A Fast Deep-Learning Signal Detection Architecture for the Short-wave Band

Keren Wang$^{1,*}$ and Weihao Li$^{1,2}$

$^1$National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu, China.
$^2$Information Engineering University, Zhengzhou, China

*Corresponding author e-mail: cfan662003@163.com

Abstract. Wideband signal detection plays an important role in wireless communication systems. In recent years, deep learning (DL) has been introduced, and many trial efforts have been posed. In this paper, a fast deep-learning signal detection architecture is proposed for the short-wave band, which consists of the GPU-based Down Digital Converter (DDC), the DL-based signal detector, and the signal data viewer & recorder. Wideband signal data are taken directly as the input, and detected signals are automatically recorded as the output of the architecture. Experimental results suggest that our architecture is capable of detecting signals in the whole short-wave band in quasi-real-time, and specifically, with single GPU device.

1. Introduction

Spectrum resources are becoming scarce due to the increasing need for spectral bandwidth and number of users. Aiming at relieve the problem, cognitive radio (CR) technology has attracted much attention [1, 2]. Wideband signal detection is a critical technology in most wireless communication systems and has been identified as one of the most challenging problems in the CR technology applications [4, 5, 6].

Various methods have been proposed to detect signal in the wideband, including energy detection (ED) [1, 7], eigenvalue-based spectrum sensing methods [8, 9], goodness-of-fit (GoF) test-based methods [10, 11, 12], student’s t distribution test-based methods, etc. Those unsupervised methods work well in certain cases, while suffer that the channel model cannot be easily estimated and detection parameters are quite difficult to decide. In recent years, deep learning technology has been introduced into signal detection. Jeon [14] proposed a blind detection method for Multi-Input Multi-Output (MIMO) systems using supervised learning. Ye [15] introduced deep learning into signal detection in Orthogonal Frequency Division Multiplexing (OFDM) systems. Lin proposed a Deep Neural Network (DNN)-based algorithm for signal detection in MIMO systems. Those deep-learning-based methods achieve favorable results while contain just a few network parameters to tune. It suggest that deep learning is a potential good tool for signal detection.

Aiming at detect signal-of-interest directly on the wide-band of shortwave in quasi-real-time, we resort to deep-learning-based object detection methods. Among those methods, Fully Convolutional Network (FCN) [16] is a classical one, which is quite simple and shallow and is possible for quasi-real-time signal detection applications. Thus in this paper, the Fast Deep-Learning Signal Detection (FDLSD) Architecture is proposed, combining the technology of GPU-based DDC and deep-learning-based signal detector. Section II describes the detail structure of FDLSD architecture, followed by various experiments evaluating mainly the time performance. The conclusion is drawn in the last section.
2. FDLSD Architecture

Figure 1 depicts the FDLSD system architecture. FDLSD takes the wide-band signal data as the input, followed by the GPU-based DDC & spectrogram calculator, which simultaneously calculates the spectrogram and implements DDC. In our architecture, the output of DDC consists of almost 29996 narrow-band channels, each of which owns the bandwidth of 4812Hz. Then, the quasi-real-time DL-based signal detector, which takes directly the wide-band spectrogram as the input, is able to detect signals of interested kinds. Appearing/disappearing signal frequencies are transferred to the signal data viewer & recorder, which starts/ stops recording of corresponding narrow-band signals.

2.1. GPU-based Digital Down Converter

As shown in figure 1, the spectrogram is the input for DL-based signal detector, while narrow-band signal data is required for signal data viewer & recorder. Thus it is intuitive to implement simultaneously the spectrogram calculator and DDC on GPU.

