface displaying the model predictions in real time.

5.0 References

[1] W. H. Clark, V. Arndorfer, B. Tamir, D. Kim, C. Vives, H. Morris, L. Wong, and W. C. Headley, “Developing rfml intuition: An automatic modulation classification architecture case study,” in MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM), 2019, pp. 292–298.

[2] S. C. Hauser, W. C. Headley, and A. J. Michaels, “Signal detection effects on deep neural networks utilizing raw iq for modulation classification,” in MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM), 2017, pp. 121–127.

[3] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1. Institute of Electrical and Electronics Engineers (IEEE), Feb. 2018, p. 168–179. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2018.2797022

[4] Y. Wu, X. Li, and J. Fang, “A deep learning approach for modulation recognition via exploiting temporal correlations,” in 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2018, pp. 1–5.

[5] L. Boegner, M. Gulati, G. Vanhoy, P. Vallance, B. Comar, S. Kokalj-Filipovic, C. Lennon, and R. D. Miller, “Large scale radio frequency signal classification,” 2022. [Online]. Available: https://arxiv.org/abs/2207.09918

[6] N. West, T. Roy, and T. O’Shea, “Wideband signal localization with spectral segmentation,” 2021. [Online]. Available: https://arxiv.org/abs/2110.00583

[7] A. Vagollari, V. Schram, W. Wicke, M. Hirschbeck, and W. Gerstacker, “Joint detection and classification of rf signals using deep learning,” in 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), 2021, pp. 1–7.

[8] H. N. Nguyen, M. Vomvas, T. Vo-Huu, and G. Noubir, “Wideband, real-time spectro-temporal rf identification,” in Proceedings of the 19th ACM International Symposium on Mobility Management and Wireless Access, ser. MobiWac ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 77–86. [Online]. Available: https://doi.org/10.1145/3479241.3486688

[9] L. Boegner, G. Vanhoy, P. Vallance, M. Gulati, D. Feitzinger, B. Comar, and R. D. Miller, “Large scale radio frequency wideband signal detection & recognition,” 2022. [Online]. Available: https://arxiv.org/abs/2211.10335

[10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 11 976–11 986.

[11] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ser. ICML’15. JMLR.org, 2015, p. 448–456.

[12] 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), 2016, pp. 770–778.

[13] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in International Conference on Learning Representations, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[14] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[15] V. Kathail, “Xilinx vitis unified software platform,” in Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ser. FPGA

’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 173–174. [Online]. Available: https://doi.org/10.1145/3373087.3375887

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov