Spectrum Painting for On-Device Signal Classification

Li, Bingyang; Huang, Weiqing; Wang, Wen; Wang, Qing

DOI
10.1109/WoWMoM60985.2024.00046

Publication date
2024

Document Version
Final published version

Published in
Proceedings - 2024 IEEE 25th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2024

2024 IEEE 25th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM) | 979-8-3503-9466-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/WoWMoM60985.2024.00046

Spectrum Painting for On-Device Signal Classification

Bingyang Li*†, Weiqing Huang*, Wen Wang*§, Qing Wang‡
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
‡Delft University of Technology, Delft, The Netherlands
Email: {libingyang, huangweiqing, wwen}@iie.ac.cn, qing.wang@tudelft.nl

***Abstract*—Achieving accurate and low-latency spectrum sensing on resource-constrained devices is essential but very difficult. Traditional In-phase and Quadrature (I/Q)-based and the Short-Time Fourier Transform (STFT)-based methods fail to balance the computational overhead and classification accuracy. In this paper, we propose a novel framework –*Spectrum Painting (SP)*– which enables on-device signal classification with low latency and high accuracy. We design new signal processing methods to compress spectrograms while keeping global signal features and augmenting the salient features of small objects. SP achieves high-accuracy signal classification, assisted further by our proposed Dual-channel Convolutional Neural Network (DualCNN). We collect diverse datasets to evaluate the proposed SP, including synthesized data, and testbed data (from up to 18 commodity devices) obtained from real-world environments in the wild and office settings. Experimental results of SP running on Raspberry Pi 4B show a great reduction in latency up to 20× while maintaining a 95% accuracy. Furthermore, SP demonstrates superior performance within both the centralized learning architecture and the Federated Learning (FL) architecture. For example, the challenging cross-environment evaluation of the SP in the iid-FL scenario yields a substantial accuracy improvement, on average from 24.6% to 83.8%.**

I. Introduction

The emerging wireless communication technologies operating in the Industrial, Scientific and Medical (ISM) frequency band, such as WiFi and Bluetooth, have significantly advanced our society. These coexisted technologies operate independently without coordination between them, leading to potential interference. To achieve reliable communications, devices can *sense the spectrum and identify the transmission signals* [1–11], preferably without decoding them to reduce the overhead on spectrum sensing. Classifying ongoing transmission signals can also help monitor the use of spectrum. Given the shared nature of spectrum resources in the ISM band among multiple users, distributed spectrum sensing systems allow adaptive and reliable transmission based on global and local information.

Deep Learning (DL) has become a widely adopted approach for accurate signal classification in frequency bands such as 900 MHz [1–3], 2.4 GHz [4–6], 3.5 GHz [7–9], and 5 GHz [10, 11]. Despite its effectiveness, *implementing DL-based methods on resource-constrained devices (e.g., Raspberry Pi) to achieve high-accuracy and low-latency signal classification in the real world, is difficult and not yet solved.*

State-of-the-art DL-empowered signal classification methods mostly use In-phase and Quadrature (I/Q) samples [3, 6] and Short-Time Fourier Transform (STFT) [2, 6, 12]. However, none of them can achieve high-accuracy and low-latency on-device signal classification. I/Q-based methods, which exploit mainly the time-domain information, enable rapid signal classification but yield lower accuracy, because signals in the shared spectrum exhibit significant similarity in the time domain [10]. For example, we find from experiments that I/Q-fed DL methods only take 0.9 ms on Raspberry Pi for the on-device inference, but the achieved accuracy is less than 40% *(cf. Observation 1 in Section II for the details)*. On the other hand, STFT-based approaches transform I/Q samples into *spectrograms*, providing a more accurate representation of the time-frequency characteristics of signals. However, they come with the drawback of the significantly increased computational time due to the large size of input spectrograms, they can improve the accuracy to be higher than 90%, but with a 100× longer latency on inference *(cf. Observation 2 in Section II)*.

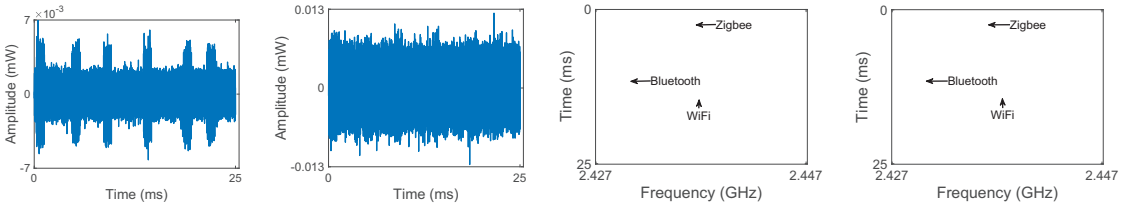
In this paper, we design a novel framework that integrates STFT and DL on embedded devices to classify the coexisting WiFi, Bluetooth, and Zigbee signals in the shared ISM bands. The fundamental idea is to *stretch and brighten small objects* (representing Bluetooth/Zigbee) in *compressed spectrograms*, followed by a classification model using a novel Dual-channel Convolutional Neural Network. Though promising, translating this concept into a functional system faces several challenges that must be addressed.

The first challenge is to reduce the spectrogram size while preserving essential signal characteristics. The commonly utilized method is downsampling. However, it suffers from losing valuable signal information. This information loss can be more severe for the Bluetooth and Zigbee signals, which have smaller bandwidths and occupy fewer bins in the spectrogram. Therefore, simply using downsampling can significantly decrease the accuracy of convolutional neural networks. Thus, we need to balance the trade-off between spectrogram compression and signal classification accuracy.

To address the first challenge, we first downsample the raw spectrograms. Downsampling will worsen the characteristics of small objects (Bluetooth/Zigbee) and increase classification error. Common solutions involve augmenting small objects with image operations [13] (flipping, cropping, or rotating)

This work is partially supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant no. 2023YFC3321401) and the EU's Horizon Europe HarmonicAI project under the HORIZON-MSCA-2022-SE-01 scheme with grant agreement number 101131117. Part of the work was done when B. Li was a visiting PhD student at TU Delft.
§Corresponding author: wwen@iie.ac.cn

979-8-3503-9466-5/24/$31.00 ©2024 IEEE
DOI 10.1109/WoWMoM60985.2024.00046



(a) I/Q at 25 dB (b) I/Q at 15 dB (c) Spectrogram at 25 dB (d) Spectrogram at 15 dB

Fig. 1: I/Q samples and spectrograms of coexisting WiFi, Bluetooth, and Zigbee signals.

or suppressing large objects to highlight small objects with matched filters in time domain [14]. However, these methods are not applicable in the shared signal classification scenario as the image operations disrupt the time-frequency characteristics of signals while filtering signals suffer from high false alarm rates and computational complexity (our tests reveal that the time cost is 10 times higher than the proposed approach.) In this paper, we propose a novel approach called ***Spectrum Painting (SP)***, inspired by the artistic technique, to create a simple background to highlight the main object in time-frequency domain. Specifically, the painting adjusts the larger objects to create a less complex background while emphasizing smaller ones, and leads to compressed spectrograms as a by-product. SP enables us to achieve high accuracy in the classification of small objects while maintaining low latency.