Figure 2 shows the processing of our GPU-based DDC, where the prefix “kernel” indicates the following computation is implemented in GPU. Firstly, 20-order FFT is carried out in the GPU on the input wide-band signal data with a sample rate of 40M, resulting in FFT data with the frequency resolution of 38.147Hz. Phase compensation is employed to ensure the continuity of the signal data for each narrow-band channel. Finally, IFFT on each narrow-band channel is computed to generate signal data of all narrow-band channels, which can be transmitted using TCP channel to any other clients, e.g., the DL-based signal detector and the signal data viewer & recorder as shown in figure 1.
2.2. DL-based Signal Detector

In most cases, the spectrogram contains enough information for signal detection and recognition. Thus here the spectrogram of the wide-band signal, collected per several seconds, is taken directly as the input of the neural network. Nevertheless, since the GPU memory is limited, the wide-band spectrogram has to be split along the time axis or the frequency axis into several small images, which are actually the input into the neural network.

Considering both the detection accuracy and the processing speed, we introduce FCN [16] here for detecting signal-of-interest. The network structure of FCN model is outlined in figure 4, where each Conv2d block is implemented by a bottleneck block of ResNet, and each DeConv2d block is implemented by a simple up-sampling block. For each Conv2d / DeConv2d block, batch normalization[17] and ReLu activation function [18] are utilized. Hidden sizes for Con2d/ DeConv2d blocks are all tunable parameters. Similar as in the domain of object detection [16], hidden sizes of blocks in upper layers are mostly smaller than lower blocks. Furthermore, the output of FCN is a relatively small image, whose size is 1/4 of the input image size.

![Figure 4. Network structure of FCN.](image)

Signal appearing in the short wave band mostly has a bandwidth smaller than 4kHz (except for several kinds of signal like the frequency hopping signal). Thus when the frequency resolution is nearly 40Hz, we can easily detect most signals in 50 pixels (i.e., half of the bandwidth) along the frequency axis. Our FCN has a reception of almost 79 pixels, which is considered enough for the task of signal detection.

Furthermore, due to the shallowness of the FCN, the inference step runs quite fast on GPU, which makes it possible to detect continuous signals of the whole wide-band spectrogram as fast as the collecting speed. Detailed experimental results are shown in Section 3.

Training of FCN is somehow different to the inference. The input images consist of the spectrogram image $S_{i,j}, 0 \leq i \leq I, 0 \leq j \leq J$ and the mask image $M_{i,j,c}, 0 \leq i \leq I/4, 0 \leq j \leq J/4, 0 \leq c \leq C$, where the latter has 0 or 1 value. Intuitively, Value 1 in the mask image indicates that signal appears at the time-frequency point, while channels of the mask image indicates various kinds of signal-of-interest. For example, when audio and Morse are two focused signal types, $M_{i,j,1}$ implies that audio appears at time $i$ and frequency $j$, while $M_{i,j,3}=1$ implies that Morse appears at time $i$ and $j$. The training target of FCN is just to minimize the cross-entropy between the mask image and the output image $O_{i,j,c}$, as shown in the following formulation.

$$\min_{i,j,c} - \sum M_{i,j,c} \log O_{i,j,c}$$

(1)
In the inference step, a averaging operation is carried out on the output image $O_{t,i,c}$ along the time axis $i$, resulting in a new matrix $A_{j,c}$. Each element of $A_{j,c}$ indicates that signal of type $c$ exists at the frequency index $j$. Furthermore, the value of each element varies from 0 to 1, indicating the confidence that signal-of-interest appears at the frequency.

An additional simple appearing/disappearing module is employed to generate results of three types. The first case is that a signal channel is newly detected at the detected frequency; the second case is that a previously detected signal still exists; while the last case is that a previously detected signal disappears.

2.3. Signal Data Viewer & Recorder

As shown in figure 5, the signal data viewer & recorder mainly focuses on the inspection of the spectrogram and the recording of the signal data. Two subscribers receive the spectrogram & signal data and the active signal frequency, separately. Two spectrogram viewers are designed to show the wide-band spectrogram and two channels of narrow-band spectrogram, separately. The signal data recorder is just responsible for recording signal data to the disk.

![Figure 5. Signal data viewer & recorder.](image)

3. Experimental Results and Analysis

Since FDLSD architecture is proposed mainly to detect short-wave signals on the wide band, our experiments here focuses on the time efficiency. Additionally, just one GPU device is utilized, which suggests that we also need to give out the memory usage for our architecture. In FDLSD architecture, DDC is a quite critical step, where the FFT points indicates the frequency resolution. Thus our experiments covers different setups both for DDC and signal detection.

