Exploring the Potential of Residual Networks for Efficient Sub-Nyquist Spectrum Sensing

Hem Regmi; Sanjib Sur
Computer Science and Engineering; University of South Carolina, Columbia, USA
hregmi@email.sc.edu; sur@cse.sc.edu

***Abstract*—We propose *ReSense*, a residual network for spectrum sensing high-frequency signals with low-frequency samplers. *ReSense* first transforms the aliased signal from low-frequency samplers into image-like inputs and uses multiple convolution layers and skip connections to predict the signal's frequency components to enable spectrum sensing. We evaluate *ReSense* on the signal dataset with single and double frequencies and achieve 95% and 40% accuracy in detecting modulation type on respective datasets, indicating accurate spectrum sensing.**

***Keywords*: Spectrum Usage and Cognitive Radio systems; Convolutional Neural Networks; Residual Networks.**

I. Introduction

The proposed 6G wireless communication system requires an extensive frequency range of 300 GHz to 10 THz to achieve peak transmission speeds of 100 Gbit/s to 1 Tbit/s and low communication delays. To manage spectrum resources efficiently in the complex and ultra-dense network, cognitive-based intelligent spectrum management systems are necessary [1]. Dynamic spectrum sharing is crucial to allow multiple devices to access different parts of the spectrum without fixed assignments. Mobile devices must be able to detect available spectrum and its utilization to reduce the chance of interfering with primary users. For detecting spectrum usage at higher frequencies, a high-bandwidth Analog-to-Digital converter (ADC) with several GHz of sampling frequency is needed. But these are costly and consume much power, hence they are often impractical for wireless and mobile devices. Low-bandwidth ADCs, on the other hand, are feasible for mobile devices as they have low cost and consume lower power, but the device can only sample the spectrum at a sub-Nyquist rate.

Spectrum recovery with sub-Nyquist is challenging for the following reasons: (1) due to the low sampling frequency of analog-to-digital converters, high-frequency components of the signal merge with the low frequency of the signals causing signal aliasing, and (2) only a few cosets, far less than the number of channels, are sampled with multi-coset sampling to reduce the number of sampling ADCs, which in turn losses the information of the signal, and needs suitable compressed sensing method to recover them.

To approximate the same functionality of high-bandwidth ADCs, low-bandwidth ADCs should be used, allowing for more effective utilization of the spectrum. Besides, multiple low-bandwidth ADCs can be combined to recover signals at high-frequencies [2]. Additionally, the signal can be sampled at specific delays, and with compressed sensing [3], and multi-cosets sampling [4], prediction of signals at other delays is possible without having to sample them. Modulation-type detection of the received signal is a crucial part of spectrum sensing for the following reasons. *First,* it affects the spectral characteristics of the signal, such as bandwidth, center frequency, and the signal's shape, which are crucial for spectrum recovery. *Second,* different modulation types have different energy around the center frequency, which controls the energy distribution of the signal. *Third,* modulation type determines the Signal-to-Noise (SNR) of the signal because some modulation types are more prone to noise than others.

In this work, we propose *ReSense*, which aims to find signal locations in the frequency domain indicated by pre-defined discrete channel indexes and the modulation types of the signals located somewhere in the baseband in two datasets of signals with single and double frequency components. The signals are aliased and do not have all cosets. To solve this problem, *ReSense* utilizes a deep learning platform that enables compressed-sensing and anti-aliasing filters. The convolution layers of the deep learning network are designed to re-sample signals at various frequencies and combine them to recover the actual frequency and modulation type. *ReSense* achieves 95% accuracy for detecting the modulation type of a single frequency and more than 40% for the signal with two different frequencies at various sub-bands.

II. Spectrum Sensing Challenges

Spectrum Sensing Fundamentals: Spectrum sensing is the presence of a signal of a specific spectrum. Frequency spectrums are not utilized all the time, and

it opens an opportunity to allow dynamic spectrum sharing. There are multiple ways we can detect the spectrum, such as energy detection at the particular band, detecting the pattern of the signal, template matching with known signal, *etc.* Currently, machine learning methods help to detect the signal. Such machine learning models can be configured to sense the spectrum based on the requirements and are more generalizable.

