Deep Learning-Based Wideband Spectrum Sensing: A Low Computational Complexity Approach

Ruru Mei and Zhugang Wang

***Abstract*— One of the key challenges in wideband spectrum sensing is that it has to be performed with extremely low latency and high accuracy over a large band to detect tiny spectrum holes. Motivated by this, we introduce a parallel convolutional neural network (ParallelCNN)-based wideband spectrum sensing approach, which aims to detect tiny spectrum holes with high sensing accuracy and low latency. The ParallelCNN contains two parallel CNNs, where the I/Q samples are split into two parts and processed concurrently by the two CNNs. Afterwards, the outputs of the two CNNs are combined to improve spectrum sensing performance. We extensively evaluate the accuracy, latency and generality performance of ParallelCNN. Performance evaluations indicate that the ParallelCNN approach outperforms other deep learning (DL)-based approaches such as the DeepSense and TCN, in terms of a higher precision, a higher recall and a lower latency, under a wide range of signal-to-noise (SNR) levels. Strikingly, the latency can be reduced by up to 94%. In addition, the ParallelCNN can be generalized to different protocols such as LTE-M uplink transmissions, 5G NR downlink transmissions and WiFi transmissions.**

***Index Terms*— Cognitive radio, wideband spectrum sensing, deep learning, convolutional neural network, latency.**

I. Introduction

THE rapid growth of new wireless devices and technologies such as smart terminals and fifth-generation (5G) technologies has caused a shortage of available radio spectrum due to the static spectrum allocation policy. To address this challenge, cognitive radio (CR) technology has been proposed to allow the coexistence of multiple wireless communication technologies (e.g., LTE, WiFi, NR, Satellite Communications) within the same spectrum band without causing interference. The CR technology enables secondary users (SUs) to detect empty channels and to opportunistically communicate within these channels, while ensuring that primary users (PUs) are not adversely affected [1]. Therefore, spectrum sensing is the foundation of CR technology, which has been widely concerned by academia and industry [2].

Previous work on spectrum sensing has been divided into two categories: narrowband and wideband [3]. Energy detection is a popular approach for narrowband spectrum sensing, but it has poor detection performance under low signal-to-noise (SNR) scenarios [4]. Other narrowband sensing methods such as matched filter and cyclostationary features detection can provide good performance in low SNR regime, but they require prior knowledge of the primary network [5].

The wideband spectrum sensing techniques usually fall into Nyquist and sub-Nyquist categories [3]. For the Nyquist-based wideband sensing techniques, most researchers divide the wide band into multiple sub-bands and adopt narrowband sensing methods to detect these sub-bands sequentially or simultaneously. However, the sequential sensing approaches occupy a long sensing time, making them hardly adaptable to real-time communications. On the other hand, the simultaneous sensing approaches such as multi-band joint detection [6] and filter bank-based sensing [7] require numerous sensors and joint synchronization functions, which lead to high implementation complexity. Compressive wideband spectrum sensing techniques utilize a few observations to reconstruct the signal, but the nonlinear reconstruction process introduces high latency and degrades the sensing performance compared with Nyquist-based approaches [8], [9].

Deep learning (DL)-based spectrum sensing technology has recently attracted considerable attention due to its superior performance over conventional algorithms [10], [11], [12], [13]. However, the current applications of DL in spectrum sensing are limited to narrowband sensing and have high computational complexity. For instance, both the CD-DetectNets proposed in [14] and the AlexNet-based scheme proposed in [15] require more than a million parameters. Additionally, the CM-CNN-based algorithm [16] and the CNN-long short term memory (LSTM)-based algorithm [11] use sample covariance matrices as the input of networks, which improves the sensing accuracy, but the preprocessing of covariance matrices increases computational complexity. These DL-based narrowband sensing algorithms do not consider the importance of latency for practical devices. This is because many DL techniques require extensive computation and data processing, resulting in high latency. To address this issue, a CNN-based framework (DeepSense) [17] was proposed for wideband spectrum sensing, which detects multiple holes at once directly from in-phase and quadrature (I/Q) samples. Such a CNN-based sensing approach presents low latency, but its accuracy decreases significantly under relatively low SNR scenarios.

