A Deep Learning Framework for Blind Time-Frequency Localization in Wideband Systems

K. N. R. Surya Vara Prasad*†, *Student Member, IEEE*, Kevin B. Dsouza*†, *Student Member, IEEE*, Vijay K Bhargava†, *Life Fellow, IEEE*, Shankhanaad Mallick‡, Hamidreza Boostanimehr‡.

***Abstract*—In this paper, we propose a blind time-frequency localization method for wireless signals present in a wideband radio frequency (RF) spectrum. The signal detection problem is transformed into an object detection problem by converting the RF time-series captures into spectrogram images. A deep learning system based on the Faster RCNN [2] is then configured to suit the signal detection task. Guidelines are provided to make design choices in terms of both data pre-processing and the FRCNN modeling, for example, on the Short Time Fourier Transform (STFT) parameters, the spectrogram sizes, and the anchor sizes. Experiments with artificially generated WiFi high throughput data [3] reveal that (i) the proposed framework can achieve up to a mean average precision (mAP) of 0.9 for captures with positive signal-to-noise ratio (SNR), (ii) the proposed framework is fairly robust to the number and size of the anchors, and (iii) the proposed framework is sensitive to the disparity in the signal sizes, giving us few insights into possible future extensions.**

I. Introduction

As we progress into the internet of things (IoT) era, there is a growing need to identify and separate out unexpected wireless devices, for example in the license-free bands, for security and spectrum management reasons. From the wireless security point of view, products can be developed to make ad-hoc security decisions such as, sending emergency alerts about transmissions from an unexpected device, employing techniques to jam signals from the detected transceiver, and also estimating the geographic location of the detected device from the incumbent signals. From the spectrum management point of view, commercial products can be built to dynamically share spectrum among the vast number and variety of heterogeneous devices in the IoT space. Knowing apriori which time-frequency resources are under-utilized and which ones have minimum interference, smart spectrum allocations can be made in densely populated scenarios. With the above applications in mind, we propose a deep learning architecture to detect any wireless transceiver in the wideband radio frequency (RF) spectrum of interest and also estimate the time and frequency span of each transmission. These two major tasks are referred to from here on as the *blind time-frequency localization* problem.

Blind time-frequency localization techniques have been investigated extensively in the literature. A multi-band joint detection technique which jointly detects signal energy levels in multiple frequency bands is introduced in [4]. The spectrum sensing problem is formulated as an optimization problem in an interference limited network. Wavelet edge detection is employed to detect the signal spectrum edge in [5]. Following this, blind source separation is done to separate the signals in the frequency domain. In both these works, although the signals can be accurately localized and separated in frequency, the joint time-frequency information is lost. In many applications such as detection of the hopping pattern of a wireless device and joint frequency and temporal optimization of the shared spectrum, it becomes a necessity to detect both the time and frequency information of the signals present. In [6], a blind energy detection method is proposed to perform channel sensing. Cyclostationarity features are then used to detect periodic signals. In all of [4]–[6], frequency span information becomes automatically available but significant post-processing is required to obtain time span information.

Recent deep learning architectures for image analysis [15], [16] allow us to extract rich features out of RF data for downstream tasks such as detection, localization and classification. Audio Event Detection is one example where the application of deep learning has been explored in the recent past. The underlying philosophy is that by converting the time series data into spectrograms and then employing deep learning techniques, we can extract certain specific patterns that help detect and localize audio events. In [7], a state-of-the-art object detection framework was adapted to detect monophonic and polyphonic audio events from the spectrograms. A similar approach is proposed in [8], with the added functionality of capturing the long-term temporal context from the extracted features through the use of a convolutional recurrent neural network (CRNN). Both [7] and [8] detect the presence of audio events by converting time series information into time-frequency spectrograms and then learning from the features present in the spectrograms. However, as

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and, in part, by the MITACS Accelerate Program with Skycope Technologies Inc. A part of this work has been filed for provisional patent at the US Patent and Trademark Office [1]. *K. N. R. S. V. Prasad and K. B. Dsouza contributed equally to this work. †K. N. R. S. V. Prasad, K. B. Dsouza, and V. K. Bhargava are with the Department of Electrical and Computer Engineering at the University of British Columbia, Canada (emails: {surya, kevin, vijayb}@ece.ubc.ca). ‡ S. Mallick and H. Boostanimehr are with Skycope Technologies Inc., British Columbia, Canada (emails: {shankha, hamidb}@skycope.com).

