Seek and Classify: End-to-end Joint Spectrum Segmentation and Classification for Multi-signal Wideband Spectrum Sensing

Prashant Subedi
Cyber-Physical Networking Lab, School of Computing
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
psubedi3@unl.edu

Sangwon Shin
Cyber-Physical Networking Lab, School of Computing
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
sshin11@unl.edu

Mehmet C. Vuran
Cyber-Physical Networking Lab, School of Computing
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
mcv@unl.edu

2024 IEEE 49th Conference on Local Computer Networks (LCN) | 979-8-3503-8800-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/LCN60385.2024.10639826

***Abstract*—The rise in the use of wireless communication has led to the problem of spectrum scarcity in licensed bands. The popularity of the Internet of Things (IoT) requires innovative solutions that maximize the use of the available spectrum to support the increasing number of connected devices. The ability to detect and classify modulation of the signals efficiently can enable a cognitive radio to monitor the spectrum activity in real-time and utilize unused frequencies. In this work, Seek and Classify, an end-to-end framework for joint spectrum segmentation and classification for narrowband signals from wideband IQ samples, is developed. Seek and Classify includes a novel intersection of unions-based training methodology and machine learning architectures that advances this unique area. Evaluations performed through both synthetically generated radio signals and over-the-air experiments with software defined radios reveal that the proposed training strategy and models increase the classification accuracy from 41% to 99%. Moreover, the end-to-end framework reduces the sensing time for narrowband signals by 2-10 times, depending on hardware capabilities. The extensive evaluations provide guidance for the choice of training methods, machine learning architectures, and preprocessing tools for the most effective joint segmentation and classification performance.**

***Index Terms*—spectrum sensing, cognitive radio, spectrum detection, spectrum segmentation, modulation classification**

I. Introduction

Automatic signal classification enables spectrum monitoring and enforcement, as well as the development of cognitive radios (CRs) that can opportunistically use idle spectrum bands. Software-defined Radio (SDR) can capture multiple signals within their bandwidth, but current work in modulation classification often assumes a single signal per sample [1]–[6].

In practical workflows, extracting the signal from a spectrum sample faces challenges because the carrier frequency, bandwidth, and structural properties may not be known in advance, rendering state-of-the-art solutions ineffective in end-to-end workflows.

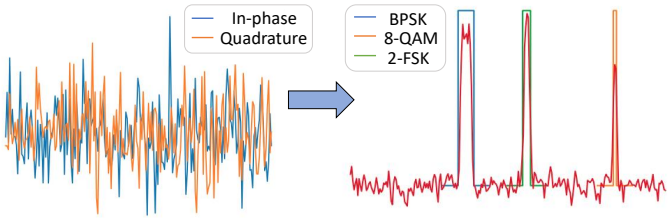


Fig. 1 Spectrum segmentation and multi-signal classification illustration.

To this end, in this work, we quantify this problem and provide the following contributions: ***(1)*** We introduce *Seek and Classify*, an end-to-end machine-learning (ML) based framework to jointly and blindly segment and classify multiple narrowband signals present in wideband samples. ***(2)*** We show that Seek and Classify can adapt to the errors in the spectrum segmentation process and, accordingly, improve the classification accuracy from 41% to 99% when compared to using a separately trained classifier. ***(3)*** While it is possible to re-tune the radio to sample narrowband signals separately, doing so increases the time spent in sensing the spectrum. We show that Seek and Classify can reduce segmentation and classification time by 2–10 times than a re-tuning and re-sampling approach. ***(4)*** We create synthetic and over-the-air datasets including wideband samples of multiple types of signals. These datasets could be used for joint multi-signal segmentation and classification in this unique area. We pledge to make the datasets available if the paper is accepted.

Seek and Classify is shown to be capable of performing high-fidelity spectrum sensing to efficiently use the unused spectrum. The rest of the paper is organized as follows: The related work in signal detection, segmentation, and classification is discussed in Section II. The spectrum segmentation and multi-signal classification problem is formally defined in Section III. Seek and Classify framework is described in Section IV, along with a novel training approach and ML architectures. The solution is evaluated through experiments

This work is supported in part by NSF CNS 2030272 and Department of Navy, Office of Naval Research, NSWC N00174-23-1-0007 grants. This work relates to Department of Navy award N00174-23-1-0007 issued by the Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

using synthetic and over-the-air datasets and compared to state-of-the-art solutions in Section V. Finally, the paper is concluded in Section VI.

II. Related Work

Autonomous spectrum activity detection and classification enable secondary users to find unused licensed band segments [7]–[9]. Standards like LTE-U, NR-U, and CBRS for dynamic spectrum sharing [10]–[12] demonstrate CR technology's potential to utilize the available spectrum effectively. The key tasks are signal detection and automatic modulation classification (AMC).

A. Signal Detection

Spectrum sensing techniques range from those with *a priori* knowledge of transmission parameters to completely blind methods. Early solutions, like matched filtering [13], [14] and cyclo-stationary feature detection [15], are impractical for real-time detection due to long observation times [8]. Energy detection (ED) methods do not require prior knowledge of the signal [16], but are sensitive to noise estimate errors [14]. Several approaches to noise level estimation have been developed [17]–[20]. Recent approaches using deep learning models improve detection accuracy compared to ED-based methods for blind spectrum detection [21]–[24]. However, they either use short-time Fourier transform (STFT) or the power spectral density (PSD) as input, adding additional processing time. The use of complex numbers obtained using fast Fourier transform (FFT) for signal detection has not been explored.

B. Automatic Modulation Classification

Automatic Modulation Classification (AMC) techniques classify modulation types of samples. In [25], a CNN trained on raw IQ signals is shown to perform $2.5-5$dB better than feature-based classifiers at lower signal-to-noise ratios. Other deep learning architectures like LSTM ([2]), ResNet [1] and complex-valued neural network [3] are also implemented for signal classification tasks. However, baseline models trained on fixed parameters like channel, sampling frequency, roll-off factor, and over-sampling factor suffer from significant performance drops when these parameters change, a phenomenon known as *domain shift* [26]. Recent studies have focused on creating generalized models that maintain accuracy despite domain shifts [4], [5], [27]. In blind spectrum sensing scenarios, classifiers trained on data from a single transmitter with a known center frequency experience domain shifts due to multiple signals and signal detector imperfections. The impact of domain shifts from imperfect extraction of signals from wideband samples remains unexplored in the current literature. Despite significant contributions to blind spectrum detection and AMC separately, integrated systems solving these problems end-to-end are lacking. Our work aims to fill this gap. Notably, a multi-signal detection and classification framework using a sliding window-based technique and separately trained Complex-ResNet for classification is developed [28]. However, this framework relies on a fixed threshold that works well with its dataset, reducing domain shift. We investigate the domain shift effects of using separately trained classifiers in Section V and show that joint training is necessary. Existing modulation classification techniques require SDR re-tuning for each signal. We show in Section V that re-tuning the radio and resampling increase the time taken by up to 10 times. To the best of our knowledge, *Seek and Classify* is the first end-to-end framework that provides joint spectrum segmentation and multi-signal classification.