In this work, we propose a parallel CNN-based wideband spectrum sensing approach, referred to as ParallelCNN, for monitoring multiple spectrum holes simultaneously. Compared with DeepSense [17], we demonstrate that the proposed approach can achieve higher spectrum sensing accuracy in the

Manuscript received 11 July 2023; accepted 28 August 2023. Date of publication 31 August 2023; date of current version 11 October 2023. This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (Class A) (Subject Number: XDA153501, Sub-Subject Number: XDA15350103). The associate editor coordinating the review of this letter and approving it for publication was C. Studer. *(Corresponding author: Zhugang Wang.)*

Ruru Mei is with the Key Laboratory of Electronics and Information Technology for Space Systems, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Electromagnetic Field and Microwave Technology, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: meiruru20@mails.ucas.ac.cn).

Zhugang Wang is with the National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangzg@nssc.ac.cn).

Digital Object Identifier 10.1109/LCOMM.2023.3310715

presence of lower latency and fewer parameters. The major contributions of this article are threefold.

1) We propose a parallel CNN (ParallelCNN)-based approach to perform wideband spectrum sensing, where I/Q samples are divided into two parts and processed in parallel by the two CNNs. Therefore, the latency of the ParallelCNN is significantly reduced.
2) We evaluate the accuracy, size and latency of the ParallelCNN on three datasets, including (i) LTE-M uplink transmissions over a 10-MHz-wide band; (ii) 5G NR downlink transmissions in a non-terrestrial network (NTN) channel over a 60-MHz-wide band; (iii) WiFi transmissions collected using software defined radio (SDR) testbed. Simulation results show that the ParallelCNN can be successfully generalized to different protocols, architectures and spectrum environments.
3) The ParallelCNN shows better performance compared with the DeepSense [17] and the temporal convolutional network (TCN)-based method [18], in terms of lower latency, higher precision and higher recall. ParallelCNN can detect spectrum holes as small as 5.4% of the bandwidth of the monitored spectrum with a latency of 0.56ms, a precision and recall of 91% and 88% respectively at SNR = 0dB.

The remaining part of this article is organized as follows. Section II introduces the proposed ParallelCNN-based spectrum sensing algorithm. The utilized datasets are described in Section III. In Section IV, extensive simulation results are provided to evaluate the performance of the ParallelCNN. Finally, our conclusions are drawn in Section V.

II. ParallelCNN Aided Spectrum Sensing Algorithm

In this section, we introduce the ParallelCNN architecture and the procedure of offline training.

A. ParallelCNN Architecture

In order to meet the strict latency and accuracy constraints of real-time spectrum sensing, we need to design a lightweight model, whilst obtaining high sensing accuracy. Spurred by these goals, we adopt three strategies: (i) A multi-label CNN is chosen to identify which portions of the wide spectrum are empty, reducing latency compared to the sequential narrowband sensing. (ii) We use unprocessed I/Q samples, which avoids the resource consumption required for fast Fourier transform (FFT). This is mainly thanks to the fact that the hidden spectrum feature information in the I/Q constellation plane can be learned by the filters in the four convolutional layers within the ParallelCNN. (iii) The I/Q samples are split into two parts as input, and a parallel CNN scheme is designed to process two I/Q sequences simultaneously, further increasing accuracy and reducing latency.

Fig. 1 depicts the architecture of ParallelCNN, while Table I summarizes its hyperparameters. In the table, $p \times q$ means there are p 1D convolution kernels and each kernel is with a size of q. $p(q)$ means there are p output features and q input features in the fully connected layer. ParallelCNN is composed

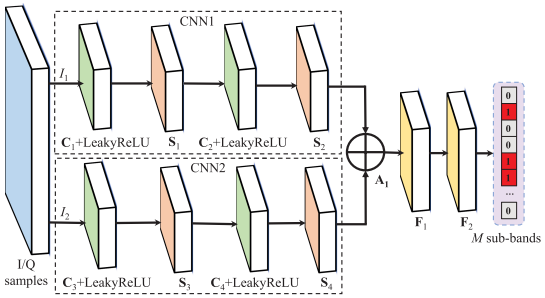

Fig. 1. ParallelCNN architecture.