978-1-7281-5207-3/20/$31.00 ©2020 IEEE

explained below, the same philosophy has not been well explored yet for detecting general purpose wireless RF signals present in wideband spectrum. The idea of using Deep Learning based frameworks to detect wireless signals has been looked into recently. The work [9] converts the time-frequency information into power spectral density (PSD) based spectrograms. The spectrogram is then fed into a five-layer CNN which is used to perform multi-class classification over different wireless technologies like WiFi, Bluetooth and ZigBee. Although the approach is able to perform classification over heterogeneous devices, it cannot localize them in time and frequency. Localization in time and frequency, if achieved, can be used to study various other properties of these devices like hopping patterns, signal bandwidth and dwell time. This information will be crucial for security purposes because it helps us perform narrow-band jamming to mitigate rogue devices without affecting the other friendly devices on the Industrial Scientific and Medical (ISM) band. A different time-frequency transformation called the Choi-Williams Distribution (CWD) is used in [10] which uses it to distinguish between different type of coding schemes like polytime codes, Frank code and Costas codes. After image preprocessing, this transformation is fed into a two layer CNN with pooling and the recorded Ratio of Successful Recognition (RSR) is about 90% for most codes. However, it faces a similar drawback of not being able to localize the signal in both time and frequency.

The problem of blind time-frequency localization in a wideband RF spectrum translates into the problem of detecting rectangular boxes in a spectrogram (c.f. Fig. 1). The state of the art in computer vision for general-purpose object detection is to employ convolutional neural networks (CNNs) to identify whether an image contains an object and predict the associated bounding boxes [11]–[13]. Previous state of the art methods, for example [11], have employed one-stage object detection using a single CNN to simultaneously obtain the category and location of the objects. Such one-stage methods have recently been outperformed by certain two-stage object detection methods [12], [13]. In these methods, the first stage generates a set of candidate bounding boxes, commonly referred to as the region proposals. The most popular region proposal methods include selective search [12], which is based on greedy superpixel merging, and EdgeBoxes [13], which is based on edge maps and edge groups. The second stage performs a classification task on the region proposals to identify the objects and a refining task on the dominant region proposals to provide the bounding boxes. A major bottleneck with the all the above CNN methods is that they perform supervised machine learning and therefore require large amounts of labelled datasets to achieve high accuracy in object detection and localization [11]–[13]. Large labelled datasets are currently available for object detection in day-to-day real-life images containing humans (c.f. MS COCO [14]). However, there are no standard labelled datasets available online for wireless signals present in the wideband radio frequency (RF) spectrum. Therefore, limited work exists on the use of supervised learning for blind time-frequency localization. We workaround the dataset issue by creating a synthetic one based on the recently proposed IEEE 802.11n high throughput (WiFi-HT) [3] protocol.