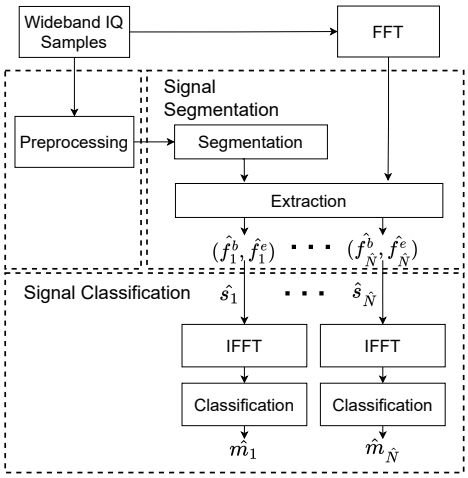


Fig. 2 Seek and Classify framework.

III. Problem Definition

Considering N transmitters transmitting narrowband signals, the composite signal received at a receiver is given by:

$$r(t) = \sum_{i=1}^{N} s_i(t) + n(t), \tag{1}$$

assuming an additive white Gaussian noise (AWGN) channel, where $s_i(t)$ is the signal transmitted from a transmitter i and $n(t)$ is the noise. Let f_i be the center frequency, b_i the bandwidth, $P_{t,i}$ the TX power, and $m_i \in \mathcal{M}$ the modulation type of signal $s_i(t)$, where $\mathcal{M}$ is the set of possible modulation types. The relation between f_i, b_i and s_i depends on the modulation type m_i. For the problem definition, we assume: (1) All the signals are in the frequency range of the receiver, where f_r is the center frequency of the receiver and F is the sampling rate. (2) There is no overlap between the narrowband signals in the frequency domain. The first assumption is relaxed in practice through windowed sampling of much wider signals. The second assumption follows the licensing mechanism in wireless systems and holds true for most practical cases. Consider that a CR collects discrete IQ samples of $r(t)$. If $r(t)$ is sampled at the receiver for a time T_s, the total number of samples received at the receiver is $Z = T_s \times F$. The collected sample $\mathbf{s}$ is a sequence of Z complex numbers representing the IQ components. Using discrete Fourier transform (DFT), the maximum resolution that can be achieved per frequency bin is $F/Z = 1/T_s$ and the number of bins is Z.

Accordingly, **the joint spectrum segmentation and multi-signal classification problem** can be framed as follows: Given Z samples, sampled at a rate F, find the set $\mathbf{T} =$

$\{(f_i^b, f_i^e, m_i) : \forall i \in \{1, ..., N\}\}$, where f_i^b and f_i^e are the first and last frequency bins and m_i is one of the modulation types in $\mathcal{M}$ for the i^{th} narrowband signal and N is the total number of narrowband signals present in the wideband sample. An illustration of the joint spectrum segmentation and multi-signal classification problem is shown in Fig. 1. The captured IQ samples (left) contain mixed signals that cannot be separated in the time domain. The desired outcome (right) shows signals separable in the frequency domain, with their positions and modulation types marked. Next, we present Seek and Classify.

IV. Seek and Classify

Combining a spectrum segmentation method with a separately trained signal classification method results in poor detection accuracy. This is because the distribution of input data to the classifiers shifts from that of the training data due to impurities in segmentation, resulting in a covariate shift. The effect of covariate shift for joint segmentation and classification is illustrated in Section V, showing that when classifiers are trained separately, end-to-end accuracy falls drastically to below 45%. Employing joint training improves accuracy by up to 56.7 percentage points (from 42.45% to 99.15%). To this end, we devise an end-to-end architecture and a ***novel methodology to train*** classifiers using the segmented signals.

An overview of the Seek and Classify (SnC) end-to-end framework is shown in Fig. 2. SnC has three stages: wideband sample preprocessing, spectrum segmentation, and multi-signal classification. In the following, we describe the novel SnC framework in each stage and the segmentation and classification architectures. It is important to note that SnC is an ML-architecture-agnostic framework, where future neural network architectures could be embedded into the segmentation and classification stages. SnC is evaluated in comparison with the state-of-the-art solutions in Section V.

The input to the framework is a sequence $\mathbf{S}$ of IQ samples and the goal is to predict the target $\mathbf{T}$. Define $\hat{\mathbf{T}}$ as the estimated target which consists of $\{\hat{f}_i^b, \hat{f}_i^e, \hat{m}_i\}$ for $i \in \{1, .., \hat{N}\}$, where $\hat{f}_i^b$ and $\hat{f}_i^e$ are the beginning and end of the extracted signal, s_i, defining both its position in the spectrum and its bandwidth, $\hat{m}_i$ is the estimated modulation of s_i, and $\hat{N}$ is the number of predicted signals.

In the framework, the prediction is done in three stages: (1) The preprocessing stage takes $\mathbf{s}$ as input and produces a preprocessed version, $\mathbf{P}$, suitable for segmentation. (2) The spectrum segmentation stage takes $\mathbf{P}$ as input and predicts $\hat{N}$, the number of signals, and $\hat{\mathbf{F}} = \{(\hat{f}_i^b, \hat{f}_i^e)\}$, the set of start and end frequency bins of the signals present in the sample. (3) The classification stage takes $\mathbf{s}$ and $\hat{\mathbf{F}}$ as inputs and predicts the set $\hat{\mathbf{C}} = \{\hat{m}_i\}$ of modulation classes. Together, these form the set $\hat{\mathbf{T}}$.