TABLE I
HYPERPARAMETERS OF THE PARALLELCNN FOR ALL DATASETS

Layers	LTE/SDR Dataset			NR Dataset		
	Filter	Stride	Out shape	Filter	Stride	Out shape
I_1 / I_2			(64, 2)			(128, 2)
$\mathbf{C}_1$ / $\mathbf{C}_3$	8×3	1	(62, 8)	8×5	1	(124, 8)
$\mathbf{S}_1$ / $\mathbf{S}_3$	2	2	(31, 8)	2	2	(62, 8)
$\mathbf{C}_2$ / $\mathbf{C}_4$	16×5	2	(14, 16)	16×7	1	(56, 16)
$\mathbf{S}_2$ / $\mathbf{S}_4$	2	2	(7, 16)	2	2	(28, 16)
$\mathbf{A}_1$			112			448
$\mathbf{F}_1$	64(112)		64	32(448)		32
$\mathbf{F}_2$	16/4(64)		16/4	14(32)		14

of CNN1, CNN2, and two fully connected layers, and the spectrum sensing using ParallelCNN is explained as follows.

1) Input data: We view N consecutive I/Q samples as input $I \in \mathbb{R}^{N\times 2}$ and divide I into $I_1 = I[0 : N/2] \in \mathbb{R}^{N/2\times 2}$ and $I_2 = I[N/2 : N] \in \mathbb{R}^{N/2\times 2}$.
2) Feature extraction of input signals: I_1 and I_2 are propossed by CNN1 and CNN2, respectively. CNN1 and CNN2 alternately stack the 1D convolutional layers (e.g. $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, $\mathbf{C}_4$) and the 1D maximum pooling layers (e.g. $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathbf{S}_3$, $\mathbf{S}_4$). The feature maps extracted by the CNN1 from I_1 can be expressed as

$$S_2 = f_{MP}(f_L(W_2 * f_{MP}(f_L(W_1 * I_1 + b_1)) + b_2)), \tag{1}$$

and the feature maps extracted by the CNN2 from I_2 can be given by

$$S_4 = f_{MP}(f_L(W_4 * f_{MP}(f_L(W_3 * I_2 + b_3)) + b_4)), \tag{2}$$

where W_i and b_i are convolution filters and bias of the i-th convolutional layer, respectively, $f_L(x) = \max(0, x) + 0.2 \times \min(0, x)$ is the LeakyReLU activation function, $f_{MP}(\cdot)$ represents the maximum pooling function, and $*$ represents the convolution operation.
3) Combination: The outputs of the CNN1 and CNN2 are fed to the add layer $\mathbf{A}_1$ for merging, defined as

$$A_1 = \text{flatten}(S_2 + S_4), \tag{3}$$

where flatten$(\cdot)$ denotes the function that flattens a tensor into a vector.

4) Classification: The add layer is followed by two fully connected layers (e.g. $\mathbf{F}_1$, $\mathbf{F}_2$) to facilitate the integration of the extracted spectrum feature information. We treat wideband spectrum sensing as an M-label classification problem, thus the output of the ParallelCNN can be defined as

$$\hat{Y} = f_S(W_{F_2}(f_L(W_{F_1}A_1 + b_{F_1})) + b_{F_2}), \quad (4)$$

where W_{F_1}, W_{F_2}, b_{F_1} and b_{F_2} are the weights and biases of layers $\mathbf{F}_1$, $\mathbf{F}_2$, respectively, and $f_S(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

B. Offline Training Strategy

For training, the labeled I/Q samples are collected to construct the training dataset and denoted by

$$(\mathbf{I}, \mathbf{Y}) = \left\{(I^{(1)}, Y^{(1)}), \cdots, (I^{(q)}, Y^{(q)}), \cdots, (I^{(Q)}, Y^{(Q)})\right\}, \quad (5)$$

where $I^{(q)} \in \mathbb{R}^{N\times 2}$ represents the input data for neural network; $Y^{(q)} = [y_1^{(q)}, \cdots, y_m^{(q)}, \cdots, y_M^{(q)}], y_m^{(q)} \in \{0, 1\}$ represents the M labels.