In this paper we introduce a deep learning framework based on the Faster RCNN [2], for blind time-frequency localization of narrowband signals present in a wideband RF spectrum. The main contributions of our work are:

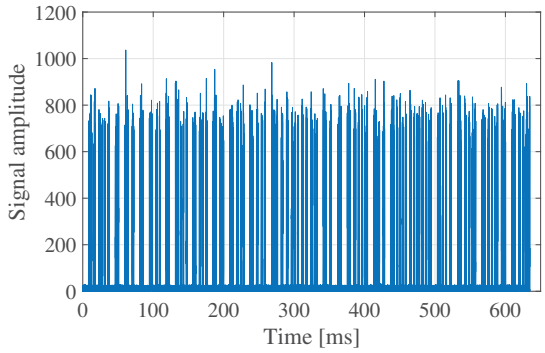
- An appropriate feature extraction model needs to be chosen to extract useful features for the blind time-frequency localization. Our experiments suggest that, while pretrained weights [14] are a good starting point for medium sized networks such as VGG-13 [15], making the weights trainable gives much better performance. It is seen that very deep feature extraction networks such as ResNet-50 [16] don't necessarily improve performance.
- We provide design insights with respect to multiple variables within the FRCNN architecture, namely the Short-Time Fourier Transform (STFT) parameters, the spectrogram resolution, anchor sizes, and various numerical thresholds in the model. We note that tuning these variables is essential to achieve satisfactory performance.
- For evaluation purposes, we generate synthetic data as per the recently proposed WiFi high throughput (WiFi-HT) protocol [3]. An mAP of up to 0.9 is recorded when the model is trained and tested on positive SNR values with single-bandwidth signals. This is seen to deteriorate as we train the model with disparate signal sizes. Few insights are provided on why this might be happening. Experiments also suggest that the model is fairly agnostic to the number of anchor boxes because of the robust regression.

The remainder of this paper is organized as follows. In section II, we present the proposed framework for signal detection and time-frequency localization. In Section III, we present details on the multiple design choices we make in adopting the FRCNN architecture. Numerical studies based on artificial WiFi-HT [3] data are presented in Section IV, followed by concluding remarks in Section V.

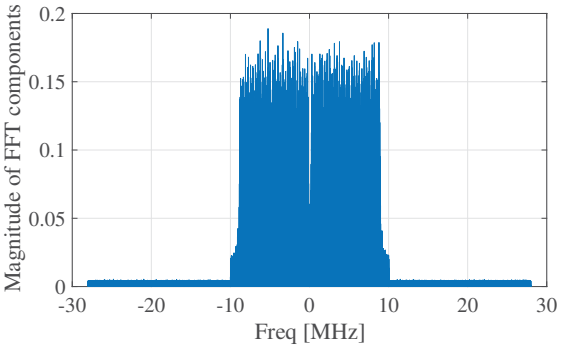
II. Blind time-frequency localization

To achieve the blind time-frequency localization objective, we propose a framework comprising three major stages, referred to as pre-processing, object detection, and post-processing, respectively. Details are presented below.

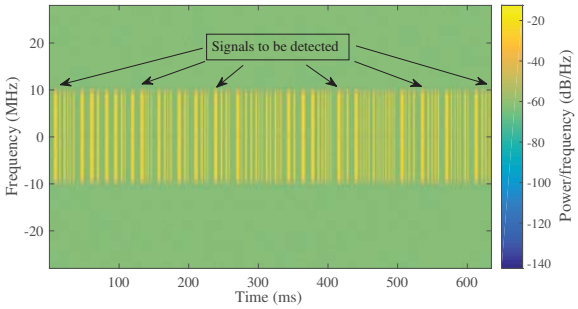
1) Time-series data pre-processing: Let us assume that a wideband sensor is employed to capture time-series RF data with center frequency f_c, bandwidth W, and duration T each. For the purpose of illustration, Fig. 1 presents a wideband capture of 56MHz taken with a sampling rate of 56 MHz for a duration of 633ms at a center frequency of 2.4GHz. While Fig. 1a plots the signal amplitude as a function of time, Fig. 1b plots the magnitude of the fast Fourier transform (FFT) components as a function



(a) Signal amplitude vs. time



(b) Magnitude of FFT components vs frequency



(c) Spectrogram image created by plotting the p.s.d values against time and frequency

