A Downscaled Faster-RCNN Framework for Signal Detection and Time-Frequency Localization in Wideband RF Systems

K. N. R. Surya Vara Prasad, *Student Member, IEEE*, Kevin B. D'souza, *Student Member, IEEE*, and Vijay K. Bhargava, *Life Fellow, IEEE*

***Abstract*— We propose a wideband spectrum sensing technique to detect and localize wireless radio frequency (RF) signals of interest in time and frequency when uninteresting signals cause RF interference (RFI). Specifically, we adopt and downscale the existing Faster-RCNN (FRCNN) framework to achieve better signal detection and localization than the state-of-the-art. For experimental evaluation, we present a data generation framework for Wi-Fi as the signals of interest and the Bluetooth and microwave oven signals as the RFI. Experiments reveal that (i) the downscaled FRCNN model can achieve up to a mean average precision (mAP) of 0.8, significantly outperforming the state-of-the-art, (ii) feature extraction with the VGG-13 architecture gives the best mAP with pretrained weights and configured as trainable, (iii) for signal detection in real RF traces, when compared to training purely with synthetic RF data, a better mAP can be achieved by training with a mixture of synthetic and real RF traces or by finetuning the synthetically-trained weights with an additional round of training on a small amount of real RF traces, and (iv) the mAP performance decreases as the signal to noise ratio (SNR) is lowered.**

***Index Terms*— Signal detection, time-frequency span estimation, short-time Fourier transform (STFT), deep learning, supervised learning.**

I. Introduction

With the emergence of the internet of things (IoT), we are currently witnessing a steep surge in the number of wireless devices around us. In future wireless systems, with the IoT devices and unmanned aerial vehicles (UAVs) coexisting with mobile phones, it becomes imperative to distinguish between these devices from a security and spectrum management point of view. Technology that can detect and categorize heterogeneous wireless signals and localize their time-frequency span can be commercialized into wireless products. For example, if we detect the presence of a narrowband transmitter and localize the wireless transmissions in time and frequency, we can build commercial wireless security products to send emergency alerts about the presence of an unexpected wireless device and/or use narrowband signal jammers to block the signals from the device. The time-frequency localization also gives us access to information on which time-frequency resources are under-utilized and are subject to minimal interference. We may then build smart spectrum allocation products to serve an increased number of devices per unit area. For example, if we decode the frequency-hopping pattern of a given device, we may use a time delay and re-allocate the same set of time-frequency resources to a different device operating with the same dwell-time, bandwidth, and hopping period. Such smart spectrum allocation products would be of immense commercial value in the license-free ISM band, especially in the internet of things (IoT) era where we are experiencing a significantly increased density of wireless devices. In order to enable the applications mentioned above, we need to develop an algorithm which can perform two major tasks simultaneously: (i) detect the presence of a wireless signal in a wideband radio frequency (RF) spectrum and (ii) localize its time-frequency span. In this regard, we propose a deep learning solution when RFI is caused by uninteresting signals.

For simultaneous extraction of the time and frequency span, we need a time-frequency representation of the spectrum where both the time and frequency spans are simultaneously exposed. We focus on the short-time Fourier transform (STFT) [7], which is a single time-frequency resolution method, i.e., the same time-frequency resolution is maintained across the wideband spectrum irrespective of the center-frequency of the narrowband transmissions. We cannot directly work with the raw STFT matrices because they are complex-valued. Instead, we work with the spectrograms, which are the log-transformed STFT magnitude matrices. While the STFT magnitudes are known to be sufficient for separating and reconstructing the underlying signals [7], a further log-transformation offers better control over the numerical range of the STFT magnitudes for different frequencies. It is therefore commonplace to work with the spectrogram matrices.

Alternatives to the STFT include the continuous wavelet transform (CWT), the constant-Q transform (CQT), the Wigner-Ville transform (WVT), and the S-method [8], [9].

Manuscript received February 5, 2019; revised August 7, 2019 and February 3, 2020; accepted April 3, 2020. Date of publication April 21, 2020; date of current version July 10, 2020. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by the Skycope Technologies Inc., BC, Canada, through the MITACS Accelerate Program. This article was accepted for presentation in part at the IEEE VTC-Spring 2020 and was submitted for presentation in part to the IEEE VTC-Fall 2020. The associate editor coordinating the review of this article and approving it for publication was P. Casari. *(Corresponding author: K. N. R. Surya Vara Prasad.)*

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, British Columbia, V6T 1Z1, Canada (e-mail: surya@ece.ubc.ca; kevin@ece.ubc.ca; vijayb@ece.ubc.ca).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2987990

The CWT and CQT have a frequency-dependant time-frequency resolution, wherein the lowest frequency components are least smeared in frequency and the highest frequency components are the most smeared in frequency. Such variable resolution methods are not preferable for spectrum monitoring and smart spectrum allocation applications, where all the frequencies within the wideband spectrum are equally important. The WVT and the S-methods, on the other hand, offer a single time-frequency resolution but with a few shortcomings over the STFT. The WVT provides a finer time-frequency resolution than the STFT but introduces undesirable artifacts (cross-terms) and an increased amount of aliasing. The S-method minimizes the cross-terms introduced by the WVT but it is non-invertible, making it difficult to recover the underlying narrowband signals. We therefore work with the STFT spectrogram matrices in this work.

Limited literature exists, however, on the use of spectrograms for joint signal detection and time-frequency localization. Existing works [18]–[22] have considered wireless captures that are limited to interference-free and/or white Gaussian noise scenarios. Also, the performance is subject to hand-crafted hyperparameters which do not generalize well across diverse noise types, fading conditions, and propagation scenarios. The resulting arbitrariness in performance may be overcome through supervised learning, wherein a machine learning model is trained with spectrograms containing the signals of interest, alongside various types of RFI and noise. The model can learn complex features which facilitate signal detection despite the imperfections caused by noise and interference. With this objective in mind, we propose a supervised machine learning framework based on a downscaled form of the Faster RCNN (FRCNN) [4] to achieve signal detection and time-frequency localization task. Multiple design insights are provided to carefully adopt the FRCNN framework. For experimental evaluations, we consider an example setup where Wi-Fi signals are to be detected under RFI from Bluetooth and microwave oven signals. The proposed deep learning system significantly outperforms state-of-the-art morphological processing [21] and support vector machine (SVM) [22] methods on both synthetic and real RF traces. A detailed literature review is presented next.

II. Literature Review

To report the presence of unexpected signals and localize them, few authors have developed changepoint detection algorithms to localize only the frequency span or time span information, while others have directly studied the joint time-frequency localization problem.

1) Changepoint Detection for Frequency-Span or Time-Span Localization: Recent works [10]–[13] have studied the problem of signal detection and frequency span localization through changepoint detection. To report the occupied frequencies over a predefined time period, an energy-based thresholding was pursued in [10], [11], an energy-based statistical significance testing procedure was pursued in [12], and a more involved multi-scale wavelet edge detection method was pursued in [13]. These methods require significant additional work to simultaneously report the time span of each signal. For example, the same changepoint detection algorithms need to be run on each detected frequency to obtain the time span. The results over multiple frequencies need to be carefully grouped into disjoint signals. In addition, due to the nature of the wireless channel, the energy-based methods [10], [12], [13] do not achieve robust performance under noisy and fading conditions, whereas the wavelet method [13] requires handcrafted hyperparameters which do not generalize well beyond a handful of captures with varied noise and interference types. Also, none of these works study RFI separation, so additional classification work is required to filter out the interference. The work [11] classifies the detected signals into different wireless technologies, but is only limited to signals which exhibit cyclostationarity.

More generally, the works [14]–[17] have studied statistical feature-based changepoint detection in sequential data. These methods perform better than simple energy-based or wavelet-based edge detection methods and can be extended for the time-frequency localization task with a certain amount of post-processing. For example, we can study one STFT time-bin (frequency-bin) at a time and detect all the frequency (time) changepoints. To achieve the time-frequency localization, we can then develop a *multi-time-bin (multi-frequency-bin) changepoint combining algorithm* to combine the changepoints over multiple time-bins (frequency-bins) and a *signal classification algorithm* to separate out the RFI from the detected signals. Such a per-time-bin (per-frequency-bin) STFT analysis, although logically valid, is not worth pursuing because the state-of-the-art changepoint detection algorithms, such as RuLSIF method [14], the binary segmentation method [15], and the kernel changepoint procedure [16], incur linear complexity in the number of samples [17] and therefore time-consuming. There are also no clear guidelines available for tuning the hyperparameters in these methods, forcing us to rely on sub-optimal and computationally exhaustive grid-search.

2) Joint Time-Frequency Localization: Regardless of the type of time-frequency representation, only a limited number of works have studied the problem of signal detection and time-frequency localization. Time-frequency representations such as the STFT simplify the signal detection task into that of detecting bounding boxes for the signals of interest. Previous works [19]–[21] have employed simple border detection methods based on digital signal processing to obtain the bounding boxes. Different sequences of morphological processing operations are proposed in these works to localize the signals in an interference-free environment. When compared to [19] and [20], the work [21] offers improved signal detection with fewer number of hyperparameters and is the state-of-the-art in morphological processing. Numerical experiments however reveal that the detection performance of [21] is poor on datasets containing more than a handful of captures (c.f. Section VI-D.1 and Fig. 7). The same is the case with other morphological processing methods [19], [20]. This is because the morphological processing, by nature, suffers from limited hyperparameter generalizability, i.e., the hyperparameters which work well for a given RF

capture and a given noise/interference level do not work well for other captures and interference levels.

Since the major concern is the limited hyperparameter generalizability across multiple captures and interference levels, a natural solution is to devise a supervised machine learning approach wherein a model is trained to detect the signals of interest in the presence of varied levels of noise and interference. Unfortunately, no previous works have explored this research direction. Deep learning methods are the state-of-the-art for supervised machine learning but we need to firstly verify if the signal detection task can be achieved using simpler and more traditional machine learning methods. For the purpose of exposition, we consider a support vector machine (SVM) classification method [22] which uses the histogram of gradient (HOG) features for object detection. The SVM-HOG method, which is quite popular in the computer vision community, can be extended for the signal detection task in spectrogram matrices. We notice through experiments on example RF traces in Section VI-D.1 and VI-E.1 that the SVM-HOG method achieves a detection performance which is much better than the morphological processing method [21] but is still worse when compared to the proposed deep learning method. Also, there are no clear guidelines available for the hyperparameter tuning. An exhaustive grid search is not only sub-optimal but also computationally prohibitive because the training HOG feature set needs to be regenerated for each hyperparameter combination. For these reasons, we consider a deep learning solution.

