

# Stitching the Spectrum: Semantic Spectrum Segmentation with Wideband Signal Stitching

Daniel Uvaydov\*, Milin Zhang\*, Clifton Paul Robinson,  
Salvatore D'Oro, Tommaso Melodia and Francesco Restuccia

Institute for the Wireless Internet of Things, Northeastern University, United States  
Email: {uvaydov.d, zhang.mil, robinson.c, s.doro, t.melodia, f.restuccia}@northeastern.edu

\* These authors contributed equally to this work

**Abstract**—Spectrum has become an extremely scarce and congested resource. As a consequence, spectrum sensing enables the coexistence of different wireless technologies in shared spectrum bands. Most existing work requires spectrograms to classify signals. Ultimately, this implies that images need to be continuously created from I/Q samples, thus creating unacceptable latency for real-time operations. In addition, spectrogram-based approaches do not achieve sufficient granularity level as they are based on object detection performed on pixels and are based on rectangular bounding boxes. For this reason, we propose a completely novel approach based on *semantic spectrum segmentation*, where multiple signals are simultaneously classified and localized in both time and frequency at the I/Q level. Conversely from the state-of-the-art computer vision algorithm, we add non-local blocks to combine the spatial features of signals, and thus achieve better performance. In addition, we propose a novel data generation approach where a limited set of easy-to-collect real-world wireless signals are “stitched together” to generate large-scale, wideband, and diverse datasets. Experimental results obtained on multiple testbeds (including the Arena testbed) using multiple antennas, multiple sampling frequencies, and multiple radios over the course of 3 days show that our approach classifies and localizes signals with a mean intersection over union (IOU) of 96.70% across 5 wireless protocols while performing in real-time with a latency of 2.6 ms. Moreover, we demonstrate that our approach based on non-local blocks achieves 7% more accuracy when segmenting the most challenging signals with respect to the state-of-the-art U-Net algorithm. We will release our 17 GB dataset and code.

## I. INTRODUCTION

The demand for mobile connectivity is increasing spectrum congestion to unprecedented levels [1], with the number of mobile devices expected to reach 64 billion by 2025 [2], and global mobile data traffic to reach 288 exabytes per month by 2027 [3]. To improve spectrum usage, technologies such as *spectrum sharing* will become fundamental components of next-generation wireless systems [4]. A fundamental problem in spectrum sharing is determining in real time which spectrum bands are currently underutilized, also referred to as *spectrum sensing* [5]. Traditionally, spectrum sensing has focused on determining whether a given channel is busy or free [6–11]. This, however, does not allow to implement effective spectrum policies where wireless technologies have different access priorities [12]. For this reason, a much more relevant – though significantly more challenging – problem is determining which wireless technology is utilizing a given portion of the spectrum over time and frequency, also known as *spectrum classification* [13]. To this end, recent work has demonstrated that data-driven

approaches based on deep learning (DL) can be effective in addressing spectrum classification tasks [11, 12].

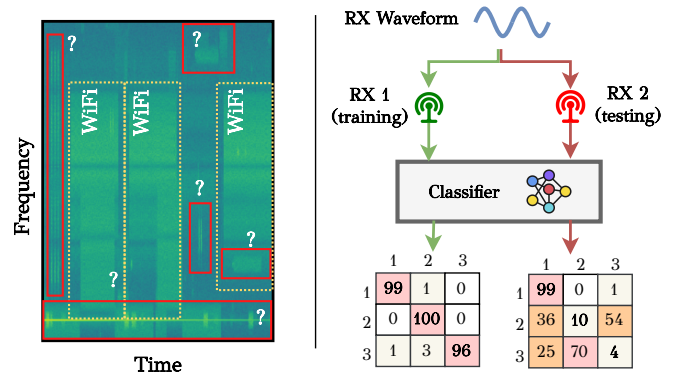


Fig. 1: (Left): Example of why labeling spectrum is a hard problem; (Right): A waveform classifier may perform well on data collected by the same device used during the training phase (RX 1), but may show poor performance when tested on data collected by a different device (RX 2).

Existing work on DL for spectrum classification—discussed in Section II—suffers from a number of critical issues. Most importantly, prior work has mostly relied on simulations and/or small-scale experimental datasets to evaluate performance [14–18], this is not without reason. Indeed, labeling real-world wideband spectrum is extremely challenging due to the coexistence of different signals in the same spectrum bands. Figure 1 (left) shows an example of a spectrum capture obtained through our testbed—described in Section V-C—in the 5 GHz industrial, scientific and medical (ISM) band. Although it is relatively easy to properly label Wi-Fi transmissions, the collected data also shows several interfering signals at different frequencies whose type and location is not available a priori, which makes their labeling hardly feasible.

In addition, the lack of diverse datasets necessarily leads to poor performance under dynamic channel conditions [19, 20]. An illustrative example is shown in Figure 1 (right), where a waveform classifier achieves very high accuracy when tested with data collected by one of the devices used to generate the training dataset (i.e., RX 1), but performs poorly when tested with data collected over-the-air (OTA) by a new and unseen device (i.e., RX 2) whose data was never included in the training dataset. Although it has been demonstrated that feeding as much diverse data (in terms of channel, mobility, and traffic

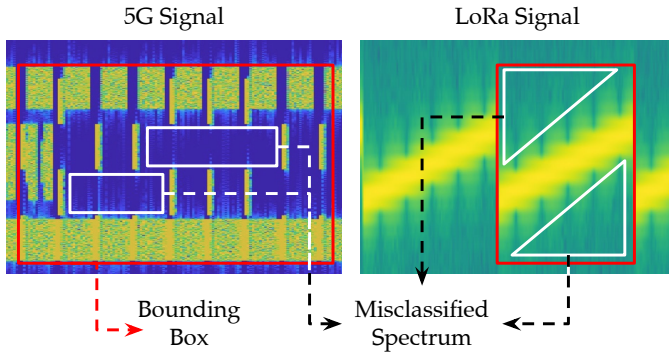


Fig. 2: Object detection requires the construction of an image and does not achieve fine-grained I/Q granularity.

conditions) makes DL solutions more reliable, how to collect such data is an open challenge [21].

Lastly, existing work mostly leverages object detection algorithms such as You Only Look Once (YOLO), which are used out-of-the-box from computer vision applications [22, 23]. This approach has several drawbacks. First, it requires the creation of an image out of in-phase/quadrature (I/Q) samples, thus incurring additional latency. This is a critical issue since spectrum needs to be classified almost instantaneously to provide meaningful and actionable inference – for example, to detect and fill a spectrum hole. Moreover, modern wireless signals such as 5G and LoRa can hardly conform into square boxes, as shown in Figure 2. Thus a significant amount of spectrum will be incorrectly classified as occupied, thus leading to poor spectrum efficiency if opportunistic and/or multi-tier spectrum access policies are being implemented [24].

