Spectrum Sensing Method Based on Residual Cellular Network

JIANXIN GAI¹, XIANFENG XUE¹, ZHEN LI², and LINGHUI ZHANG¹

¹The Higher Educational Key Laboratory for Measuring & Control Technology and Instrumentations of Heilongjiang Province, Harbin University of Science and Technology, Harbin 150080, China (e-mail: jxgai@hrbust.edu.cn)
²School of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China. (e-mail: lizh0019@gmail.com)

Corresponding author: Jianxin Gai (e-mail: jxgai@hrbust.edu.cn).

This work is supported by National Natural Science Foundation of China (No.61501150), Natural Science Foundation of Heilongjiang Province (No.QC2014C074), and Fundamental Research Funds for the Universities in Heilongjiang Province (No.2018-KYYWF-1656).

ABSTRACT

The traditional spectrum sensing method based on convolutional neural network (CNN) has the single-branch convolutional network structure and the shallow network structure which limits the ability of extracting the Primary User (PU) feature. Aiming at these problems, a spectrum sensing method based on the residual cellular network (ResCelNet) is proposed in this work, involving a structure of "dual-branch convolution plus summation operation plus residual learning plus dual-branch convolution plus summation operation". Specifically, the dual-branch convolution improves the feature extraction ability, the addition operation enhances the micro-feature information, and the residual learning is adopted to facilitate training the deep spectrum sensing network. This method transforms the spectrum sensing problem into the image binary classification problem. Firstly, the received signals are reshaped into a matrix and normalized to gray levels, which is used as the input of the network. Secondly, the feature information of gray-scale images is extracted and the network is trained through dual-branch convolution and residual learning. Finally, the test data is input into the trained model and spectrum sensing based on image classification is implemented. The experimental results demonstrate that the proposed method exhibits a higher detection probability and a lower false alarm probability as well as better generalization ability as compared with the traditional methods under low signal to noise ratio (SNR) circumstance.

INDEX TERMS

Spectrum sensing, residual cellular network, dual-branch convolution, residual learning.

I. INTRODUCTION

Spectrum sensing is of vital importance to realize cognitive functions in the area of cognitive radio [1]. Conventional spectrum sensing methods can be broadly categorized into Energy Detection [2], Cyclostationary Feature Detection [3], Matched Filter Detection [4], and Eigenvalue Detection [5]. As a common issue of these methods, a pre-defined threshold is calculated theoretically based on a White Gaussian Noise (WGN) assumption of the noise distribution. However, due to the fact that the noises in practical environments are not necessarily following WGN, the pre-defined threshold is neither accurate enough nor dynamically adaptive to the volatility of the complicated circumstance.

On the contrary, modern methods [6, 7, 8, 9, 10] mainly focus on machine learning based solutions that circumvent the presumed threshold of WGN distribution to achieve a reasonable Primary User (PU) signal recognition rate, with high-dimensional feature representation and higher discriminative power. Support Vector Machine (SVM) by Awe [6], Bao [7], and Chen [8] is an early foray in the area of machine learning for spectrum sensing. In their SVM-based methods, the PU signal recognition is achieved by training SVM classifiers based on the energy vectors sampled from Secondary User (SU). However, SVM is time-consuming due to the fact that the optimal decision boundary computation is of cubic complexity in the training phase and the online detection is also not real-time. Thus, the performance of the SVM group is not able to satisfy the real-time requirements although the performance is reasonable compared to conventional methods. Whereas these conventional non-deep-learning methods heavily rely on handcrafted features, the feature representations are not jointly learned with neural network weights. Thus the solution is not optimal. Enlightened by the popular Artificial Neural Network (ANN) applications in computer vision tasks, Tang [9] and Vyas [10] proposed ANN-based spectrum sensing methods, where the optimal PU signal de-
tection model is trained using signal energy features together with cyclostationary features via back-propagating and updating through ANN weights and biases. A common issue of this kind of methods is the vanishing gradient problem when the ANN structure is deep.

In view of the above-mentioned issues, we propose a residual cellular network called ResCelNet, which produces representative and discriminative feature extraction for PUs. The ResCelNet network is carefully designed so that its dual-branch convolution is able to enhance the feature representation. In addition, a residual deep structure is also incorporated to alleviate the gradient vanishing problem. Analyses and experimental results demonstrate that the proposed method produces remarkably higher detection probability at reasonably low SNR and low false-positive rate, and exhibits significantly better generalization under various noise conditions.

The remainder of this paper is organized as follows. In Section 2, we review the related works. The proposed ResCelNet is elaborated in Section 3, followed by the extensive experiments in Section 4 which validates the superiority of ResCelNet in the balance of accuracy and resource requirement. Finally, we conclude this work in Section 5.

The main contributions of our paper are summarized as follows:

- We proposed a novel spectrum sensing algorithm based on residual cellular network.
- We presented a novel way to combine multiple-branch structure and shortcut connection to learn better PU feature embedding.
- Our method achieves balance between computational complexity and accuracy, outperforming the state-of-the-art methods in real scenarios.

II. RELATED WORK

In general, existing spectrum sensing modeling can be categorized into traditional statistics methods and neural network methods. Traditional statistical methods utilize only binary hypothetical tests, typically with a constant threshold of likelihood by assuming a distribution of the signal and noise. In contrast, neural network models typically learn various discriminative features to classify signals accurately and dynamically.

A. TRADITIONAL METHODS

In a cognitive network, whether the PU signal is detected by the SU can be represented in a binary hypothetical test problem:

\[
\begin{align*}
\mathcal{H}_0: x(n) &= u(n) \\
\mathcal{H}_1: x(n) &= s(n) + u(n)
\end{align*}
\]

where \( \mathcal{H}_0 \) represents absence of PU signal, \( \mathcal{H}_1 \) represents occupancy of PU signal, \( n \) is the time series of the discrete signal, \( x(n) \) is the received signal, \( s(n) \) is the emitted signal from the PU through the Rayleigh fading channel, and \( u(n) \) represents the noise following the Gaussian distribution with mean value 0 and standard deviation \( \sigma \).