Recent advances in deep learning for time-series and matrix analysis [4] allow us to extract rich features out of RF data and utilize convolutional neural networks (CNNs) for improved and fairly generalizable signal detection. In this work, we propose a downscaled form of the Faster region-based convolutional neural network (FRCNN) [4] architecture for the signal detection and time-frequency localization under RFI. The downscaled FRCNN comprises of two CNN stages: a region proposal network (RPN) and a Detector network. While the RPN provides a set of candidate regions within the spectrogram for the signal of interest, the Detector network classifies and regresses upon the candidate regions to provide tight bounding boxes for the signals. Our main contributions are as follows:

- We identify that simple border detection algorithms such as the morphological processing method [21] achieve poor performance on datasets containing more than a handful of captures because they suffer from limited hyperparameter generalizability, i.e., the hyperparameters which work well for a given RF capture do not work well for other captures. To overcome this issue and achieve uniformly good performance on a wide range of captures, we propose a deep learning solution based on the FRCNN framework.
- To perform supervised learning on the signal detection task, we need an accurately labelled dataset. To facilitate the same, we present a detailed data generation framework for an example setup where the signals of interest are from the IEEE 802.11g (Wi-Fi) protocol and the RFI is from the Bluetooth low energy [26] and microwave oven [27] signals. Experiments reveal that the proposed deep learning framework achieves a mean average precision (mAP) of upto 0.8 for a synthetic dataset containing 150 training and 50 test captures, each spanning a bandwidth of 56MHz, duration of 90ms, and containing an average of 4.5 Wi-Fi signals in the SNR range [15, 50]dB.
- Adopting the FRCNN framework for the time-frequency localization task is not straightforward because multiple design choices need to be made, for example, on the type and the depth of the feature extraction network, the number of anchors and their sizes, the size of the RPN and the Detector, and on the multiple numerical thresholds within the framework. We provide crucial insights on making these design choices, all based on the type of signals we are interested in. The best performance is achieved when the feature extraction network is a pretrained VGG-13 configured as trainable, the anchor sizes are the same as those of the signals of interest, the RPN and Detector networks are downscaled from the default architecture by a factor of 2, and the numerical thresholds within the RPN and Detector networks are set as (*RPN min overlap, RPN max overlap, Detector min overlap, Detector max overlap*) $= (0.1, 0.7, 0.1, 0.5)$. The proposed deep learning method significantly outperforms the state-of-the-art morphological processing [21] and SVM classification [22] methods.
- For real test captures, directly deploying the synthetic-trained model results in significant performance loss due to the inherent data mismatch between the training and test captures. To alleviate the performance loss, we consider two solutions, namely *data mixing* and *model finetuning*. In *data mixing*, we train with a mixture of synthetic and real measurement captures. While the synthetic data allows us to confidently establish the ground truth, the real data exposes the model to a variety of noise and interference types that cannot be modelled or are not accounted for in the synthetic data. In *model finetuning*, the synthetic-trained model is finetuned through an additional round of training on a small number of real RF traces. *Model finetuning*, which is the more effective one, is seen to offer an mAP of up to 0.6204 on the real test captures.
- Before we proceed to on-field deployment, we need to further study whether the proposed framework generalizes well across different signal-to-noise ratios (SNRs). Our experiments with the downscaled FRCNN reveal that the mAP decreases as the SNR is lowered, calling for further investigation on custom-made denoising algorithms prior to signal detection.

The rest of the paper is organized as follows. In Section III, we propose a framework for signal detection and time-frequency localization. Section IV presents an overview of the downscaled FRCNN architecture, with insights on the design choices to be made for the signal detection. A data generation framework is presented in Section V, followed by numerical evaluations in Section VI and concluding remarks in Section VII.

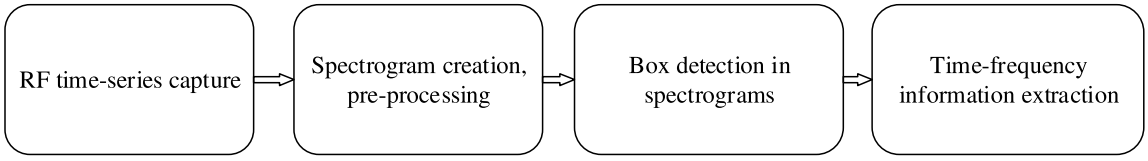
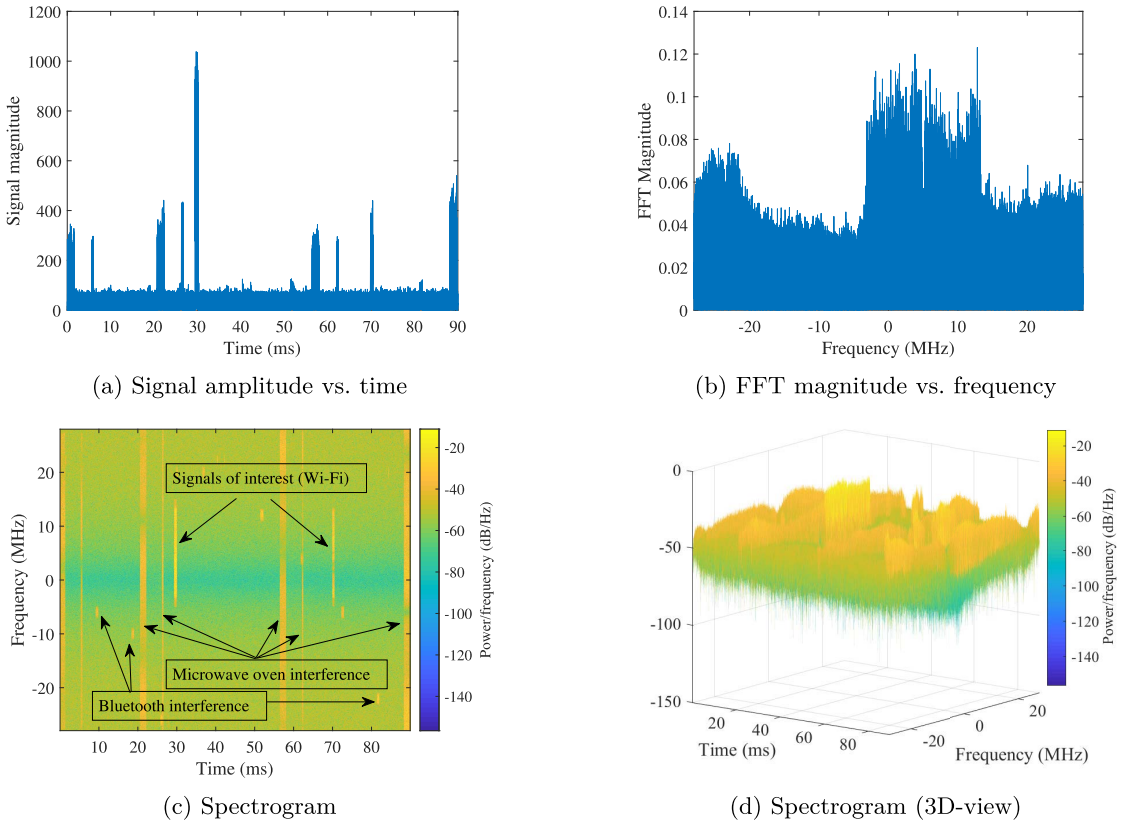


Fig. 1. Proposed framework for signal detection and time-frequency localization.



(a) Signal amplitude vs. time

(b) FFT magnitude vs. frequency

(c) Spectrogram

(d) Spectrogram (3D-view)

Fig. 2. The time content, frequency content, and spectrogram of an example wideband RF capture of duration 90ms, center frequency 2.457GHz, bandwidth 56MHz, and sampling rate 56MHz respectively. The signals of interest are the Wi-Fi signals and the radio frequency interference constitutes Bluetooth low energy and microwave oven signals.

III. Framework for Signal Detection and Time-Frequency Localization

We propose a deep learning framework to detect and estimate the time-frequency span of the signals of interest in a wideband RF spectrum. The framework takes the wideband RF time-series data as the input and provides the time and frequency information of the interesting signals as the output. An outline of the proposed framework is presented in Fig.1.

3) RF Time-Series Capture: In the first stage, we employ a wideband sensor with center frequency f_c and bandwidth W to record time-series RF data in fragments of T milliseconds each. The time and frequency content of an example wideband capture with $f_c = 2.457$GHz, $W = 56$MHz, $T = 90$ms, and a sampling rate of 56MHz is given in Fig. 2. We need to localize the Wi-Fi signals in time and frequency when RFI exists in the form of Bluetooth and microwave oven signals. The signal amplitude is plotted as a function of time in Fig. 2a and the FFT magnitudes are plotted as a function of frequency in Fig. 2b.

4) Spectrogram Creation and Pre-Processing: In order to detect the signals of interest and extract their joint time-frequency information, we use the spectrogram matrices as the input format. An illustration corresponding to Figs. 2a-2b is presented in Figs. 2c-2d, with the STFT obtained using a Hann-type window of size 1400, 50% window overlap, and FFT size of 1024. As seen in Fig. 2c, the problem of signal detection and time-frequency localization boils down to that of obtaining rectangular bounding boxes in the time-frequency domain.

5) Bounding Box Detection in Spectrograms: To detect bounding boxes for the signals of interest, we take a supervised learning approach, wherein, we train a downscaled FRCNN model [4] to localize Wi-Fi signals under noise and RFI. The trained model, when input with a test spectrogram matrix, detects the bounding boxes for Wi-Fi signals and reports their

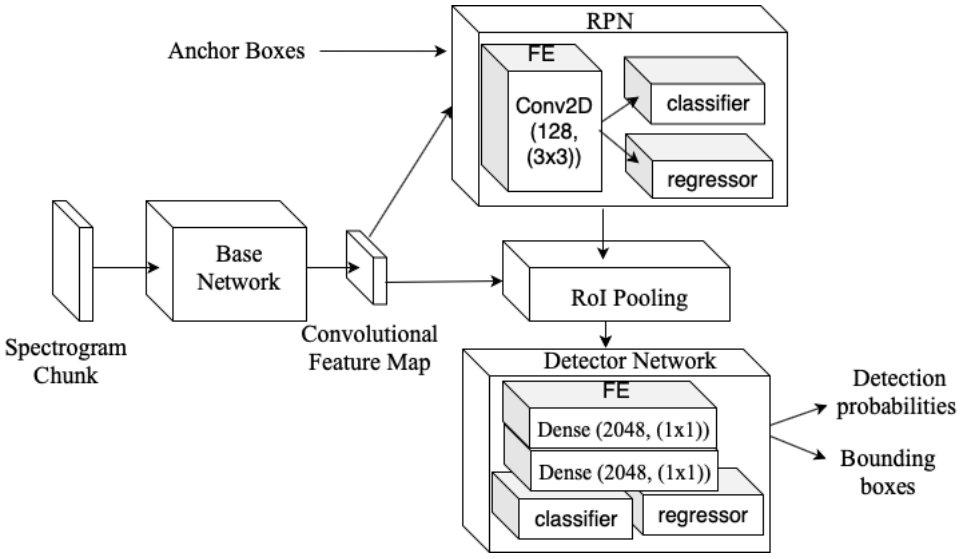


Fig. 3. The downscaled FRCNN model. For each spectrogram chunk, the base network yields a feature map which is fed into the region proposal network (RPN) and the Detector network. The RPN and the Detector host a bunch of feature extraction (FE) layers, the outputs of which are used for classification and regression to yield bounding boxes.

time-frequency span. Details on the downscaled FRCNN are provided in Section IV.

6) Time-Frequency Information Extraction: The machine learning model is trained to report the dimensions of the bounding box of the signals and not the absolute time and frequency span information, because the attributes to be learned need to be in a similar scale for the training to be successful. The absolute time and frequency attributes have vastly different scales, for example, in the order of milliseconds and MHz respectively. Therefore, as the final step, we use the time and frequency resolution information from the STFT parameters to linearly scale each predicted bounding box into the start time, end time, start frequency, and end frequency of the detected Wi-Fi signal. Next, we present insights on tailoring the downscaled FRCNN model to suit the time-frequency localization task.