#### Summary of Novel Contributions

In this paper, we address the above three issues by proposing the following novel contributions:

(1) We present a novel dataset generation pipeline for wide-band spectrum sensing applications, which allows us to generate large-scale datasets that (i) contain signals collected OTA and are affected by real-world channel conditions; (ii) can be completely labeled; and (iii) can be generated in a time-efficient manner. Our approach is based on “stitching” different signals together to create samples where signals are overlapping and affected by real-world noise and interference. We show that this pipeline can generate a virtually infinite amount of labeled data starting from only 17 GB of OTA data captured using only two radios in 3 hours and for 5 different wireless technologies (i.e., WiFi, LTE, BLE, LoRa, and ZigBee);

(2) We develop a novel custom DL algorithm for multi-label multi-class spectrum sensing based on *semantic segmentation*. Although our algorithm is structured based on the widely known U-Net network for image segmentation [25], it presents a few key differences that make our network perform better for spectrum segmentation. Conversely from existing spectrum sensing work, our approach (i) operates at the I/Q level, instead of creating images; (ii) classifies each and every I/Q sample incoming from the ADC, without creating any bounding boxes,

thus increasing classification accuracy significantly; (iii) it only uses 1024 I/Q samples as input, which leads to very low inference time. Moreover, we show that our network achieves 7% more accuracy than U-Net in the most challenging protocols (Wi-Fi and LTE) while maintaining similarly low latency;

(3) We demonstrate experimentally how the proposed semi-augmented dataset generation pipeline can make DL-based solutions more robust against changing conditions and capable of generalizing across different scenarios and deployments. Specifically, we show how the pipeline makes the above wide-band spectrum sensing algorithm more accurate and able to deliver high accuracy even when operating with data collected by different devices, sampling rates, antennas, and under previously unseen network conditions. Our results show that we localize and classify 5 wireless protocols with a mean Intersection over Union (IoU) of 96.70%. We also performed real-world experiments with GPUs and we show that our algorithm takes only 2.6 ms of latency to process 100 MHz of spectrum.

## II. BACKGROUND AND RELATED WORK

DL has been identified as the ideal candidate to solve a variety of challenges in the wireless domain, where the RF environment is hard to model using closed forms due to its non-deterministic behavior [26, 27].

Since DL-based solutions require large quantities of data to be effective, one approach that finds broad application in the literature is that of using simulators and emulators to generate large synthetic datasets aiming to capture the different behavior of waveforms transmitted OTA under diverse channel conditions [28]. Although entirely relying upon synthetic data is sufficient to validate complex DL algorithms and demonstrate their potential in solving complex networking tasks, this approach does not necessarily transfer well to real-world applications where the input data consists of signals collected OTA [14–18].

Indeed, it has been shown that utilizing signals collected via experimental data collection campaigns allows DL models to better generalize and achieve higher accuracy [29, 30]. For example, [29, 30] design and train a DL classifier that uses OTA data to perform waveform classification. Despite demonstrating the benefits and importance of using OTA data, these works only produce a single-class classification outcome and focus on modulation recognition tasks only, thus not providing complete information on spectrum utilization and resulting in lower accuracy when handling interfering and coexisting signals.

Some more recent work aims at localizing wireless signals in the spectrum using well-established computer vision approaches meant for object detection, such as YOLO [22, 23]. As mentioned before, while object detection via YOLO is indeed a viable method for computer vision tasks, this approach does not achieve the level of resolution or granularity required for wireless signals. Wireless signals do not necessarily fit neatly into square bounding boxes (i.e. chirp spread spectrum signals). Furthermore, there are wireless technologies whose signals do not fully utilize their allocated spectrum (i.e. LTE, 5G) and bounding boxes can not accurately depict the under-utilization of these technologies, as shown in Fig. 2. This is especially

crucial in applications of opportunistic spectrum sharing where underutilized portions of the spectrum can be taken advantage of to increase spectrum efficiency.

Another line of work focuses on using data augmentation techniques to diversify and extend the scope of datasets collected OTA. Specifically, [31, 32] develop data augmentation pipelines for wireless signal classification with a specific focus on single-label classification tasks, where the augmentation process is designed to emulate the effect of diverse channel effects and noise levels on collected signals. If compared to ours, the above works focus on single-label classification tasks and do not address the problem of locating and characterizing multiple potentially overlapping wireless signals in wideband applications.

The closest paper to ours is the very recent work from Nguyen *et al.* [23], where authors curate a OTA dataset that is extended via data augmentation techniques to address wideband spectrum sensing tasks. In [23], authors develop a spectrum sensing algorithm that uses YOLO-based object detection to identify and localize different waveforms in real-time and in-the-wild (ITW). Despite the approach being very similar to ours at a high level, [23] utilizes YOLO which produces extensive bounding boxes that might include either empty portions of spectrum or signals belonging to a different class. Moreover, despite the authors of [23] demonstrating exceptional accuracy results, they do not evaluate the portability and generalization of the solution against different radios, sampling rates, and in environments where transmissions are generated by devices not being controlled by the authors.

**Our work separates itself from the existing literature in that:** (i) it introduces a semi-augmented data generation pipeline that has been designed to facilitate the generation of OTA wideband datasets including waveforms with diverse technologies, channel and noise conditions, sampling rates, transmission antennas, and center frequency; (ii) it presents a novel multi-class DL-based semantic spectrum segmentation algorithm capable of delivering high-resolution spectrum sensing capabilities that go beyond inaccurate bounding boxes used in YOLO-based solutions, thus offering a more accurate classification and localization of signals and unused portions of the spectrum; and (iii) we validate our work following a purely OTA approach with signals collected ITW with multiple radios, sampling rates, center frequencies, and RF environments.

### III. SEMI-AUGMENTED DATASET GENERATOR

Accurate DL-based spectrum sensing heavily relies on the availability of properly labeled and diverse datasets. However, how to build such datasets via OTA data collection is resource- and time-consuming. This problem is further exacerbated when the inference objective is not simply classification (e.g., recognizing the modulation of a signal, or determining the presence of a specific waveform), but requires a more fine-grained output. Among others, a relevant spectrum sensing task is *multi-label multi-class spectrum sensing*, where the goal is to monitor a portion of the spectrum and identify all signals that are being transmitted, their bandwidth, technology, and center frequency.

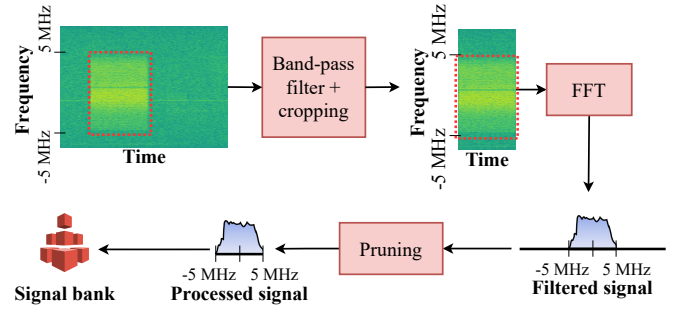


Fig. 3: Pre-processing for frequency synchronized signals

The main challenge is that data must be collected OTA so as to capture realistic channel conditions and prepare the Artificial Intelligence (AI) to operate correctly in a wireless deployment with real radios. Additionally, nodes performing spectrum sensing might not always be aware of where signals are being transmitted, and their center frequency will rarely match that of the signals the sensing node is trying to detect, classify and locate. In Section V-C, we demonstrate how training on data that always assumes synchronization between sensing and transmission center frequencies results in poor accuracy.