Multi-Coset Sampling: Multi-coset sampling is a popular technique in different disciplines to sense the signal with low-cost samplers. The key idea is to beat the Nyquist requirement, which requires the sampler with a frequency at least more than double of maximum frequency components of the signal. However, designing such a high-frequency sampling analog-to-digital converter (ADC) is costly and unavailable on mobile devices. But, to utilize the available spectrum, devices first need to sense the spectrum and ensure it is currently not utilized by other devices. Since the frequency spectrum is at higher frequencies, there is a need to achieve high-frequency sensing with low-sampling ADCs, which can be achieved with Multi-Coset sampling with properly designed cosets.

Figure 1 describes the process of recording data samples with multi-coset sampling. The original signal $x(t)$ has frequency f_{max}, and requires sampling frequency of f_{NYQ}, such that $f_{NYQ} \geq 2 \times f_{max}$. Now, we divide the signal into 40 different channels, where each channel samples the signal with low sampling frequency ADC. Since we divide it into L different channels, we record signals with sampling frequency $f_s = \frac{f_{NYQ}}{L}$. These L channels do not sample the signal simultaneously; instead, they sample the signal with a certain delay to capture the samples not captured by other channels. The key idea here is to arrange L channels and design delay in such a way as to record the frequency-aliased signals with Nyquist period $T = \frac{1}{f_{NYQ}}$. However, we might not need samples from all the channels to reconstruct the signal because signals are sparse, and only a few frequency components can faithfully reconstruct the signal. We store C cosets, $C \lll L$, and reconstruct estimate L channels using compressed sensing. Figure 1 shows multi-coset sampling, where $x_0(t)$, $x_1(t)$, $x_{11}(t)$, and $x_{37}(t)$ are four different cosets with signal samples. We can observe the sampling frequency is reduced for cosets.

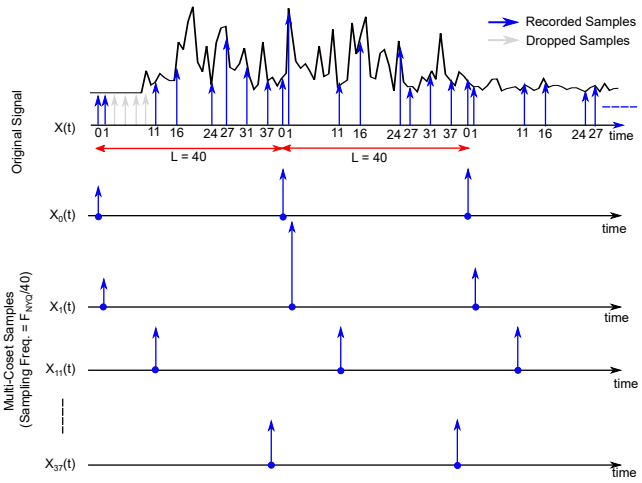

Fig. 1: *Multi-coset sampling principle and multi-cosets from the original signal.*

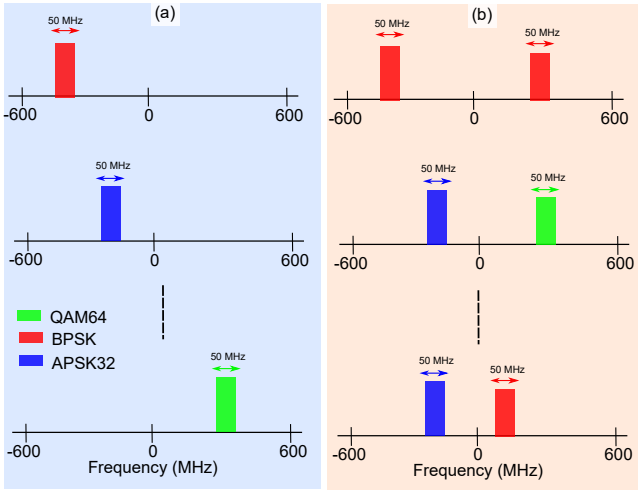

Fig. 2: *Frequency occupancy of input signals for QAM64, BPSK, and APSK32 on: (a) single frequency signal. (b) two frequency signal.*