IV. A Downscaled Faster RCNN Framework for Signal Detection and Localization

The downscaled FRCNN framework is composed of three major modules, as illustrated in Fig 3. The first module, which is the base network (BN), takes the 2D matrix as the input (spectrogram in our case), extracts features that are relevant to the signal detection task and outputs a downscaled feature matrix. The second module, which is the region proposal network (RPN), takes as input the down-scaled feature matrix, a set of anchors and the ground truths (GTs). The RPN firstly performs feature extraction (FE) on the input feature map using a 2D convolution layer containing 128 filters of size 3×3. On each pixel of the output feature map, the RPN positions the anchors and performs a classification and a regression task to output region proposals, which are the likely candidate regions for the signals to be present, along with the probabilities of signal presence. The region proposals from the RPN, along with the feature map from the BN, are fed into the Detector network, which is the third module. The Detector network performs feature extraction on the region proposals using two densely-connected neural network layers, each containing 2048 filters of size 1×1. The Detector then assigns object class labels to each region proposal and also performs regression on the positive region proposals to tighten them into bounding boxes. Also provided are the probabilities with which the assigned labels are true. When spectrograms are provided as input, the FRCNN can be thought of as a single unified framework for detecting and localizing the bounding boxes for the signals of interest. Next, we present design insights on tailoring each module within the downscaled FRCNN model for the signal detection and localization task.

7) Spectrograms: As inputs, we need to generate spectrograms such that the time and frequency resolution is sufficient to capture the time-frequency span of the signals of interest. Also, care should be taken to ensure that the size of the input matrix is not too large. Firstly, when we use large matrices as input to the model, each pass of training takes a long time to complete because of the large input feature map, resulting in long convergence times. Secondly, the base network that extracts the features has a particular receptive field depending on the choice of the base network, for example, the VGG-13 and RESNET-50. When the receptive fields are large when compared to the size of the signals to be detected, it is difficult for the base network to make sense of the features to be extracted because the input would mostly contain background noise and only a small portion is occupied by the signals of interest. In light of these two observations, we restrict the length of our input matrix to 600 pixels in time. This is achieved by chopping the input spectrogram into chunks of 600 time-pixels each and working with the relative positions of the ground truths. Doing so ensures that the size of the signals of interest is comparable to the size of the input spectrogram chunk. This approach is also seen to decrease the training time significantly.

8) The Base Network: The base network (BN) in the FRCNN model handles the crucial task of feature extraction and needs to be chosen based on the input format and the type of signals to be detected. Typically, the BN is a CNN whose depth depends on the complexity of features to be extracted. Standard feature extraction models in computer vision include VGG-13 [6] and RESNET-50 [28], among others. It is unclear whether these standard models can extract necessary features for signal detection in RF datasets. We therefore conduct numerical experiments in Section VI to choose an appropriate BN.

9) Region Proposal Network: In the RPN, the feature map obtained from the BN is passed through a 2D convolution layer, which contains 128 filters of size 3×3, to obtain a low-dimensional feature map. The RPN positions the anchors on each pixel in the input feature map to create raw region proposals. The low-dimensional feature map, along with the raw region proposals created from the anchors, are fed into two 2D convolution layers to perform a classification and a regression task respectively. The classification task assigns probabilistic labels to each raw proposal as positive or negative, to denote whether the proposal contains the signal of interest or not. For proposals that are deemed positive, the regression task fine-tunes the size to suit the dimensions of the signal.

To train the RPN, if any given anchor has an intersection over union (IoU) greater than a certain threshold with the ground truth, namely the *RPN max overlap*, we treat that anchor as a positive proposal. Similarly, if the IoU is smaller than a certain threshold, namely the *RPN min overlap*, we treat the anchor as a negative region proposal. For pixels that are on or close to the feature map boundary, we clip the anchors to limit their span to the matrix size. The clipped sizes are used for the IoU calculations. On the other hand, ground truths which are on the feature map boundary are ignored from the training procedure because we experienced convergence issues when they were included. This is not a major concern though, because we can always detect signals that have been abruptly cut at the boundary of a spectrogram chunk by feeding an overlapping spectrogram chunk to the FRCNN model and combining the results through simple post-processing. The classification layer is then trained to separate positive and negative proposals. The regression layer tightens the positive proposals to bound the ground truths. Since many of the positive proposals would have a high overlap with each other, a non-max suppression (NMS) operation is employed.

10) Anchors: The anchors serve as raw region proposals for the RPN. We may therefore choose the anchor sizes and aspect ratios to match the dimensions of the signals of interest. For example, if we are interested in the IEEE 802.11g (Wi-Fi) protocol, we know that the signals have a bandwidth of 20MHz and duration from about 0.05 to 15 milliseconds [5]. We may therefore choose multiple anchors, each with a frequency span of 20 MHz and a time-span chosen uniformly randomly in the range [0.05, 15] ms.

11) Detector Network: The RoIs from the RPN may have varied sizes, depending on our choice of the anchors and the output of the regression. An RoI convolution pooling operation [4] is performed to convert the RoIs into a fixed size, typically 7×7, so as to be able to use convolution layers for further classification and regression. After the RoI convolutional pooling, the fixed size RoIs are fed into two densely-connected layers, each containing 2048 filters of size 3×3, to convert them into low-dimensional feature vectors. These feature vectors are then input to two densely-connected layers to perform a classification and a regression task. The classification assigns probabilistic positive and negative labels to each RoI, depending on the presence of the signal. The regression fine-tunes each positive-labelled RoI to match the signal dimensions and outputs the four corner coordinates of the regressed RoI. When training, any RoI whose IoU with the ground truth is greater than *Detector max overlap* is a positive target. If the IoU is less than the *Detector max overlap* but greater than *Detector min overlap*, the RoI is a negative target.

12) Downscaling the Default FRCNN: The default FRCNN model hosts (i) one 2D convolution layer of 256 3×3 filters for feature extraction within the RPN, and (ii) two densely-connected neural network layers of 4096 1×1 filters each for feature extraction within the Detector. While these sizes were configured for general-purpose object detection in the ImageNet dataset, it is unclear whether such large feature extraction layers are required for the signal detection and localization task at hand. The signal detection task requires detecting signals of a much smaller variety than in general-purpose object detection. Also, unlike in ImageNet, the input matrices for signal detection are predominantly occupied by different types of background noise. In addition, the numerical range of values is much smaller in spectrograms and the numerical fluctuations within the signal are much more rapid and predominant (c.f. Fig. 2(d) for example). This follows from the nature of how wireless signals are transmitted. We therefore conduct numerical experiments on the default FRCNN architecture in Section VI with downscaling factors of 2, 4, and 8 for the RPN and the Detector. Downscaling by a factor of 2 is seen to give the best performance, thus leading us to choose 128–filter feature extraction layer in the RPN and 2048–filter feature extraction layers in the Detector.

13) RPN and Detector Thresholds: The *RPN max overlap* needs to be chosen such that at least a few anchors per spectrogram have a high enough IoU with the ground-truth to be considered as positive targets for the RPN training. Typically, the *RPN max overlap* is chosen to be greater than or equal to 0.5 because an IoU of 0.5 or more gives us confidence that the anchor indeed contains the signal of interest. Similarly, the *RPN min overlap* needs to be chosen such that at least a few anchors per spectrogram have a low enough IoU with ground-truth to be considered as negative targets for the RPN training. Typically, the *RPN min overlap* needs to be less than 0.5 because an IoU less than 0.5 denotes that the anchor may not contain the signal of interest. Higher *RPN max overlap* and lower *RPN min overlap* values would deem fewer anchors as positive and negative targets respectively.

The *Detector max overlap* needs to be chosen so that at least a few RoIs per spectrogram have a high enough IoU with the ground-truth to be considered as positive targets for training the Detector. Typically, the *Detector max overlap* is chosen

to be at least 0.5 because an IoU of 0.5 or more gives us confidence that the RoI indeed contains the signal of interest. On a similar note, the *detector min overlap* needs to be chosen such that at least a few RoIs per spectrogram have an IoU value between the *Detector max and min overlap* to be considered as negative targets for the training. Typically, the *Detector min overlap* needs to be less than 0.5 because an IoU less than 0.5 denotes that the RoI may not contain the signal of interest. The higher the *Detector max overlap*, the fewer the number of positive target RoIs. Similarly, the smaller the difference between the *Detector max and min overlap*, the fewer the number of negative target RoIs. In Section VI, we attempt a simple one-pass search algorithm to choose the *RPN and Detector min and max overlap* values.

14) Training Procedure and Dataset: For the classification and regression tasks in both the RPN and the Detector networks, we use the standard weighted-sum multi-task loss function proposed in [4]. The loss function is a weighted sum of a binary cross-entropy loss term for classification and a robust loss term for regression. Similar to [4], we employ the *approximate joint training* method to train our system. To perform supervised learning for the Wi-Fi signal detection, we need a dataset comprising a variety of time-series RF captures measured in the ISM 2.4 GHz wideband spectrum. Each RF capture needs to contain the RFI experienced in the ISM band and needs to be accurately labeled for the Wi-Fi signals. Training with real RF traces is obviously the most desired option because it allows us to build a tailor-made model for field deployments, with the added convenience of not having to explicitly build mathematical simulation models for the noise and interference. However, a major challenge with training on real RF traces is the data availability. Extensive measurement campaigns need to be conducted in order to capture a sufficiently wide variety of RF traces. If we do not capture enough data and instead choose to train with a small amount of real data, the model would simply overfit and provide poor test performance. In the event that one is able to collect a large amount (and variety) of real RF traces, an additional challenge is to accurately establish the ground truths and label them. Standard packet demodulation procedures are non-ideal and their performance is sensitive to the received SNR [25]. Missed or incorrect labels would confuse the model during the training and therefore result in poor test performance. Manual labeling is an alternative solution but is not scalable. Consequently, in order to pursue supervised learning, we need to train with synthetic traces.

Training with synthetic data, although more of a requirement than a choice we make, offers a variety of benefits. Firstly, synthetic traces allow us to confidently establish the ground truths. The ability to establish ground truths empowers us to create large amounts of accurately labelled data, which is a fundamental requirement for supervised learning. Secondly, by using synthetic traces for training, one can expose the model to a wider variety of noise, interference, and fading types that are difficult in general to capture through a finite amount of field measurements. Lastly, one may note that the synthetic traces are generated using simulation models which were originally developed from real measurement studies. Consequently, training with synthetic data tailors the model to the task at hand, albeit with a certain data mismatch that needs to be overcome prior to field deployment. For on-field deployments, we consider two solutions to overcome the data mismatch between synthetic and real RF traces: (i) *data mixing*, wherein, we train the model with a mixture of synthetic and real traces, and (ii) *model finetuning*, wherein, we further train the synthetically-trained model on a small amount of real measurements.

In the next section, we present a detailed data generation framework to create synthetic RF captures containing Wi-Fi signals in the 2.4GHz ISM band. For the RFI, we consider the ubiquitous Bluetooth low energy [26] and microwave oven [27] signals. Multiple fading and coloured noise types are considered for the signal degradation. While the basic simulation models for Wi-Fi, Bluetooth, and microwave oven signals are available online, there is no available framework which connects these models together in a meaningful manner, with appropriate numerical values for the multiple variables involved, for example, the transmission powers, narrowband frequencies, inter-packet delay times, frequency hopping indices, and the microwave oven properties. The proposed framework bridges this gap.