The hypothetical test problem can be naturally regarded as a classification problem of two classes \( \mathcal{H}_0 \) and \( \mathcal{H}_1 \): \( \mathcal{H}_0 \) is merely noise, and \( \mathcal{H}_1 \) contains both noise and the signal emitted from PUs. In order to evaluate the performance, detection probability \( P_d \) and false alarm probability \( P_f \) are commonly used as evaluation criteria, where \( P_d \) indicates the successful detection rate with the presence of the PU, while \( P_f \) indicates the false detection of a PU with absence of the PU. The hypothetical test classification problem can be formulated as:

\[
\begin{align*}
P_d &= P\{\mathcal{H}_1 | \mathcal{H}_1\} \\
P_f &= P\{\mathcal{H}_0 | \mathcal{H}_0\}
\end{align*}
\] (2)

The SU receives a one-dimensional signal vector of \( N \) samples \( x(n) = [x(1), x(2), \ldots, x(N)] \), which will be transformed into two-dimensional gray images by data preprocessing in subsection III-B. Since the two-dimensional gray images of the decomposed signal matrices are essentially equivalent to the original one-dimensional signal, the hypothetical test problem can be further converted to image classification problem.

B. NEURAL NETWORK METHODS

Convolutional Neural Network (CNN) is widely used in image feature extraction and classification due to its multi-layer structure and powerful end-to-end feature learning capability. Chen et al. [11] proposed a spectrum sensing method based on Short-Time Fourier Transform (STFT) and CNN, in which the spectrogram matrix is normalized to gray levels before input to CNN, followed by convolutional feature extraction and classification to identify the presence of PU signals. In this method, STFT is essential to convert the PU signals to spectrogram matrices datasets which is fed into the CNN training process, rendering generalization of communication scenarios due to its independence to various modulation mechanisms used by PUs. Liu et al. [12] proposed a spectrum sensing method based on covariance matrix and CNN, where the input signals are transformed into covariance matrices as the input to CNN and then a CNN classification model is trained on a dataset of PU signal matrices and Gaussian white noise matrices. In this method, the prior information of PUs is not required in the process of transforming the original signal to the covariance matrices, resulting in good generalization of the classification model. Zhang et al. [13] proposed a novel spectrum sensing method using Orthogonal Frequency Division Multiplexing (OFDM) and CNN, where the cyclic auto-correlation of the OFDM signal is normalized to gray levels as input to CNN, followed by the cyclic auto-correlation feature extraction through the LeNet-5 network, and finally robust spectrum sensing is achieved even in low signal to noise ratio (SNR) scenarios. The above-mentioned CNN methods achieved reasonable performance compared to traditional hypothetical test methods by improving the gener-
alization and robustness. However, the discriminate capability is still limited. Due to the vanishing gradient problem, these CNN network structures typically consist of a single branch network with no more than 5 layers.

In view of the limited learning capability of the above-mentioned CNN methods, multiple-branch architectures and deep networks are specially designed to improve performance while alleviating the vanishing gradient problem.

The main idea of multiple-branch structure originates from the inception module design in GoogleLeNet [14], where the convolutional feature maps coming from multiple branches are concatenated as input feature map of the inception module in next layer. The feature discriminability is improved by increasing the depth and width of the GoogleLeNet network. Li et al. [15] proposed a dual-branch fusion method based on hyper-spectal imagery. In this method, global and local feature maps are extracted from the two network branches, respectively, and the classification layer is trained based on the concatenated feature vector. Zhao et al. [16] proposed a single-frame image super-resolution method based on multi-scale residual fusion, where three scales of image features are extracted from three network branches, respectively, before the fusion features are input to the final classifier to achieve a super-resolution image. Although this method is not directly applicable to spectrum sensing tasks, it sheds light on the future direction of CNN-based spectrum sensing. The above-mentioned methods have proven its superiority of discriminative feature extraction of multiple-branch CNN network structures.

A popular idea of ameliorating the vanishing gradient problem in deep network is shortcut connection which was originally proposed in the Hopfield Network [17]. The main idea is connecting a shortcut between any two neurons, enabling the gradient propagation to any neuron to accelerate information transmission. In [18], the shortcut connection is also utilized in multi-layer perceptron (MLP) to alleviate the vanishing gradient problem. Shortcut connection has proven its superiority by increasing the depth of CNN to extract more discriminative feature for PUs, which is orthogonal to the above-mentioned multiple-branch approach.

III. SPECTRUM SENSING BASED ON RESCELNET

A. FRAMEWORK

Traditional CNN-based spectrum sensing methods adopt the canonical LeNet-5 network structure, which takes input of the gray-level images and passes them through a series of operations including convolution, pooling, rectified linear units (ReLU), fully connection (FC) and Softmax, and finally classifies signals into \( \mathcal{H}_0 \) and \( \mathcal{H}_1 \). The main drawback of this kind of methods is the limited learning ability due to the single branch structure and shallow network caused by the vanishing gradient problem.

In view of these problems, a new approach is proposed in this work, in which the observed vector from the received signal is segmented into signal matrices and then normalized to gray levels as input to the proposed ResCelNet network. Offline dataset is created to train the ResCelNet model and then online dataset can be inferred by the trained model. The main flow of the proposed approach is shown in Fig. 1.

B. DATA PREPROCESSING

The received one-dimensional signal can be decomposed into matrices \( X_I \) and \( X_Q \) and transformed into signal matrix gray image [11]. Specifically, the received signal goes through orthogonal sampling, low-pass filtering and sampling decision, producing two signals \( I \) and \( Q \) which serves for the data source of spectrum sensing models:

\[
X(n) = I(n) + jQ(n)
\]

where \( I(n) \) and \( Q(n) \) are the \( N \)-time sampled signals received by SUs. Then they are reshaped into \( M \) segments with the length of each segment being \( N/M \), represented as follows:

\[
X_I = \begin{bmatrix}
I(1) & I(2) & \cdots & I(N) \\
\frac{N}{M} + 1 & \frac{N}{M} + 2 & \cdots & \frac{2N}{M} \\
\vdots & \vdots & & \vdots \\
\frac{(M-1)N}{M} + 1 & \frac{(M-1)N}{M} + 2 & \cdots & I(N)
\end{bmatrix}
\]

\[
X_Q = \begin{bmatrix}
Q(1) & Q(2) & \cdots & Q(N) \\
\frac{N}{M} + 1 & \frac{N}{M} + 2 & \cdots & \frac{2N}{M} \\
\vdots & \vdots & & \vdots \\
\frac{(M-1)N}{M} + 1 & \frac{(M-1)N}{M} + 2 & \cdots & Q(N)
\end{bmatrix}
\]

where \( M \) and \( N/M \) represents the number of rows and columns of signal matrices \( I \) and \( Q \), respectively.