The second challenge is to design a low-complexity neural network that can effectively extract features from the spectrograms with extremely imbalanced object-occupied bins. The Convolutional Neural Network (CNN) has shown superior learning ability for large objects compared to small objects in images/spectrograms. However, in signal classification, the characteristics of small objects such as Bluetooth and Zigbee signals, are crucial in distinguishing between the signal classes with or without them. Compared to WiFi signals, the narrower bandwidths of Bluetooth and Zigbee signals result in a smaller number of bins being occupied in the spectrograms, posing a significant challenge in designing a neural network that can learn the features of both large objects like WiFi signals, and small objects such as Bluetooth and Zigbee signals.

To address the second challenge, we design an SP-enhanced *Dual-channel Convolutional Neural Network (DualCNN)* that extracts features from two channels with different inputs. The first channel takes an augmented spectrogram as input and focuses on extracting global features of all signals, while the second channel takes a painted spectrogram as input and learns the details of small objects. By concatenating the output of two channels, our model generates a comprehensive representation of the signal that captures both global information and local details. With a shallow architecture, our approach enables high accuracy of on-device signal classification with low latency.

We summarize our contributions in this paper as follows:

- We design a novel spectrogram enhancement framework – Spectrum Painting (SP)– that creates a simple background to highlight small objects, and by-products can compress spectrograms for resource-constrained embedded devices. This approach enhances the distinguishable features of various signals and reduces computational overhead.
- We design the dual-channel convolutional neural network –DualCNN– that solves the problem of simultaneously learning the features of both large and small objects in the spectrograms. DualCNN extracts global information and detailed features of the spectrogram through dual-channel inputs, achieving high accuracy with low inference latency.
- We implement our SP on Raspberry Pi 4B and thoroughly validate its performance with synthesized data and testbed data. Experiment results demonstrate that our approach reduces the latency from 100 ms to 5 ms while achieving an accuracy of up to 95%. Our proposed SP also exhibits robust performance in real-world scenarios, even when directly transferred to a new environment.
- To evaluate the performance in distributed spectrum sensing systems, we verify SP using testbed data in the Federated Learning (FL) architecture with both independent and identically distributed (iid) and non-iid data distributions. Our SP, integrated into common FL frameworks such as FedAvg, FedProx, and Scaffold, improves the accuracy largely to 95% and 68% in iid and non-iid scenarios, respectively. Cross-environment results show that SP can improve the accuracy on average from 24.6% to 83.8% compared to the baseline in the iid-FL scenario.

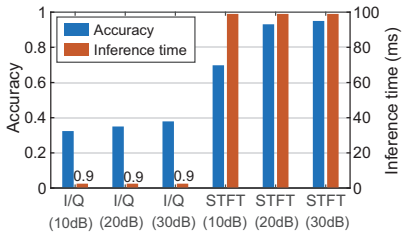
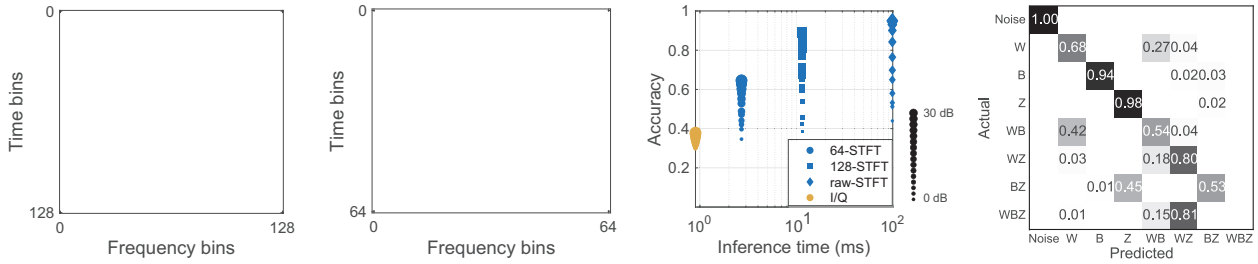


Fig. 2: Signal classification results under different SNRs.

II. Background and Motivation

Short-Time Fourier Transform (STFT) is a widely adopted signal processing technique that segments the I/Q samples with a window function and performs discrete Fourier transform in each window. STFT transforms the signal $x(t)$ from the time domain to the time-frequency domain as follows:

$$\textbf{STFT}\{x(t)\}(\tau,\omega) = \int_{-\infty}^{+\infty} x(t) f(t-\tau) e^{-j\omega t} dt, \quad (1)$$



(a) 128×128 spectrogram (b) 64×64 spectrogram (c) Accuracy vs. latency (d) Confusion matrix (64-STFT)

Fig. 3: Downsampled spectrograms and classification results.

where $f(\tau)$ is the window function, the size that determines the time and frequency resolution of STFT. The spectrogram is the magnitude squared of STFT, whose elements represent the signal's amplitude in a specific time-frequency bin.

Fig. 1 shows I/Q samples and spectrograms of coexisting WiFi, Bluetooth, and Zigbee signals when signal-to-noise ratios (SNRs) are 25 dB and 15 dB. There is one WiFi signal transmitting at 2.437 GHz with a 20 MHz bandwidth; one Bluetooth transmitting with frequency hopping between 2.4 GHz to 2.483 GHz and a 1 MHz bandwidth; and one Zigbee transmitting at 2.435 GHz with a bandwidth of 2.5 MHz. Obviously, spectrograms show distinct time-frequency characteristics, especially at low SNRs such as 15 dB, as shown in Fig. 1(d). Furthermore, STFT transforms the task of classifying signals into image classification, which can be effectively accomplished using CNN models.

However, using I/Q samples and raw spectrograms as inputs to CNN models to perform *on-device signal classification* is not practical. We conduct an experiment to evaluate this by considering the transmissions of WiFi (W), Bluetooth (B), and Zigbee (Z) signals at the 2.4 GHz band. We activate/deactivate the devices of each communication technology independently. Consequently, there are 8 signal combinations to be classified: only noise, W, B, Z, WB, WZ, BZ, and WBZ. *We run a CNN model for signal classification on the resource-constrained Raspberry Pi 4B.* Fig. 2 shows the accuracy and inference time of I/Q samples and spectrograms as the CNN model's input [5]. We have the following observations:

Observation 1: I/Q-fed CNN is fast but inaccurate. We use an energy threshold-based method to detect signals and feed the I/Q samples to the CNN with a dimension of 2×1024. It is clear from Fig. 2 that the I/Q-fed CNN only needs 0.9 ms for the on-device inference time, but its accuracy performance is very poor: achieving less than 40% accuracy under all SNRs.