V. Data Generation Framework for Wi-Fi Signals in the 2.4GHz ISM Band

We consider RF transmissions in the 2.4 GHz ISM band as per the IEEE 802.11g (Wi-Fi) protocol [5] and generate the time-series data synthetically using MATLAB WLAN toolbox. Each generated RF capture is centered in the 2.4 GHz range, has a wideband bandwidth of 56 MHz and spans a duration of 90ms. Since the channels 1, 6, and 11 are the most popular and each channel spans 20 MHz, depending on the center frequency, the capture may contain one or two channels of Wi-Fi transmissions. On an average, each RF capture contains about 4.5 Wi-Fi signal packets, with the average packet spacing obtained upon observing real Wi-Fi captures from a NI USRP-2901 device [31]. We also randomly add radio frequency interference (RFI) comprising Bluetooth low energy (BLE) and microwave oven (MO) signals. All the signals, whether Wi-Fi or the RFI, are subject to distance-dependent signal attenuation, small-scale fading, and additive white/colored noise. The small-scale fading effects are simulated using the MATLAB LTE toolbox to be one among the tap-delay line implementation of the SISO Extended ITU outdoor to indoor and pedestrian model or the SISO Extended typical urban model [30]. The additive noise is randomly chosen to be one among the white, pink, red, and purple types and is generated using the MATLAB DSP System toolbox. Details on the data generation are presented in **Algorithm 1**. As inputs to the **Algorithm 1**, we provide $f_s = 56$MHz, $l = 90$ms, $N_c = 200$ captures, $P_{\text{Wi-Fi}} = -10$dB, $P_{\text{BLE}} \in \{-30, -26, -10\}$dB, $P_{\text{MO}} = -10$dB, $d_{\text{Wi-Fi}} \in \mathcal{U}[5, 50]$m,[1] $d_{\text{BLE}} \in \mathcal{U}[2, 30]$m, and $d_{\text{MO}} \in \mathcal{U}[2, 30]$m. The 200 captures amount to a total of 885 Wi-Fi signals. For this $(P_{\text{Wi-Fi}}, d_{\text{Wi-Fi}})$ combination, the received signal-to-noise ratio spans the range [15, 50]dB. From the total dataset, 150 captures are used for training

Algorithm 1 : RF Data Generation With Wi-Fi Signals and Interference From Bluetooth Low Energy (BLE) and Microwave Oven (MO) Signals

Require: Sampling rate f_s, wideband capture length l, number of wideband captures N_c.

Require: Transmit powers and distances of the Wi-Fi source (P_{WiFi}, d_{WiFi}), BLE source (P_{BLE}, d_{BLE}), and microwave oven (P_{MO}, d_{MO}).

Choose a center frequency for the wideband RF capture f_c (GHz) $\leftarrow$ $2.412 + 5e6 * \mathcal{UI}[0,\ 10]$

Choose one or more of the channels $\{1, 6, 11\}$ for Wi-Fi transmissions

for each Wi-Fi channel, until the capture length is l, **do**

  generate Wi-Fi packets as per IEEE 802.11g [5] with power P_{WiFi} and inter-packet delay times $\in \mathcal{U}[0.001, 30]$ms

  record the start and end points in time and frequency for each p Preliminary work on the topic has been patented [1].et

end for

Generate and add BLE packets as per Bluetooth 5 [26] with power P_{BLE} by randomly:

- alternating between advertising channel and data channel types,
- alternating between the LE1M and LE2M physical layer modes
- introducing delays $\in \mathcal{U}$[0.1, 1]ms between successive packets, and
- frequency-hopping among the available channels (79 channels are available for LE1M and 40 for LE2M respectively).

Generate and add MO signal bursts with power P_{MO}, following [27], with

- voltage frequency $f_{AC} = 60$Hz,
- signal inter-arrival time parameters $\alpha_1 = -72$ and $\alpha_0 = 180$
- magnetron on-state times $T_{M1} \in \mathcal{U}$[1.4, 1.9]ms and $T_{M2} \in \mathcal{U}$[0.3, 0.9]ms, and magnetron frequency-drift times $T_{FD} \in \mathcal{U}$[4, 4.25]ms.

Apply distance-based signal attenuation, small-scale fading, and additive white/colored noise to all of the Wi-Fi, BLE, and MO signals.

return Wideband RF captures with recorded time-frequency labels for the Wi-Fi signals

and the remaining 50 for test purposes. For the purpose of generalizability, the following composition is used when creating both the training and test datasets: 16% contain Wi-Fi only, 16% Wi-Fi with BLE RFI, 16% Wi-Fi with MO RFI, and the remaining 52% Wi-Fi with BLE and MO RFI.

VI. Numerical Studies

We now present numerical studies on the performance of the downscaled FRCNN model for the signal detection and time-frequency localization task. Firstly, we provide details on the training and test datasets, the spectrogram generation, the numerical thresholds chosen for the FRCNN model, and the metric for performance evaluation.

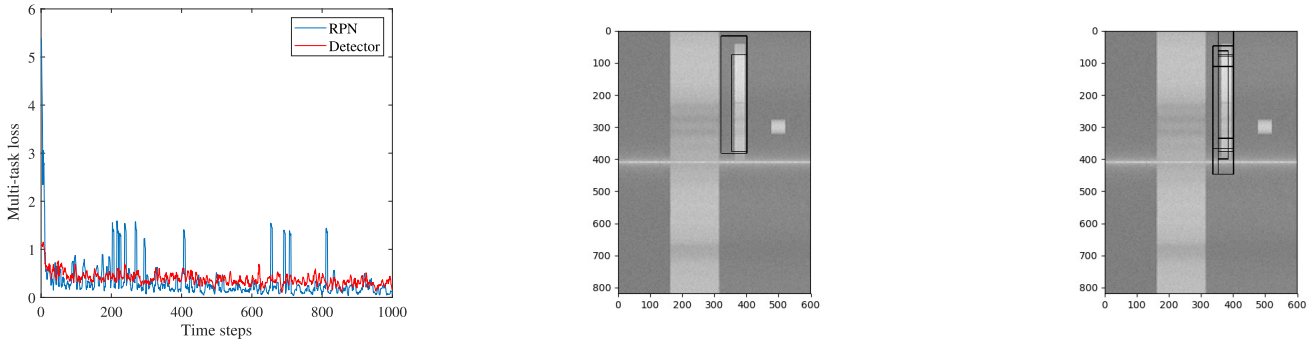
A. Spectrogram Generation and Numerical Choices for the Downscaled FRCNN

STFT is applied on each RF capture with a Hann-type window of size 1400 with 50% overlap and an FFT size of 1024. The resulting spectrograms have 7199 time bins and 1024 frequency bins each, i.e, with a resolution of 0.0125ms and 54.7kHz respectively. We chop the spectrograms into fixed chunks of 600 time-bins in order to speed up the training. Each RF capture is therefore split into 12 input matrices for the FRCNN model. When presenting example spectrograms, we focus on one input at a time, i.e, one spectrogram of time-span 7.5ms at a time. Regarding the numerical choices for the FRCNN, we consider $N_a = 3$ anchors defined such that the time-axis sizes are chosen from the set {2, 10, 80} time-bins, to represent signal time-spans of {0.025, 0.125, 1}ms respectively, and the frequency-axis size is chosen to be 366 frequency-bins, to represent a frequency-span of 20 MHz. The *RPN max overlap* and the *Detector max overlap* are chosen from the set {0.5, 0.7, 0.9 }. The *RPN min overlap* and the *Detector min overlap* are chosen from the set { 0.1, 0.3 }. Also, following [4], we set the number of RoIs N_r from the NMS operation to 300, the output RoI size from the RoI pooling network to 7×7, and the *Detector positiveness threshold* to 0.5. The λ value in the multi-task loss function (c.f. [4]) is set to 1 for the classification and regression. The mini-batch size is set to one spectrogram chunk.

B. Training Convergence

To verify the training convergence, we consider an example experiment with the base network set to VGG-13 initialized with pre-trained weights and configured as trainable. Each training mini-batch contains one spectrogram chunk and we set *RPN min overlap* to 0.3, *RPN max overlap* to 0.5, *Detector min overlap* to 0.1, *Detector max overlap* to 0.5, and the number of training epochs to 10. Stochastic gradient descent is used to optimize both the RPN and the Detector networks, with a learning rate of 10^{-5}, zero momentum and zero learning rate decay. Optimal tuning of these hyperparameters can lead to improved performance of the downscaled FRCNN, which is an interesting avenue for future work. In Fig. 4a, we plot the multi-task loss for the RPN and Detector networks as a function of time. Moving average was applied over a window of 5 time steps to focus on the trends. We notice that the sum-loss converges with time to zero. The convergence rates and the loss fluctuations depend on the learning rate of the stochastic gradient descent algorithm and the number of positive and negative targets available per mini-batch. Example positive targets on a spectrogram chunk for the RPN and the Detector networks are shown in Figs. 4b and 4c respectively. As may be noted from Sections IV-.9 and IV-.11, the RPN targets are the anchors whose IoU with the ground truth

[1]The notations $\mathcal{U}[x,\ y]$ and $\mathcal{UI}[x,\ y]$ refer to uniformly random values and uniformly random integers in $[x,\ y]$ respectively.



(a) Training loss convergence (b) RPN Targets (c) Detector Targets

Fig. 4. (a) Training loss convergence for the classification and regression tasks in the RPN and the Detector networks. (b)-(c) Example positive targets for the RPN and the Detector networks. The rectangular boxes are the targets, while the signals underneath the boxes are the ground truths. To illustrate noise and RFI, we consider colored noise and a signal each from Bluetooth (c.f. right of the targets) and microwave oven (c.f. left of the targets).

is ≥ 0.5 and the detector targets are the RoI proposals chosen after NMS operation.

C. Prediction Performance Metrics and Baseline Methods

To evaluate the performance of the FRCNN, we consider the mean average precision (mAP) metric [4], which is a standard and widely-used metric for evaluating object detection algorithms in computer vision. An overview of the mAP calculation is presented here. We begin by sorting all the bounding box predictions from the FRCNN model in the decreasing order of the Detector classification probabilities. Next, we assign a True or False label for each prediction, depending on whether it has an IoU ≥ 0.5 with any ground truth or not, and calculate the precision and recall values until the current prediction. We then consider 11 recall levels ranging from 0 to 1 in steps of 0.1 and record the maximum achieved precision for each recall level. The mAP is then obtained as the average of the precision values recorded for the 11 recall levels. Higher mAP values denote better prediction performance. The precision vs. recall curves may also be used to evaluate the performance of the model, with larger area under the curves denoting better prediction performance.