Fig. 1: A simulated wideband RF capture, when the duration $T = 633$ms, center frequency is $f_c = 2.4$GHz, wideband bandwidth $W = 56$MHz, and sampling rate $f_s = 56$MHz.

of frequency. For a joint time-frequency representation of the wideband captures, we apply short-time Fourier transform (STFT) on the time-series captures and plot the power spectral density (p.s.d.) as a function of time and frequency to obtain spectrogram images. Fig. 1c illustrates the spectrogram created for the RF capture in Fig. 1a-1b, when the STFT parameters are chosen as follows: number of frequency bins is 4096, number of time bins is 4096, the STFT window is of Hann-type, and the window overlap is of 2048 time bins. Few insights on the choice of STFT parameters are given in Section III-A. As may be noted from Fig. 1c, the signals to be detected appear in the form of rectangular boxes in the spectrogram.

2) Bounding box detection: From the spectogram in Fig. 1c, we may note that the dimensions of each rectangular box within the spectrogram give us the time and frequency information of the corresponding wireless signal. The problem of signal detection and time-frequency localization therefore boils down to the problem of detecting and estimating the dimensions of rectangular-shaped boxes present in the spectrogram. Before attempting box-detection in the spectrogram image, we may employ some pre-processing steps. For example, we may remove out-of-band transmissions to eliminate unreliable information. We may also employ denoising methods, such as wavelet denoising, to improve the signal-to-noise ratio (SNR) of the spectrogram. To detect the rectangular-shaped boxes present in the spectrograms, we train a Faster RCNN (FRCNN) model [2] with a labelled dataset.

3) Post-processing the bounding boxes: The final stage in the proposed framework converts the dimensions of each bounding box reported by the FRCNN model into time and frequency information. For example, using the STFT parameters employed in the spectrogram creation stage, we may scale the x and y dimensions of each box into the time and frequency span of the corresponding signal. The same approach may also be followed to obtain the narrowband center frequency of the signal. In the next section, we present insights on several design choices that need to be made within the Faster RCNN model to perform the blind time-frequency localization task.

III. Design Choices to Adopt FRCNN for Blind Time-Frequency Localization

The Faster RCNN is an object detection framework that contains a base network (BN), a region proposal network (RPN), and a detection network (DN). The BN performs feature extraction on the raw spectrograms and yields a feature image. The RPN generates region proposals, which are nothing but candidate boxes that are likely to contain the objects of interest. In our case, the objects of interest are the rectangular boxes in the spectrogram. The region proposals and the feature image are the inputs to the DN. The DN assigns object class labels to each region proposal from the RPN, performs a regression task to localize the object within the region proposal, and also provides probabilities with which the assigned labels are true. Essentially the RPN acts as an attention mechanism over the feature image to help the DN with the object detection and localization task. The entire FRCNN model can be thought of as a single unified framework for detecting and localizing the rectangular boxes (which are nothing but a manifestation of the signals of interest) present in the spectrograms. For details on each module in the FRCNN, please see [2].

While we directly use the FRCNN model for blind time-frequency localization, we need to make several design choices in order to make it suitable for the task. Below we provide insights on the major design choices to make.

A. Choice of STFT parameters

To generate the spectrogram images from the raw RF time-series data, we need to apply discrete-time STFT and this in turn requires us to choose a few hyperparameters, namely, the window size, the window type, the window overlap, and the FFT size. We may choose these STFT parameters based on the minimum time and frequency resolutions that we need to achieve. The window size governs the maximum achievable frequency resolution. If the window is T seconds long, the minimum detectable narrowband bandwidth is $1/T$ Hz. For example, when the sampling rate is $f_s = 56 \times 10^6$Hz and the window size is 5600, the minimum detectable bandwidth is 10 kHz. While respecting this lower limit, the FFT size allows us to control the number of frequency bins in the spectrogram. For example, an FFT size of 1500 divides the spectrogram into 1500 frequency bins. Therefore, if the wideband bandwidth is 56MHz, the minimum detectable signal bandwidth would be $\approx$ 37.3kHz. The window size, the window overlap, and the sampling rate govern the maximum achievable time resolution. For example, if the capture signal duration is $T_{sig} = 0.633$ seconds, the sampling rate is $f_s = 56{\times}10^6$Hz, the window size is $N_{win} = 5600$ and the window overlap is $N_{ov} = 2800$, the maximum achievable time resolution is $(N_{win}-N_{ov})/f_s = 50\mu s$. Lastly, the window type governs the amount of discontinuities between successive window segments.