III. *ReSense* Design

A. Dataset Description & Preprocessing

We use low-sampling ADCs to collect a data sample with 8 cosets out of 40 channels. Each coset has I and Q components of the signal and comprises 1024 samples, with one of 13 different modulation types (APSK16, APSK32, APSK64, ASK8, BPSK, OQPSK, PSK16, PSK8, QAM128, QAM16, QAM256, QAM64, QPSK) (see Figure 2), resulting in 1024 $\times$ 16 real numbers. To predict modulation type from cosets, we convert the signal 1024 $\times$ 16 into 128 $\times$ 128 images to utilize 2D convolution neural networks to extract features from the aliased signal. This has proven to yield better results than 1D signal [5]. We then up-sample the 128 $\times$ 128 images to 244 $\times$ 224 and concatenate 3 times to create a 3 $\times$ 224 $\times$ 244 RGB color image (see Figure 3 [a–b]). The converted RGB image is fed into the ResNet Network [6]. Predicting the modulation type is formulated as a classification problem using one-hot coding, which allows us to code each class with orthogonal coding [7].

B. Anti-aliasing with ResNet

Low-sampling ADCs often mix high-frequency components with low-frequency ones. As a result, it is not easy to construct an algorithm that can recover

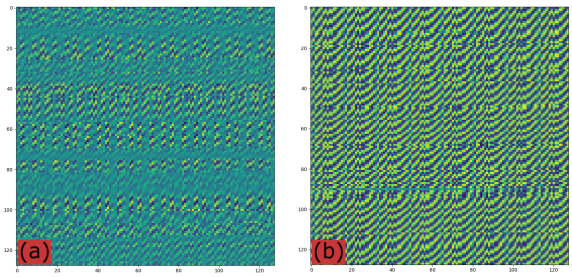

Fig. 3: *Image representation of input signals: (a) single frequency signal. (b) two frequency signal.*

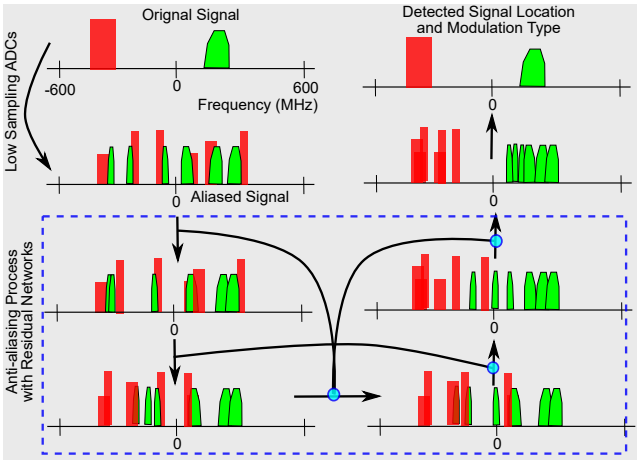

Fig. 4: *Anti-aliasing process with residual networks.*

the original signal from the aliased signal, with the recovered signal being an approximation of the original. In the past, multiple methods have been proposed to achieve this goal [8], [9]; however, only a few of these involve deep-learning techniques [10]. Deep learning networks have revolutionized the vision domain in a way that was unimaginable a decade ago, as they can approximate any mathematical relationship between input and output. The core method for recovering the aliased signal involves sampling it at different frequencies to separate and accurately identify the frequency signal.

The solution to recovering aliasing can be best described with the mod function. For example, consider an array, A = [5 6 4 7 11 12 8]. If we perform the *B = (A mod 2)* then [6 4 12 8] fall into the same space in new array B. However, when we perform *C = (A mod 3)*, then only [6 12] falls into the same space. Similarly, we can construct D, E, F, *etc.*, with different mod functions. Next, we can analyze B, C, D, E, F... to recover A. Analogous to the mod function, we perform multiple sampling operations on the aliased signal to create sub-signals S_i = downsample (S, f_i), where S_i are downsampled signals with corresponding sampling frequency f_i. Next, we analyze S_i to recover the original signal's different frequency components. This formulates the hard optimization problem. In *ReSense*, we don't solve this problem as the explicit optimization problem but use the suitable deep learning framework to achieve a similar result. Residual network [6] allows the connection between different convolution layers, where each successive layer contains the max-pooling function that acts as the down-sampling layer. Skip connection then allows the connection between down-sampled data and original data, which is equivalent to mixing different S_i to obtain the actual frequency components. Figure 4 illustrates the anti-aliasing process and Figure 5 shows the residual network architecture with fully connected layers.