The binary cross-entropy is adopted as the loss function of ParallelCNN. A stochastic gradient descent (SGD)-based Adam optimizer is used. In addition, to prevent overfitting, an EarlyStopping function is deployed to monitor the loss of the validation set during the training process.

III. Experimental Datasets

In this section, we provide an overview of the three datasets with different transmission protocols and spectrum environments.

A. Three Datasets

LTE dataset comprises approximately 520,000 transmissions and is created through the simulation of LTE-M uplink transmissions in the physical uplink shared channel (PUSCH) over a 10-MHz-wide band. The 10-MHz-wide spectrum band is divided into 16 sub-bands. Each sub-band is 3 resource blocks wide which is 540 kHz in bandwidth. In addition, each transmission within the LTE dataset experiences Rayleigh fading channel and Additive White Gaussian Noise (AWGN).

NR dataset contains about 520,000 transmissions of 14 non-overlapping channels, occupying a total of 60 MHz bandwidth. This dataset is created by simulating 5G NR link transmissions in the physical downlink shared channel (PDSCH). Each transmission within the NR dataset experiences NTN tapped delay line (TDL)-A propagation channel and AWGN. Meanwhile, the Doppler shift induced by satellite movement is as high as 29637 Hz, significantly increasing the difficulty of classification. Furthermore, a subcarrier spacing of 30 kHz is selected and each sub-band is configured to comprise 11 resource blocks, resulting in a sub-bandwidth of 3.96 MHz.

SDR dataset provided in [17] was collected using 5 USRP N210s SDRs. It consists of roughly 500,000 transmissions spread across four channels each with a bandwidth of 5 MHz, occupying a collective bandwidth of 20 MHz. The data was collected on two separate days with different transmitter orientations to provide diversity in terms of SNR and channel effects within the dataset.

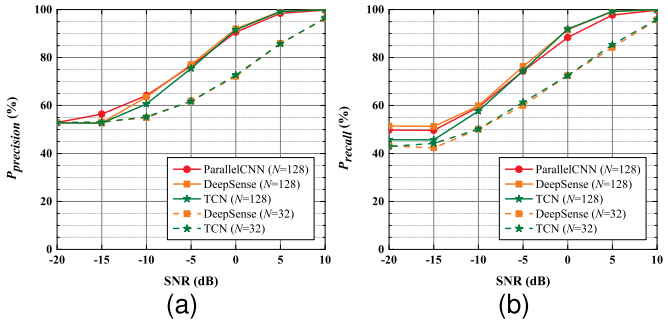

Fig. 2. $P_{precision}$ and P_{recall} versus SNR on LTE dataset, $M = 16$. (a) $P_{precision}$ versus SNR. (b) P_{recall} versus SNR.

B. Performance Metrics

We treat wideband spectrum sensing as a multi-label classification problem, whose goal is to classify multiple frequency bands in the spectrum as busy or idle. Precision, recall and F1-score are general metrics used for evaluating the performance of classifiers. To obtain an overall performance for a multi-label classifier, these values need to be averaged amongst classes. The micro-averaging for precision, recall and F1-score are

$$P_{precision} = \frac{\sum_{c=1}^{C}(\text{True Positives})_c}{\sum_{c=1}^{C}(\text{True Positives})_c + (\text{False Positives})_c}, \quad (6)$$

$$P_{recall} = \frac{\sum_{c=1}^{C}(\text{True Positives})_c}{\sum_{c=1}^{C}(\text{True Positives})_c + (\text{False Negatives})_c}, \quad (7)$$

$$P_{F1} = 2 \times \frac{P_{precision} \times P_{recall}}{P_{precision} + P_{recall}}, \quad (8)$$

where $P_{precision}$ is essentially the measure of false positives, P_{recall} is the measure of false negatives, P_{F1} is the harmonic mean of the precision and recall, and C is the total number of classes for our application.