B. Choice of base network

The base network is a CNN that can be either shallow or deep depending on the complexity of features that need to be extracted from the input image. The convolutional layers are interleaved with max pooling layers and the combination of these layers decide the total down-scaling factor. Standard feature extraction models such as VGG-13 [15] and ResNet-50 [16], have been built to detect objects in benchmark datasets for computer vision, such as MC COCO [14]. Example objects that are to be detected from these datasets include humans, animals, and automobiles. Whether these standard models can perform feature extraction for the blind time-frequency localization task, is unclear. We can conduct a few experiments in this context. Firstly, we can consider the publicly available pretrained weights in the VGG-13 and ResNet-50 models and verify if the extracted features are useful for the signal detection task. Secondly, we can set the pretrained weights as initialization points to optimize the weights in the VGG-13 and ResNet-50 models to perform tailor-made feature extraction for the task at hand. Thirdly, we can consider the architecture of the VGG-13 and ResNet-50 models, randomly initialize the weights and optimize the weights for the feature extraction. We pursue these three experiments in Section IV.

C. Anchor box sizes and aspect ratios

For each pixel of the feature image output from the BN, the RPN generates a predefined number N_a of raw region proposals centered at the pixel, where N_a is the number of anchors. Anchors are user-defined raw region proposals whose size and aspect ratio needs to be specified before the training process begins. Anchor boxes aid the RPN in generating region proposals for the DN. When training the RPN, anchor boxes whose intersection over union (IoU) with the GT are larger than a certain numerical threshold are treated as positive targets and the ones having IoU lower than a certain threshold are treated as negative targets. Consequently, in order for the training to be successful, we need to carefully choose the anchor sizes and aspect ratios.

Since the anchor boxes serve as raw region proposals for the RPN to act upon, we may choose the anchor sizes and aspect ratios to be of similar time-frequency span as the signals of interest. For example, if we are interested in the IEEE 802.11n high throughput (HT) signals, they are bound to span a bandwidth of either 20MHz or 40 MHz. The time span is between 300 microseconds to about 15 milliseconds [3]. Consequently, the anchor boxes may be defined such that the frequency span is 20 or 40 MHz, and the duration is within $[0.3, 15]$ ms.

IV. Numerical Examples

A. Dataset for training and testing

We consider the IEEE 802.11n high-throughput mode protocol [3], popularly known as the WiFi-HT protocol, as the signal of interest and utilize MATLAB WLAN toolbox to simulate artificial wideband time-series captures. All the captures are in the 2.4GHz license-free ISM band with a duration of 630ms, a wideband bandwidth of 56 MHz, and an SNR drawn from the set $\{0, 10, 20, 30\}$ dB. The data sampling rate is 56MHz. The captures contain approximately 90 WiFi-HT signal packets each and all these signals have a bandwidth of 20MHz. All the signals are subject to small-scale fading effects, which are generated via tap-delay line implementation of randomly chosen one among the SISO Extended ITU outdoor to indoor and pedestrian and the SISO Extended typical urban model settings [17]. In total, we generate a dataset of 7 captures per SNR, amounting to around 2520 signals. Out of the 7 captures per SNR, we randomly choose 5 for training and the rest for test.