► **Network for Single Frequency:** Data samples with single frequency component [11] has a center frequency ranging between [-600, 600] MHz with a wider bandwidth. Only 8 cosets are used to record data samples, which creates an input size of 1024 × 16. Subsequently, the I-Q signal from these cosets is resized and passed through a feature extraction network composed of convolution and pooling layers. The output features are then passed through multiple fully connected layers to predict a one-hot coded vector (1 × 13) that is generated by the softmax function. This vector is then used to determine the modulation type of the signal.

► **Network for Two Frequencies:** We also have two signals with different frequencies sampled with low sampling ADCs. The range [-600, 600] MHz is divided into 24 sub-bands where two distinct signals are placed in two of them. Sampling ADCs provide the same sampling frequency and the same number of cosets of a single frequency; thus, the number of samples remains the same (1024 × 16). However, the sampled signal includes the aliased version of both signals 1 and signal 2 (see Figure 2). First, we use the residual network to extract features from the input signal and tune network parameters to recognize both signals' location and type of modulation. The output prediction is a 1 × 24 vector where only two elements have a value between 1 and 13, the rest being zero. This vector is then converted into a 1 × 4 vector by recording the index and value of each modulation type. This reduces the size of the output and the burden of the network. Additionally, we convert a vector of 1 × 4 into one-hot coding to enhance classification. The network's output is [1 × 24, 1 × 13, 1 × 24, 1 × 13].

► **Loss Functions & Training:** We train neural networks using Mean Squared Error (MSE) loss to compare predicted and actual values. Non-linear Relu activation functions help adjust the network's parameters for accurate modulation type prediction. AdamW optimizer is used for training, and the best model is stored for inference. The training process took about 72 hours on a server with NVIDIA RTX A6000 GPU machine and 256 GB RAM.