As the first baseline, we consider the morphological processing method [21]. In [21], the spectrograms are processed for time-frequency localization through (i) binarization with an energy threshold γ_{bin}, (ii) a modified morphological opening operation on the binary image with n_{erode} successive erode and n_{dilate} successive dilate operations, (iii) a connected-component labelling operation on the opened image, and (iv) an energy filtering step where the isolated regions with the top K sum-energies are reported as the signals of interest. The first step serves as an energy-based denoising step. The second step compensates (partially) for the loss of important information from the binarization step and also serves as a second level of denoising. The amount of denoising and energy compensation is governed by n_{erode}, n_{dilate}, and s. The third step isolates high-energy regions corresponding to the signals present in the capture. The final energy filtering step serves as an interference removal step. Unfortunately,

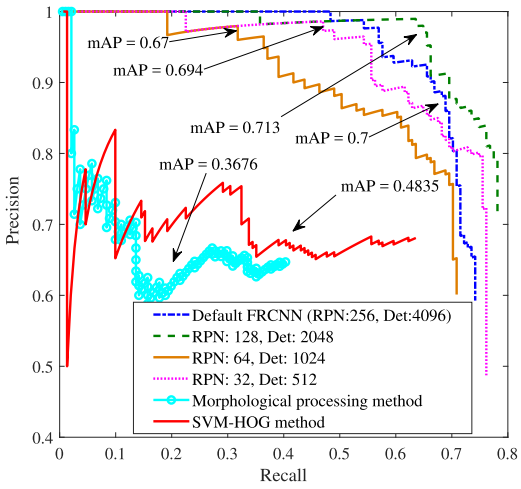


Fig. 5. Precision vs recall when the RPN and Detector in the FRCNN are downscaled by a factor of 2, 4, and 8 respectively. Downscaling by 2 gives the best mAP performance. The downscaled FRCNN model is also seen to significantly outperform the two baseline methods.

the authors in [21] do not provide much insight on how these hyperparameters can be chosen for a dataset containing more than a handful of captures with varied SNRs. Hyperparameter tuning is simply acknowledged as a crucial task but is left out for the future. Similar is the case with other morphological processing methods [18]–[20]. We therefore pursue a computationally intensive grid search with the following range of values: $\gamma_{\text{bin}} \in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$, $n_{\text{erode}} \in [1, 2, 3.4, 5, 6, 7, 8, 9, 10, 11, 12]$, $n_{\text{dilate}} = n_{\text{erode}} + 1$, $s \in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, and $m = 4$. To separate out the noise and interference in the final step, unlike in [21], we have filtered out regions that are similar in size as the Bluetooth and microwave oven signals, and also the regions whose frequency spans are more than 50% different from the signals of interest.

As the second baseline, we consider the SVM-HOG classification method [22]. The SVM-HOG method runs a sliding detection window of size $wf \times wt$ across each spectrogram, in steps of length w_{step} along each axis. A histogram of ordered gradients (HOG) feature vector is assigned to each detection

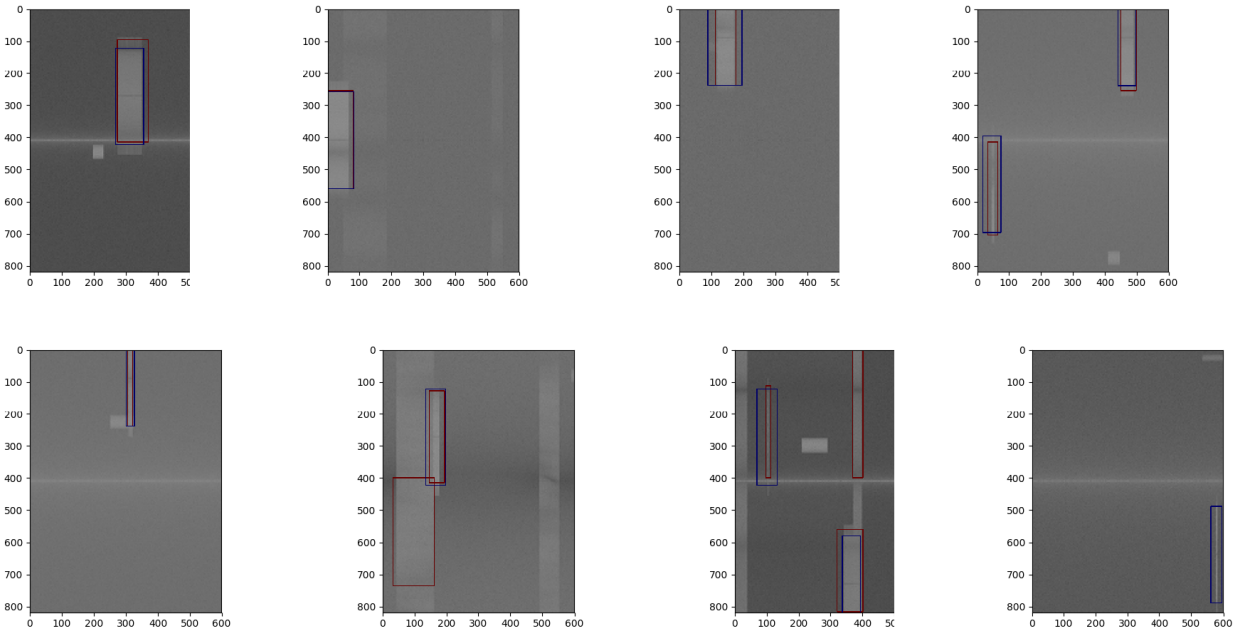
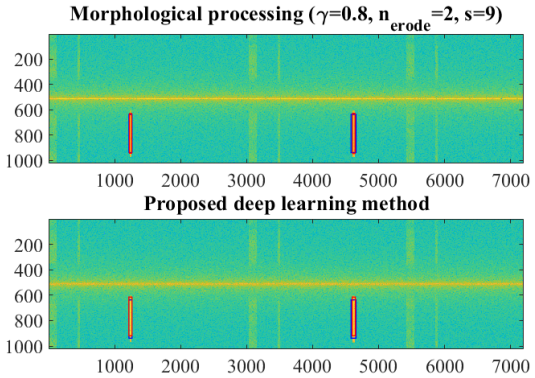


Fig. 6. Example predictions on the test spectrograms with the downscaled FRCNN model when the BN is the VGG-13 (trainable, initialized with pretrained weights). The blue and red rectangles indicate the signal ground truth and the predictions respectively. The model successfully separates out the RFI and provides tight bounding boxes for most signals of interest, even if there is an overlap with the RFI signals. The loss in mAP occurs mainly due to false positives arising from microwave oven signals.

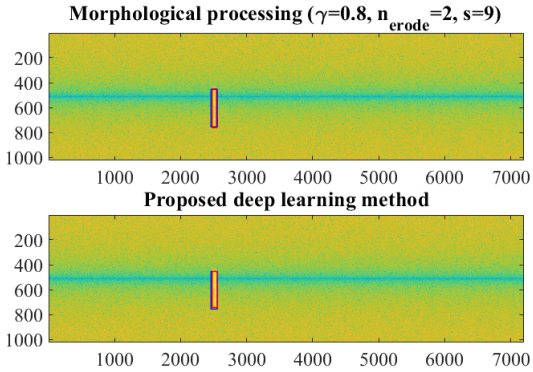
window as follows. Each window is divided into smaller regions, called cells, of size $pix \times pix$. For each cell, an HOG vector is formed by calculating the gradients at each pixel and quantizing them into ori orientation bins (see [22] for details). To reduce the impact of illumination and shadowing, the HOG of each cell is locally normalized over somewhat larger regions, called blocks, of size $blk \times blk$. The HOG descriptor for the window is then obtained by accumulating the HOGs of all the blocks within itself. Typically, we overlap the blocks with a stride of length b_{stride}, so that each cell contributes multiple HOGs to the final feature vector, each normalized with respect to a different block. Positive labels are then assigned to the windows which have IoU $\geq iou_{\text{min}}$ with the signal of interest, while negative labels are assigned to the ones having zero IoU with all the signals of interest. A linear SVM classifier is then trained with the positive and negative labelled feature vectors, in order to predict whether a window contains the signal of interest or not. Non-max suppression is then employed on the positive windows because multiple overlapping prediction windows will be encountered for each signal.

To adapt the SVM-HOG method for the signal detection and time-frequency localization task, we make the following extensions. For assigning the positive labels, while it is commonplace to use $iou_{\text{min}} = 0.5$, we have set iou_{min} to a significantly smaller value of 0.05 because unlike in [22], our ground truths have variable time-span. Since we consider much larger detection windows than in [22], the HOG vectors end up being too long for the training to converge. A commonplace solution is to transform the HOG vectors into much smaller length len_{ny} using the Nystroem approximation [23] with an RBF kernel of width γ_{ny}. For the training, the authors in [22] have considered a ratio of $1 : 10$ for the positive to negative training samples. This ratio is suitable for the images of size 320×240 considered in [22], but the same ratio results in memory problems when training the SVM for our spectrograms of size 600×1024 because of the significant increase in the size of the training dataset. We have reduced the ratio to $1 : 2$ in order to be able to fit the training dataset into the 8 Gb of RAM on our computer. We have later experimented with the ratio $1 : 10$ by partitioning the training dataset into 5 subsets and sequentially training the SVM model with one subset at a time, but this did not yield any improvement in performance. We have now also included *hard negative mining* [24] so as to achieve improvements in performance beyond regular hyperparameter tuning. In this procedure, an SVM model is initially trained with the original training feature set and all the false positives from the training features are extracted using the trained model. We then re-train the same model on an augmented dataset containing both the original feature set and the false positives. This is iterated a few times until the number of false positives ceases to decrease. Significant improvements in the performance are observed from such *hard negative mining*.

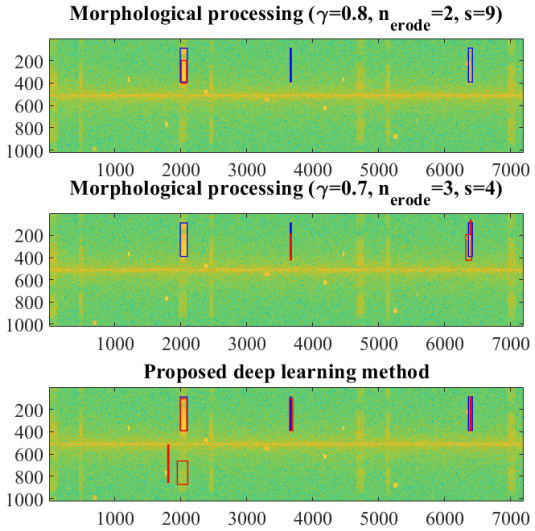
For the hyperparameter tuning, the authors in [22] do not provide much insight on how the hyperparameters can be automatically chosen and have instead relied on experimental evaluations over a range of numerical values. We have therefore resorted to an exhaustive grid search and have considered the following range of values: (i) $wf \in [320, 400, 480, 640, 800]$, corresponding to signals of bandwidth [17.5, 21.875, 26.25, 35, 43.75] MHz respectively,