Undoubtedly, one could perform a large OTA data collection campaign. However, such a task would be extremely hard and time-consuming due to (i) the several many possible combinations to be covered in time and frequency, which exponentially grow in the case of wideband spectrum sensing; and (ii) the need for large portions of spectrum without any interference from external systems. Our goal is to mitigate the complexity via a combination of OTA data collection and data augmentation.

**Approach in a nutshell.** First, we create a *signal bank* (Section III-A), i.e., a dataset containing individual signals collected OTA in the absence of interference. This serves as a seed to create much larger and more diverse datasets. Then, our generator (Section III-B) extracts multiple signals from the signal bank and combines them together (via time and frequency shift mechanisms) to create a semi-augmented “stitched signal” portraying a more realistic OTA data capture. These signals will interfere with each other, be in randomized frequency locations, and will have already experienced the common channel effects (fading, multipath, etc.) during OTA collection.

#### A. Signal Bank Generation

The signal bank consists of a labeled collection of individual signals collected OTA when (i) only one signal is transmitted at a time; (ii) the center frequency and bandwidth are known a-priori; and (iii) data collection is performed over a limited and small portion of the spectrum that is constantly monitored to ensure the lack of interference from external systems.

To be added to the signal bank, signals first go through a pre-processing pipeline shown in Fig. 3. Specifically, signals are broken up into shorter signals that are (i) cropped to contain only the actual signal transmission (e.g., by removing the silence period before and after data transmission); and (ii) bandpass-filtered to only extract the signal of interest and re-



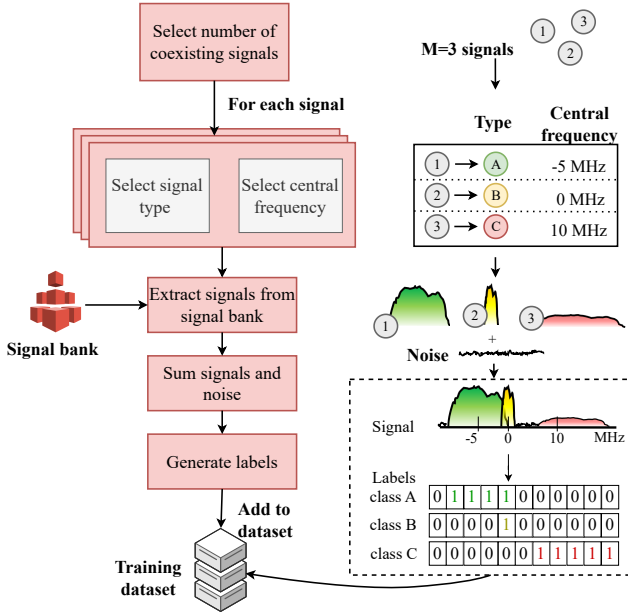


Fig. 4: Generating a single training sample for the dataset.

move any undesired signals that may have been recorded. Once signals have been pre-processed, they are (iii) converted to the frequency domain through a Fast Fourier Transform (FFT). Finally, being in the frequency domain, (iv) any frequency components outside of the band of interest occupied by the signal are pruned, and the remaining I/Qs are added to the signal bank. This procedure is repeated multiple times for each signal type so as to generate multiple instances of the same signal with diverse duration and spectrum occupancy. Note, we do not remove any small interference or channel effects that may overlap with the signal of interest during recording, we consider this to be commonplace effects that may happen on a signal and will add to the generalization ability later. The combination of signals in the dataset generator will account for the more large scale interference that will be experienced OTA.

## B. Dataset Generation

The next step of the semi-augmented dataset generator pipeline is combining multiple signals to generate a “stitched” wideband signal to be added to the training dataset. The steps are outlined in Fig. 4 (left), while Fig. 4 (right) shows an illustrative example of a sample and its corresponding label.

The Dataset Generator is given a set of parameters such as the total number  $C$  of signal types (i.e., WiFi, LTE, BLE, among others) that are present in the signal bank; the desired observable bandwidth  $B$  (i.e., the field of view) of the receiver; the maximum number  $n_s$  of signals which can be simultaneously present in  $B$  at a given time; the probability  $p_e$  that the entire observable bandwidth is empty; and the probability  $p_c$  that any one of the signals is located at the center frequency. Similarly to real-world spectrum, our pipeline allows signals to overlap both partially or completely with other signals.

For each composite signal, we first generate the number  $M \leq n_s$  of signals that will be injected into the observable

bandwidth. This number is generated at random as follows:

$$M = \begin{cases} 0, & p_e \\ U^{\text{int}}(1, n_s), & 1 - p_e \end{cases} \quad (1)$$

where  $U^{\text{int}}(a, b)$  represents the value of a uniformly distributed integer random variable taking values in the range  $[a, b]$ . From (1), the band is completely empty with probability  $p_e$ , or it contains a number  $M$  of signals uniformly chosen between 1 and  $n_s$ . The probability  $p_e$  makes it possible to generate examples that mimic diverse traffic conditions. To emulate a congested spectrum like the one in the ISM band,  $p_e$  should be low so as to reflect the fact that such bands are frequently used by several devices at the same time. Similarly, less crowded scenarios can be emulated using a value of  $p_e$  closer to 1.

Upon randomly generating the number  $M$  of signals to be included in the current example, the pipeline proceeds in assigning a target class to each of them. Specifically, let  $\mathcal{C} = 0, 1, \dots, C$  be the set of  $C + 1$  possible class labels (e.g., where the class 0 is reserved for empty portions of the spectrum and is not included in the signal bank), the target class  $c_m$  of the  $m$ -th signal is randomly generated according to  $c_m \sim U^{\text{int}}(1, C)$ . Note that  $c_m$  does not include the class “0” as unused portions of the spectrum will be computed in the last step of the pipeline. Upon determining the class  $c_m \in \mathcal{C} \setminus \{0\}$  of signal  $m$ , the next step consists in extracting at random one instance from the signal bank and placing it in the spectrum. The positioning of the signal strongly depends on its bandwidth  $b_m$ , and its center frequency  $f_m$ .  $f_m$  is randomly chosen so as to ensure that at least a portion of the signal appears within the observable band  $[-B/2, B/2]$ . Specifically,

$$f_m = \begin{cases} 0, & p_c \\ U^{\text{cont}}(-\frac{B}{2} - \frac{b_m}{2}, \frac{B}{2} + \frac{b_m}{2}), & 1 - p_c \end{cases} \quad (2)$$

where  $U^{\text{cont}}(a, b)$  represents the value of a uniformly distributed continuous random variable in the range  $(a, b)$ .

From (2), the signal is centered at 0 Hz with probability  $p_c$ , or it is centered at any frequency in  $(-\frac{B}{2} - \frac{b_m}{2}, \frac{B}{2} + \frac{b_m}{2})$  to ensure that the signal is at least partially present in the observable band.