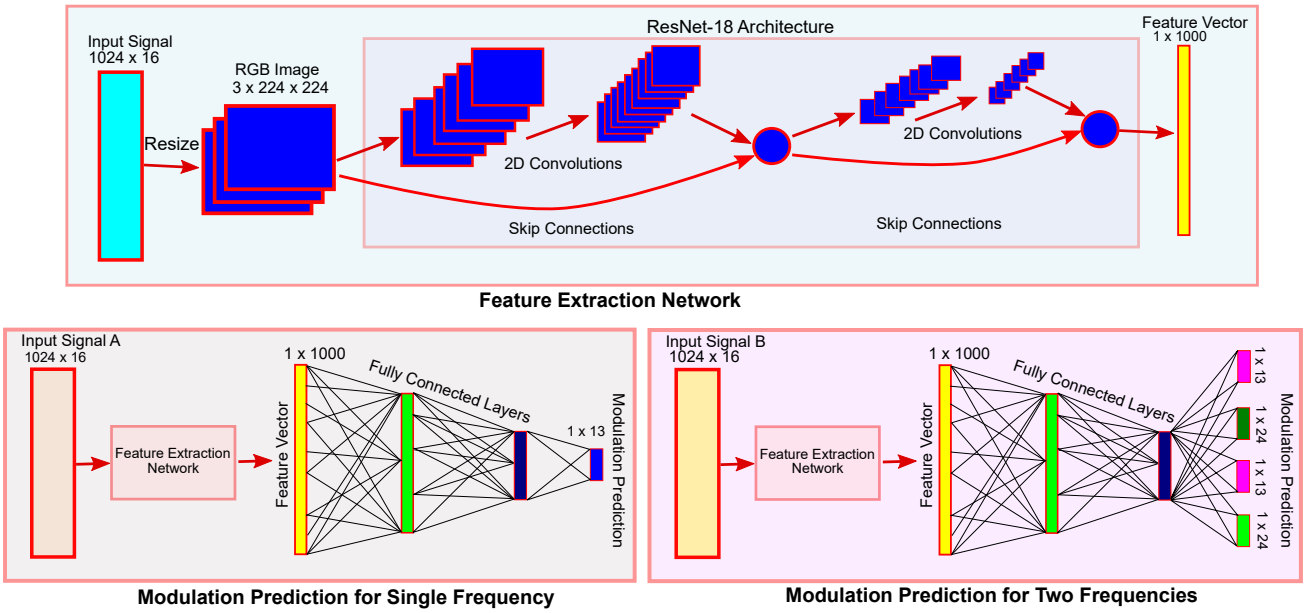

Fig. 5: ReSense*'s residual network architecture for anti-aliasing and compressed sensing.*

IV. Results & Discussion

We assess the performance of the trained models for datasets with single and double frequencies. With the test dataset, we first predict the modulation type and then compare the predicted modulation type to the ground-truth modulation type. This allows us to calculate the accuracy. For datasets with two frequencies, we separate the accuracy into two categories; index accuracy and value accuracy. The index accuracy measures how accurately the trained model can identify the signal position in the 24 sub-bands. The value accuracy calculates whether the modulation types are correctly predicted. We only count accurate classification for two frequency signals when we accurately predict index and value. We use $\sim$ 31K test samples for single frequency and predict modulation type with more than 95% accuracy. Similarly, we use 10K test samples for two frequencies and get an overall accuracy of 40%. However, we separate the accuracy into index and value accuracy, where we determine the index of frequencies with 99% accuracy and modulation type with 41% accuracy. We can improve the two frequencies with other attention-based networks that can separate and recover two signals; however, they might require more memory and inference time than our current implementation.

V. Conclusion

In this work, we present a deep learning-based spectrum sensing system for signal with one and two frequencies. *ReSense*'s residual network architecture combines the aliased signal at various sub-sampling spaces to predict the actual frequency component of the signal for spectrum sensing. High accuracy in spectrum sensing allows *ReSense* to be deployed in mobile devices to detect unused frequency with low-cost sampling ADCs of the mobile devices for better spectrum utilization.

VI. Acknowledgments

We sincerely thank the reviewers for their comments. This work is partially supported by the NSF under grants CAREER-2144505, CNS-1910853, and MRI-2018966.

References

[1] Z. Song, et al., "Real-time Multi-Gigahertz Sub-Nyquist Spectrum Sensing System for mmWave," in *ACM mmNets*, 2019.

[2] H. Hassanieh, et al., "GHz-wide sensing and decoding using the sparse Fourier transform," in *IEEE INFOCOM*, 2014.

[3] Z. Tian, et al., "Compressed sensing for wideband cognitive radios," in *IEEE ICASSP*, 2007.

[4] D. Arianandа, et al., "Multi-coset sampling for power spectrum blind sensing," in *IEEE DSP*, 2011.

[5] A. Estebsari, et al., "Single residential load forecasting using deep learning and image encoding techniques," *Electronics*, vol. 9, no. 1, 2020.

[6] K. He, et al., "Identity mappings in deep residual networks," in *ECCV*, 2016.

[7] B. Gu, et al., "Enhanced reinforcement learning method combining one-hot encoding-based vectors for CNN-based alternative high-level decisions," *Applied Sciences*, vol. 11, no. 3, 2021.

[8] GP Aswathy, et al., "Sub-Nyquist wideband spectrum sensing techniques for cognitive radio: A review and proposed techniques," *AEU-International Journal of Electronics and Communications*, vol. 104, 2019.

[9] M. Mishali, et al., "Wideband spectrum sensing at sub-Nyquist rates [applications corner]," *IEEE Signal Processing Magazine*, vol. 28, no. 4, 2011.

[10] X. Meng, et al., "End-to-end deep learning-based compressive spectrum sensing in cognitive radio networks," in *IEEE ICC*, 2020.

[11] Z. Song, et al., "GBSense," 2022. [Online]. Available: http://www.gbsense.net/challenge/