A. Preprocessing

Existing energy detection and deep learning approaches to recognize received signals utilize the frequency domain information of signals [20], [24], [29]. Our preliminary evaluations

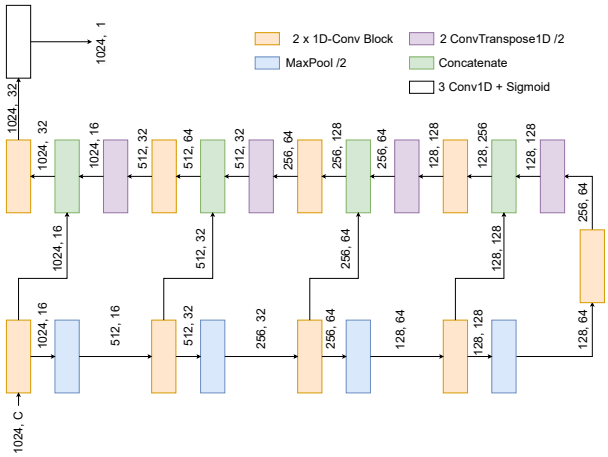


Fig. 3 SnC Segmentation (SnC-Seg) Architecture.

by using $\mathbf{s}$ directly also did not show good results. Hence, in the preprocessing stage, the frequency domain representation of $\mathbf{s}$ is calculated, producing $\mathbf{P} = \{p_1, ..., p_L\}$ where L is chosen based on the detection method used.

We consider three preprocessing methods: complex FFT (cFFT), absolute FFT (aFFT), and power spectral density (PSD) calculated using Welch's method [30]. For FFT-based preprocessing methods, only the first L samples are taken. Under the assumption of continuous transmission of narrowband signals during the short sampling time, if L is sufficiently large, the information of the narrowband signals is present in L samples. For PSD, the size of the window is taken to be L and the entire input of size Z is used.

B. Spectrum Segmentation

The output $\mathbf{P}$ of preprocessing is passed to the spectrum segmentation stage to predict the set $\{(f_i^b, f_i^e)\}$. This goal can be modeled as predicting whether there is a signal in each frequency bin. Depending on the segmentation model used, the resolution of the model ρ is defined by $\rho = Z/O$, where Z is the length of the signal and O is the length of the output of the segmentation stage. So, the target output of the model can be represented as $\mathbf{B} = \{b_1, .., b_O\}$, where

$$b_j = \begin{cases} 1 & \text{if } \exists (f_i^b, f_i^e) \text{ s.t. } \lfloor f_i^b/\rho \rfloor < j \leq \lceil f_i^e/\rho \rceil \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

Then, a prediction of $\hat{\mathbf{B}} = \{\hat{b_1}, ..., \hat{b_M}\}$ is made. The set of start and end frequency bins, $\mathbf{F}$, can be constructed by taking the start and end indices of contiguous segments of '1's in $\hat{\mathbf{B}}$ and scaling the indices by ρ. This is the reverse of the operation used to derive $\mathbf{B}$ from $\mathbf{F}$ in (2).

The mapping from $\mathbf{P}$ to $\mathbf{B}$ can be approached as a one-dimensional semantic segmentation problem with two classes. Semantic segmentation models predict each input value, making the length output of the segmentation stage O the same as the preprocessing stage, L. Models for semantic segmentation have been modeled for spectrum detection [23], [24]. However, they are usually trained with real-valued inputs. Semantic segmentation has been investigated within the context of image processing and U-Net-like architectures have been shown to

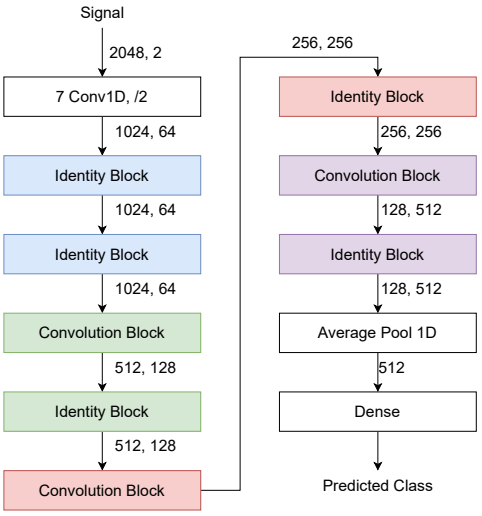


Fig. 4 SnC Classifier (SnC-Clf) Architecture.

perform well [23], [31]. Given that spectrum signals are one-dimensional, we propose a one-dimensional neural network model for spectrum segmentation.

The architecture of the SnC spectrum segmentation stage (SnC-Seg) is shown in Fig. 3. SnC-Seg can utilize any preprocessing methods described in Section IV-A. For real-valued inputs (i.e., PSD and absolute FFT), inputs are treated as one-dimensional sequences with one channel. For complex-valued inputs (i.e., complex FFT), the input is mapped as one-dimensional sequences with two channels. Section V demonstrates that using the same number of samples in preprocessing shows an advantage in using the complex representation.

The learning task is to take $\mathbf{P}$ as input and predict $\mathbf{B}$. The model is trained using a training set of $\mathbf{P}$ and corresponding $\mathbf{B}$ from the dataset, consisting of samples $\mathbf{S}$ and targets $\mathbf{T}$. The problem requires a binary class classification for each $b_i \in \mathbf{B}$. Given the significant imbalance between bins with signals and noise known as *class imbalance problem*, we use the Focal loss function [32]. Focal loss improves detection performance by 0.5% over binary cross entropy loss [23].

Using preprocessing methods and the proposed model, the SnC segmentation stage predicts the signal locations in a wideband sample. The positional information is used to extract signals for classification by converting the sample to the frequency domain and isolating the signal:

$$\mathbf{S} = FFT_Z(\mathbf{s}), \quad \text{and} \quad \hat{\mathbf{S}}_\mathbf{i} = \mathbf{S}[\hat{f}_i^b, ..., \hat{f}_i^e], \quad \forall i \tag{3}$$

where $\hat{\mathbf{S}}_\mathbf{i}$ is the frequency domain representation of the i^{th} extracted signal and FFT_Z is Z-point FFT operation. For computationally efficient FFT operation, the number of IQ samples, Z, is a power of 2. The classification stage is described next.

C. Multi-Signal Classification

Existing deep learning solutions for modulation classification (e.g., [1], [3]–[6], [25]) focus on single signal samples and *are unsuitable for an end-to-end framework for multiple signal classification.* Moreover, due to inherent errors in the spectrum segmentation process, models need to be trained to classify the modulation m_i of the imperfectly extracted signals $\hat{\mathbf{S}}_\mathbf{i}$ from a wideband sample. Hence, innovations in ML architecture and training methodology are needed. We present the former in the following and discuss the latter in Section IV-D.