Once all of the  $M$  signals have been generated and positioned in the spectrum, they are all combined together via an additive operation (or stitching). Finally, background noise measured OTA from an empty wireless channel is also added as a “background” to fill empty spectrum portions. Upon generation of the sample, we produce its label (see the bottom right part of Fig. 4) and both are stored in the dataset. Since our goal is to produce a high-resolution and accurate labeling, labels were structured as a matrix  $\mathbf{L}$  of dimension  $C \times n_{iq}$ , where  $n_{iq}$  is the number of I/Qs in frequency fed as input to the DL model. For any given label, each row  $i$  corresponds to a class in  $\mathcal{C}$ , and each column  $j$  corresponds to a sub-band  $k = 1, 2, \dots, n_{iq}$  of the observable bandwidth. Therefore, when a class  $i$  is present in a specific sub-band  $k$ , the generic element  $l_{i,j}$  of the matrix  $\mathbf{L}$  is such that  $l_{i,j} = 1$  if a signal of class  $i$  is present in sub-band  $j$ ,  $l_{i,j} = 0$  otherwise. Thus, the resolution of our classification (e.g., how small of a sub-band we are able to classify) is



therefore determined by  $n_{iq}$ . For example, if  $B = 25\text{MHz}$  and  $n_{iq} = 1024$  then we break the observable band into  $n_{iq}$  bins of size  $F = B/n_{iq} \approx 24\text{kHz}$ .

#### IV. SEMANTIC SPECTRUM SEGMENTATION

In this section, we showcase one particular application of practical relevance where the proposed semi-augmented data generation pipeline can effectively enable the training of DL-based solutions for spectrum sensing tasks that are capable of generalizing and delivering high accuracy and reliability. Specifically, we focus on the case of wide-band *multi-label, multi-class spectrum sensing* where the goal is to accurately detect, characterize and localize multiple wireless signals with possibly diverse waveforms, bandwidths, power levels, and center frequencies but all coexisting (and possibly overlapping) within the same band of interest.

Since multi-label, multi-class spectrum sensing aims at producing an accurate report of spectrum utilization, in practical applications it can be used to identify which portions of the spectrum are being used by a specific technology (e.g., WiFi, Bluetooth, and/or incumbents), while at the same time detect spectrum holes for opportunistic spectrum access in real-time.

##### A. Proposed Approach

In contrast to conventional wideband spectrum sensing, we pursue a different approach and apply *semantic segmentation* [25]. This approach is well-established in the computer vision community as an effective tool to determine the shape of different elements in images or video frames. If compared to YOLO, which outputs a rectangular box around a specific object, semantic segmentation offers a much higher resolution by identifying all pixels that represent the object, therefore better conforming to the object's shape.

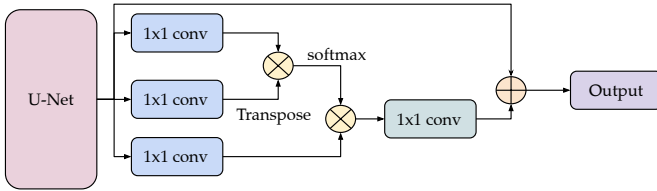


Fig. 5: Proposed Semantic Spectrum Segmentation architecture utilizing U-Net [25] with a non-local block

In the context of spectrum sensing, semantic segmentation allows to detect multiple signals simultaneously and in one stage. Among other advantages, semantic segmentation is implemented as a single DL classifier that can be optimized end-to-end by feeding I/Q samples in the frequency domain as input.

The semantic spectrum segmentation model architecture is shown in Fig. 5. Our architecture is inspired by U-Net [25] to handle the majority of the feature extraction. To feed I/Q in the frequency domain to the network, we adapt U-Net by converting 2D Convolutional Neural Network (CNN) kernels to 1D. Our U-Net-based semantic spectrum sensing network consists of five encoding and decoding blocks. The encoding block is a stack of two 1D convolutional layers with a kernel

size  $1 \times 3$  followed by batch normalization and ReLU activation. A maxpooling layer is used to downsample the encoded features. The decoding block takes both output features from its previous layer and the encoded features from a skip connection. To match the dimensionality of both inputs, an upsampling layer followed by a convolutional kernel is applied to the downsampled features. Feature dimensions are 64, 128, 256, 512, and 1024 for the five encoding and decoding blocks, respectively.

1) **Non-local Block:** One of the issues with CNNs is that they tend to process groups of features locally, which makes it difficult to capture spatially distant information without the use of feature merging techniques such as the use of pooling layers. Since U-Net has neither global pooling layers nor fully connected layers to mix up features globally, utilizing U-Net in its original form results in performance loss, especially when dealing with wideband signals in the presence of interference. To improve the system performance and increase accuracy, we apply a Non-local block [33] after the final decoding block of U-Net so as to integrate a self-attention process. This is achieved via the architecture shown after U-Net in Figure 5. Specifically, three  $1 \times 1$  convolutional kernel are used to encode input as Queries  $\mathbf{Q}$ , Keys  $\mathbf{K}$ , and Values  $\mathbf{V}$ , respectively. The self-attention is instead defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right) \mathbf{V} \quad (3)$$

where  $\sqrt{d}$  is the embedding dimension of  $\mathbf{Q}$  and  $\mathbf{K}$ . The self-attention computes a weighted average of the whole encoded input features  $\mathbf{V}$  based on an attention map  $\text{softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d})$ .

Such an attention technique can apply to every encoding and decoding block in U-Net to mix up features globally and improve performance. However, the non-local block requires two matrix multiplication over the whole spatial and feature dimension, which is a relatively computationally heavy task. This complexity might result in long inference times that might make the sensing output obsolete by the time it is computed. Since our goal is to develop a system that can operate in real-time and in an actual wireless deployment, we have decided to add a non-local block between the last decoding block and the final multi-labeling layer only.

2) **Extension to multi-label semantic segmentation:** One of the main aspects that make semantic segmentation for spectrum sensing applications significantly different from its traditional application to computer vision tasks is that objects in the foreground always hide those in the background, but the same does not hold for Radio Frequency (RF) applications. Pixels in a picture only carry information about foreground objects. However, two wireless signals can overlap in both time and frequency domains and coexist at the same time without necessarily being hidden by other signals in the same frequency bands. The conventional semantic segmentation approach can only assign a single class to each pixel (e.g., the class of the object in the foreground or the signal with the highest power), therefore it cannot be used directly to perform spectrum sensing tasks as it would necessarily result in misclassifications when applied to overlapping wireless signals

To solve this problem, we have extended the traditional semantic segmentation architecture to consider the much more complex task of multi-label multi-class classification where the output of the system is a multi-dimensional binary mapping between each I/Q sample and all possible classes. More specifically, we have extended the U-Net architecture such that the input is represented by a sequence of I/Q samples in the frequency domain with size  $2 \times n_{iq}$ , where  $n_{iq}$  denotes the number of frequency bins or I/Qs (i.e., the resolution of the classifier), and 2 denotes the real and imaginary part of the complex-values I/Q samples. A  $1 \times 1$  convolutional kernel at the last layer will generate a  $C \times n_{iq}$  output, where  $C$  denotes the number of classes. Thus, each row of the output is a binary segmentation map where, for each I/Q sample, we can extract information on which classes are present or absent in the specific frequency bin.