Observation 2: STFT-fed CNN is more accurate but too slow. We fed the CNN with spectrograms of a 512×256 size. STFT-fed CNN can increase the accuracy up to 93%, but it incurs a latency almost 100× more than that of I/Q-fed CNN. These constraints imposed by high latency and computational overhead significantly hinder the practical use of STFT-fed CNN models on resource-constrained devices.

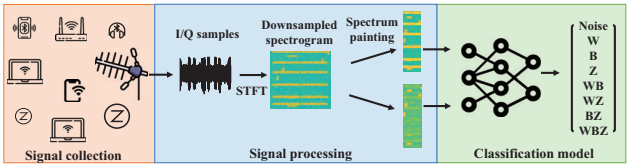


Fig. 4: The overview of our proposed framework for on-device signal classification.

III. System Design

In this section, we present our designs for achieving accurate and fast signal classification on resource-constrained devices.

A. Can We Simply Downsample the Spectrogram?

From Sec. II we know that classifying signals on resource-constrained devices using the STFT-fed CNN models leads to high latency, which is impractical. A straightforward method to address this problem is downsampling. Fig. 3(a) and 3(b) show the compressed images with sizes of 128×128 and 64×64 by downsampling the raw spectrogram with a size of 512×256. The inference times are shown in Fig. 3(c). We can observe that the inference time has been reduced from 100 ms to 11 ms and 2.7 ms, respectively. However, downsampling results in information loss. For example, the distinct characteristic of the prolonged duration of Bluetooth signals relative to other signals becomes indistinguishable in Fig. 3(b), particularly when Bluetooth overlaps with WiFi. This leads to significant drops in signal classification accuracy, for example, from 93.1% (raw-STFT) to 81% (128-STFT) and 60.1% (64-STFT) at 20 dB SNR, as shown in Fig. 3(c). We particularly analyze the performance gap between 64-STFT and raw-STFT through the confusion matrix, as shown in Fig. 3(d). We find that the classification errors of 64-STFT *primarily occur in the presence or absence of Bluetooth/Zigbee signals.* This issue is attributed to the extreme bins imbalance between Bluetooth/Zigbee and WiFi signals when downsampling the raw spectrogram to 128×128 or 64×64. Typically, CNN learns features of large objects (e.g., WiFi) better than small objects (e.g., Bluetooth). To improve accuracy, we thus should ensure the network can learn both characteristics of Bluetooth/Zigbee (small object) and WiFi (large object) signals.

B. Overview of the Proposed Framework

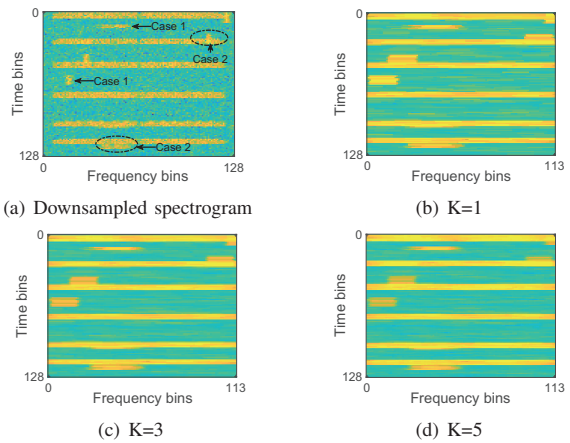
We design a framework to achieve the above goal, as shown in Fig. 4. Upon obtaining I/Q samples, we first utilize STFT to obtain the time-frequency spectrogram. Then, we downsample the original spectrograms into smaller sizes to reduce the computational overhead. Beyond these preliminary signal preprocessing, we design two key components to achieve high-accuracy signal classification with the downsampled spectrogram. **(1) Spectrum painting:** Downsampling the spectrogram results in information loss and bins imbalance, particularly for Bluetooth and Zigbee signals, which manifest as *small objects* in the spectrograms. This module adopts two methods to address these issues: augmenting the time-frequency features of Bluetooth/Zigbee signals in the spectrogram and adjusting the amplitude of WiFi signals to paint the spectrum and highlight small objects. **(2) Dual-channel CNN enabled signal classification:** Following the spectrum painting module, we obtain an augmented spectrogram and a painted spectrogram. They comprise augmented global information about time-frequency distribution of signals and highlighted small objects with a 'simple' painting background, respectively. We design a *DualCNN* integrating global and small-object signals' features for effective on-device signal classification.

C. Painting the Spectrogram

We analyze the downsampled spectrograms and find two cases that degrade the accuracy of signal classification.

- *Case 1: Bluetooth and Zigbee signals occupy significantly fewer bins compared to WiFi signal.* This stems from the narrower bandwidth and weaker received power of Bluetooth/Zigbee signals, leading to an extreme bins imbalance between Bluetooth/Zigbee and WiFi signals. Moreover, the downsampling process exacerbates this issue. For instance, one Bluetooth signal occupies only 30 bins while one WiFi signal encompasses around 460 bins in the 128×128 spectrogram, as shown in Fig. 5(a). This challenges the ability of DL models such as CNN to effectively detect and learn the features of small objects (Bluetooth/Zigbee) amidst the presence of prominent large objects (WiFi).
- *Case 2: Bluetooth/Zigbee signal overlaps with WiFi signal.* This overlap scenario becomes especially problematic when Bluetooth/Zigbee's appearance is completely covered by that of WiFi. However, due to their low data rates, Bluetooth/Zigbee usually needs more time to transmit a packet compared to WiFi. Thus, when they co-exist, (part of) Bluetooth/Zigbee signal could look like appended heads/tails to WiFi on the spectrogram, as outlined in Fig. 5(a). This poses a significant challenge for DL models to learn features of these appended Bluetooth/Zigbee signals, reducing signal classification accuracy.

In light of the above findings, we design a signal processing method –*Spectrum Painting (SP)*– to enable high-accuracy and low-latency on-device signal classification. First, SP augments characteristics of small objects (Bluetooth/Zigbee) to address the bins imbalance issue (Case 1). Then, SP adopts a painting-like method to simplify the spectrogram background, to mitigate the overlap issue (Case 2). Next, we present the details.



(a) Downsampled spectrogram (b) K=1 (c) K=3 (d) K=5

Fig. 5: Spectrum painting with different top K values.