The matrices \( X_{I,M,N/M} \) and \( X_{Q,M,N/M} \) are then normalized to gray levels according to [16], and used as input of the ResCelNet spectrum sensing model.

C. SHORTCUT CONNECTION

In traditional CNN-based spectrum sensing methods, each convolution layer consists of a series of non-linear functions including \( BN, ReLU, Pooling \) and \( Conv \), represented as:

\[
x_l = H_l(x_{l-1}), l = 1, 2, \cdots, L
\]

where \( x_l \) represents the input of the \( l \)-th layer which is the output of the \( (l-1) \)-th layer.

Traditional spectrum sensing methods based on simply stacked convolution layers cause severe vanishing gradient problem upon updating neural network weights especially when the network is too deep. The difficulty lies in the fact that back-propagation applies chain-rule differentiation to calculate the gradient, resulting in near-zero values in shallow layers and thus the neural network weights are disabled to learn any more.

When introducing shortcut connections to traditional CNN network, one or more convolution layers can be skipped and the information flow is altered, as shown in Fig. 2, where the output feature map \( x_{l-1} \) is directly transmitted to the input.
feature map $x_l$ via shortcut connection, and $F_l(\cdot)$ represents the residual feature map between $x_{l-1}$ and $x_l$. The feature map $x_l$ which is supposed to serve as input of the $l$-th layer in traditional CNN methods is now substituted with $F_l(x_{l-1}) + x_{l-1}$. Intuitively, since the residual feature map $F_l(x_{l-1})$ has lower variance compared to $x_{l-1}$, the learning procedure converges faster to reach optimal network weights.

**D. DUAL-BRANCH CONVOLUTION**

Traditional CNN-based spectrum sensing methods adopts single branch architecture to extract PU features. However, due to the fact that global optimum of neural network is hard to reach, the single branch feature map learned is only a local optimal solution. By using multiple-branch architecture, multiple independent local optimal feature maps from different branches can be integrated to achieve a better estimation of the global optimal solution.

In this work, a dual-branch architecture is proposed, as shown in Fig. 3, where the feature maps from branches $a$ and $b$ are integrated via the summation function, producing the feature map of the PU. The main reason that the summation function is adopted is the feature maps can be integrated no matter the signal intensity is strong or weak while the original gray image dimension is preserved.

The dual-branch summation process is represented as follows:

$$H(x^{a,b}) = \sum_{j=1}^{n} (x^a_j + x^b_j) \ast K_i$$

$$= \sum_{j=1}^{n} x^a_j \ast K_i + \sum_{j=1}^{n} x^b_j \ast K_i$$

where $\sum_{j=1}^{n} x^a_j$ and $\sum_{j=1}^{n} x^b_j$ represents the input data from the two branches, $K_i$ represents convolution kernel, $\ast$ represents convolution, and $H(x^{a,b})$ represents the integrated feature map after the dual-branch convolution.

**E. RESIDUAL CELLULAR BLOCK**

In this work we not only integrates convolutional feature maps from dual branches but also introduces shortcut connection within each ResCelNet module. The two enhancements intertwine together and improve the feature extraction ability significantly compared to the plain CNN architecture. The unique residual cellular block (RCB) network architecture is proposed in this work, as shown in Fig. 4, where the dual-branch structure, shortcut connection and summation operations are all involved.

In a RCB block, the two feature maps are extracted through the dual-branch structure, and then integrated using the summation function, producing the feature map of the PU. The main reason that the summation function is adopted is the feature maps can be integrated no matter the signal intensity is strong or weak while the original gray image dimension is preserved.

The dual-branch summation process is represented as follows:

$$x_2 = H(x_1)$$

where $H(\cdot)$ represents the non-linear functions of the corresponding layer, including convolution, batch normalization
Conv4 takes $x_2$ as well as $x_{4a}$, i.e. a non-linear mapping of the summation of $x_2$ and $x_3$, as input, thus the residual learning is achieved by learning $F(x^{2,3}) := x_{4,a} - x_2$ and $F(x^{2,3}) := x_{4,b} - x_3$. Conv5 and Conv6 impose new non-linear mappings on $x_{4,a} = F(x^{2,3}) + x_2$ and $x_{4,b} = F(x^{2,3}) + x_3$ as follows:

$$\begin{align*}
x_5 &= H(F(x^{2,3}) + x_2) \\
x_6 &= H(F(x^{2,3}) + x_3)
\end{align*}$$

Through RCB, the original gray image is transformed to:

$$x_7 = F(x^{5,6})$$

The "dual-branch convolution plus summation operation plus residual learning plus dual-branch convolution plus summation operation" of the RCB block combines the superiority of multiple-branch structure and shortcut connection together, and greatly ameliorate the vanishing gradient problem that inhibits the feature learning capability of the traditional CNN structures.

F. PARAMETERS OF RESCELNET ARCHITECTURE

The RCB block is cascaded to extract shallow to deep feature representation gradually. After the first RCB block, the gray level is converted to a shallow feature map $F_0$ as follows:

$$F_0 = H(W * x^{(i)})$$

where $x^{(i)}$ is the $i$-th input gray level image, $W$ is the neural network weights to extract features, and $H(\cdot)$ represents the non-linear functions of the current block.

$F_0$ is fed into cascaded RCB blocks to produce an intermediate feature representation $F_d = H_{RCB,d}(F_{d-1})$.