3) **Achieving Generalization:** While the dataset generator helps generalize across different bandwidths, center frequencies, and wireless protocols, noise estimation helps generalize across different wireless devices or radios. Specifically, during training, we record the minimum values of the smoothed signal power in the frequency domain of each sample and use the average of those values as a reference estimate of the noise floor. When testing the model, we multiply the input signal by a factor that, similarly to normalization, is computed to shift the minimum signal power to the same level as that we have estimated across the training dataset. However, such an estimate assumes that there will be an empty hole in the spectrum that only contains noise (this is very common in wideband spectrum sensing scenarios). A more sophisticated noise estimation approach may help to further improve the generalization.

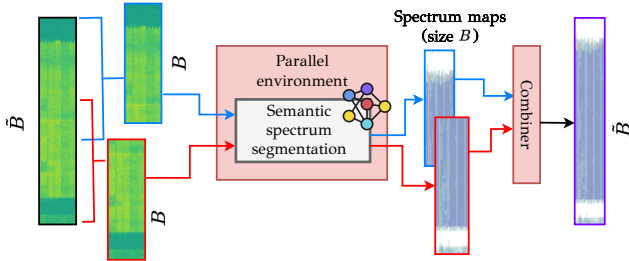


Fig. 6: Scalable and portable pipeline for processing larger bandwidths without retraining the segmentation classifier.

4) **Scalable and Portable Wide-band Processing:** As mentioned in previous sections, the resolution of our approach is  $F = B/n_{iq}$  and depends on both the size of the observable bandwidth  $B$  and a number of I/Q samples in the frequency domain (e.g., the number of sub-bands used to split the bandwidth  $B$ ). Since the semantic spectrum segmentation classifier is trained to process an observable band  $B$ , it can indeed process signals with smaller bandwidths, but cannot process those with a larger bandwidth due to a mismatch between the size of the input signal, and that of the classifier. To support increasingly larger bandwidths, there are two approaches. A naive one would be to generate a new dataset with a larger observable

bandwidth and train a new classifier. A more scalable and portable approach, which is the one we follow in our architecture, is instead to leverage parallel processing (e.g., GPU parallelism) and developed a pipeline that can be adapted to any observable bandwidth without requiring to retrain the classifier. The pipeline is illustrated in Fig. 6. Whenever we receive an input covering a portion of spectrum  $\tilde{B}$  that is larger than the observable bandwidth  $B$  of the classifier, we first collect  $\tilde{n}_{iq} = \tilde{B}/B \cdot n_{iq}$  samples to process. We then generate a set  $N$  of partially overlapping samples from the original input such that each sample covers only a portion of size  $B$  of the input signal. Each individual sample is then processed individually by our classifier to obtain  $N$  outputs of size  $n_{iq}$ . These are then combined together via averaging to produce a final output of size  $\tilde{n}_{iq}$ . The advantage of this pipeline is that (i) it is scalable in that it can leverage parallel processing to speed up the inference time for inputs covering larger bandwidths than  $B$ ; and (ii) it is portable and general as it can be used for virtually any input of variable bandwidth  $\tilde{B}$ , thus making it portable.

## V. PERFORMANCE EVALUATION

### A. Generating the experimental dataset

To demonstrate the effectiveness of our approach, our signal bank is generated by collecting data for three hours across three days and using only two radios, a USRP X310 (i.e., the transmitter) and a USRP N320 (i.e., the receiver). Data is collected using five classes ( $C = 5$ ) where each class corresponds to a wireless protocol/technology: WiFi, LTE, BLE, LoRa, ZigBee. The spectrum sensing receiver utilizes a sampling rate of 25MHz which is also our chosen observable bandwidth ( $B = 25MHz$ ). We set  $n_s = 2$  and  $p_e = 0.05$  to mimic the case of a congested ISM band where the chances of having an empty channel are very small. Finally, we set  $p_c = 0.5$  as we aim at emulating receivers tuned on commonly used center frequencies (e.g., 2.437 GHz for WiFi channel 6, 2.655 GHz for LTE band 7).

For our data collection, we consider a laboratory area that hosts Arena [34], a 64-antenna indoor testbed with ceiling-mounted antennas distributed according to an  $8 \times 8$  grid covering a 2240 square feet office space in our campus. In Arena, antennas are connected to USRP X310 and N320 radios which, together with the ceiling-mounted antennas, are used to collect the OTA data. We also consider two other areas open to the public, i.e., a kitchen area and the main lobby of a building that hosts classes, a cafeteria, and several study areas. These three areas are characterized by diverse RF channel conditions.

The hardware setup used to collect the data and the data collection locations are illustrated in Fig. 7. The data used to generate our signal bank has been collected in the laboratory setup only, while the data we use in this section to evaluate our solution has been collected in all of the three locations. For the signal bank, we collect data around the 900 MHz, 2.4GHz, and 2.6GHz bands. We find 25MHz wide bands with minimal interference and collect data while monitoring the spectrum to remove data collected in the presence of strong interference.

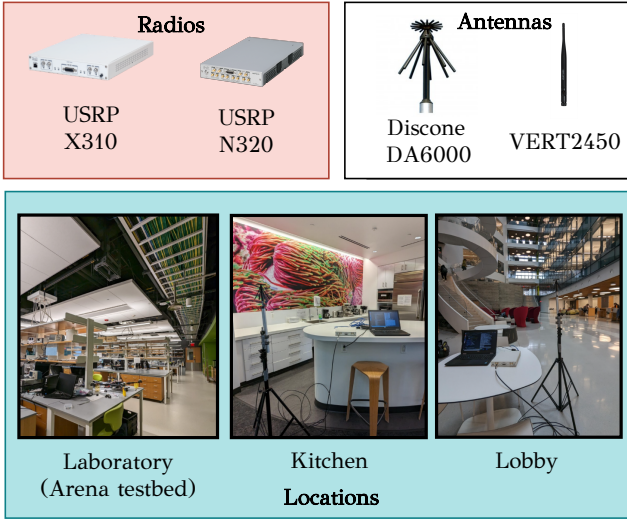


Fig. 7: Setup used for data collection with different radios, antennas, and in different locations. The signal bank has been generated in the laboratory location only, while testing data has been collected in all locations.

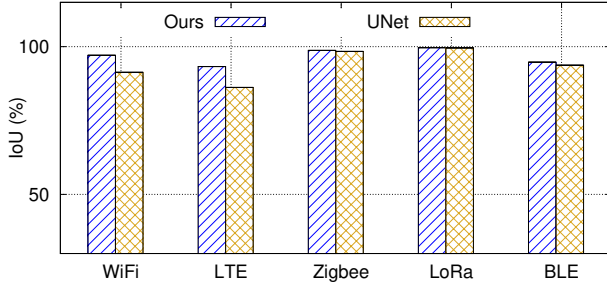


Fig. 8: IoU comparison for different models and wireless technologies.