B. Spectrogram generation

For each RF capture, we apply STFT with the following choice of parameters: window size is 5600, window overlap is 2800, window type is Hann, and the FFT size is 1500. The resulting spectrogram, after removal of out-of-band transmissions, contains 1200 frequency bins and 12599 time bins. Each spectrogram image is chopped into fixed chunks of 600 bins in time so as to reduce the overall training time, as bigger spectrograms take longer time to train and also to maximize the performance of FRCNN. This allows us to detect signals that span a minimum of 0.05ms in time and 37.3kHz in frequency. Each spectrogram image therefore results in a total of 21 input images.

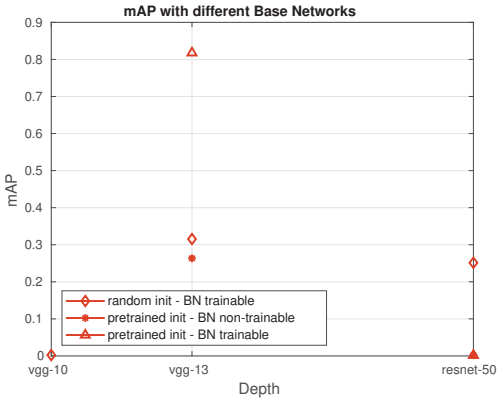


Fig. 2: mAP for different base networks

Consequently, the FRCNN model encounters a total of 420 input images during training and a total of 168 input images during testing.

C. Numerical thresholds for the FRCNN model

The convolutional layer at the start of the RPN, which is used to create a low-dimensional feature vector from the feature map generated by the BN, is chosen as per [2] to be of size 256 3×3 filters. The anchors are defined to be $\{20, 40, 80, 120\}$ time pixels and 17.92 frequency pixels, corresponding to time durations of $\{1, 2, 4, 6\}$ms respectively and frequency spans of 90% useful bandwidths for the 20 MHz narrowband transmissions. In total, we therefore have a maximum of $N_a = 4$ anchor boxes. The remaining numerical thresholds are chosen as per [2]. We train with 1 spectrogram chunk at a time. To evaluate the detection and localization performance of the proposed system, we adopt the mean average precision (mAP) metric, which is the most widely used metric for object detection in computer vision literature [2], [15], [16]. The mAP is calculated for each SNR level and the average over all the considered SNR levels is reported.

1) Impact of different base networks: In Fig. 2, we plot the mAP values achieved by the FRCNN model when the base network is chosen to be VGG-10, VGG-13, and ResNet-50 respectively. The values 10, 13, and 50 represent the count of convolutional layers. The VGG-13 and Resnet-50 feature extraction models are available in [15] [16], whereas the VGG-10 model is a subset of the VGG-13 with the three convolutional layers at the end discarded. We consider the following variants: (i) initialize and fix the BN with pretrained weights, (ii) initialize the BN with pretrained weights and make the layers optimizable, and (iii) initialize the BN with random weights and make the layers optimizable. With the VGG-10, we only attempt the third method because there are no pretrained weights available for this architecture. We notice that the VGG-13, with the BN set as trainable and the pretrained weights used as the initialization, performs the best in terms of the mAP. The mAP is slightly lower when the initialization is random and even more when the BN is set as non-trainable. Also, we notice that ResNet-50 is too deep for our task and therefore is an overkill, resulting in worse performance than the VGG-13. The VGG-10 achieved poor mAP because it fails to extract the necessary features for our task. We therefore recommend VGG-13 as the BN, initialized with pretrained weights and set as optimizable.