3.1. Experimental Setup

**Machine configuration:** Since the efficiency of the FDLSD architecture mainly depends on the GPU, other than the CPU, we take Nvidia Tesla V100 16GB as our GPU device. The CPU device is Intel Xeon CPU E5-2667, and the memory size is 64GB.

**Receiver:** Considering that the bandwidth of short-wave is almost 30MHz, here we choose WiNRADiO G35DDC HF receiver [19] as our receiver. The output of “DDC1” in the receiver is taken as our wide-band signal input, which has a sample rate of 40M, and a bandwidth of 32MHz.

**DDC setups:** Since the DDC setup has a critical effect on the speed of DDC, we evaluate following four setups for our short-wave signal detection experiments.

| Setup Id | FFT Order | Frequency Resolution | Overlap Ratio | Equivalent Hop Time |
|----------|-----------|----------------------|---------------|---------------------|
| 1        | 20        | 38.15Hz              | 61.54%        | 0.01s               |
| 2        | 20        | 38.15Hz              | 23.71%        | 0.02s               |
| 3        | 19        | 76.29Hz              | 61.54%        | 0.005s              |
| 4        | 19        | 76.29Hz              | 23.71%        | 0.01s               |
Input image size: Signal detection is carried out per 5 seconds, thus the first dimension of the image size is $2^{\log_{10}(5/0.01)} = 512$. The second dimension of the image size is just the FFT points of 30MHz, i.e. $\frac{3}{4} \cdot N_{\text{fft}}$.

Setups for splitting the wide-band spectrogram: Restricted by the GPU memory, we have to split the wide-band spectrogram image into smaller images. Considering the expensive cost of copying data between GPU and CPU, it is favourable to input as many small images as possible, which implicitly makes use of GPUS’s data parallel capability.

Four cases are evaluated, as demonstrated in table 2. Note that the iteration indicates the number of input for FCN, the batch size indicates the number of image per input, while the image size indicates the actual input size per sample for FCN. As shown in the table, only setups for DDC setup 1 is considered, of which the reason is given in the next sub-section.

| Setup Id | Iteration | Batch Size | Image Size |
|----------|-----------|------------|------------|
| 1        | 12        | 512        | 512*128    |
| 2        | 24        | 256        | 512*128    |
| 3        | 12        | 256        | 512*256    |
| 4        | 6         | 512        | 512*256    |

Model setups: For FCN, we use a simple group of hidden sizes for the evaluation. As shown in figure 4, hidden sizes for left-half blocks are 64, 16, 32, 64, 128, separately, which are from up to down. For the right-half blocks, hidden sizes are 64, 32, 16, which are from down to up.

3.2. Experimental Results for Different DDC Setups
The GPU memory usage, time elapsed for processing 1-second signal data and the real-time capability of DDC are demonstrated in table 3, where each value is averaged through 20 tests. It is obvious that in four tested cases, data from the receiver can be processed in real-time. Intuitively, the elapsed time of setup 1 is similar as setup 3, while setup 2 is similar as setup 4.

Note that time-domain data of 29996 narrow-band channels are all calculated, which can then be transmitted to the signal viewer & recorder if necessary.

Memory usages of different setups differ less than 150MB, which is quite smaller (less than 1%) to the total 16GB GPU memory. Since the post-processing of DDC’s results also costs much time, including copying data and transmitting data, we apply DDC setup 1 for following experiments.

| DDC Setup Id | Memory Usage (MB) | Time elapsed for DDC (s) | Real-time Capability |
|--------------|-------------------|--------------------------|----------------------|
| 1            | 706               | 0.472                    | Yes                  |
| 2            | 706               | 0.282                    | Yes                  |
| 3            | 586               | 0.485                    | Yes                  |
| 4            | 586               | 0.296                    | Yes                  |

3.3. Experimental Results for Different Image Splitting Setups
The GPU memory usage, time elapsed for detecting signal from spectrogram of every 5.12 seconds, and the real-time capability are given in table 4, where each value is averaged through 20 tests. In four tested cases, wide-band spectrogram with bandwidth of 30MHz can be processed in real-time. In fact, elapsed times for four cases differ quite small.