### B. Classification Performance

To evaluate the performance of our semantic segmentation model, we compare it with the original U-Net [25]. The mean IoU are 96.70%, 93.44% for ours and U-Net respectively. Figure 8 demonstrates the IoU performance for each wireless technology, in which ours can achieve 97.08%, 93.29%, 98.72%, 99.65% and 94.74% for WiFi, LTE, Zigbee, LoRa and BLE signals. On the other hand, the U-Net has 90.31%, 85.20%, 98.40%, 99.59% and 93.72% IoU for five technologies respectively. Although our model is 3% better than U-Net in general, it can achieve 7% accuracy better than U-Net on signals with larger bandwidth such as WiFi and LTE due to the non-local block that we have introduced exactly to improve the detection of signals that occupy large portions of spectrum.

As mentioned in Section IV-A, our semantic spectrum segmentation classifier outputs a binary array of size  $C \times n_{iq}$  to determine which type of signal is present in each I/Q sample. In Fig. 9, we show two examples of inputs and outputs to illustrate the ability of our solution in detecting and individually classifying with high-resolution overlapping signals of different technologies. We show the class-specific output for the classes related to the overlapping signals only. Fig. 9 (left) shows how the classifier is able to effectively classify simultaneously overlapping WiFi and Zigbee signals, while Fig. 9 (right) shows the same but for LTE and BLE overlapping signals.

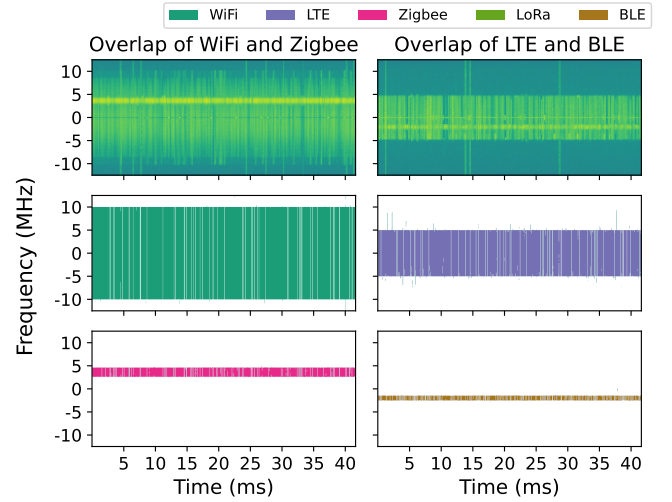


Fig. 9: Two examples of classification for overlapping signals: (Left) WiFi and Zigbee; (Right) LTE and BLE.

### C. Experimental Validation In the Wild

Our main goal is to attain reliable and effective spectrum sensing in practical use cases OTA and ITW. For this reason, we have performed an extensive data collection campaign, *purely for testing not for training*. This data spans multiple days at different locations where we have collected a variety of OTA signals being transmitted at different center frequencies, and with different sampling rates, radios, and antennas. The goal of this campaign is to collect as much data as possible to validate our solution in a variety of real-world deployments, as well as to showcase the benefits that our system design brings to the final accuracy of the spectrum sensing task.

Since the signals we have collected in this case are collected purely ITW in a completely uncontrolled RF environment shared by students, workers, visitors, and other wireless de-

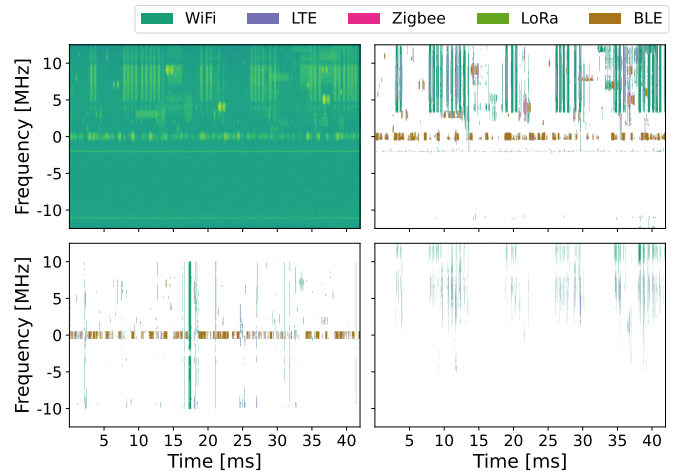


Fig. 10: Spectrum maps of our classifier trained on three different datasets and tested on data collected OTA: (Top Left) Spectrogram of data collected at 2.402 GHz (i.e., BLE advertisement channel); (Top Right): Proposed semi-augmented data generator model output; (Bottom Left): Un-augmented OTA data model output; (Bottom Right): Synthetic data model output.



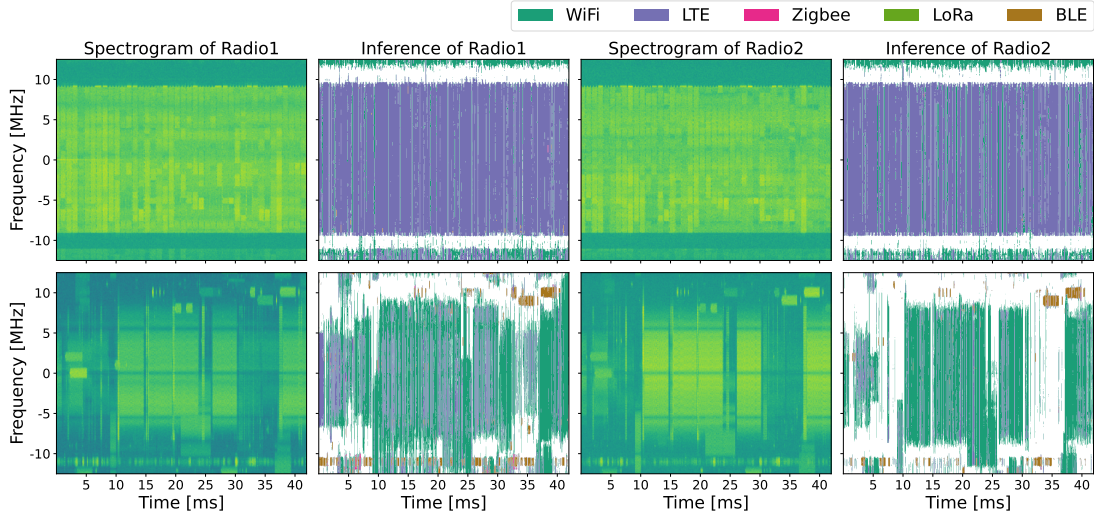


Fig. 11: Spectrum maps for data collected OTA with different radios and antennas: (Top) Data collected at 748 MHz (i.e., around LTE band 28); (Bottom): Data collected at 2.437 GHz (i.e., WiFi Channel 6).