IV. Simulation Results

In this section, we evaluate the sensing performance in terms of the precision and recall, through comparing the proposed ParallelCNN method with the DeepSense [17] and TCN methods [18]. The TCN consists of four residual blocks and a fully-connected layer, with each convolutional layer utilizing 16 1D convolution kernels of size 2. For $N = 32$, the filter sizes for the four convolutional layers in the DeepSense are set to 16×3, 16×3, 32×5, and 32×5, respectively. For $N = 128$, the filter sizes for the four convolutional layers in the DeepSense are set to 16×5, 16×5, 32×11, and 32×11, respectively.

A. LTE Dataset Results

Fig. 2(a) and Fig. 2(b) show the performance of different methods on the LTE dataset in terms of $P_{precision}$ and P_{recall},

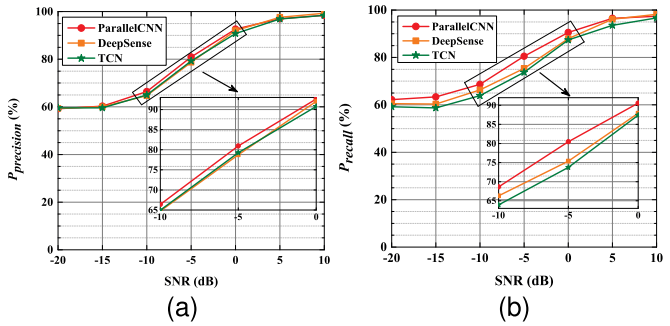

Fig. 3. $P_{precision}$ and P_{recall} versus SNR on NR dataset, $M = 14$. (a) $P_{precision}$ versus SNR. (b) P_{recall} versus SNR.

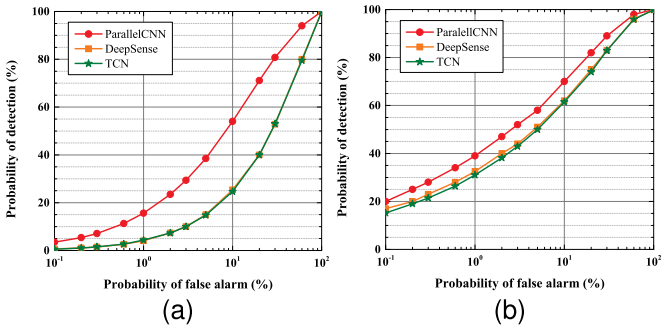

Fig. 4. ROC curves, SNR= -5dB. (a) LTE dataset. (b) NR dataset.

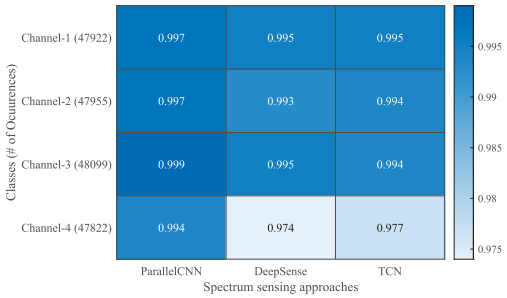

Fig. 5. F1-score per channel on SDR dataset, $M = 4$.

respectively, at different SNR values. The solid and dashed lines refer to $N = 128$ and $N = 32$. From the solid lines in Fig. 2, it can be seen that ParallelCNN, DeepSense, and TCN algorithms exhibit similar sensing performance when they all utilize 128 I/Q samples. As shown by the dashed lines in Fig. 2, the performance of DeepSense and TCN algorithms deteriorates significantly when the number of I/Q samples used is reduced to 32. For example, the $P_{precision}$ of the ParallelCNN is 15% higher than DeepSense and TCN, and the P_{recall} of the ParallelCNN is 16% higher than DeepSense and 15% higher than TCN at SNR=5dB. This is mainly due to the fact that using a larger number of I/Q samples can lead to improved classification accuracy as it provides more information about the signal. However, this can come at the cost of increased latency. The results in Table II suggest that DeepSense and TCN algorithms incur extremely large latency when they take 128 I/Q samples as input, and thus cannot meet the strict real-time constraints. To mitigate this issue, the ParallelCNN method leverages parallel processing to enable faster classification with minimal latency. Therefore, the results in Fig. 2 and Table II suggest that the ParallelCNN is a more efficient and effective method for detecting spectrum holes compared to the DeepSense and TCN methods.