Obviously, the memory usage is quite correlated to the batch size and the image size. In Setup 4, the GPU memory is almost fully utilized, which may lead to instability in the run time. Thus, Setup 1, 2, and 3 are more applicable for real-world applications.
5. Conclusion

In this work a fast architecture is proposed for detecting signals in short-wave band, which consists of the GPU-based DDC, DL-based signal detector, and the signal data viewer & recorder. Experiments are carried out for several DDC setups and spectrogram splitting setups, which suggests that our architecture is able to detect signals of the whole short-wave band in quasi-real-time, with single GPU device. We are researching on DL models more suitable for signal detection, and plan to further improve the detection accuracy, while retaining the quasi-real-time capability.

5. References

[1] Khan A A, Rehmani M H and Reisslein M 2016 IEEE Commun. Surv. Tutorials. 18(1) 860-898
[2] Prasad J G, Yeob N S, and Won K S 2013 Sensors. 13(9) 11196-11228
[3] Men S, Charge P, Wang Y, I and Li J 2019 EURASIP journal on advances in signal processing. 2019(1) 2
[4] Sun H, Nallanathan A, Wang C X and Chen Y 2013 IEEE Wirel. Commun. 20(2) 74-81
[5] Ma Y, Gao Y, Liang Y C and Cui S 2016 IEEE J. Sel. Areas Commun. 34(10) 2750-2762
[6] Bogale T E, Vandendorpe L and Le B 2015 IEEE Trans. Commun. 63(4) 1091-1105
[7] Urkowitz H 2005 Proc. IEEE. 55(4) 523-531
[8] Sharma S K, Chatzinotas S and Ottersten B 2013 IEEE Trans. Veh. Technol. 62(8) 3671-3684
[9] Lin F, Qiu R C, Hu Z, Hou S, Browning J P and Wicks M C 2012 IEEE Commun. Lett. 16(5) 604-607
[10] Denkovski D, Atanasovski V and Gavrilovska L 2012 IEEE Commun. Lett. 16(3) 310-313
[11] Shen L, Wang H, Zhang W and Zhao Z 2011 IEEE Trans. Wirel. Commun. 10(6) 1721-1724
[12] Patel D K and Trivedi Y N 2015 Electron. Lett. 51(5) 419-421
[13] Xu Y, Li D and Wang Z 2017 Proc. MLICOM vol 226 (Cham: Springer) pp 373-381
[14] Jeon Y, Hong S N and Lee N 2017 Proc. IEEE Commun.(Paris: IEEE) pp 1-6
[15] Wang C 2018 Proc. ICRIS2018 (Changsha: IEEE) pp 150-152
[16] Long J Shellhammer E and Darrell T 2015 Proc. Int. Conf. on Computer Vision and Pattern Recognition (Boston: IEEE) pp 3431-3440
[17] Loffe S and Szegedy C 2015 Int. Conf. on Machine Learning (Lille: ACM) pp 3431-3440
[18] Glorot X, Bordes A and Bengio Y 2011 Proc. Int. Conf. on Artificial Intelligence and Statistic (Fort Lauderdale: PMLR) pp 315-323
[19] Programming Information for WINRADIO G35DDC HF Receiver in Coherent Mode URL: http://www.winradio.com/home/g35ddc_coh_sdk.htm

| Table 4. Experimental results for different splitting setups. |
|-----------------|-----------------|-----------------|-----------------|
| Splitting Setup Id | Memory Usage (MB) | Time elapsed (s) | Real-time Capability |
| 1 | 9228 | 2.791 | Yes |
| 2 | 9042 | 2.831 | Yes |
| 3 | 9106 | 2.711 | Yes |
| 4 | 15204 | 2.854 | Yes |