Since extracted signals are obtained in the frequency domain, classifying them directly avoids the need for an inverse Fourier transform as in [28]. However, the set of modulations selected in [28] are distinctly different in the frequency domain. We show in Section V that classifiers using frequency domain representation fail to distinguish amplitude and phase shift keying modulation schemes such as QAM and PSK, highlighting the benefits of time domain representation.

Before classification, the extracted signals $\hat{\mathbf{S}}_\mathbf{i}$, are converted to time domain representation using inverse fast Fourier transform (IFFT), with zero-padding for computational efficiency as:

$$z_i = 2^{\lceil log_2(\hat{f}_i^e - \hat{f}_i^b)\rceil}, \quad \text{and} \quad \hat{\mathbf{s}}_\mathbf{i} = IFFT_{z_i}(\hat{\mathbf{S}}_\mathbf{i}) \tag{4}$$

where $\hat{\mathbf{s}}_\mathbf{i}$ is the i^{th} extracted signal in time domain and $IFFT_{z_i}$ is a z_i-point IFFT operation. Despite possible distortions by forward and inverse Fourier transforms, as discussed in Section V, high accuracy is achievable.

The SnC classifier stage (SnC-Clf) architecture, shown in Fig. 4 utilizes an 18-layer one-dimensional ResNet. Existing ResNet [33] models perform well in signal classification [1], [28] but are not trained to work on extracted signals. ResNet adds the original input back to the result of two convolutional operations, mitigating the vanishing gradient problem. For complex signals, it is treated as 2-channel inputs, with one-dimensional residual and convolution blocks.

The classifiers applied on each $\hat{s}_i$ produce the modulation label $\hat{m}_i$. To accommodate fixed-length input requirements, each extracted signal is zero-padded. Inputs are sequences of complex numbers, with real and imaginary parts as 2-channel inputs. Models optimize cross-entropy loss for classification [34], as $J = -1/n \sum_{i=1}^{n} \sum_{j=1}^{|\mathbf{M}|} y_{ij} \log(p_{ij})$, where y_{ij} is the target probability which is equal to 1 if the i^{th} signal is of j^{th} modulation type, and 0 otherwise and p_{ij} is the predicted probability.

D. IOU-based Multi-Signal Classifier Training

Next, we introduce a novel method to train the classifiers. Due to imperfections in segmentation, classifiers trained on signals extracted using target locations $\{(f_i^b, f_i^e)\}$ fail to generalize when using predicted locations to generalize when signals extracted using predicted locations $\{(\hat{f}_i^b, \hat{f}_i^e)\}$. The classification accuracy of the classifiers decreases from 99% to 41% without joint segmentation and classification, as shown in Section V.

To train the classifiers with extracted signals, we assign labels by matching predicted signals to target signals, inspired by end-to-end object detection methods [35]. Predicted signals are assigned labels from matched target signals to maximize the intersection over union (IOU) of matched pairs.

TABLE I: Datasets.

	Synthetic	OTA
No. of signals, N	1 – 10	1 – 3
Sampling rate, F	20 MHz	20 MHz
Sample SNR [dB]	(-10) – 10	(-16) – 12
Signal BW [MHz]	0.1, 0.2, 0.5, 1, 2	
No. of samples	55,000	8,400
Modulation	QPSK, BPSK, 8-PSK, 8-QAM, 16-QAM, GMSK, 2-FSK	QPSK, BPSK, 2-FSK

IOU measures object detection performance in computer vision. For one-dimensional (1D) frequency domain signals, IOU between i^{th} target and j^{th} prediction is calculated as

$$\text{IOU}(i,j) = \frac{\min(\hat{f}_j^e, f_i^e) - \max(\hat{f}_j^b, f_i^b)}{\max(\hat{f}_j^e, f_i^e) - \min(\hat{f}_j^b, f_i^b)} \tag{5}$$

Let $\mathbf{G} = \{(i,j)\}$ match i^{th} target signal with j^{th} predicted signal. The optimal matching $\mathbf{G}^*$ maximizes the IOU sum as:

$$\mathbf{G}^* = \arg\max_{\mathbf{G}} \sum_{i,j \in G} \text{IOU}(i,j) \tag{6}$$

The optimal matching uses the modified Jonker-Volgenant algorithm [36]. Unmatched predicted signals are labeled "none". This strategy creates a training set of extracted signals and modulation labels for the modulation classifiers.

V. Performance Evaluations

We evaluate the performance of Seek and Classify using synthetic and over-the-air (OTA) data. Evaluations are in three parts: First, we compare S&C with other solutions for spectrum segmentation and signal classification. Then, we evaluate end-to-end joint segmentation and classification. Synthetic data evaluations control channel impurities and SNR range, while OTA experiments reveal impacts of real-life radios and channels. We use a two-stage training process, transferring the network trained on synthetic data (AWGN channel) to a network trained on experimental data (real channel) for OTA experiments.

A major contribution is creating synthetic and OTA data for wideband samples with multiple signal types for joint multi-signal segmentation and classification.

A. Evaluation Setup

1) Synthetic Dataset: We use Matlab to create a representative dataset for spectrum segmentation and signal classification tasks. We ensure all signals are within the receiver's frequency range with a guard band b_g between the narrowband signals in the frequency domain. Generated signals are interpolated, shifted to correct center frequencies, and combined into samples. These samples and their generating values are saved as labels for training deep learning models.

Due to multiple signals, labeling a single signal-to-noise ratio (SNR) value to a sample is impractical. However, it is necessary to test the framework in different conditions. To this end, two metrics are used to study channel conditions: (a) Sample SNR and (b) Signal SNR. Sample SNR is the SNR of the signal with the lowest power in the wideband spectrum sample. The transmit power is randomly chosen. We define Sample SNR and use it to calculate the noise power for the AWGN channel such that NP $= P_{t,min} -$ Sample SNR. The Signal SNR is then defined for each signal as Signal SNR$_i$ $= P_{t,i} -$ NP (all in dB). The parameters used to simulate the dataset are shown in Table I.

TABLE II: Spectrum Segmentation Methods.