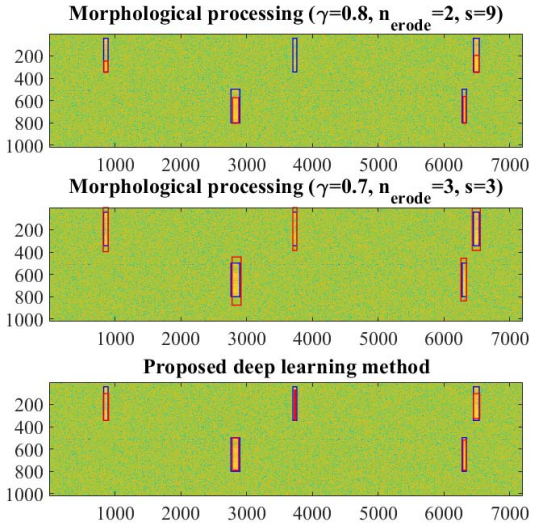
(a) Example capture 1



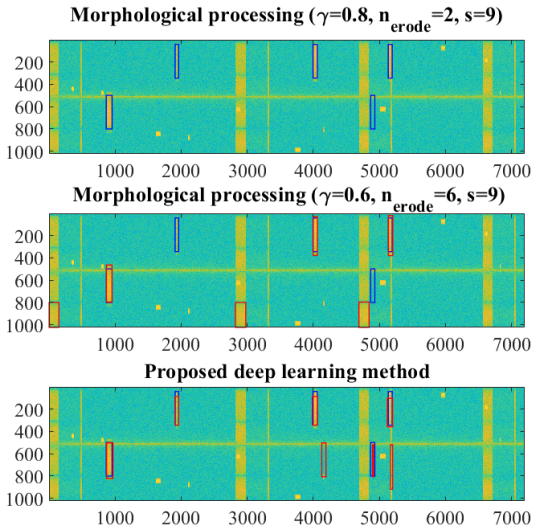
(b) Example capture 2



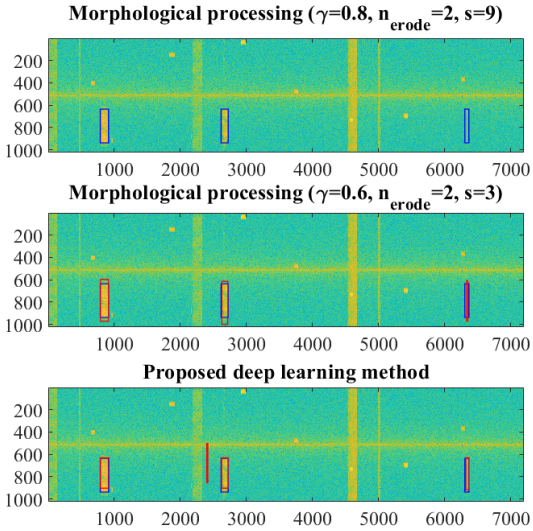
(c) Example capture 3



(d) Example capture 4



(e) Example capture 5



(f) Example capture 6

Fig. 7. Example predictions from the morphological processing method [21] and the proposed deep learning method. The optimized hyperparameters in [21] do not generalize well across captures with different noise and interference types. As a result, we observe that the optimized hyperparameter combination ($\gamma_{\text{bin}} = 0.8, n_{\text{erode}} = 2, s = 9$) results in missed detections, partially as in captures 3 and 4 or completely as in captures 5 and 6. Handcrafted hyperparameters, found through trial and error, can provide improved performance but handcrafting is an impractical solution. The proposed deep learning method, on the other hand, provides uniformly good performance.

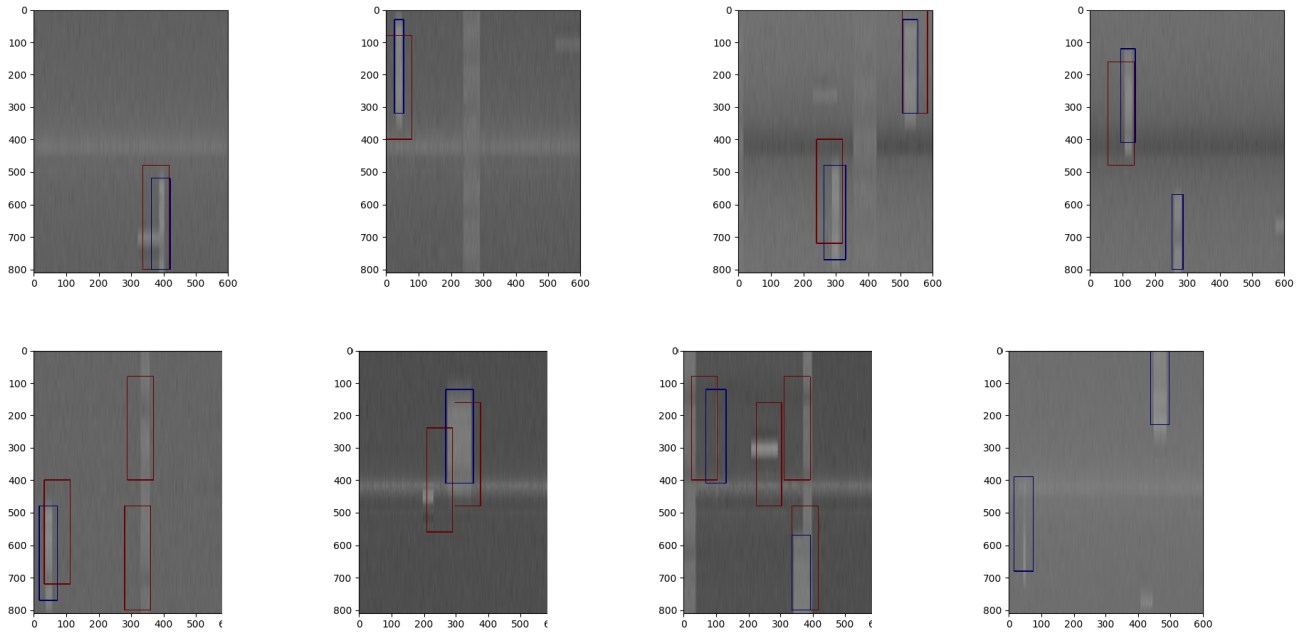
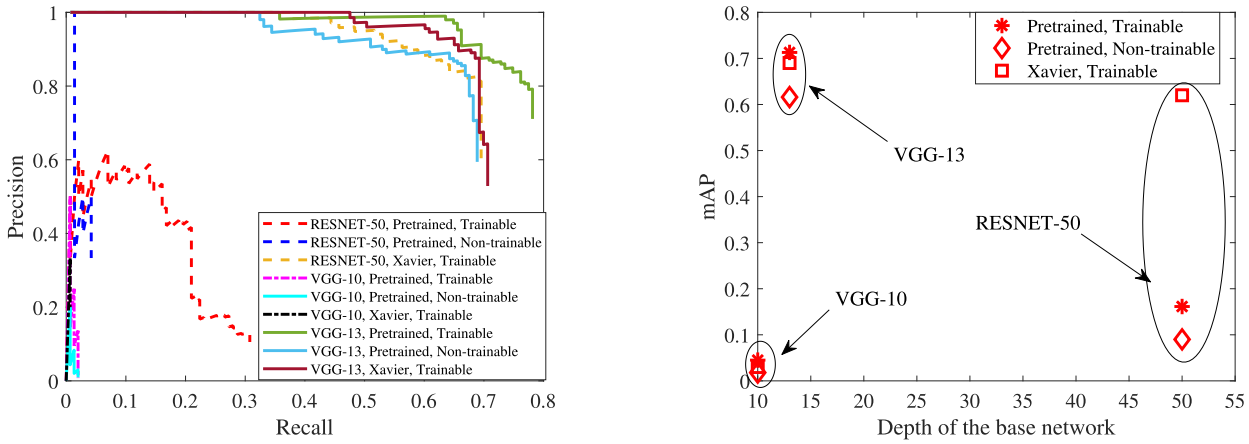


Fig. 8. Example predictions from the SVM-HOG method [22] with the optimized hyperparameters ($wt = 32, wf = 80, ori = 7, w_{\text{step}} = 8, \gamma_{\text{ny}} = 0.2, len_{\text{ny}} = 100$). We observe much fewer number of missed detections than with [21] due to the supervised nature of the method. However, the SVM-HOG method does not seem to perform very well in separating out the RFI, resulting in a significant number of false positives.



(a) Precision vs. Recall for different BNs

(b) mAP for different BNs

Fig. 9. Impact of depth of the base network. The legend indicates the BN name, whether the initialization is pretrained or Xavier normal [29], and whether the BN is trainable or not. We note that the best performance is achieved with the VGG-13 model initialized with pretrained weights and when configured as trainable.

(ii) $wt \in [24, 32, 40, 48, 56, 64, 72, 80, 160]$, corresponding to signals of time-span [0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2]ms respectively, (iii) $pix \in [2, 4, 6, 8, 16]$, (iv) the block size $b = 2 \times pix$ (each block is therefore a grid of 4 cells), the block stride $b_{\text{stride}} = 0.5 \times b$, the window step size $w_{\text{step}} = 8$, and the block normalization scheme to *L2-Hys*, all as done in [22], (v) the number of orientation bins $ori \in [3, 5, 7, 9, 11, 13]$, (vi) the Nystroem approximation kernel width $\gamma_{ny} \in [100, 10, 1, 0.2, 0.1]$ and (vii) the feature vector length $len_{ny} \in [200, 180, 150, 120, 100, 80]$.

D. Design Choices

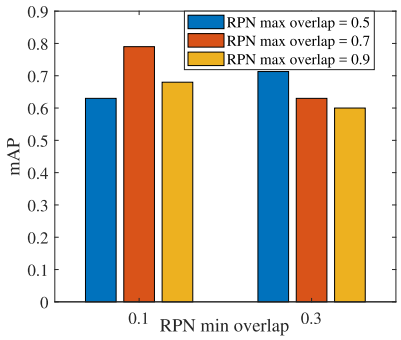
Before evaluating the performance of the proposed model, we need to make three important design choices: (i) downscale the default RPN and Detector networks by an appropriate factor, (ii) choose an appropriate BN for feature extraction, and (iii) choose an appropriate combination of the *(RPN min, RPN max, Detector min, Detector max)* overlap values.

1) Size of the RPN and Detector Networks: We downscale the size of the feature extraction layers in the RPN and the Detector from the default (256, 4096) filters to (128, 2048), (64, 1024), and (32, 512) respectively, i.e., by downscaling factors of 2, 4, and 8 respectively. The precision vs. recall performance is plotted in Fig. 5. We note that the best mAP is achieved when the default FRCNN is downscaled by a factor of 2. Downscaling is expected to provide performance improvement because the signal detection task at hand is simpler than the general-purpose object detection task for

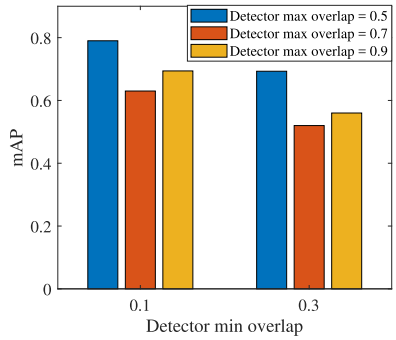
which the FRCNN was originally designed. We stick to the (128, 2048) combination for the remainder of the experiments. For comparison with the state-of-the-art, in Fig. 5, we also plot the precision vs. recall performance of the two baseline methods. We note that the downscaled FRCNN significantly outperforms the morphological processing and the SVM-HOG methods, which achieve an mAP of 0.3676 and 0.4835 respectively when their hyperparameters are optimized through exhaustive grid search. The hyperparameter combinations which delivered the best mAP for the two baselines were $(\gamma_{\text{bin}} = 0.8, n_{\text{erode}} = 2, s = 9, m = 4)$ and $(wf = 32, wt = 80, ori = 7, pix = 8, \gamma_{\text{ny}} = 0.2, len_{\text{ny}} = 100)$ respectively. Example predictions from the FRCNN and the two baseline methods are illustrated in Figs. 6, 7, and 8 respectively. The superior performance of the downscaled FRCNN may be attributed to the sophisticated feature extraction procedure and the superior learning ability of the deep neural networks when compared to the two baselines. The loss in mAP with the SVM-HOG method occurs mainly due to an excessive number of false positives that result from the model's limited ability in learning and separating the signals of interest from the RFI. The loss in mAP in the morphological processing method comes mainly from a large number of missed detections and false positives, both of which result from the limited generalizability of the hyperparameters involved.