Algorithm 1: Signal augmentation

Input: Spectrogram $\mathbf{S}_{M\times N}$, number of maximum values K, sliding window size L, step size D.
Output: Augmented spectrogram $\hat{\mathbf{S}}_{M\times P}$

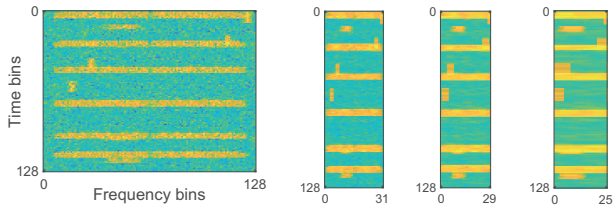
Calculate the average m of $\mathbf{S}_{M\times N}$
for $i \in \{1, 2, \cdots, M\}$ **do**
  while $j \leq N - L$ **do**
    $S_{i,j} \leftarrow$ Sort $[S_{i,j}, S_{i,j+1}, \cdots, S_{i,j+L}]$ in descending order, select the top K values, and then calculate their average
    $\hat{\mathbf{S}}_{i,j} \leftarrow S_{i,j} - m; \quad j \leftarrow j + D$
  end
end
return $\hat{\mathbf{S}}_{M\times P}$

1) Augmenting the Bluetooth/Zigbee signals in the spectrogram: To increase their bins ratios with respect to the bins of WiFi signals in the downsampled spectrogram, we *stretch and brighten* the Bluetooth/Zigbee signals (i.e., small objects). Let the matrix $\mathbf{S}_{M\times N}$ denote the downsampled spectrogram:

$$\mathbf{S}_{M\times N} = \begin{bmatrix} S_{1,1} & \cdots & S_{1,N} \\ \cdots & S_{i,j} & \cdots \\ S_{M,1} & \cdots & S_{M,N} \end{bmatrix}, \tag{2}$$

where M and N denote the spectrogram's width (time bins) and length (frequency bins), respectively; $S_{i,j}$ is jth frequency bin at time slot i. Our investigation reveals that stretching signals along the frequency domain can make small objects (Bluetooth/Zigbee) more visible. Thus we use a sliding window move along the frequency axis and calculate the mean of the top K maximum values in each windows to expand and brighten Bluetooth/Zigbee signals. The augmented spectrogram can be expressed as

$$\hat{S}_{i,j} = \text{Average}\big(\text{TOP}_K(\{S_{i,j-L/2}, ..., S_{i,j+L/2}\}\big), \tag{3}$$



(a) Downsampled spectrogram (b) L=8 (c) L=16 (d) L=32

Fig. 6: Spectrum painting with different sliding window sizes.

where $\text{TOP}_K(\{\cdot\})$ returns a $1 \times K$ vector containing the K maximum values in the sliding window. The augmented spectrogram $\hat{\mathbf{S}}_{M\times P}$ has a M length and a $P = (N - L)/D + 1$ width, where L is the window size and D is the step size of the sliding window. In the augmentation of small objects, the choice of K, L and D determines the augmentation extent. As shown in Fig. 5, when K is large (such as $K = 5$), the maximum values may include signal bins from the background (WiFi signals or noise), which would reduce the amplitude of the signal by averaging K values. While a small K value (such as $K = 1$) would amplify the fluctuation of the background, making the signals indistinguishable from the background.

Fig. 6 shows the results with different window sizes, with a step size of 4. The window length determines the range when selecting the maximum value of K. As Fig. 6(b) shows, a smaller window leads to a reduced number of sliding windows containing the signal, thereby compressing the signal representation in the frequency axis. Conversely, a longer window increases the number of windows capable of accommodating the signal, expanding the bins in the frequency axis. However, the extended window could also include more signals of WiFi, thereby decreasing the discernment between Bluetooth/Zigbee and the background (WiFi and/or noise).

Fig. 6(c) shows the augmented spectrogram using the sliding window with a window length of 16 and a step size of 4 and top 3 values in each window. The augmented spectrogram we obtained in this step contains extended and brighter Bluetooth/Zigbee signal with increased bins ratio, where the neural network can learn the features of all signals in the spectrograms. Therefore, we adopt the augmented spectrogram as one of the inputs of the neural network.

2) Learning the characteristics of Bluetooth/Zigbee signals (small objects) in the spectrogram: Intuitively, a conventional method to address the problem of learning features of small objects in the spectrogram (from the overlapped signals, i.e., Cases 2), is to flip and crop images, which are widely used in small object detection field. However, applying these techniques to spectrograms may not be ideal, as they can distort or mask critical features integral to signal classification, such as the characteristics of channel occupation, bandwidth, or transmission duration. Thus, an advanced approach is essential to capture the features of small objects in the spectrogram.

Drawing inspiration from the art of painting, we observe that painting with a relatively simple or uncluttered background allows the focus to be solely on the subject. Similarly, we want the network to focus on the small objects, rather than large objects or noise in the spectrogram. Thus, besides augmenting the small objects (cf. Sec. III-C1), we seek to create a "simple" painting background where WiFi signals and noise serve as the background while small objects such as Bluetooth and Zigbee signals are highlighted. By doing so, the Bluetooth/Zigbee signals that overlap with WiFi signals can also stand out.

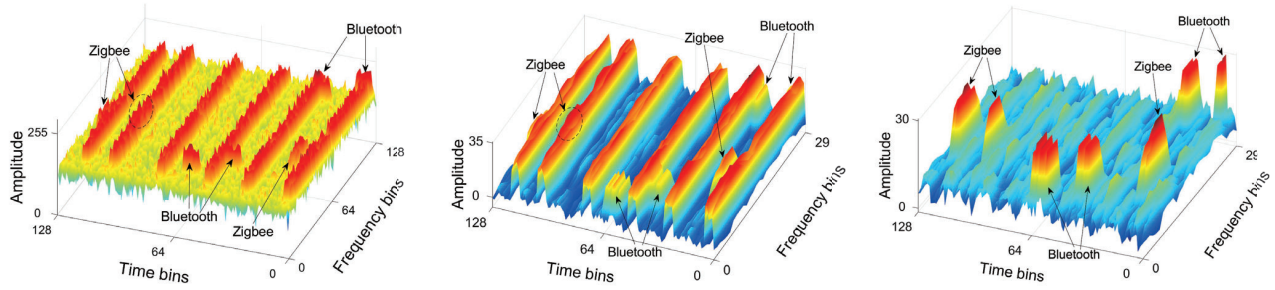
To achieve our objective, we manipulate the amplitude of the large objects while preserving its essential features, thus enabling it to function as the painting background alongside the noise. Upon observing Fig. 5(a), we deduce that WiFi signals consume significantly more bins than Bluetooth and Zigbee signals. Furthermore, we note that the WiFi signal is uniformly distributed along the entire y-axis, akin to the noise signal. Another key difference between WiFi signals and noise is the much higher amplitude of the former, despite waveform variations. By reducing the power of WiFi to correspond with that of noise, we can obtain a simplistic background that accentuates small objects. We can derive a painted spectrogram with a simple spectral background $\check{\mathbf{S}}_{M\times P}$, and each element in this matrix $\check{\mathbf{S}}_{M\times P}$ is calculated as follows:

$$\check{S}_{i,j} = \hat{S}_{i,j} - \frac{1}{N}\sum_{j=1}^{N} S_{i,j}. \tag{4}$$

In Fig. 7, we illustrate the downsampled 128×128 spectrogram and the two outputs of spectrum painting: augmented spectrogram and painted spectrogram at 20 dB. The left spectrogram has a high resolution but a weak distinguishability of signals and extreme bins imbalance. The augmented spectrogram shows enhanced time-frequency characteristics of signals but also suffers from a bins imbalance for WiFi signals and Bluetooth/Zigbee signals as shown in Fig. 7(b). By the second step of spectrum painting, the problems of bins imbalance and overlap are effectively addressed as shown in Fig. 7(c). The WiFi signals retain the fluctuation characteristics in the same horizontal line as the noise, ensuring a simplified spectrum background. As a result, the Bluetooth/Zigbee signals emerge as prominent features on the painted spectrogram, where a CNN model can easily extract features and reduce the classification error of the presence/absence of Bluetooth signals.