Name	Preproc. Method	Sample length (Z)	Input (L)
SnC-Seg(PSD)	Welch	16,384	1,024
SnC-Seg(cFFT)	FFT	1,024	1,024
SnC-Seg(aFFT)	$\lvert\text{FFT}\rvert^2$	1,024	1,024
ResNet(cFFT)	FFT	16,384	16,384
ResNet(aFFT)	$\lvert\text{FFT}\rvert^2$	16,384	16,384
LAD(PSD)	Welch	16,384	1,024
LAD(aFFT)	$\lvert\text{FFT}\rvert^2$	1,024	1,024

2) OTA Dataset: The OTA dataset is collected using a software-defined radio testbed setup as shown in Fig. 5.The testbed includes three Ettus USRP B200 transmitters, each with a sub-6 GHz wideband antenna transmitting BPSK, QPSK, and 2-FSK signals. A USRP B200 receiver with a sub-6 GHz wideband antenna captures the concurrently transmitted signals. The transmitters utilize generated IQ samples using MATLAB and then transmit them in the 900–920 MHz Industrial, Scientific, and Medical (ISM) band with bandwidths ranging from 0.1 MHz to 2 MHz with the parameters shown in Table I. The receiver collects the transmitted OTA dataset with a sampling rate of 20 MHz. The OTA dataset is similar to the synthetic dataset but includes real channel effects. Noise power is measured from samples when transmitters are off. Signal power is collected from known segmented signals, and SNR is calculated for each signal.

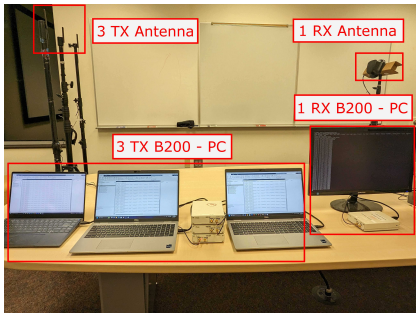
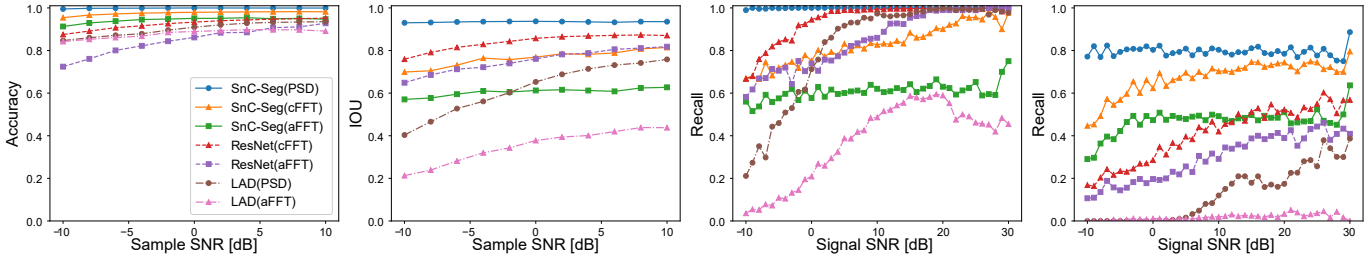


Fig. 5 Seek-and-Classify USRP testbed setup.

3) Evaluation Setup: Evaluations are performed on a server with two Intel Xeon Silver 4110 CPUs, an NVIDIA Tesla V100 GPU, and 187 GB of memory. Datasets are split into training, validation, and testing sets at an 80%-10%-10% split. Models are trained on the training set until validation loss does not increase for 10 consecutive epochs. The initial learning rate is 0.01, reduced by a factor of 10 if validation loss does not improve for 3 epochs. Performance graphs are generated using the testing set.

We first evaluate segmentation and classification separately. Then, *Seek and Classify* framework is evaluated end-to-end. Performance metrics are explained in each section.



(a) Accuracy (b) IOU (c) Recall (threshold=0.5) (d) Recall (threshold=0.90)

Fig. 6: Spectrum segmentation performance under different SNR conditions.

B. Spectrum Segmentation Performance

We evaluate SnC segmentation stage (SnC-Seg) performance and compare with two state-of-the-art methods. Furthermore, the preprocessing methods described in Section IV-A are included, resulting in seven combinations of preprocessing and spectrum segmentation methods (Table II).

1) Comparison Set:

a) ResNet: ResNets trained for modulation classification [1], [28], cannot be directly used for spectrum segmentation due to output averaging [33]. For segmentation, we predict the presence and absence of signals in each frequency domain section. A 2-layer fully connected network is applied to each element across the length. The convolutional part produces a 1024×512 output, and the fully connected network producing 1024 predictions. The network, trained with complex and absolute FFT preprocessing, uses $L/0 = 16$ for frequency resolution.

b) Energy Detection: We use an existing ED-based method called localization algorithm based on double-thresholding (LAD) [20], [29]. Since LAD cannot use the complex input directly, it uses energy values.

These approaches are trained and evaluated using the wideband dataset. Performance is evaluated across a range of sample SNR and signal SNR conditions using sample metrics, signal metrics, and timing metrics.

2) Evaluation Metrics: (1) Accuracy: Accuracy for a single sample is the fraction of bins that are correctly predicted, $1/M \sum_{i=1}^{M} [b_i = \hat{b_i}]$. The values for multiple samples can be averaged to get an aggregate value.

The rest of the metrics evaluate performance by matching predicted and target signals. The matching is selected to maximize the sum of IOU scores. IOU, recall for signals, and extracted signals per target are the signal metrics. Let σ be the set of all the target signals and $\hat{\sigma}$ be the set of all the segmented signals in all the samples being considered. Now, consider $\mathbf{Q}$ to be the set of mapping between elements of σ and $\hat{\sigma}$. The signal-based metrics are defined as: *(1) Average IOU* as: Average IOU $= 1/|\sigma| \sum_{(i,j)\in\mathbf{Q}} \text{IOU}(\sigma_i, \hat{\sigma}_j)$. IOU metric shows the quality of extracted signals. *(2) Recall for signals* metric shows the fraction of target signals that are extracted with a certain quality threshold. Quality of signal prediction is measured with IOU and the threshold is represented by θ so that Recall at $\theta = 1/|\sigma| \sum_{(i,j)\in Q} [\text{IOU}(\sigma_i, \hat{\sigma}_j) > \theta]$. In the experiments, θ values of 0.5 and 0.90 are used.