2) Depth of the Base Network: In Fig. 9(a)-(b), we plot the precision vs recall curves and the mAP vs depth values respectively, when the BN is chosen to be VGG-10, VGG-13, and RESNET-50, where the numbers $10, 13$, and 50 denote the number of weight layers in the BN. We try three different combinations for feature extraction: (i) use pretrained weights as the initialization for the BN and configure it as trainable (Pretrained, Trainable), (ii) use pretrained weights as the initialization for the BN and configure it as non-trainable (Pretrained, Non-trainable), and (iii) use Xavier normal initializer [29] for the BN and configure it as trainable (Xavier, Trainable). The VGG-13, set as trainable and initialized with the pretrained weights, gives the best precision vs recall as well as mAP performance. When the depth is appropriate, as with VGG-13, there is only a marginal drop in the mAP if the BN weights are initialized using the Xavier normal method, instead of the pretrained weight initialization. In other words, we note that the VGG-13 provides an appropriate architecture for feature extraction, but the FRCNN model only benefits marginally when the starting weights are the pretrained weights. The VGG-10, which is a slightly shallower network than the VGG-13, achieved lower mAP because it has insufficient depth for feature extraction. Henceforth, we fix the BN to VGG-13, initialize it with pretrained weights and set it as trainable.

3) RPN and Detector Min and Max Overlaps: We need to choose an appropriate combination of the *RPN and Detector min and max overlap* values. Since a grid search can be computationally exhaustive, we pursue a simple alternate-once strategy for choosing these numerical thresholds. We first fix the *Detector Min and Max overlaps* to be 0.1 and 0.5 respectively and search over different combinations of *RPN min and max overlap* values as shown in Fig 10a. Among the different



(a) *Detector (min, mix)* overlap = (0.1, 0.5)



(b) *RPN (min, max)* overlap = (0.1, 0.7)

Fig. 10. mAP with different RPN and Detector thresholds. An mAP of upto 0.8 is seen to be achievable through a simple alternate-once search strategy.

thresholds, we observe that the (*RPN Min, RPN Max*) overlap combination (0.1, 0.7) achieves better mAP values than the rest. We therefore fix this combination and search over the different *Detector min and max overlap* values, as shown in Fig 10. We notice that the best (*Detector min, Detector max*) overlap combination is the (0.1,0.5), with an achievable mAP of 0.8. This simple search method exposes an important avenue to improve the mAP of the model. Note that the mAP achieved with the different combinations of the *RPN and Detector Min and Max overlaps* are consistently better than those achieved by the two baselines. Naturally, a more exhaustive grid search method may help us choose better threshold values but would require many more experimental runs.

E. Performance Evaluation on Real Traces and Varied SNRs

We now proceed to evaluate the performance of the downscaled FRCNN on real RF traces and also validate the robustness with regard to variations in the SNR. For all the results presented henceforth, we set the RPN feature extraction layer to 128 3×3 filters, the Detector feature extraction layer to 2048 1×1 filters, the BN to VGG-13 configured as trainable and initialized with pretrained weights, and the combination (0.3, 0.5, 0.1, 0.5) for the *(RPN min, RPN max, Detector min, Detector max)* overlap values.

1) Real RF Captures: The downscaled FRCNN model, when trained on synthetic datasets, will exhibit a drop in mAP performance when the test captures are from real measurements because of the inherent mismatch in the data type.

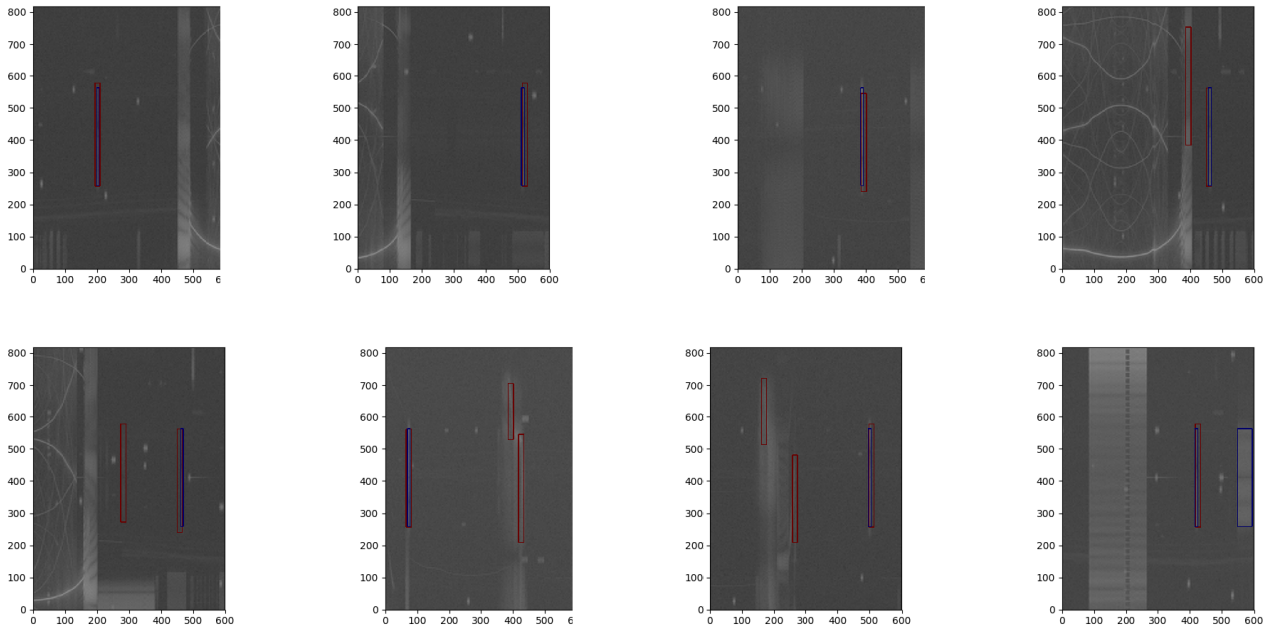


Fig. 11. Example predictions on real test captures when the synthetically-trained FRCNN model is finetuned with real RF traces. When compared to the case of synthetic test traces, the real test captures contain new types of background noise and RFI (compare the spectrogram backgrounds with Fig. 6)). The model is still able to successfully detect the signals of interest, but experiences a loss in mAP due to an increase in the number of false positives and a marginal increase in the missed detections.

To verify the same, we create a test dataset using the NI USRP-2901 with the Wi-Fi source located about 20m away and a microwave oven source about 2m away from the receiver. Multiple Bluetooth devices are present within a 5m radius around the receiver. The receiver gain is set to 20dB and a total of 50 test captures are recorded, each containing RFI from both the microwave oven and the Bluetooth signals. In Fig. 12, we plot the precision vs. recall performance when the model is trained on the synthetic dataset generated as per **Algorithm 1**. We notice that the mAP drops significantly from 0.713 for synthetic test data to 0.125 for real test data. The drop is mainly because the model is not exposed to the varied types of background noise and interference experienced in real measurements (c.f. the background in Fig. 11).

To alleviate the drop in mAP, we propose two solutions, namely *data mixing* and *model finetuning*. In *data mixing*, we train the downscaled FRCNN with a mixture of synthetic and real measurement data. We expect improvements in the mAP because the model benefits from the accurately labelled ground truths in the synthetic data and also from the exposure to the varied types of background noise and unknown interference which cannot be modelled accurately or have not been accounted for in the synthetic data. Synthetic traces would still constitute a majority of the training dataset because it is not scalable to manually label a large amount of real RF traces. In Fig. 12, we plot the precision vs recall when the model is trained with 120 synthetic and 30 real captures, i.e., 80% synthetic and 20% real captures. The test captures are fully from real measurements. We notice that the mAP improves from 0.125 to 0.484, corroborating our hypothesis that the model would perform better with *data mixing* because it is now exposed to the variety of background noise and unknown interference experienced in real RF traces.

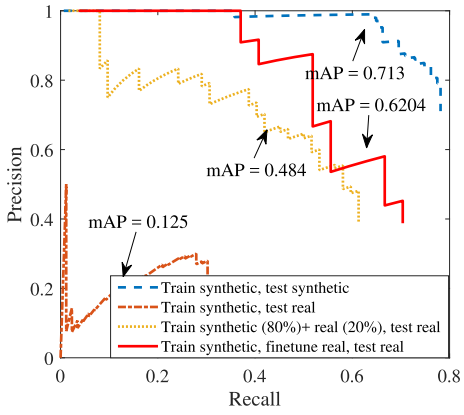


Fig. 12. Precision vs recall when the test captures are obtained through real measurements. We notice a significant loss in the mAP if the model is trained purely on synthetic data. *Data mixing* and *model finetuning* are seen to offer significant improvements in the mAP.

In *model finetuning*, we initially train the downscaled FRCNN with synthetic RF traces and then finetune the trained weights through an additional round of training on a small amount of real RF traces. Since the synthetic RF traces are similar to the real RF traces to a certain extent, the model from the initial synthetic data training serves as a warm start for the finetuning and therefore facilitates an improvement in the mAP performance. The additional computational workload from *model finetuning* is only marginal because the size of the real training dataset is much smaller than the synthetic training dataset. Also, a smaller number of training epochs is sufficient to achieve convergence. In Fig. 12, we plot the precision vs recall performance when *model finetuning* is performed with the same 30 real RF traces as used in the *data mixing*. For the finetuning, convergence was achieved with just 2 epochs.

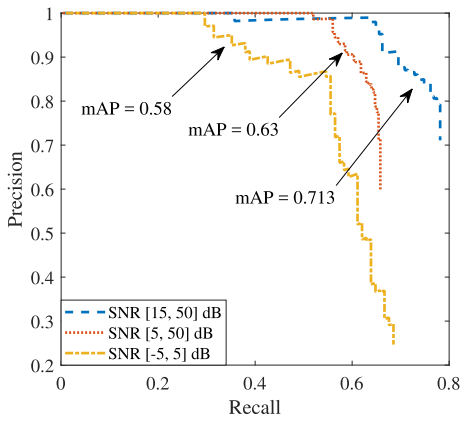


Fig. 13. Precision vs. recall curves for datasets with different SNR ranges. We notice a drop in mAP as we widen the SNR range and enter into negative SNR values.

We notice that the mAP improves from 0.125 to 0.6204. This indicates that the *model finetuning* is a viable solution for field deployments, whereby, we train a downscaled FRCNN model offline with large amounts of synthetic data and then finetune the weights over a small amount of real measurements to achieve a good mAP performance.