G. LOSS FUNCTION

Based on a standard backbone of deep neural network such as LeNet-5, the loss function can be designed to optimize the CNN weights. A classification loss is an early choice [19, 20], and contrastive loss is an alternative as well [21]. Many works [19, 20, 22] consider a multi-task objective functions to update the CNN weights. These approaches train a deep CNN...
TABLE 1: ResCelNet architecture parameters.

| Block | Function | Kernel | Padding | Stride | Output |
|-------|----------|--------|---------|--------|--------|
| Conv 40 × 40 | Conv | 80 (3 × 3) | 0 | 1 | 38 × 38 |
| | Conv (1, 1) | 80 (5 × 5) | 1 | 1 | 36 × 36 |
| | Conv (1, 2) | 80 (3 × 3) | 0 | 1 | 36 × 36 |
| | Conv (1, 3) | 80 (1 × 1) | 0 | 1 | 36 × 36 |
| | Conv (1, 4) | 80 (5 × 5) | 1 | 1 | 34 × 34 |
| | Conv (1, 5) | 80 (3 × 3) | 0 | 1 | 34 × 34 |
| | Conv (1, 6) | 80 (1 × 1) | 0 | 1 | 34 × 34 |
| maxpool | 2 × 2 | 0 | 2 | 17 × 17 |
| Conv (2, 1) | 80 (5 × 5) | 1 | 1 | 15 × 15 |
| | Conv (2, 2) | 80 (3 × 3) | 0 | 1 | 15 × 15 |
| | Conv (2, 3) | 80 (1 × 1) | 0 | 1 | 15 × 15 |
| Conv (2, 4) | 80 (5 × 5) | 1 | 1 | 13 × 13 |
| | Conv (2, 5) | 80 (3 × 3) | 0 | 1 | 13 × 13 |
| | Conv (2, 6) | 80 (1 × 1) | 0 | 1 | 13 × 13 |
| maxpool | 2 × 2 | 0 | 1 | 12 × 12 |
| Conv (3, 1) | 80 (5 × 5) | 1 | 1 | 10 × 10 |
| | Conv (3, 2) | 80 (3 × 3) | 0 | 1 | 10 × 10 |
| | Conv (3, 3) | 80 (1 × 1) | 0 | 1 | 10 × 10 |
| Conv (3, 4) | 80 (5 × 5) | 1 | 1 | 8 × 8 |
| | Conv (3, 5) | 80 (3 × 3) | 0 | 1 | 8 × 8 |
| | Conv (3, 6) | 80 (1 × 1) | 0 | 1 | 8 × 8 |
| maxpool | 2 × 2 | 0 | 2 | 4 × 4 |
| Output | FC | | | | 2 |

with the cross-entropy loss or integrate it with other losses, and then use the network as a feature extractor, followed by combining it with a metric learning approach to produce final feature embedding and classification results. However, the cross-entropy loss has become the de facto standard of the loss function in spectrum sensing deep learning networks due to the nature of binary classification of the hypothetical test problem.

The cross-entropy loss is implemented with Softmax function which is defined as:

\[ L_{\text{softmax}}(W, x) = -\sum_{i=1}^{2} \ln \left( \frac{e^{W_{yi}x_i + b_{yi}}}{\sum_{j=1}^{2} e^{W_{yj}x_j + b_{yj}}} \right) \]  

where \( W \) represents the \( 1 \times 1 \) convolution weights for Softmax function and \( b \) denotes the bias.

We use Adam optimizer [23] and parameters are set as \( \beta_1 = 0.9 \) within 150 epochs and \( \beta_2 = 0.5 \) for remaining epochs, and \( \beta_2 = 0.999 \). We adjust the learning rate training schedule as proposed by [24]:

\[ \alpha(e) = \begin{cases} \alpha_0 & \text{if } e \leq e_0 \\ \alpha_0 \times 0.001^{\frac{e - e_0}{e_0}} & \text{else} \end{cases} \]  

where we set \( \alpha_0 = 3 \times 10^{-4} \), \( e_0 = 150 \) epochs. Batch size in the training process is set to be 128.

H. ALGORITHM OF RESCELNET NETWORK LEARNING

The datasets for ResCelNet algorithm includes the offline dataset \{\( x^{(m)}(1), y^{(m)}(1) \), \( x^{(m)}(2), y^{(m)}(2) \), \ldots, \( x^{(m)}(n), y^{(m)}(n) \)\} and online dataset \{\( x^{(m+1)}, y^{(m+1)} \), \( x^{(m+2)}, y^{(m+2)} \), \ldots, \( x^{(m+n)}, y^{(m+n)} \)\}, where \( x^{(*)} \) is the received signal matrix and \( y^{(*)} \) is the ground-truth classification label of the received signal.

In this work, the mapping from input to output of the proposed ResCelNet algorithm is formulated as follows:

\[ F_{RCB}(x^{(i)}, [W]) = \hat{y}^{(i)} \cong y^{(i)} \]  

Algorithm 1: Spectrum Sensing Algorithm of ResCelNet.

Input: Offline and online datasets.
Output: Detection probability \( P_d \) and false-alarm probability \( P_{fa} \).
Initialization: Gaussian initialized weights \( W \) and maximum iteration number \( \text{IterMax} \).
Offline training:
Repeat
Compute first layer feature map according to Eq. 11;
Compute intermediate layer feature maps according to Eq. 12;
Compute final layer feature map according to Eq. 13;
The predicted label \( \hat{y}^{(i)} \) is updated according to Eq. 16;
Back-propagate to update CNN weights for the feed-forward loss \( L \) according to Eq. 14;
Save the model with lowest loss \( L \);
Until maximum epochs (\( \text{IterMax} = 500 \)) reached
Online inference:
Apply the trained ResCelNet model to the online dataset.
Calculate the correct number \( k_{\text{signal}} \) of PU identification and the correct number \( k_{\text{noise}} \) of noise sample identification.
Performance evaluation:
Calculate the detection probability \( P_d = \frac{k_{\text{signal}}}{n} \) and the false-alarm probability \( P_{fa} = \frac{k_{\text{noise}}}{n} \).

IV. EXPERIMENTS

In this section, we first introduce the data simulation protocol used in our experiments, then ablation analyses are provided to show the effects of various factors. This is followed by experimental results compared with other state-of-the-art methods.

A. DATA SIMULATION

In this work, we use the QPSK modulation/demodulation platform of MATLAB that is commonly used in communication system simulation to carry out the experiments. Both offline and online datasets are simulated on transceivers.