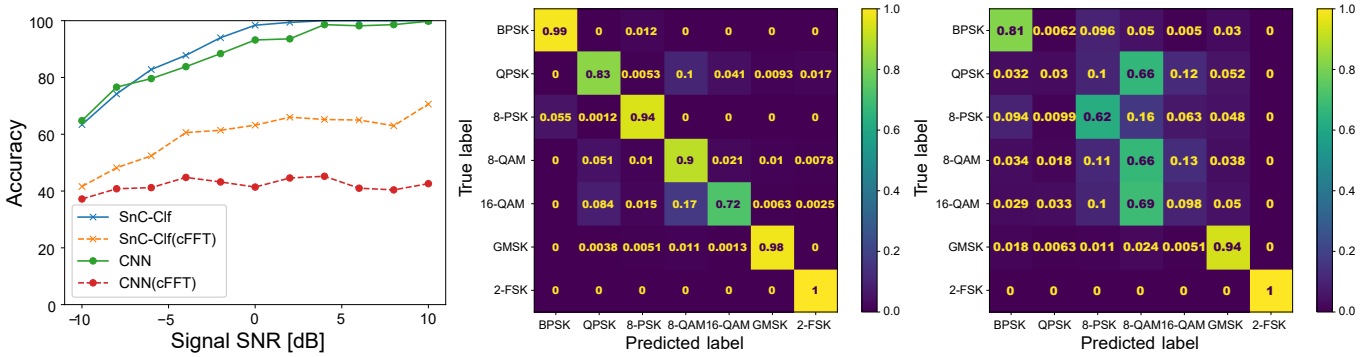
The execution time for preprocessing and segmentation steps are analyzed for all combinations. The deep learning models are evaluated at batch sizes of 1 and 32. A batch size of 1 is useful for real-time scenarios, while a batch size of 32 is useful for scenarios where batches of $\mathbf{S}$ can be collected before predicting the entire batch.

3) Evaluation Results: The segmentation performance in different sample SNR conditions is shown in Fig. 6. The SnC-Seg(PSD) model shows an overall accuracy of 99.8%, and an average IOU score of 0.93. The scores are consistent across different sample SNR conditions in the simulated dataset. SnC-Seg(cFFT) model shows an overall accuracy of 97.5%. However, the slight change in accuracy results in an 18% reduction of the IOU score. The PSD calculation requires 16 times more samples than FFT to reduce the variance due to noise showing performance improvement. SnC-Seg(aFFT) model has an accuracy of 94.0% and an IOU score of 0.60, indicating a 20% reduction in the IOU score compared to using the complex values directly. SnC-Seg can perform better with complex values that retain the phase information lost when calculating absolute values.

The ResNet(cFFT) has 5% lower accuracy than SnC-Seg(cFFT), but it has a higher IOU score of 0.84 compared to 0.76. ResNet suffers a 10% IOU score reduction when absolute values are used instead of complex numbers.

The difference between SNR vs recall at 0.5 and 0.9 threshold (Fig. 6c and Fig. 6d) for all cases shows the difficulty in creating segmentation solutions that perfectly extract target signals from a wideband sample. The SnC-Seg(PSD) shows a recall of 99.9% at a threshold of 0.5 but it drops to 79.61% for the threshold of 0.9. The model-driven LAD performs poorly compared to data-driven solutions at a threshold of 0.9. The recall for ResNet(cFFT) drops from 83.8% to 41.8% when the threshold is changed from 0.5 to 0.9. The 50% drop is due to the tendency of the model to break target signals into multiple signals, only one of which is matched during the evaluation.

The preprocessing and prediction time for different segmentation models is shown in Fig. 8. Due to the lower preprocessing time, SnC-Seg(cFFT) is evaluated in 0.35 ms compared to 1.94 ms for SnC-Seg(PSD). The time difference is only for evaluation - since SnC-Seg(PSD) utilizes more samples, the sampling time would also be increased. The ResNet models



(a) Classifier accuracy on different SNR (b) Time domain input (SnC-Clf). (c) Frequency domain input [SnC-Clf(cFFT)].

Fig. 7: Spectrum classifier performance.

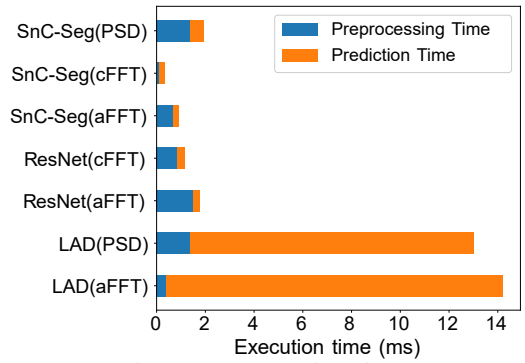


Fig. 8: Segmentation time.

utilize the same number of samples as SnC-Seg(PSD) and take 1.14 ms and 1.75 ms, respectively. Overall, **SnC-Seg(cFFT) is the fastest** among the evaluated segmentation methods, while **SnC-Seg(PSD) has the highest accuracy**, with the choice of preprocessing techniques striking a balance between delay and accuracy. Next, we discuss the classification performance.

C. Multi-Signal Classification

We evaluate the SnC classifier stage (SnC-Clf) using time and frequency domain representation of the signals, justifying the introduction of IFFT in the end-to-end framework (Fig. 2). Models are trained using a synthetic dataset containing samples with a single signal. We first discuss the comparison set, evaluation metrics, and results.

1) Comparison Set - CNN: SnC-Clf is compared with a state-of-the-art CNN developed in [1]. CNN consists of 13 1D convolution layers followed by two fully connected layers. Classifiers with frequency domain representation inputs are identified with '(cFFT)'.

2) Evaluation Metrics: Classification performance is measured using accuracy across different signal SNRs and confusion matrices. *(1) Classification Accuracy:* Accuracy over a set of signals $\{S_1, ..., S_N\}$ is given by Classification Accuracy $= 1/N \sum_{i=1}^{N} [m_i = \hat{m}_i]$. *(2) Confusion Matrix:* Normalized confusion matrix $\mathbf{CM}$ is calculated as: $\mathbf{CM}_{ij} = \sum_{n=1}^{N} [m_n = i \wedge \hat{m}_n = j] / \sum_{n=1}^{N} [m_n = i]$.

3) Evaluation Results: Classification accuracy across different signal SNR is shown in Fig. 7a. The SnC-Clf has 100.0% accuracy at 10dB SNR, decreasing to 63.4% at −10dB. On the other hand, when frequency domain input is utilized, SnC-Clf(cFFT) the accuracy reduces to 70.6% at 10dB. Overall accuracy for SnC-Clf is 90.1%. In comparison, CNN accuracy drops from 99.8% at 10dB SNR to 64.8% at −10dB with an overall accuracy of 88.65%. The frequency domain input results in a drastic decrease in CNN(cFFT) accuracy of 42.6% at 10dB.