B. NR Dataset Results

Fig. 3(a) and Fig. 3(b) present the $P_{precision}$ versus SNR curves and P_{recall} versus SNR curves, respectively, for the simulated NR dataset. The number of I/Q samples used in ParallelCNN, DeepSense, and TCN models is increased to 256, 128, and 128, respectively, to mitigate the impact of Doppler shift up to 29637Hz in NR transmissions. It is shown that the proposed ParallelCNN method still maintains optimal detection performance compared to DeepSense and TCN methods. However, the gap between the three methods is narrowed, which implies that the accuracy of DeepSense and TCN is improved compared to previous implementations on the LTE dataset. For example, at SNR=-5dB, the ParallelCNN can achieve a $P_{precision}$ and P_{recall} of 80.9% and 80.5% respectively; the DeepSense can achieve a $P_{precision}$ and P_{recall} of 78.7% and 75.3% respectively; the TCN can achieve a $P_{precision}$ and P_{recall} of 79.2% and 73.7% respectively. The reason for the narrowing of the gap could be attributed to the fact that both DeepSense and TCN methods trade extremely high implementation complexity for accuracy, resulting in unacceptable latency. In comparison, ParallelCNN delivers higher accuracy while being less complex to implement.

In addition, to demonstrate the characteristics of the proposed ParallelCNN method under different probability of false alarm (Pfa), we compare the receiver operating characteristics (ROC) curves of the ParallelCNN method along with that of the benchmark methods in Fig. 4, and the SNR is set to -5dB. It can be observed that for the LTE dataset, the ParallelCNN method outperforms the DeepSense and TCN methods with a rather large gap. For the NR dataset, the ParallelCNN method is superior to DeepSense and TCN methods, with a probability of detection (Pd) gap of roughly 7% when $0.005 \leq$ Pfa ≤ 0.2.

C. SDR Dataset Results

Compared with the LTE dataset and NR dataset, the number of sub-bands in the SDR dataset is reduced to 4, which can make classification somewhat easier. However, as the SDR dataset is collected from real-world signals, it is subject to the variability and unpredictability of the wireless environment, including fluctuations in SNR. This means that the SNR cannot be fully controlled during data collection.

Fig. 5 suggests that ParalellCNN achieves near perfect performance on each channel, whereas the P_{F1} of DeepSense and TCN declines on the last channel. ParallelCNN achieves the highest P_{F1} of 99%, and both DeepSense and TCN achieve a lower P_{F1} of 97% on the last channel.

D. Real-Time Hardware Performance

The CNNs are implemented on an FPGA through High Level Synthesis, then run on a hardware platform of xc7z045ffg900-2 to test latency. Table II appears to provide

TABLE II
COMPLEXITY ANALYSIS OF DIFFERENT MODELS

Algorithms	Datasets	# of CNN Parameters	Latency (ms)	FPGA Utilization
DeepSense	LTE($N = 128$)	39472	6.14	4.9%
	LTE($N = 32$)	12272	0.79	4.3%
	SDR	15108	0.89	4.8%
	NR	39406	6.14	4.9%
TCN	LTE($N = 128$)	36608	9.24	7.3%
	LTE($N = 32$)	12032	2.35	7.2%
	SDR	5876	2.10	7.5%
	NR	32510	9.07	7.3%
ParallelCNN	LTE($N = 128$)	9696	0.56	4.4%
	SDR	8916	0.52	4.9%
	NR	16830	1.03	6.6%

information on the number of learnable parameters of the ParallelCNN, DeepSense, and TCN models, as well as their respective performances on FPGA for three different datasets.