During the simulation, the emitted signals go through a series of alterations including serial-parallel conversion, Rayleigh channel fading, and AWGN noise (additive white Gaussian noise), finally producing the real-scenario signal at the receiver end. The detailed processes at the transmitter and receiver ends are described as follows.
For the transmitter end, first the serial-parallel conversion module converts the transmitted binary NRZ symbol sequence into two-channel sequences, followed by modulating each sequence with sine and cosine signals via modulator. Then the two sequences are superimposed together and go through the Rayleigh fading channel. After AWGN noising, the to-be-emitted signals are obtained.

At the receiver end, the received signal is preprocessed using orthogonal sampling, low-pass filtering and sampling decision, forming signal vectors \( I \) and \( Q \) with \( N \) samples, which are further reshaped and normalized to gray level matrices of \( M \) rows by \( N/M \) columns. In the process, WGN with mean value 0 and standard deviation 1 (or pink noise optionally) is used as noise signal. In this work, the QPSK-based radio transceiver system is briefed as “QPSK system” for short following the naming in [25, 26], and the simulation flow of the QPSK system is illustrated in Fig. 6.

Note that the standard processes of PU transmitting signal and SU receiving signal implementations are already embedded in the physical transmitter and receiver devices, while all the data processing and model inference steps proposed in this work are implemented at the receiver end. Therefore, there is no need to generate these data through simulation in the actual scenario of spectrum sensing, which can save some time overhead.

By configuring the parameters of the AWGN module of Simulink, the emitted signals are obtained from multiple PUs under various SNR values. The simulation environments are given as follows: Matlab R2020b, Intel® Core® CPU i7-1065G7 @ 1.30GHz, 32GB RAM, and NVIDIA® GeForce®MX350.

### B. ABLATION STUDY

In order to validate the effectiveness of the proposed ResCelNet spectrum sensing method, three ablation studies are carried out:

1) Effects of multiple branches on the performance of ResCelNet spectrum sensing method.
2) Effects of the number of network layers on the performance of ResCelNet and traditional CNN methods.
3) Effects of the number of samples on the performance of ResCelNet method.

**Effects of multiple branches.** In this experiment, 1000 received signal samples and 1000 received noise samples are collected from SUs for the training purpose, and 100 received signal samples and 100 received noise samples are collected from the same group of SUs for the validating purpose. In order to validate the effects of multiple branches, we set 1-branch, 2-branch and 3-branch structures and compare the performance under 8, 9 and 10 network layers. The SNR in this experiment is fixed to -19dB as a low SNR environment.

Table 2 shows the offline training time and accuracy rates of ResCelNet under different numbers of branches and numbers of layers. It is observed that when the number of layers are fixed, our detection accuracy increases as number of branches increases. When the number of ResCelNet layer is set to 8, the accuracy are 87.5%, 93.5% and 94.0% for 1-branch, 2-branch and 3-branch, respectively. The relevant improvements of 2-branch and 3-branch accuracy rates compared to the single branch is 6.9% and 7.4%. The reason for the significant improvement is that multiple-branch network extracts features independently through different branches, and the integrated feature reduces the variance of local optimal features and becomes a better estimation of the global optimum. However, the accuracy improvement of multiple-branch network is achieved at the cost of multiple computation and memory, reducing its utility in practical scenarios, which is evident in Table 2. A superior method is desired to be outstanding in its accuracy and resource balance: the models with less resource requirements exhibit obvious lower or unstable accuracy while those with slightly higher accuracy will consume unnecessarily much more computational cost and inference memory. To balance the accuracy and computational cost, we choose 2-branch structure for ResCelNet.

### TABLE 2: Performance comparison of ResCelNet with various layers and branches.

| Algorithm          | Training Time (s) | Accuracy (%) |
|--------------------|-------------------|--------------|
| ResCelNet_8_Multi_1| 22.5              | 87.5         |
| ResCelNet_8_Multi_2| 23.9              | 93.5         |
| ResCelNet_8_Multi_3| 24.8              | 94.0         |
| ResCelNet_9_Multi_1| 26.2              | 90.0         |
| ResCelNet_9_Multi_2| 26.4              | 94.0         |
| ResCelNet_9_Multi_3| 27.8              | 94.5         |
| ResCelNet_10_Multi_1| 26.9             | 92.0         |
| ResCelNet_10_Multi_2| 27.7             | 95.5         |
| ResCelNet_10_Multi_3| 28.4             | 96.5         |

**Effects of Number of Network Layers.** In this experiment, the training dataset of PU is obtained under various SNR values: collect 50 pairs of PU signals with SNR ranging from -19dB to 0dB with 1dB interval, forming 1000 pairs of PU signals as the signal training dataset, and in the meantime collect 1000 pairs of noise signals as training dataset for SUs. Similarly, the validation dataset of PU is obtained under mixed SNR scenario: collect 5 pairs of PU signals under various SNR values, forming 100 pairs of PU signals as the signal validation dataset, and in the meantime collect 100 pairs of noise signals as validation dataset for SUs.

For a fair comparison, the network parameters are kept identical for both ResCelNet and the traditional CNN. However, the network structures are different: ResCelNet is comprised of multiple RCB blocks, while the traditional CNN consists of stacked convolution and pooling layers.

Fig. 7 shows the effects of the number of layers \( L \) on ResCelNet and the traditional CNN. It is evident that the accuracy of ResCelNet is significantly outperforming that of the traditional CNN, with roughly 50% improvement. Specifically, ResCelNet outperforms CNN by 8%, 35% and 55% when \( L \) is 5, 20, 68, respectively. However, the accuracy...
of ResCelNet is not following a mono-increasing trend, but roughly following a bell curve, with peak accuracy value reaching 94.5% when $L = 20$. The main reason is too deep neural networks are prone to gradient vanishing. As a result, we set $L = 20$ as an optimal layer parameter.