D. Dual-channel Convolutional Neural Network Structure

Our proposed approach utilizes the proposed spectrum painting module to generate two new spectrograms $\hat{\mathbf{S}}_{M\times P}$ and $\check{\mathbf{S}}_{M\times P}$. The former $\hat{\mathbf{S}}_{M\times P}$ enhances the SNR by stretching and brightening small objects and preserving global time-frequency features, while the latter $\check{\mathbf{S}}_{M\times P}$ highlights small objects by featuring a simplified background. To effectively capture both global and detailed features of signals in spectrograms, we propose a dual-channel convolutional neural network (DualCNN) that integrates two distinct channels. One channel is dedicated to capturing the global features of signal distribution across both time and frequency axis, while the other channel focuses on extracting fine-grained details of small objects (Bluetooth/Zigbee).



(a) Downsampled spectrogram (b) Augmented spectrogram (c) Painted spectrogram

Fig. 7: Spectrum painting: (a) downsampled spectrogram with a size of 128×128; (b) augmented spectrogram with amplified signals; (c) painted spectrogram with highlighted small objects.

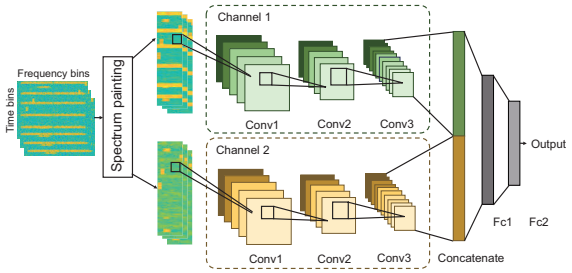


Fig. 8: The proposed SP-enhanced DualCNN structure.

As illustrated in Fig. 8, DualCNN consists of two parallel channels for feature extraction, feature fusion based on concatenation, and a fully connected network for classification. The two channels have identical structures and adopt only three convolution layers in each channel, which enable the model to be applied to resource-constrained devices. These layers have kernel sizes of 3×3, 5×5, and 7×7, providing the feature extraction network with a larger view to learn spatial characteristics. Batch normalization layers, rectified linear unit (ReLU) activation functions, and max pooling layers are used after each convolution layer.

The first channel of DualCNN, denoted as Channel 1, accepts a $M \times P$ augmented spectrogram as its input and generates a feature map that captures the global information of the signals present in the spectrogram. Channel 2 receives a $M \times P$ painted spectrogram as input and produces a feature map that specifically characterizes the small objects represented in the spectrogram. By combining the extracted features from both channels through concatenation and flattening operations, the network effectively integrates the distinguishing characteristics of both small and large objects. This discriminative feature representation significantly contributes to the improvement in classification accuracy.

IV. Performance Evaluation

A. Evaluation Setup

1) Synthesized data: We simulate the coexistence of WiFi, Bluetooth and Zigbee signals using *Matlab WLAN Toolbox, Bluetooth Toolbox, and Communications Toolbox*. We generate the waveform with different parameters, as listed in Table I. To simulate the wireless path fading, we utilize the *wlanTGnChannel* function for WiFi, and Rayleigh fading for Bluetooth and Zigbee. We set the transmission power of WiFi, Bluetooth and Zigbee to 15 dBm, 5 dBm and 0 dBm, respectively. Besides, we add an Additive White Gaussian Noise (AWGN) channel to simulate various SNRs. We activate/deactivate the transmission of three technologies independently to generate eight classes in non-coexisting and coexisting scenarios.

We use the pass loss model in [15] to simulate the various distances between the spectrum analyzer and devices. In our experiments, unless otherwise specified, we assume a fixed distance of 2 m between the spectrum analyzer and devices, which yields a received power of -31 dBm, -42 dBm and -46 dBm for WiFi, Bluetooth and Zigbee, respectively.

2) Testbed: We further use commodity devices to evaluate the performance of our Spectrum Painting (SP). We utilize the spectrum analyzer EMCAS6 to collect data in two distinct environments: *in the wild* and *in an office*. We strategically position *18 devices (six devices for each technology)* at varying distances from the spectrum analyzer as shown in Fig. 9 and Fig. 10. We also evaluate SP in a scenario involving a pair of devices to establish a single link for each technology. We collect in total 9600 spectrograms with a sampling rate of 20 MHz and a center frequency of 2.447 GHz. The dataset includes 8 types of signals corresponding to the presence/absence of each device type: only noise, WiFi (W), Bluetooth (B), Zigbee (Z), WB, WZ, BZ, and WBZ.

3) Baselines: For comparison with our proposed SP, we implement three baseline methods. All these baseline methods adopt the CNN model with four convolutional layers [5], each with modifications based on the input.

- **Raw-STFT**, which utilizes STFT to obtain the raw spectrogram with a size of 512×256. One more max-pooling layer is introduced in the last convolutional layer to reduce the feature map dimension.
- **128-STFT**, which employs downsampled 128×128 spectrogram as the input of CNN.

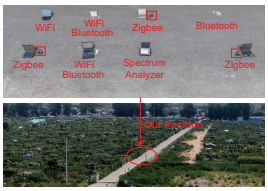


Fig. 9: Data collection in the wild with commodity devices.

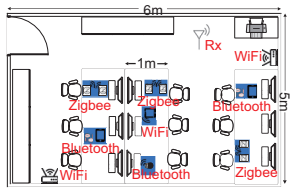


Fig. 10: Layout for data collection in the office.

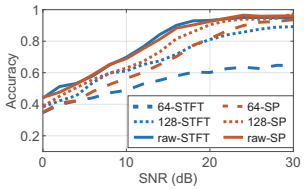


Fig. 11: Classification accuracy vs. various SNRs.

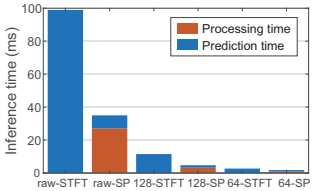


Fig. 12: On-device inference time (incl. processing time).

TABLE I: Parameters for waveform generator.

Protocol	Parameter	Values
Bluetooth	PHY modes, bandwidth	Basic rate, 1 MHz
	Frequency channel	Randomly hop for 79 channels
	Modulation scheme	GFSK
WiFi	Bandwidth, coding	20 MHz, BCC
	Frequency channel	2.447 GHz
	Modulation schemes	BPSK, QPSK, 16/64/256-QAM
Zigbee	Samples per chip	4
	Bandwidth, modulation	2 MHz, oQPSK
	Frequency channel	2.445 GHz
	Signal filter	Half-sine pulse shaping filter

- **64-STFT**, which downsamples the raw spectrogram to the size of 64×64 that is used as the input of CNN.