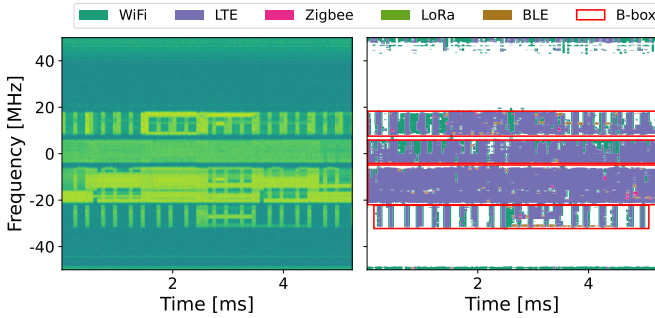


Fig. 12: (Left) Spectrogram of OTA data at 100 MHz sampling rate in 700-800MHz. (Right) Our model output with ideal object detection bounding box.

vices, it is challenging (if not impossible) to build an accurate and comprehensive ground truth for signals. Therefore, we follow the approach used in computer vision problems with unlabeled test data and resort to graphical comparison of the outputs only.

**1) Performance with Varying Training Datasets:** In Fig. 10 we show the output of our classifier for OTA spectrum measurements when the classifier is trained using three different datasets: an augmented dataset obtained using the proposed semi-augmented dataset generation pipeline described in Section III-B, the un-augmented OTA dataset used to generate the former, and a synthetic one generated in MATLAB and without any OTA data, containing artificial channel effects.

Signals are collected on the ISM band when the receiver center frequency is on a BLE advertisement channel (2.402 GHz). We see that the classifier trained on data produced by our augmented dataset generator is able to classify and localize the centered BLE signals better than the same classifiers trained using both the un-augmented and synthetic datasets. Moreover, we notice that the classifier trained over the augmented data is able to classify and localize signals not necessarily centered at the center frequency of the receiver. The classifier trained on the un-augmented dataset can classify the BLE signal to some extent but misclassifies WiFi signals even in those areas of the

spectrum where there is no activity. The classifier trained on the synthetic dataset, instead, is completely unable to classify any of the OTA signals since synthetic data does not properly capture the characteristics of wireless signals collected ITW.

**2) Performance with Varying Radios:** Fig. 11 shows the performance of the classifier on testing data collected from two different radios. Radio 1 being the USRP N320 equipped with the DA6000 antenna (which is also the same setup we used to collect the data used to generate the augmented dataset), and Radio 2 being the USRP X310 with a VERT2450 antenna. Similarly to the previous case, each row corresponds to an individual signal collected at different center frequencies and bands. In this case, the top figure shows the spectrogram of a signal collected in the LTE band at 748 MHz, while the bottom one has the same center frequency as WiFi channel 6. Indeed, we can see that the output of the classifier for both radios is very similar and show good classification performance in both cases, but with Radio 1 (i.e., the USRP N320) detecting more signals than Radio 2 (i.e., the USRP X310). For example, Radio 1 produces I/Q samples that make it easier to detect the BLE signals, while Radio 2 can detect them only partially.

**3) Performance with Varying Sampling Rates:** We also evaluate the generalization capability of the classifier by testing it on bandwidths larger than the ones used during the training process. Specifically, we utilize data sampled at 100MHz centered at 750MHz, a known LTE band. We broke up the larger bands into overlapping pieces and averaged the classifications of the overlapping segments. We decided to take this approach to show that you do not necessarily need to retrain the network for higher bandwidths and can either divide and process pieces in parallel if you have the higher sampling capability or even sweep across large bands to cover them if you do not. Fig. 12 shows the model output when tested on this data. We see there is still good localization and classification performance even as we process these higher sampling rates. The model can accurately detect and localize the LTE signals which are common in this range of frequencies. Furthermore, we also

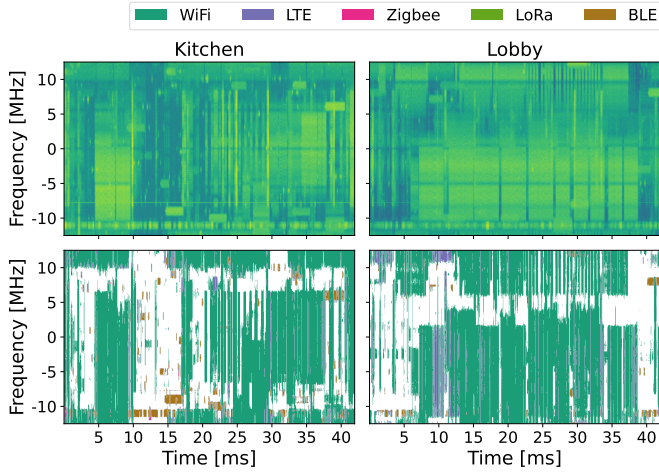


Fig. 13: Spectrum maps for data collected OTA at different locations with  $B = 25$  MHz and at 2.437 GHz: (Left) kitchen area; (Bottom): main lobby.

show a set of red bounding boxes that reflect a perfect object detection output from YOLO. It is clear to see that rectangular bounding boxes do well in locating LTE transmissions, but mark spectrum holes as LTE signals, whereas our segmentation outputs are more accurate in detecting such holes.

4) **Performance with Varying Locations:** Fig. 13 shows our model performance in two other physical locations at 2.437 GHz (i.e., WiFi channel 6). The two columns show the spectrogram and output of the kitchen and lobby. In both cases, the model accurately detects the WiFi signals dominating this center frequency and is able to also detect the BLE signals off to the side of the band. Therefore our model is resilient to changes in physical location as well.

#### D. Evaluation of Inference Time

To assess the suitability of our model to deliver real-time spectrum sensing capabilities, we evaluate its inference time on both CPU and GPU and compare it with YOLOv3 (as used in [23]) and the original U-Net model. Our results have been obtained by averaging 1000 independent measurements. Fig. 14 shows the latency tested on different bandwidths. Because YOLOv3 is generally used on 2D images, we assume an input size of  $512 \times 512$  for 100MHz, as done in [23], and scale it down linearly to  $128 \times 128$  for 25MHz. As expected, the YOLO model has the highest latency since it deals with larger input sizes and requires more complex computations. We only report the latency to compute an output from YOLOv3 without including the time needed to perform non-maximum suppression, i.e., the final step of object detection tasks where the most appropriate bounding box is chosen (which would further increase latency). The difference in latency between our model and YOLOv3 is substantial, with our model being able to compute an output up to 31% faster than YOLOv3 when running on a GPU at 100 MHz. However, when the models are executed on CPUs, our model is 72% faster than YOLOv3. On multiple ends, our model significantly outperforms YOLOv3 by a wide margin both on a CPU or GPU. Interestingly, although our model extends U-Net by adding a non-local block and a multi-label

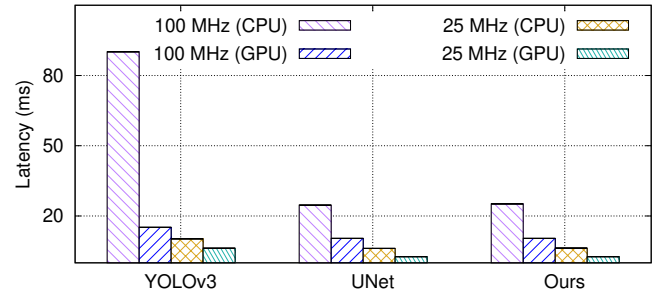


Fig. 14: Inference latency comparison on CPU and GPU for different observable bandwidth size values.

segmenter, its inference time is almost identical to that of the original U-Net model. Specifically, if compared to U-Net, our model is 0.126 ms slower on CPU (2% increase), and 0.007 ms slower on a GPU (0.25% increase).