We also note that the two baselines perform much worse than the proposed deep learning method on the real RF traces. The morphological processing method [21] achieves an mAP of 0.1 when the combination $(\gamma_{\text{bin}} = 0.4, n_{\text{erode}} = 9, s = 6)$, obtained via grid search over the 30 real RF traces within the *data mixing* dataset, is employed. If we instead employ the combination $(\gamma_{\text{bin}} = 0.8, n_{\text{erode}} = 2, s = 9)$ optimized for synthetic traces, none of the signals in the real RF traces are detected. The SVM-HOG method, on the other hand, provides a non-zero mAP of 0.0114 when tested directly on real RF traces. For better mAP, we attempted *model finetuning* and only experienced a marginal increase from 0.0114 to 0.134. Since the two baselines perform significantly worse than the proposed deep learning method, we avoid clutter and do not include their precision vs. recall curves in Fig. 12.

2) Variations in the Signal-to-Noise Ratio: We now proceed to evaluate the mAP performance for different SNR levels in the dataset. In the synthetic dataset so far, the signals span the SNR range of [15, 50]dB. We now create two additional datasets, with the received SNR in the range [5, 50]dB and [−5, 5]dB respectively. The same training/test composition as given in Section V is maintained. In Fig. 13, we plot the precision vs recall achieved with the three datasets. We notice that the mAP drops marginally to 0.63 when the SNR range is expanded from [15, 50]dB to [5, 50]dB. The drop becomes more significant as we move to the [−5, 5]dB SNR range, with the mAP being 0.58. These trends expose the limited generalizeability of the model across wider SNR ranges and negative SNR regions. Experiments with wavelet denoising [32] as a simple pre-processing step reveal no significant improvement in the mAP. A customized and more advanced denoising mechanism is therefore required as a preprocessing step to improve the SNR before using the model.

VII. Conclusion and Future Work

In this work, we have proposed a downscaled Faster RCNN (FRCNN) [4] framework to perform signal detection and time-frequency localization in a wideband radio frequency (RF) spectrum under interference from uninteresting signals. Multiple design insights are provided on adapting the downscaled FRCNN for the task at hand, all based on the signals of interest. For synthetic traces, a mean average precision (mAP) of up to 0.8 is observed when the signals of interest are from Wi-Fi and the RF interference is from Bluetooth and microwave oven signals, significantly outperforming the state-of-the-art. The best performance is achieved when the feature extraction model is a pretrained VGG-13 [6] configured as trainable, the anchor sizes are the same as those of the signals of interest, the region proposal network (RPN) and Detector networks are downscaled from the default architecture by a factor of 2, and the numerical thresholds within the RPN and Detector networks are optimized through grid search. For real RF traces, we propose to (i) train with a mixture of synthetic and real RF traces, or (ii) finetune the synthetic-trained model through an additional round of training on a small amount of real RF traces. Both strategies offer significant improvements in the mAP when compared to directly using the synthetic-trained model. Interesting avenues for the future are to: (i) build a general-purpose non-max suppression that is robust to different anchor sizes, (ii) build a custom-made denoiser for improved performance under low SNR, and (iii) conduct a detailed study to determine the most suited time-frequency representation for the spectrum monitoring applications.

Acknowledgment

Preliminary work on the topic has been filed for a patent [1]. The authors would like to thank Skycope Technologies Inc., particularly Dr. H. Boostanimehr and Dr. S. Mallick, for identifying the time-frequency localization problem and its equivalent box detection in a spectrogram. The authors would also like to thank them for the inputs on spectrogram and Wi-Fi signal simulation, and for the feedback during preliminary investigations leading to this research.

References

[1] K. N. R. S. V. Prasad, K. B. D'Souza, H. Boostanimehr, and S. Mallick, "Wireless threat detection device, system and methods to detect signals in wideband RF systems and localize related time and frequency information based on deep learning," U.S. Patent 16/567 630, Sep. 11, 2019.

[2] K. N. R. S. V. Prasad, K. B. D'Souza, V. K. Bhargava, H. Boostanimehr, and S. Mallick, "A deep learning framework for blind time-frequency localization in wideband systems," accepted for presentation at the IEEE Veh. Tech. Conf. (VTC-Spring), Antwerp, Belgium, May 2020.

[3] K. N. R. S. V. Prasad and V. K. Bhargava, "A classification algorithm for blind UAV detection in wideband RF systems," submitted for publication, Mar. 2020.

[4] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Neural Inf. Process. Syst. (NIPS) Conf.*, Montreal, QC, Canada, Dec. 2015, pp. 91–99.

[5] *IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band*, IEEE Standard 802.11g-2003, Oct. 2003. [Online]. Available: https://standards.ieee.org/standard/802_11g-2003.html

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.
[7] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: A state of the art," in *Proc. Int. Conf. Digit. Audio Effects (DAFx)*, Paris, France, Sep. 2011, pp. 375–386.
[8] B. Boashash, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference* New York, NY, USA: Academic, 2015.
[9] L. Stankovic *et al.*, *Time-Frequency Signal Analysis With Applications*, Norwood, MA, USA: Artech House, 2013.
[10] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Wideband spectrum sensing in cognitive radio networks," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 901–906.
[11] M. Bkassiny, S. K. Jayaweera, Y. Li, and K. A. Avery, "Wideband spectrum sensing and non-parametric signal classification for autonomous self-learning cognitive radios," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2596–2605, Jul. 2012.
[12] S. Liu, Y. Chen, W. Trappe, and L. J. Greenstein, "ALDO: An anomaly detection framework for dynamic spectrum access networks," in *Proc. 28th Conf. Comput. Commun. (IEEE INFOCOM)*, Rio de Janiero, Brazil, Apr. 2009, pp. 675–683.
[13] D. Liu, C. Li, J. Liu, and K. Long, "A novel signal separation algorithm for wideband spectrum sensing in cognitive networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Miami, FL, USA, Dec. 2010, pp. 1–6.
[14] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Netw.*, vol. 43, pp. 72–83, Jul. 2013.
[15] P. Fryzlewicz, "Wild binary segmentation for multiple change-point detection," *Ann. Statist.*, vol. 42, no. 6, pp. 2243–2281, Dec. 2014.
[16] D. Garreau and S. Arlot, "Consistent change-point detection with kernels," *Electron. J. Statist.*, vol. 12, no. 2, pp. 4440–4486, 2018.
[17] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107299.
[18] O. Berder, C. Bouder, and G. Burel, "Identification of frequency hopping communications," in *Proc. Problems Modern Appl. Math. (WSES)*, 2000, pp. 259–264.
[19] S. Phonsri, S. S. Mukherjee, and M. Sellathurai, "Computer vision and bi-directional neural network for extraction of communications signal from noisy spectrogram," in *Proc. IEEE Conf. Antenna Meas. Appl. (CAMA)*, Chiang Mai, Thailand, Nov. 2015, pp. 1–4.
[20] X. Mankun, P. Xijian, L. Tianyun, and X. Mantian, "A new time-frequency spectrogram analysis of FH signals by image enhancement and mathematical morphology," in *Proc. 4th Int. Conf. Image Graph. (ICIG)*, Sichuan, China, Aug. 2007, pp. 610–615.
[21] H. Zhuang, B.-S. Oh, D. Lin, K.-A. Toh, and Z. Lin, "Multicomponent signal decomposition using morphological operations," in *Proc. IEEE 23rd Int. Conf. Digit. Signal Process. (DSP)*, Shanghai, China, Nov. 2018, pp. 1–5.
[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR05)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.
[23] C. K. I. Williams and M. Seeger, "Using the Nystroem method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2001, pp. 682–688.
[24] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 89–96.
[25] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[26] The Bluetooth SIG. *Bluetooth 5 Core Specifications*. Accessed: Jun. 2019. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/
[27] M. Nassar, X. E. Lin, and B. L. Evans, "Stochastic modeling of microwave oven interference in WLANs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, Jun. 2011, pp. 1–6.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[29] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artificial Intell. Stat.*, Sardinia, Italy, May 2010, pp. 249–256.
[30] IEEE Standards Association. (2012). *SISO ITU Extended Models, Channel Models for TG8*. Accessed: Jun. 2019. [Online]. Available: https://mentor.ieee.org/802.15/dcn/12/15-12-0459-07-0008-tg8-channel-models.doc
[31] National Instruments. (Jul. 2019). *Specifications: USRP-2901 Software-Defined Radio Device*. Accessed: Jun. 2019. [Online]. Available: http://www.ni.com/pdf/manuals/374925c.pdf
[32] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.

K. N. R. Surya Vara Prasad (Student Member, IEEE) received the B.Tech. degree in electrical engineering from the IIT Bhubaneswar, Bhubaneswar, India, in 2012, and the M.A.Sc. degree in electrical and computer engineering from The University of British Columbia (UBC), Vancouver, Canada, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, UBC. At the IEEE COMSNETS 2014, he was awarded the Best Demo and Exhibits award. His current research focus is on machine learning methods and their applications in wireless communications systems. He was an NSERC Alexander Graham Bell Scholar with the Department of Electrical and Computer Engineering, UBC.

Kevin Bradley D'souza (Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology Karnataka, India, the M.A.Sc. degree in electrical and computer engineering from The University of British Columbia (UBC), where he is currently pursuing the Ph.D. degree. At UBC, he worked on precoding schemes for 5th generation wireless systems and computational genomics jointly under the Information Theory Group. He worked on Computational Biology Group, The Simon Fraser University. His work focuses on building machine learning tools to aid in the understanding of structural and functional genomic data.

Vijay K. Bhargava (Life Fellow, IEEE) was born in Beawar, India, in 1948. He received the B.A.Sc., M.A.Sc., and Ph.D. degrees from Queen's University at Kingston, Canada, in 1970, 1972, and 1974, respectively.

He is currently a Professor with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, where he served as the Department Head, from 2003 to 2008. Previously, he was with the University of Victoria from 1984 to 2003, Concordia University, from 1976 to 1984, the University of Waterloo in 1976, and the Indian Institute of Science from 1974 to 1975. He has held visiting appointments at the Ecole Polytechnique de Montreal, NTT Research Lab, Tokyo Institute of Technology, the University of Indonesia, the Hong Kong University of Science and Technology, Tohoku University, and Friedrich Alexander University, Germany. He is in the Institute for Scientific Information (ISI) Highly Cited list. He has served as the Founder and President of Binary Communications Inc., from 1983 to 2000. He is a coauthor/co-editor of seven books the latest of which is *Wireless-Powered Communication Networks* (Cambridge University Press, 2016).

Dr. Bhargava is a fellow of the Royal Society of Canada, The Canadian Academy of Engineering, and the Engineering Institute of Canada. He is a Foreign Fellow of the National Academy of Engineering, India. He has received awards for his teaching, research, and service to the IEEE. The latest awards are the Killam Prize in Engineering awarded by the Canada Council for the Arts and the Humboldt Research Prize awarded by the Alexander von Humboldt Foundation of Germany. He has served as a Distinguished Visiting Fellow of the Royal Academy of Engineering, U.K. A long-time volunteer of the IEEE, he has served as the Director of Region 7 from 1992 to 1993, the Vice President of Regional Activities Board-RAB (now MGA) from1994 to 1995, the President of the Information Theory Society in 2000 and the the IEEE Communications Society from 2012 and 2013, and the Director for Division III in 2018. He has served as an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and the Editor-in-Chief of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.