To illustrate the reason behind the inferior accuracy of the traditional CNN with deep layers, we recorded the loss values over all iterations with 44 layers and 68 layers for both ResCelNet and CNN, as shown in Fig. 8. It is observed that the loss values of both ResCelNet_44L and ResCelNet_68L fluctuate more than those of CNN_44L and CNN_68L. For CNN, the loss values do not change noticeably (from 0.70 to 0.71) when the $L$ increases from 44 to 68. According to Eq. 16, the loss value is related to the predicted label $\hat{y}^{(i)}$, and according to Eq. 14, $\hat{y}^{(i)}$ depends on the network weight $W$. This indicates that in the case the loss value always stuck to a fixed value, it is highly probable that the network weights lost its capability to update. We can conclude that vanishing gradient problem occurs when the traditional CNN is deep, and the proposed ResCelNet algorithm effectively alleviate this issue.

**Effects of Number of Samples.** In real spectrum sensing scenarios, the sampling rate in QPSK can be configured for signals $I$ and $Q$, rendering different signals, which is potentially affecting the detection accuracy of ResCelNet. In this experiment, we test the effects of different number of samples. Without loss of generality, we collect 64, 81, 100, 400, 900, and 1600 samples under SNR values of -19dB, -14dB, -9dB, and -4dB, and study the effects on accuracy, respectively.

Fig. 9 shows the effects of different number of samples on accuracy of ResCelNet. Under $SNR = -4dB$, the accuracy is 97%, 98%, 100%, 98%, 97%, and 96%, when the number of samples are 64, 81, 100, 400, 900, and 1600. It can be noticed that the accuracy increases to its peak when the number of samples is 100, and then drops slowly as the number of samples increases to 1600. To balance the accuracy and computation cost, the optimal number of samples is set to be 100 for ResCelNet.

### C. COMPARE WITH THE STATE-OF-THE-ARTS

We compare the proposed ResCelNet approach with state-of-the-art methods, and five experiments are carried out:

1) Comparison of convergence rates of ResCelNet and traditional CNN.

2) Comparison of computational complexity of SVM, CNN and ResCelNet
3) Comparison of efficiency of ResCelNet, CNN [13] and SVM [8].
4) Comparison of detection probabilities of ResCelNet, CNN and SVM under both WGN and pink noise environments.
5) Comparison of ROC (receiver operating characteristics) curves of ResCelNet, CNN and SVM.

Comparison of convergence rates. In this experiment, the number of layers of the traditional CNN is set to be 5, and the data collection protocol for training and validating is kept identical as in the second ablation study.

The accuracy of ResCelNet and CNN over iterations up to 500 is shown in Fig. 10. It can be observed that ResCelNet accuracy achieves a reasonably high value 93.5% when iteration is 50, and saturates around 94% when iteration increases to 500. In contrast, the accuracy of the traditional CNN increases from 48% at iteration 50 to 84% at iteration 400. Whereas this seems to be a good trend, there is no more improvement of CNN even the iteration reaches 500. The fast convergence of ResCelNet is attributed to its shortcut connection structure which accelerates gradient information back-propagation. In the meantime, the low accuracy of the traditional CNN is due to its vanishing gradient problem with the stacked convolution structure.

Comparison of computational complexity Here we analyze the computational complexity of the model training and inference. Let $n$ denote the number of training samples, according to [27], the complexity of training a SVM is proportional to $n_f^2$, $n$ and $S$, where $n_f$ is the number of free support vectors, $n$ is the number of training samples, and $S$ is the number of support vectors. Given the upper bound $C$ of the support vectors as well as a fixed kernel, the complexity ranges from $n^2$ to $n^3$ depending on the value of $C$. For SVM inference, the time complexity is linearly proportional to the number of support vectors.

Generally speaking, the complexity of backward propagation (training) and forward propagation (inference) of an artificial neural network is proportional to $O(m^3)$ and $O(m^4)$, respectively, where $m$ denotes the upper bound of the number of neurons in a layer [28]. Specifically, for a convolutional neural network, the complexity of inference is $O \left( n \sum_{l=1}^{L} F_l^2 K_l^2 Q_l (Q_{l-1}) \right)$ [29], where $L$ is the total number of network layers, $F_l$, $K_l$ and $Q_l$ are the length of output feature map, length of convolution kernel and number of output channels of convolutional layer $l$, respectively. Although CNN inference complexity looks higher than that of SVM, the computational cost of SVM is normally higher in practice due to the relatively higher number of support vectors to achieve a reasonable accuracy.

The inference complexity of ResCelNet is essentially the same as that of CNN since ResCelNet is a variant of baseline CNN with a negligible extra addition operation introduced by shortcut. The difference between algorithms of ResCelNet and CNN lies in that RCB in ResCelNet contains residual feature output map which is first order difference of $x_{l-1}$ and $x_l$, resulting in 1-order smaller magnitude than CNN’s straightforward feature map $x_l$. Therefore, ResCelNet calls for remarkably less training iterations/epochs to achieve the same training effect, which results in that ResCelNet converges dramatically faster compared to CNN to learn the output of the feature map with comparable performance [30]. In conclusion, the computational cost of these three algorithms is ranked as SVM>CNN>ResCelNet.

Comparison of efficiency. In this experiment, the number of layers of the traditional CNN is set to be 5 and 20 (same as the optimal $L$ for ResCelNet), and offline dataset is kept identical as in the second ablation study, the online dataset of PU is obtained under mixed SNR scenario: collect 5 pairs of PU signals under various SNR values, forming 100 pairs of PU is obtained under mixed SNR scenario: collect 5 pairs of PU signals as the signal test dataset, and in the meantime identical as in the second ablation study, the online dataset of PU is obtained under mixed SNR scenario: collect 5 pairs of PU signals under various SNR values, forming 100 pairs of PU signals as the signal test dataset, and in the meantime collect 100 pairs of noise signals as test dataset for SUs.

Table 3 reports the detection probability $P_d$, the false-
alarm probability $P_{fa}$, training time and test time of SVM, traditional CNN and the proposed ResCelNet, and the best figures are embolden. Theoretically, the proposed ResCelNet should save computation time since it has shortcut connection which facilitate information propagation, and also increase detection accuracy due to multiple-branch structure and shortcut connection. This is consolidated by the experimental results: compared to CNN_20L, ResCelNet is outperforming by 0.37 in detection probability and reducing by 0.16 in false-alarm probability, while the offline training time and online inference time are also saved by 3.32s and 0.68s. Although ResCelNet introduces extra 7.31s in training and 0.96s in testing compared to CNN_5L, it exhibits significant improvement of 0.15 in $P_d$ and reduction of 0.09 in $P_{fa}$ due to its superior feature extraction capability of the 20 shortcut-connected and multiple-branch layers.