We use PyTorch for data processing and model construction. The model training is performed on an NVIDIA A10 GPU using 80% of the dataset; testing is conducted *on the embedded device Raspberry Pi 4B*, utilizing 20% of the dataset.

B. Preliminary Results

Comparison under different SNRs. We first compare SP against three baseline methods under varying SNRs. We train the models on samples with all SNRs and test them with samples under each SNR separately. The accuracy results in Fig. 11 shows that the accuracy of STFT-fed CNN models decreases from 95% to 89% and 64% when downsampling the raw spectrogram into 128×128 and 64×64, respectively. In contrast, SP achieves an accuracy consistently above 90% at high SNRs across all input sizes. Fig. 12 shows the inference time of all models executed on the Raspberry Pi 4B. *Note that for a fair comparison, the on-device inference time of our SP includes the model prediction time as well as the additional processing time introduced by our method.* It can be observed that downsampling the raw spectrograms reduces the inference time of STFT-fed CNN models from 99 ms to 11.5 ms and 2.7 ms for 128×128 and 64×64 spectrograms, respectively. SP achieves even lower inference times, reducing them further to 4.7 ms and 1.8 ms for 128×128 and 64×64 spectrograms, respectively. These results highlight the effectiveness of SP in improving the accuracy and reducing the latency.

Fig. 13 shows the confusion matrix of 64-SP at 30 dB SNR (please refer to Fig. 3(d) in Sec. III for the confusion matrix of 64-STFT for a comparison). The overall accuracy of 64-STFT and 64-SP is 68.4% and 92.9%, respectively. We can find that 64-STFT exhibits inferior performance in classes involving small objects of Bluetooth and Zigbee. Specifically, 64-STFT misclassifies 42% of WB as W, and 45% of BZ as Z. WBZ is completely misclassified to W, WZ and WB. Such suboptimal performance is due to the limited capacity of the STFT-fed CNN models in learning the distinguishing features of small objects. In contrast, 64-SP significantly improves the accuracy of WB, BZ, and WBZ from 54%, 53% and 0% to 86%, 93% and 80%, respectively. These results clearly demonstrate the effectiveness of our SP in correctly classifying classes that involve small objects, thereby improving overall accuracy.

Comparison with device motion of Bluetooth. To investigate the impact of Bluetooth device motion on signal classification, we simulate varying distances between the spectrum analyzer and Bluetooth devices, by modifying the received signal power. Results at 30 dB SNR (Fig. 14) indicate a decrease in accuracy with increasing distance. However, SP consistently outperforms STFT-fed CNN models, maintaining accuracy above 80% for all distances when using raw and 128×128 spectrograms as input. Even at a 10 m distance, SP achieves 8.5%, 15.8%, and 6.6% higher accuracy than STFT-fed CNN models with raw, 128×128, and 64×64 spectrograms as input, respectively. The results validate that our SP effectively enhances the visibility and quality of small objects, even when they exhibit extremely weak signals.

C. Impact of Different Settings in Spectrum Painting

The performance of SP depends on the selected parameters: the length L, step size D, and the top K values of the sliding window. We systematically evaluate various (L, D) combinations and diverse top K values to identify optimal settings.

In Fig. 15, we present results for five (L, D) settings with $K = 3$ and 128×128 spectrograms as input. Larger step sizes (D) induce information loss, resulting in decreased accuracy. For example, with a window size of 16, accuracy with $D = 8$ is 3.8%, 9.4%, and 1% lower than $D = 4$ at 10 dB, 20 dB, and 30 dB, respectively. Conversely, small step sizes like $D = 2$ can lower the SNR and reduce accuracy. For example, $(L, D) = (16, 2)$ yields 7.7% and 1.7% lower accuracy than $(L, D) = (16, 4)$ at 20 dB and 30 dB. The window size also influences top K selection and frequency-axis extension. Fig. 16 shows the impact of different top K values with $(L, D) = (16, 4)$. When K is set to 3, SP efficiently amplifies signals against overlapping and noise. Thus, 128-SP with $K = 3$ achieves the highest accuracy of 68%, 90% and 95.6% at 10 dB, 20 dB and 30 dB SNRs, respectively. In contrast, $K = 1$ or 5 causes an increase in background

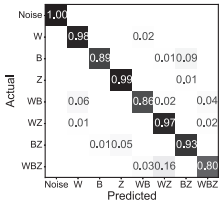


Fig. 13: Confusion matrix of 64-SP (SNR: 30 dB).

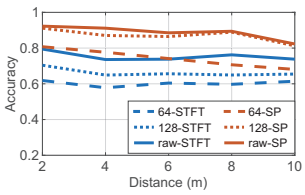


Fig. 14: Accuracy vs. positions of Bluetooth device.

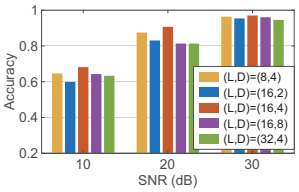


Fig. 15: Impact of window size L and step size D.

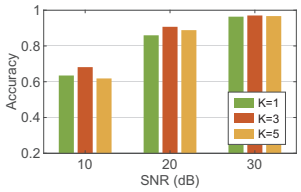


Fig. 16: Impact of selecting different top K values.

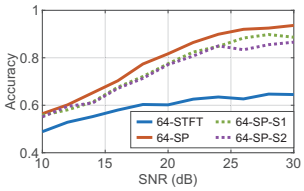


Fig. 17: Ablation study under different SNRs.

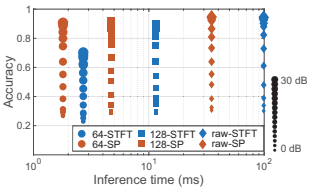


Fig. 18: Scalability performance with 30 devices.

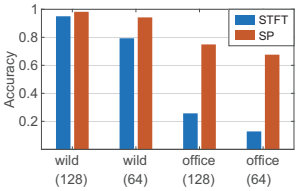


Fig. 19: Cross-environment testbed results (6 devices).

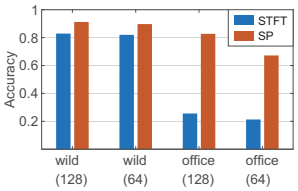


Fig. 20: Cross-environment testbed results (18 devices).

noise or a decrease in the signal amplitude, leading to suboptimal performance. With comprehensive analysis, we set $L = 16, D = 4, K = 3$ for subsequent experiments.