TABLE III: End-to-end Evaluation Models

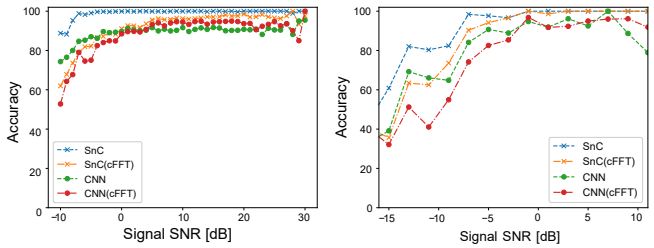
Name	Segmentation	Classifier
SnC	SnC-Seg(PSD)	SnC-Clf
SnC(cFFT)	SnC-Seg(cFFT)	SnC-Clf
CNN	SnC-Seg(PSD)	CNN
CNN(cFFT)	SnC-Seg(cFFT)	CNN

Confusion matrices in Fig. 7 provide insight into decreased performance with FFT input. SnC-Clf(cFFT) has 100% and 94% recall on 2-FSK and GMSK modulations. SnC-Clf(cFFT) shows 81% accuracy for BPSK, but misclassifies 66% of QPSK and 16-QAM signals as 8-QAM. With time domain samples, recall for 2-FSK remains 100%, GMSK accuracy improves to 98%, and recall for other modulations is above 72%. Improved performance with time domain input in frequency and amplitude/phase modulations (e.g., FSK, PSK, QAM) demonstrates **the need for IFFT.**

SnC outperforms compared models in segmentation and classification separately. However, classification models trained on sanitized datasets do not perform well in an end-to-end workflow, highlighting the need for the novel method in Section IV-D. Next, we evaluate the end-to-end framework.

D. End-to-End Evaluations

Integrating separately trained models into an end-to-end workflow drastically reduces performance due to imperfections in the segmentation process. Novel training methods are needed, as discussed in Section IV-D. To evaluate end-to-end SnC performance, we compare IOU-based training with model architectures. We introduce perfect segmentation, using true segmented signals to train the classifier stage, representing state-of-the-art training approaches [4], [5], [27]. Then, SnC-Clf and comparison sets are evaluated. Finally, we introduce IOU-based training in the end-to-end workflow and evaluate its impacts.



(a) Synthetic data (b) OTA data

Fig. 9: End-to-end results.

1) Perfect Segmentation Training: We train and validate two classifiers with true segmented signals: SnC-Clf* and CNN*. Note that SnC* is distinct from the SnC in Section IV because the classifier is trained using perfect segmentation instead of IOU-based training. The true segmented signals are extracted with prior knowledge of the positions of the signals. The classifiers are then evaluated using SnC-Seg. The accuracy of classifiers on classifying perfectly segmented signals is as high as 99.9% for SnC-Clf* and 85% for CNN*. However, when a practical segmentation method is used, the accuracy plummets to $37\% - 41\%$, irrespective of the ML architecture. **This result validates the key motivation of SnC**: End-to-end segmentation and classification require tailored training methods because training based on sanitized datasets suffers from imperfections in signal extraction during inference.

2) Seek and Classify: We train and validate two classifiers with true segmented signals: SnC-Clf and CNN. The resulting segmentation-classifier combinations are shown in Table III. We evaluate SnC using synthetic and OTA data. For comparison, CNN-based classifier with two preprocessing methods (cFFT and PSD). In Fig 9a, the classification accuracy of synthetic and OTA data is shown. Comparing CNN results with SnC illustrates the ML-agnostic behavior of SnC, where future ML frameworks could be integrated without significant performance degradation. CNN results 9.98 p.p. degradation compared to SnC (89.5% vs. 99.48%).

E. OTA Experiment Results

To illustrate the feasibility of SnC utilizing OTA signals, we collect $8,600$ OTA signals featuring three modulation types: BPSK, QPSK, and 2-FSK (Table I). Each model is trained using transfer learning from a neural network initially trained on synthetic data over an AWGN channel.

The results shows that **SnC outperforms the comparison models in terms of accuracy under real radio channel conditions**, as shown in Fig. 9b. SnC achieves the highest accuracy of 92.6%, followed by SnC(cFFT) at 87.6%, CNN at 83.3%, and CNN(cFFT) at 78%. Compared to synthetic data results in Fig. 9a, accuracy degrades at lower SNR values for OTA data. For example, at the SNR of -10dB, SnC accuracy decreases by 4.31 p.p. (from $88.8\%1$ to 84.50%) when synthetic and OTA datasets are compared.

The importance of IOU-based training is validated through OTA data. SnC joint training improves classification accuracy by 25.3 p.p. (from 67.3% to 92.6%). Furthermore, adaptation to adverse effects of real environments is better. Specifically, transfer learning and IOU-based training increase average accuracy at low-SNR ($[-15, -5]$dB) by 21.81 p.p. (from 62.06% to 83.87%). OTA data results also indicate that FFT preprocessing impacts performance by up to 5.89 p.p., which could be traded off for delay (Fig. 8)

F. Delay Performance

Existing modulation classification approaches are not suitable for end-to-end workflow where signals extracted using IFFT are to be classified. Instead, the results of the segmentation stage can be used to change the center frequency and sampling rate of the radio to align with each detected signal. Using this resampling method enables existing modulation classification approaches designed for single signals. We calculate the overhead associated with the resampling method and compare it to SnC.

The additional overhead for collecting narrowband signals includes actual collection time and the time to retune the radio. We measure radio retuning times of two SDR hardware: USRP N310 and B200, using Python UHD API. Radio retuning times for N310 and B200 are 118 ms and 13.47 ms, respectively. Approximately 1 ms is required to collect samples in both devices. Assuming an average of 5 signals in a wideband sample, the total time for narrowband signal collection in N310 and B200 are 595 ms and 72.35 ms, respectively.

In SnC, FFT and IFFT are used for extracting the signals (Fig. 2). The average time for FFT and IFFT across all samples is approximately 5.5 ms. The average time for SnC over synthetic data is 66 ms. The resampling method requires both segmentation and classification, so the total time required is 654.6 ms for N310 and 131.95 ms for B210. SnC, which takes only 66 ms, is around 2 times faster when using B210 and 10 times faster with N310. The saved time can be used for communication purposes or sensing more frequently.

VI. Conclusion

In this study, we introduce *Seek and Classify*, an end-to-end framework for joint spectrum segmentation and classification of multiple signals within wideband spectrum samples. Seek and Classify significantly outperforms existing approaches where machine learning solutions are trained separately for segmentation and classification, boosting the classification accuracy from 41% to 99%. Additionally, Seek and Classify offers speed benefits when classifying narrowband signals directly instead of re-sampling. The utilization of Seek and Classify in sensing leads to 2–10 times reduction in time required for sensing, thereby providing additional time for communication or improving the sensing frequency. When evaluating signal segmentation using various preprocessing methods, up to 20% improvement can be achieved in the segmentation performance of deep learning models when employing *complex-valued* FFT directly rather than using its absolute value. The extensive evaluations reveal important results for the choice of training methods, machine learning architectures, and preprocessing tools for the most effective joint segmentation and classification performance.