Compared to SVM, ResCelNet achieves significant improvement of 0.56 in $P_d$ and reduction of 0.09 in $P_{fa}$. In the meanwhile, the training time is raised by 17.73s while the online detection time is reduce by 6.96s. The reason for longer training time of ResCelNet is that the 20 layers of neural network weights need much long time to converge compared to SVM. Nevertheless, in real spectrum sensing scenarios, the optimal spectrum sensing classifiers are generated beforehand, so that the training process is merely a one-time task while inference timing is the most important efficiency factor. It is concluded that the proposed ResCelNet is more efficient compared to SVM and CNN given a reasonable detection probability.

### TABLE 3: Performance comparison of ResCelNet against SVM and CNN.

| Algorithm | $P_d$ | $P_{fa}$ | Training Time (s) | Test Time (s) |
|-----------|-------|----------|-------------------|---------------|
| SVM       | 0.42  | 0.36     | 15.24             | 10.72         |
| CNN_5L    | 0.83  | 0.19     | 25.66             | 2.80          |
| CNN_20L   | 0.61  | 0.26     | 36.29             | 4.44          |
| ResCelNet | 0.98  | 0.10     | 32.97             | 3.76          |

**Comparison of detection probabilities.** In this experiment, the offline dataset of PU is obtained under various SNR values: collect 50 pairs of PU signals with SNR ranging from -19dB to 0dB with 1dB interval, forming 1000 pairs of PU signals sampling from SU as the signal training dataset, and in the meantime collect 1000 pairs of WGN signals as training dataset for SUs. Similarly, another offline dataset of PU is obtained under mixed SNR scenario: collect 5 pairs of PU signals sampled from SU under various SNR values, forming 100 pairs of PU signals as the signal validation dataset, and in the meantime collect 100 pairs of WGN signals as validation dataset for SUs. For online dataset, 100 pairs of PU signals sampled from SU and also collect 100 pairs of WGN or pink noise data.

When the environmental noise is WGN, the detection probabilities of ResCelNet, CNN and SVM with SNR ranging from -19dB to 0dB are compared in Fig. 11. In low SNR scenarios, ResCelNet is outstanding in terms of detection probability, especially when $SNR < 10dB$. When $SNR$ is as low as -19dB, the detection probabilities of ResCelNet, CNN and SVM are 0.96, 0.70 and 0.60, respectively. When $SNR \geq -10dB$, ResCelNet and CNN performs closely in detection probability, and still outperforming SVM by about 10%. The superiority of ResCelNet is attributed to the superior feature extraction based on the combination of shortcut connection and dual-branch structure.

ResCelNet under different communication modulation mechanisms, e.g. QPSK and BPSK, and different decaying communication channels, e.g. Rayleigh and AWGN, are also depicted in Fig. 11. It is noticed that the ResCelNet detection probability is not sensitive to these factors, indicating its good robustness and generalization.

**FIGURE 11:** Detection probabilities of ResCelNet, SVM and CNN under various $SNR$ in WGN noise environment.

Most of the traditional spectrum sensing methods are evaluated in WGN environment when simulated. However, this is not necessarily the real scenario situation. In real communication channels, it is not unusual to have pink noise present. In this work, we study the effects of pink noise on the ResCelNet performance for the purpose of evaluating its robustness and generalization in a practical spectrum sensing system.

Without loss of generality, we select the noise power uncertainty factor of $1dB$, $1.1dB$, and $1.2dB$ for pink noise, and compare the detection probabilities $P_d$ of ResCelNet, CNN and SVM with SNR ranging from -19dB to 0dB in Fig. 12. It is observed that the detection probability is extremely robust to the uncertainty of the noise power. It is obvious that ResCelNet exceeds SVM and CNN in terms of detection probability, especially when $SNR < -10dB$. When $SNR$ is as low as -19dB, the detection probabilities of ResCelNet, CNN and SVM are 0.82, 0.66 and 0.40, respectively. When $SNR \geq -10dB$, ResCelNet and CNN performs closely in detection probability, and still outperforming SVM by about 20%.

---

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
It is evident that under the situation the offline training dataset is derived from WGN environments while online test dataset is derived from pink noise environments, the proposed ResCelNet still exhibits reasonably high detection performance. We can conclude that the proposed ResCelNet shows great robustness under various external conditions compared to the existing methods.

Comparison of ROC. In this experiment, the dataset is collected in the same manner as in the experiment of detection probability $P_d$. $P_{fa}$ is also an important evaluation criterion for spectrum sensing methods. In order to evaluate performance comprehensively, we aligned the $P_d$ and the corresponding $P_{fa}$ in all the 1000 spectrum sensing cases, forming the ROC curves of ResCelNet, CNN and SVM, as shown in Fig. 13. It is obvious that given any $P_{fa}$, the detection probability $P_d$ of ResCelNet is outperforming against SVM and CNN, and vice versa, given any $P_d$, the false-alarm probability $P_{fa}$ of ResCelNet is the lowest. For instance, given $P_{fa} = 0.1$, $P_d$ is 0.98, 0.75 and 0.46 for ResCelNet, CNN and SVM, respectively. We can safely conclude that the proposed ResCelNet shows superior performance compared to the existing methods.

V. CONCLUSION

Traditional CNN-based spectrum sensing methods are limited in their capability of feature extraction due to the single branch convolution structure, while the learning ability is not simply improved by increasing the network depth because of the vanishing gradient problem. In this work, we innovatively incorporate dual-branch convolution and shortcut connection to CNN, proposing a ResCelNet spectrum sensing method based on a novel residual cellular block structure. A significantly deeper and richer feature map representation is produced by the proposed ResCelNet algorithm. Experimental results show that the proposed ResCelNet remarkably outperforms state-of-the-arts spectrum sensing approaches such as SVM and CNN in terms of detection probability and false-alarm probability. In addition, ResCelNet achieves an excellent trade-off between accuracy and efficiency while maintaining good robustness and generalization under various external conditions. We hope this work could motivate more research work to find more effective and efficient models in the field of spectrum sensing. Future work may include exploring composite loss function for learning better feature map representation to produce more superior ROC performance. Additionally, a practical verification based on specific application may be carried out using specific types of transceiver devices.