D. Ablation Study

We further study the individual contributions of each channel in our SP. We obtain two outputs from spectrum painting: an augmented spectrogram **(SP-S1)** containing global information of WiFi, Bluetooth, and Zigbee; and the painted spectrogram **(SP-S2)** containing highlighted small object details. To assess their performance, we evaluate them separately using the corresponding single channel of the proposed DualCNN. The results, using 64×64 spectrograms as input, are presented in Fig. 17. Remarkably, both 64-SP-S1 and 64-SP-S2 outperform 64-STFT, exhibiting accuracy improvements of 24.1% and 22% at 30dB SNR, respectively. However, due to the lack of global information, the accuracy of 64-SP-S2 is 7.1% lower than that of the complete 64-SP model. Conversely, 64-SP-S1 shows a small accuracy gap when compared to 64-SP, as it lacks the highlighted characteristics of small objects. The results demonstrate the importance of incorporating both the global information and the highlighted small objects to achieve high-accuracy on-device signal classification.

E. Scalability Evaluation

Next, we evaluate the performance of SP in a large scenario with 10 WiFi devices, 10 Bluetooth devices, and 10 Zigbee devices, all transmitting data simultaneously. We test different SNR scenarios by varying the positions of the devices. There are situations that WiFi signals partially appear on the spectrograms due to the random channel configuration.

The results in Fig. 18 illustrate that SP demonstrates superior performance with low latency, achieving an accuracy of 94.7%, 91.5%, and 90.6% using raw, 128×128 and 64×64 spectrograms as input, respectively. It is worth noting that the performance of the STFT-fed CNN models in the multi-device scenario is better at high SNRs than the single-device scenario discussed in Section IV-B. This improvement can be attributed to the increased number of Bluetooth and Zigbee signals captured, which effectively alleviates the network's burden in learning the characteristics of small objects. Nevertheless, for SP, the benefits of the multi-device scenario are not as pronounced because the compression in the frequency axis in SP leads to a higher similarity between Bluetooth and Zigbee signals. Nonetheless, it is observed that SP shows a significant reduction in the inference time and an accuracy improvement to above 90% at high SNRs across all input sizes.

F. Cross-environment Evaluation with Testbed Data

We also evaluate the performance of SP in different environments (*in the wild* and *in the office*) with real-world testbed data. We train the models on the data collected in the wild. For testing the model, we use two settings: on the data collected in the wild (a less challenging scenario since both training and test data are from the same environments), and on the data collected in the office (a more challenging scenario since the test data is from a different environment).

Fig. 19 shows the accuracy in a scenario with six devices. The two leftmost groups of bars represent the outcomes of training and testing the models exclusively on the wild dataset. In comparison to 128-STFT and 64-STFT, 128-SP and 64-SP achieve 4% and 8% higher accuracy, respectively. Furthermore, when deploying the trained models for classifying data in the office (see the two rightmost groups of bars in Fig. 19), the accuracy of 128-STFT and 64-STFT substantially decreases to 25.7% and 12.8%, respectively. In contrast, our SP shows a significantly higher accuracy, achieving 74.9% and 67.6% for 128-SP and 64-SP. This is because that the distinct surroundings and device placements between the wild and the office environments cause variations in signal propagation and reception. However, our SP addresses this challenge by effectively highlighting environment-independent signal features

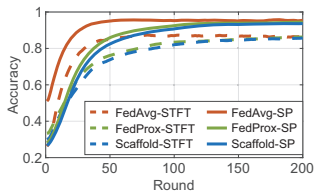


Fig. 21: Testbed results in the iid FL scenario.

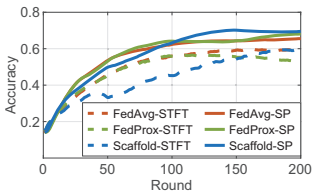


Fig. 22: Testbed results in the non-iid FL scenario.

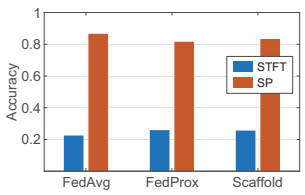


Fig. 23: Cross-environment testbed results (iid FL).

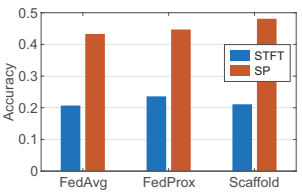


Fig. 24: Cross-environment testbed results (non-iid FL).

through the process of spectrum painting, thereby reducing the impact of environmental variation on classification accuracy.

Fig. 20 illustrates the consistent performance trend observed in the multi-device scenario. Notably, 128-SP demonstrates a marginal decrease in accuracy by only 8.5% when transferred from wild to office. Conversely, the accuracy of 128-STFT experiences a significant drop from 82.9% to 25.6% when transferring the models to a new environment. This is also because the propagation of WiFi, Bluetooth, and Zigbee signals is environment dependent. Additionally, it is worth noting that the accuracy of SP in the multi-device scenario exhibits a comparatively smaller decrease compared to the single-device scenario, when transferring the trained models from wild to office. Remarkably, the correct classification within the classes of B and BZ in the multi-device scenario demonstrates improvement. This is because the one spectrogram in the multi-device scenario tends to captures multiple Bluetooth signals. Consequently, the presence of more Bluetooth signals within a given spectrogram simplifies the classification task for the models. The results show that SP can still work in new environments but the accuracy can be decreased.

V. Evaluation within the Federated Learning Architecture

Background. Spectrum sensing often relies on distributed systems to provide better coverage and reliability [16]. Nevertheless, conventional distributed spectrum sensing systems with DL algorithms require uploading local data to the central server, raising privacy issues. Furthermore, training a DL model at a local node requires a substantial amount of labeled data, usually unavailable to a single node. *Federated Learning (FL)* addresses these challenges by allowing local clients (i.e., nodes) to collaboratively train a global model without data sharing [17]. Local clients independently train their own models using local data and then upload the computed gradients to the server. The server aggregates these gradients to formulate a global model. To confront the challenges inherent in on-device signal classification and privacy concerns in distributed spectrum sensing systems, we evaluate the performance of our SP within the FL architecture in the this section.

Setup. We use the testbed data collected from 18 devices. We consider three prominent FL architectures: FedAvg [18], FedProx [19], and Scaffold [20]. Both *independent and identically distributed (iid)* as well as *non-iid* data distributions are tested. All methods adopt the same network architecture to ensure fairness in our assessment. We use eight clients and run 200 communication rounds between local clients and the server. In the iid scenario, we evenly distribute the entire training dataset among the clients, with each client holding all the classes. In the non-iid scenario, each client is assigned two classes, and the number of samples within each client is proportional to 1/8 of the total sample count. We configure the local epochs and batch size to 5 and 50, respectively. In each communication round, all clients are involved in model training. We use Adam as the optimizer to update the models with a learning rate of 0.0001.