TABLE I: mAP with different number of anchors

Anchors	Anchor dimensions	mAP
1	[1ms, 20MHz]	0.7913
2	[1ms, 20MHz], [2ms, 20 MHz]	0.7944
3	[1ms, 20MHz], [2ms, 20 MHz], [3ms, 20MHz]	0.8176
4	[1ms, 20MHz], [2ms, 20 MHz], [3ms, 20MHz], [4ms, 20 MHz]	0.8172

2) Impact of the number of anchor boxes: We now analyze the effect of using different number of anchors on the mAP of the FRCNN model. In Table I, we list out the mAP values achieved with $N_a = 1, 2, 3, 4$. It is observed that the mAP values are fairly constant across different N_a values, with only marginal improvements when N_a is increased from 1 to 4. This reveals that the regression tasks in the RPN and the Detector network are powerful enough to regress from arbitrarily close anchors to the ground truths. Also, the chosen anchors need not have very high IoU with the ground truths for the training to be successful. In the ensuing experiments, we fix the number of anchors to 3, with the dimensions as per Table I.

3) Impact of SNR: We now study the performance when the SNR of the data is varied. We begin with a training dataset comprising 5 captures per SNR level in the set $\{0, 10, 20, 30\}$ dB. In Fig. 3, we plot the mAP values achieved when the test dataset comprises of captures with SNR $= -10, 0, 10, 20,$ and 30 dB respectively. We notice that the mAP values are consistently around 0.9 for positive SNR levels while dropping to a little over 0.5 for -10 dB. To verify whether this trend is universal, we next consider a dataset comprising 5 SNR levels, namely, $\{-10, 0, 10, 20, 30\}$ dB, i.e., we have now added one negative SNR level to the dataset considered so far. The mAP values for each SNR level in the test dataset are given in Fig. 3. When compared to the case with non-negative SNR levels, we notice a drop in the mAP performance for all the SNR levels except for -10 dB. This is expected as the model should perform well on the data it has seen before, however, the drop in mAP values corresponding to positive SNR when trained with images of negative SNR points to the lack of generalization of the model across positive and negative SNR. It is also observed that the mAP on test captures with SNR less than -10 dB is very poor irrespective of the training strategy. This exposes the need for a denoising mechanism as a preprocessing step on the spectrograms before we feed them into the FRCNN.

4) Impact of disparity in signal sizes: We now consider an artificially generated WiFi-HT dataset comprising two

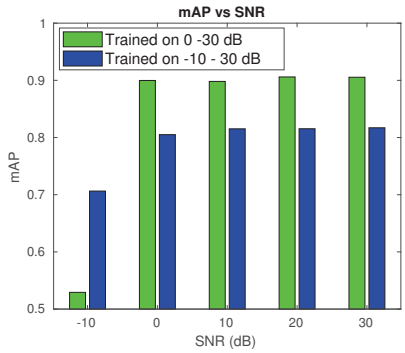


Fig. 3: mAP vs SNR

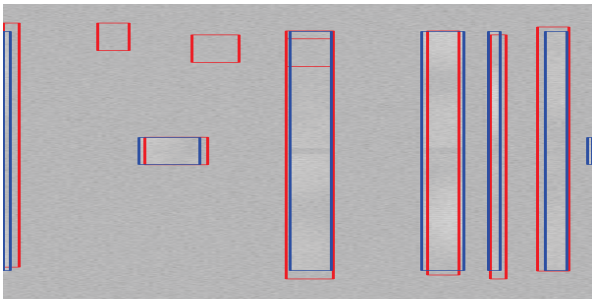


Fig. 4: Test image of model trained on disparate anchor sizes. The x and y axes denote time and frequency respectively. Blue and red boxes are the ground truths and predictions respectively.

different signal bandwidths, namely, the 5 MHz and 40 MHz, and SNR levels in the set $\{-10, 0, 10, 20, 30\}$dB. The 5MHz and 40MHz signals span bandwidths that are disparate by a multiplication factor of 8. We conduct this experiment to verify whether the trained FRCNN model is indifferent to the disparity in the ground truth sizes. A total of 6 anchors are considered, with time durations taken from the set $\{1, 2, 3\}$ms and the narrowband bandwidths taken from the set $\{5, 40\}$MHz. The mAP values experience a considerable fall from 0.79 to 0.53. Example predictions are illustrated in Fig 4. The performance loss may be attributed to one or more of the following reasons: (i) the BN may be struggling to extract all the necessary features for such disparate signals of interest, i.e., the BN fails to differentiate between background and signals whose size is much smaller or larger than it's receptive field, (ii) the number of region proposals are not balanced for each signal size, thus favoring one over the other. Further experiments need to be conducted to identify the root cause and to develop methods to overcome the drop.