Based on Table II, it can be noted that ParallelCNN has the lowest latency among the three models for all three datasets. For the LTE dataset and $N = 128$, the latency for ParallelCNN is 0.56ms, which is 91% and 94% lower than DeepSense and TCN, respectively. When the number of I/Q samples used in DeepSense and TCN is reduced to 32, the latency for ParallelCNN is still 29% and 76% lower than DeepSense and TCN, respectively. For the SDR dataset, the latency for ParallelCNN is 0.52ms, which is 41% and 75% lower than DeepSense and TCN, respectively. For the NR dataset, the latency for ParallelCNN, DeepSense and TCN is increased to 1.03ms, 6.14ms and 9.07ms respectively. Nevertheless, the latency for ParallelCNN is still much lower than the latency for DeepSense and TCN, with a reduction of up to 83% and 88%, respectively. In particular, the ParallelCNN models for the LTE and SDR datasets experience lower latency than the length of one LTE sub-frame, while for the NR dataset, the latency is similar to the length of one NR sub-frame. Overall, it can be concluded that ParallelCNN is more efficient for spectrum sensing tasks compared to DeepSense and TCN.

V. CONCLUSION

In this article, we propose a ParallelCNN-based approach for real-time wideband spectrum sensing. ParallelCNN relies on the parallel processing of I/Q samples divided into two parts by two CNNs to detect empty bands with the lowest latency and highest accuracy. The ParallelCNN is validated through three different datasets under various SNR conditions, and the real-time latency is measured with an FPGA implementation. Furthermore, the performance of the DeepSense and TCN is also evaluated for comparison purposes. Simulation results demonstrate that the ParallelCNN method can achieve the lowest latency among the three methods for all three datasets. Strikingly, compared to DeepSense, ParallelCNN can increase precision and recall by 25% and 22% respectively, while reducing latency by 29%; compared with TCN, ParallelCNN can increase precision and recall by 26% and 22% respectively, while reducing latency by 76%. As a consequence, ParallelCNN can be widely used in real-time wideband spectrum sensing tasks. However, due to the limitations of high-speed ADCs, the proposed ParallelCNN becomes prohibitively expensive when the frequency range is excessively wide, for example, a few GHz. We are currently working on extending the ParallelCNN architecture to sub-Nyquist sampling-based wideband spectrum sensing.

REFERENCES

[1] J. Lunden, V. Koivunen, and H. V. Poor, "Spectrum exploration and exploitation for cognitive radio: Recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 123–140, May 2015.

[2] L. Zhang, M. Xiao, G. Wu, M. Alam, Y.-C. Liang, and S. Li, "A survey of advanced techniques for spectrum sharing in 5G networks," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 44–51, Oct. 2017.

[3] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, p. 126, Jan. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/1/126

[4] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 116–130, 1st Quart., 2009.

[5] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1277–1304, 2nd Quart., 2017.

[6] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Wideband spectrum sensing in cognitive radio networks," in *Proc. IEEE Int. Conf. Commun.*, Mar. 2008, pp. 901–906.

[7] B. Farhang-Boroujeny, "Filter bank spectrum sensing for cognitive radios," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1801–1811, May 2008.

[8] H. Sun, W.-Y. Chiu, and A. Nallanathan, "Adaptive compressive spectrum sensing for wideband cognitive radios," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1812–1815, Nov. 2012.

[9] C.-P. Yen, Y. Tsai, and X. Wang, "Wideband spectrum sensing based on sub-Nyquist sampling," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3028–3040, Jun. 2013.

[10] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, Mar. 2019.

[11] J. Xie, J. Fang, C. Liu, and X. Li, "Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2196–2200, Oct. 2020.

[12] X. Ding, T. Ni, Y. Zou, and G. Zhang, "Deep learning for satellites based spectrum sensing systems: A low computational complexity perspective," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1366–1371, Jan. 2023.

[13] D. R. Rao, T. J. Prasad, and M. N. G. Prasad, "Deep learning based cooperative spectrum sensing with crowd sensors using data cleansing algorithm," in *Proc. Int. Conf. Edge Comput. Appl. (ICECAA)*, Oct. 2022, pp. 1276–1281.

[14] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1727–1730, Dec. 2019.

[15] D. Chew and A. B. Cooper, "Spectrum sensing in interference and noise using deep learning," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6.

[16] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[17] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "DeepSense: Fast wideband spectrum sensing through real-time in-the-loop deep learning," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[18] T. Ni, X. Ding, Y. Wang, J. Shen, L. Jiang, and G. Zhang, "Spectrum sensing via temporal convolutional network," *China Commun.*, vol. 18, no. 9, pp. 37–47, Sep. 2021.