Evaluation with iid and non-iid data. Fig. 21 and Fig. 22 respectively show the performance of 128-STFT and 128-SP in iid and non-iid scenarios. The training and testing procedures are conducted both on the data collected in the wild. From Fig. 21 we can find that our SP achieves an accuracy of 93.5% after 100 communication rounds, which is 9.3% higher than that of 128-STFT. Also, we note that FedAvg converges at 50 communication rounds, whereas both FedProx and Scaffold reach convergence after 100 rounds. This is because the corrective measures on the weight updates of local models in FedProx and Scaffold slow down the model convergence. In the non-iid scenario shown in Fig. 22, we observe a significant decline in accuracy for both the baseline and our SP model, reaching 59.3% and 65.5%, respectively. This decline can be attributed to the limited efficacy of features learned solely from two classes within each local client, thereby hampering the global model's ability to classify mixed signals accurately. The results validate the superiority of our SP over the baselines within the FL architecture, in both iid and non-iid scenarios.

Cross-environment evaluation. To further assess the cross-environment performance within the FL architecture, we conduct evaluations by employing global models trained on the wild dataset and testing them on data collected in the office. The results in the iid scenario, given in Fig. 23, show that 128-SP exhibits remarkable accuracy above 80% across all three FL methods when deployed in a new environment. Conversely, 128-STFT demonstrates lower accuracy, achieving only 22.5%, 25.9%, and 25.6% accuracy, respectively. These findings emphasize the robustness of 128-SP in effectively adapting to a new environment and surpassing the performance of 128-STFT by a considerable margin. Comparatively, the global models in the non-iid scenario depicted in Fig. 24 exhibit relatively poorer performance. This is due to the difficulties the global models encounter in learning distinctive features in the non-iid scenario. Nonetheless, even in this

scenario, when the global models are transferred to a new environment, 128-SP demonstrates noteworthy improvements in accuracy, surpassing the performance of 128-STFT by 20.8%, 21.1%, and 27.1%, respectively. These findings emphasize the robustness of SP in effectively adapting to a new environment within the FL architecture, in both iid and non-iid scenarios.

VI. Related work

Signal classification using I/Q samples. A prior work [6] has addressed the classification of Bluetooth, WiFi, and Zigbee signals in the 2.4 GHz band. They utilize I/Q samples as input to develop a CNN that achieved an accuracy of 99% at high SNRs and 85% at low SNRs. In a more recent study [10], the authors considered the classification of coexisting WiFi, LTE, and 5G signals in both non-coexisting and coexisting scenarios. They designed a CNN and Long Short-Term Memory combined deep neural network for classification. The accuracy of the I/Q-based model drops by 20% in coexisting scenarios. However, these works have not studied on-device signal classification where computing resource is limited.

Signal classification using STFT. The authors [2] employed STFT for the classification of Low-Power Wide-Area Network technologies. The results indicate that the STFT-based CNN outperforms the I/Q-based CNN, particularly at low SNR scenarios, achieving a 20% improvement in accuracy. Additionally, the authors in [10] have shown that the use of STFT as input can enhance classification accuracy by over 10% in coexisting scenarios. Despite the promising results, researchers have highlighted that the STFT-based models require more resources and time for classification [3].

Reducing the latency of STFT-based signal classification with cropping. Commonly, to ensure that the signals on the spectrogram are sufficiently clear for neural networks to learn their characteristics, the size of the spectrogram is typically set to at least 128×128. However, such large spectrograms can be impractical to deploy on resource-constrained devices. The authors in [8] proposed a Quarter-spectrogram approach to eliminate redundant areas in original spectrograms, compressing its size from 128×128 to 64×64. Similarly, [7] cropped the full radar band into a band of interest for radar detection. However, in certain scenarios, signals may randomly appear in the spectrogram, making it difficult to remove unwanted regions. Thus, a trade-off must be made between accuracy and computation complexity when using spectrograms for neural network training in signal classification.

VII. Conclusion

In this paper, we have proposed a novel framework named Spectrum Painting (SP) to classify signals with low latency and high accuracy on resource-constrained devices. We utilized downsampled spectrogram to represent the time-frequency characteristic of signals. We proposed to paint the compressed spectrograms for signal augmentation and small object highlights, to compensate for the information loss and pixel imbalance in the compressed spectrograms. We designed a Dual-channel Convolutional Neural Network (DualCNN) to learn the enhanced spectrogram's global information and detailed features of small objects. Extensive experiments conducted on synthesized data and testbed data within both the centralized learning architecture and the Federated Learning architecture show that our SP framework can achieve accurate on-device signal classification on resource-constrained devices and significantly reduces the signal classification time.

References

[1] A. Shahid and et. al., "Demo abstract: Identification of lpwan technologies using convolutional neural networks," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019.

[2] ——, "A convolutional neural network approach for classification of lpwan technologies: Sigfox, lora and ieee 802.15. 4g," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2019.

[3] J. Fontaine, A. Shahid, R. Elsas, A. Seferagic, I. Moerman, and E. De Poorter, "Multi-band sub-ghz technology recognition on nvidia's jetson nano," in *IEEE Vehicular Technology Conference (VTC)*, 2020.

[4] M. He and et. al., "Identification of ism band signals using deep learning," in *Wireless and Optical Communications Conference (WOCC)*, 2020.

[5] N. Bitar, S. Muhammad, and H. H. Refai, "Wireless technology identification using deep convolutional neural networks," in *IEEE Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

[6] M. Schmidt, D. Block, and U. Meier, "Wireless interference identification with convolutional neural networks," in *IEEE international conference on industrial informatics (INDIN)*, 2017.

[7] W. M. Lees and et. al., "Deep learning classification of 3.5-ghz band spectrograms with applications to spectrum sensing," *IEEE transactions on cognitive communications and networking*, 2019.

[8] F. Bhatti and et. al., "Shared spectrum monitoring using deep learning," *IEEE Transactions on Cognitive Communications & Networking*, 2021.

[9] A. Selim and et. al., "Spectrum monitoring for radar bands using deep convolutional neural networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2017.

[10] W. Zhang, M. Feng, M. Krunz, and A. H. Y. Abyaneh, "Signal detection and classification in shared spectrum: A deep learning approach," in *IEEE Conference on Computer Communications (INFOCOM)*, 2021.

[11] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "Deepsense: Fast wideband spectrum sensing through real-time in-the-loop deep learning," in *IEEE Conference on Computer Communications (INFOCOM)*, 2021.

[12] S. R. Shebert, A. F. Martone, and R. M. Buehrer, "Wireless standard classification using convolutional neural networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2021.

[13] B. Zoph and et. al., "Learning data augmentation strategies for object detection," in *ECCV*, 2020.

[14] D. Bhargavi and C. R. Murthy, "Performance comparison of energy, matched-filter and cyclostationarity-based spectrum sensing," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2010.

[15] T. Janssen and et. al., "Lora 2.4 ghz communication link and range," *Sensors*, 2020.

[16] Y. Jing, T.-Y. Wang, and X. Yu, "A blind distributed spectrum sensing scheme with homogeneity test," *IEEE Transactions on Wireless Communications*, 2022.

[17] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *IEEE conference on computer communications (INFOCOM)*, 2021.

[18] H. B. McMahan and et. al., "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics*, 2016.

[19] T. Li and et. al., "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, 2020.

[20] S. P. Karimireddy and et. al., "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. ICML, 2020.