## VI. CONCLUSIONS AND REMARKS

In this work, we have proposed a novel approach based on *semantic spectrum segmentation*. Moreover, we have proposed a novel data generation approach where a limited set of easy-to-collect real-world wireless signals are “stitched together” to generate large-scale, wideband, and diverse datasets. We have extensively evaluated our approach through experiments on multiple testbeds (including the Arena testbed) using multiple antennas, multiple sampling frequencies, and multiple radios over the course of 3 days. Our results have shown that our approach classifies and localizes signals with a mean intersection over union (IOU) of 96.70% across 5 wireless protocols while performing in real-time with a latency of 2.6 ms. Moreover, we have demonstrated that our approach based on non-local blocks achieves 7% more accuracy when segmenting the most challenging signals with respect to the state-of-the-art U-Net algorithm. Public access to our code and data is available at [https://github.com/uvaydovd/spectrum\\_sensing\\_stitching](https://github.com/uvaydovd/spectrum_sensing_stitching).

#### ACKNOWLEDGMENT OF SUPPORT AND DISCLAIMER

This work is funded in part by the National Science Foundation grants CNS-2134973, CNS-2120447, ECCS-2229472, ECCS-2146754, CNS-2112471 and ECCS-2329013, and in part by funds from OUSD R&E, NIST, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program and in the Future of Semiconductors (FuSe) program, by the Air Force Office of Scientific Research (AFOSR) under contract number FA9550-23-1-0261, by the Office of Naval Research (ONR) under award number N00014-23-1-2221, and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via [2021-2106240007]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, AFOSR, ONR, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

## REFERENCES

- [1] Federal Communications Commission (FCC), "Spectrum Crunch," <https://www.fcc.gov/general/spectrum-crunch>.
- [2] Ericsson, "Ericsson Mobility Report," Tech. Rep., November 2021. [Online]. Available: <https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2021>
- [3] 5G Americas, "5g: The future of iot," 5G Americas, Bellevue, Washington, Tech. Rep., July 2019. [Online]. Available: [https://www.5gamericas.org/wp-content/uploads/2019/07/5G\\_Americas\\_White\\_Paper\\_on\\_5G\\_IOT\\_FINAL\\_7.16.pdf](https://www.5gamericas.org/wp-content/uploads/2019/07/5G_Americas_White_Paper_on_5G_IOT_FINAL_7.16.pdf)
- [4] L. Zhang, M. Xiao, G. Wu, M. Alam, Y.-C. Liang, and S. Li, "A Survey of Advanced Techniques for Spectrum Sharing in 5G Networks," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 44–51, 2017.
- [5] Y. Arjoun and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, p. 126, 2019.
- [6] X. Jin and Y. Zhang, "Privacy-preserving Crowdsourced Spectrum Sensing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1236–1249, 2018.
- [7] H. Qi, X. Zhang, and Y. Gao, "Channel Energy Statistics Learning in Compressive Spectrum Sensing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 7910–7921, 2018.
- [8] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for Spectrum Sensing in Cognitive Radio," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2306–2321, 2019.
- [9] H. Qi, X. Zhang, and Y. Gao, "Low-Complexity Subspace-Aided Compressive Spectrum Sensing Over Wideband Whitespace," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 762–11 777, 2019.
- [10] D. Chew and A. B. Cooper, "Spectrum Sensing in Interference and Noise Using Deep Learning," in *Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2020, pp. 1–6.
- [11] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "DeepSense: Fast Wideband Spectrum Sensing Through Real-Time In-the-Loop Deep Learning," in *Proc. of IEEE INFOCOM*, 2021, pp. 1–10.
- [12] L. Baldesi, F. Restuccia, and T. Melodia, "ChARM: NextG Spectrum Sharing Through Data-driven Real-time O-RAN Dynamic Control," in *Proceedings of IEEE INFOCOM*. IEEE, 2022, pp. 240–249.
- [13] Q. Mao, F. Hu, and Q. Hao, "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2595–2621, 2018.
- [14] G. C. Sobabe, Y. Song, X. Bai, and B. Guo, "A cooperative spectrum sensing algorithm based on unsupervised learning," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–6.
- [15] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.
- [16] Z. Chen, H. Cui, J. Xiang, K. Qiu, L. Huang, S. Zheng, S. Chen, Q. Xuan, and X. Yang, "Signet: A novel deep learning framework for radio signal classification," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 529–541, 2022.
- [17] 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)*. IEEE, 2021, pp. 1–7.
- [18] N. Soltani, V. Chaudhary, D. Roy, and K. Chowdhury, "Finding waldo in the cbirs band: Signal detection and localization in the 3.5 ghz spectrum," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4570–4575.
- [19] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2019.
- [20] S. D'Oro, F. Restuccia, and T. Melodia, "Can You Fix My Neural Network? Real-Time Adaptive Waveform Synthesis for Resilient Wireless Signal Classification," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, Vancouver, BC, Canada, May 2021.
- [21] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, K. Chowdhury, S. Ioannidis, and T. Melodia, "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, 2020.
- [22] S. Kayraklik, Y. Alagöz, and A. F. Coşkun, "Application of object detection approaches on the wideband sensing problem," in *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2022, pp. 341–346.
- [23] H. N. Nguyen, M. Vomvas, T. D. Vo-Huu, and G. Noubir, "Wrist: Wideband, real-time, spectro-temporal rf identification system using deep learning," *IEEE Transactions on Mobile Computing*, 2023.
- [24] Federated Wireless, "Citizens Broadband Radio Service (CBRS) Shared Spectrum: An Overview," <https://www.federatedwireless.com/wp-content/uploads/2017/09/CBRS-Spectrum-Sharing-Overview.pdf>, 2018.
- [25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III* 18. Springer, 2015, pp. 234–241.
- [26] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: A comprehensive survey," *Ad Hoc Networks*, vol. 93, p. 101913, 2019.
- [27] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- [28] X. Li, F. Dong, S. Zhang, W. Guo *et al.*, "A survey on deep learning techniques in wireless signal recognition," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.
- [29] 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.
- [30] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davaslioglu, H. Deng, and T. Erpek, "Real-time and embedded deep learning on fpga for rf signal classification," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6.
- [31] N. Soltani, K. Sankhe, J. Dy, S. Ioannidis, and K. Chowdhury, "More is better: Data augmentation for channel-resilient rf fingerprinting," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 66–72, 2020.
- [32] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE access*, vol. 8, pp. 1498–1506, 2019.
- [33] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.
- [34] L. Bertizzolo, L. Bonati, E. Demirors, A. Al-shawabka, S. D'Oro, F. Restuccia, and T. Melodia, "Arena: A 64-antenna sdr-based ceiling grid testing platform for sub-6 ghz 5g-and-beyond radio spectrum research," *Computer Networks*, vol. 181, p. 107436, 2020.