REFERENCES

[1] Timothy James O'Shea, Tamoghna Roy, and T. Charles Clancy. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[3] Yihui Ren, Wen Jiang, and Ying Liu. Complex-valued Parallel Convolutional Recurrent Neural Networks for Automatic Modulation Classification. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 804–809, 2022.

[4] Ke Bu, Yuan He, Xiaojun Jing, and Jindong Han. Adversarial Transfer Learning for Deep Learning Based Automatic Modulation Classification. *IEEE Signal Processing Letters*, 27:880–884, 2020.

[5] Qing Wang, Panfei Du, Xiaofeng Liu, Jingyu Yang, and Guohua Wang. Adversarial unsupervised domain adaptation for cross scenario waveform recognition. *Signal Processing*, 171:107526, 2020.

[6] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *IEEE CVPR)*, volume 2, pages 1735–1742, 2006.

[7] Joseph Mitola. Cognitive radio for flexible mobile multimedia communications. In *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)(Cat. No. 99EX384)*, pages 3–10. IEEE, 1999.

[8] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13):2127–2159, 2006.

[9] Qing Zhao and Brian M. Sadler. A Survey of Dynamic Spectrum Access. *IEEE Signal Processing Magazine*, 24(3):79–89, 2007.

[10] Gaurang Naik, Jung-Min Park, Jonathan Ashdown, and William Lehr. Next generation wi-fi and 5g nr-u in the 6 ghz bands: Opportunities and challenges. *IEEE Access*, 8:153027–153056, 2020.

[11] FCC Authorizes Full Commercial Deployment in 3.5 GHz Band. https://www.fcc.gov/document/fcc-authorizes-full-commercial-deployment-35-ghz-band/. Accessed: 2023-07-18.

[12] Haijun Zhang, Xiaoli Chu, Weisi Guo, and Siyi Wang. Coexistence of wi-fi and heterogeneous small cell networks sharing unlicensed spectrum. *IEEE Communications Magazine*, 53(3):158–164, 2015.

[13] Anant Sahai, Niels Hoven, and Rahul Tandra. Some fundamental limits on cognitive radio. In *Allerton Conference on Communication, Control, and Computing*, volume 16621671. Monticello, Illinois, 2004.

[14] Beibei Wang and K.J. Ray Liu. Advances in cognitive radio networks: A survey. *IEEE Journal of Selected Topics in Signal Processing*, 5(1):5–23, 2011.

[15] W.A. Gardner. Signal interception: a unifying theoretical framework for feature detection. *IEEE Transactions on Communications*, 36(8):897–906, 1988.

[16] Jun Ma, Geoffrey Ye Li, and Biing Hwang Juang. Signal Processing in Cognitive Radio. *Proceedings of the IEEE*, 97(5):805–823, 2009.

[17] M.P. Olivieri, G. Barnett, A. Lackpour, A. Davis, and Phuong Ngo. A scalable dynamic spectrum allocation system with interference mitigation for teams of spectrally agile software defined radios. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, pages 170–179, 2005.

[18] H. Saarnisaari and P. Henttu. Impulse detection and rejection methods for radio systems. In *IEEE Military Communications Conference, 2003. MILCOM 2003.*, volume 2, pages 1126–1131 Vol.2, 2003.

[19] Janne J. Lehtomaki, Johanna Vartiainen, Markku Juntti, and Harri Saarnisaari. Spectrum Sensing with Forward Methods. In *IEE MILCOM 2006*, pages 1–7, 2006.

[20] J. Vartiainen, J.J. Lehtomaki, and H. Saarnisaari. Double-threshold based narrowband signal extraction. In *2005 IEEE 61st Vehicular Technology Conference*, volume 2, pages 1288–1292 Vol. 2, 2005.

[21] Ye Yuan, Zhonghua Sun, Zhihao Wei, and Kebin Jia. DeepMorse: A Deep Convolutional Learning Method for Blind Morse Signal Detection in Wideband Wireless Spectrum. *IEEE Access*, 7:80577–80587, 2019.

[22] Xiong Zha, Hua Peng, Xin Qin, Guang Li, and Sihan Yang. A deep learning framework for signal detection and modulation classification. *Sensors*, 19(18), 2019.

[23] Hao Huang, Jian-Qing Li, Jiao Wang, and Hong Wang. FCN-Based Carrier Signal Detection in Broadband Power Spectrum. *IEEE Access*, 8:113042–113051, 2020.

[24] Meiyan Lin, Xiaoxu Zhang, Ye Tian, and Yonghui Huang. Multi-signal detection framework: A deep learning based carrier frequency and bandwidth estimation. *Sensors*, 22(10), 2022.

[25] Timothy J. O'Shea, Johnathan Corgan, and T. Charles Clancy. Convolutional radio modulation recognition networks. In *Engineering Applications of Neural Networks*, pages 213–226. Springer, 2016.

[26] Erma Perenda, Sreeraj Rajendran, Gerome Bovet, Sofie Pollin, and Mariya Zheleva. Learning the unknown: Improving modulation classification performance in unseen scenarios. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[27] Erma Perenda, Sreeraj Rajendran, Mariya Zheleva, Gérôme Bovet, and Sofie Pollin. Contrastive learning with self-reconstruction for channel-resilient modulation classification. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023.

[28] Changbo Hou, Guowei Liu, Qiao Tian, Zhichao Zhou, Lijie Hua, and Yun Lin. Multisignal Modulation Classification Using Sliding Window Detection and Complex Convolutional Network in Frequency Domain. *IEEE Internet of Things Journal*, 9(19):19438–19449, 2022.

[29] Johanna Vartiainen, Heli Sarvanko, Janne Lehtomaki, Markku Juntti, and Matti Latva-aho. Spectrum Sensing with LAD-Based Methods. In *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2007.

[30] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio and Electroacoustics*, 15(2), 1967.

[31] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[34] Aurélien. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, chapter Chapter 4. Training Models. O'Reilly Media, Inc., 3rd ed. edition, 2022.

[35] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision – ECCV 2020*, page 213–229, 2020.

[36] David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.