V. Conclusion and Future Work

We have proposed a deep learning framework to perform blind time-frequency localization, for commercialization into wireless security and spectrum management products. We apply short-time Fourier transform on the raw time-series radio frequency data and convert them into spectrogram images, so as to transform the blind time-frequency localization into an object detection problem. For positive signal-to-noise ratio (SNR) images, we notice that the model achieves a mean average precision (mAP) of up to 0.9, but the performance drops for negative SNR images and when multiple signal sizes co-exist within the same spectrogram. Avenues worth exploring in the future would be (i) we note that the trained FRCNN model is able to generalize over small intervals in the positive SNR regime, but not over the entire SNR range of interest; custom-made models corresponding to different SNR ranges can be introduced and their performance improvement over general-purpose detectors can be analyzed and reported, and (ii) FRCNN is capable of detecting signals of one modulation type, i.e., OFDM in the case of Wi-Fi-HT; a natural extension is to classify multiple modulation types.

References

[1] K. N. R. S. V. Prasad, K. B. D'Souza, H. Boostanimehr, and S. Mallick, "A 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," patent filed at the US PTO, appl. id. 62800401, Feb. 2019.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. of NIPS, 2015.

[3] E. Perahia, and R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac.* 2nd Ed., United Kingdom, Cambridge University Press, 2013.

[4] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Wideband spectrum sensing in cognitive radio networks," in *Proc. of IEEE ICC*, May 2008, pp. 901-906.

[5] D. Liu, C. Li, J. Liu, and K. Long, "A novel signal separation algorithm for wideband spectrum sensing in cognitive networks," in *Proc. IEEE Globecom*, Dec. 2010, pp. 1-6.

[6] M. Bkassiny, *et. al.*, "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.

[7] P. Pham, J. Li, J. Szurley, and S. Das, "Eventness: Object Detection on Spectrograms for Temporal Localization of Audio Events," in *Proc. of IEEE ICASSP*, Apr. 2018, pp. 2491-2495.

[8] C. Kao, *et. al.*, "R-CRNN: Region based convolutional recurrent neural network for audio event detection," in *Proc. of Interspeech*, Sept. 2018, pp. 1-6.

[9] N. Bitar, S. Muhammad, H. H. Refai, "Wireless Technology Identification Using Deep Convolutional Neural Networks," in *Proc. of IEEE PIMRC*, 2017.

[10] M. Zhang, M. Diao, and L. Guo, "Convolutional neural networks for automatic cognitive radio waveform recognition," *IEEE Access*, vol. 5, pp. 11074-11082, 2017.

[11] P. Sermanet, *et al.*, "Overfeat: integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, Feb. 2014.

[12] J. R. Uijlings, *et. al.*, "Selective search for object recognition," *Int. J. Computer Vision (IJCV)*, vol. 104, no. 2, pp 154-171, Apr. 2013.

[13] C. L. Zitnick and P. Dollar,"Edge boxes: Locating object proposals from edges," in *Proc. of IEEE ECCV*, Sept. 2014, pp. 391-405.

[14] T.-Y. Lin, *et. al.*, "Microsoft COCO: common objects in context," in *Proc. of IEEE ECCV*, 2014.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in International Conference on Learning Representations (ICLR), 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv:1512.03385, 2015.

[17] SISO ITU Extended Models, Channel Models for TG8 - IEEE Standards Association, [Online] *https://mentor.ieee.org/802.15/dcn/12/15-12-0459-07-0008-tg8-channel-models.doc*, 2012.