References
[1] Q.-H. Wang, B.-L. Ye, Y. Tian, W.-Z. Li, S.-L. Lu, and D.-X. Chen, “Survey on spectrum allocation algorithms for cognitive radio networks,” ACTA ELECTRONICA SINICA, vol. 40, no. 1, p. 147, 2012.
[2] L. D. S. Costa, D. A. GUIMARÃES, and F. U.-F. Bartolomeu, “On the signal-to-noise ratio wall of energy detection in spectrum sensing,” IEEE Access, vol. 10, pp. 16499–16511, 2022.
[3] M. Yang, Y. Li, X. Liu, and W. Tang, “Cyclostationary feature detection based spectrum sensing algorithm under complicated electromagnetic environment in cognitive radio networks,” China communications, vol. 12, no. 9, pp. 35–44, 2015.
[4] A. Brito, P. Sebastião, and F. J. Velez, “Hybrid matched filter detection spectrum sensing,” IEEE Access, vol. 9, pp. 165504–165516, 2021.
[5] W. Zhao, H. Li, M. Jin, Y. Liu, and S.-J. Yoo, “Eigenvalues-based universal spectrum sensing algorithm in cognitive radio networks,” IEEE Systems Journal, vol. 15, no. 3, pp. 3391–3402, 2021.
[6] O. P. Awe, A. Deligiannis, and S. Lambotharan, “Spatio-temporal spectrum sensing in cognitive radio
networks using beamformer-aided svm algorithms,” *IEEE Access*, vol. 6, pp. 25377–25 388, 2018.

7. J. Bao, J. Nie, C. Liu, B. Jiang, F. Zhu, and J. He, “Improved blind spectrum sensing by covariance matrix cholesky decomposition and rbf-svm decision classification at low snrs,” *IEEE Access*, vol. 7, pp. 97 117–97 129, 2019.

8. S. Chen, X. Wang, and B. Shen, “A support vector machine based spectrum sensing for cognitive radio,” *Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition*, vol. 31, no. 3, pp. 313–322, 2019.

9. Y.-J. Tang, Q.-Y. Zhang, and W. Lin, “Artificial neural network based spectrum sensing method for cognitive radio,” in *2010 6th international conference on wireless communications networking and mobile computing (WiCOM)*, IEEE, 2010, pp. 1–4.

10. M. R. Vyas, D. K. Patel, and M. Lopez-Benitez, “Artificial neural network based hybrid spectrum sensing scheme for cognitive radio,” in *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, IEEE, 2017, pp. 1–7.

11. Z. Chen, Y.-Q. Xu, H. Wang, and D. Guo, “Deep stft-cnn for spectrum sensing in cognitive radio,” *IEEE Communications Letters*, vol. 25, no. 3, pp. 864–868, 2020.

12. C. Liu, J. Wang, X. Liu, and Y.-C. Liang, “Deep cm-cnn for spectrum sensing in cognitive radio,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2306–2321, 2019.

13. M. Zhang, L. Wang, and Y. Feng, “Ofdm spectrum sensing method based on convolutional neural network,” *Syst. Eng. Electron*, vol. 41, no. 1, pp. 178–186, 2019.

14. C. Szegedy et al., “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

15. X. Li, M. Ding, and A. Pižurica, “Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 4, pp. 2615–2629, 2019.

16. B. Zhao, R. Hu, X. Jia, and Y. Guo, “Multi-scale residual fusion network for super-resolution reconstruction of single image,” *IEEE Access*, vol. 8, pp. 155 285–155 295, 2020.

17. C. M. Bishop et al., *Neural networks for pattern recognition*. Oxford university press, 1995.

18. K. Kafadar and J. R. Koehler, “Modern applied statistics with s-plus,” *The American Statistician*, vol. 53, no. 1, p. 86, 1999.

19. L. Zheng, Y. Yang, and A. G. Hauptmann, “Person re-identification: Past, present and future,” *arXiv preprint arXiv:1610.02984*, 2016.

20. W. Li, X. Zhu, and S. Gong, “Person re-identification by deep joint learning of multi-loss classification,” *arXiv preprint arXiv:1705.04724*, 2017.

21. N. Xue, L. Niu, X. Hong, Z. Li, L. Hoffaeller, and C. Pöpper, “Deepsim: Gps spoofing detection on uavs using satellite imagery matching,” in *Annual Computer Security Applications Conference*, 2020, pp. 304–319.

22. Z. Li, H. Shao, L. Niu, and N. Xue, “Progressive learning algorithm for efficient person re-identification,” *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 16–23, 2020.

23. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

24. A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *arXiv preprint arXiv:1703.07737*, 2017.

25. I. Fatadin, S. J. Savory, and D. Ives, “Compensation of quadrature imbalance in an optical qpsk coherent receiver,” *IEEE Photonics Technology Letters*, vol. 20, no. 20, pp. 1733–1735, 2008.

26. C. Crognaule, “Sensitivity and power budget of a homodyne coherent dp-qpsk system with optical amplification and electronic compensation,” *Journal of lightwave technology*, vol. 32, no. 7, pp. 1295–1306, 2014.

27. L. Bottou and C.-J. Lin, “Support vector machine solvers,” *Large scale kernel machines*, vol. 3, no. 1, pp. 301–320, 2007.

28. Kasper, *Introduction To The Basics Of Neural Networks*, 2018. [Online]. Available: https://kasperfired.com / series / introduction - to - neural - networks / computational-complexity-of-neural-networks.

29. Y. Sun and G. Li, “Detection of crowd abnormal behavior based on nested model of convolutional neural network,” *Computer Application and Software*, vol. 36, no. 3, pp. 196–201, 2019.

30. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016, pp. 770–778.
JIANXIN GAI received the B.S. and M.S. degrees in Instrumentation Science and Technology from Harbin University of Science and Technology, China, in 2003 and 2006, respectively, and the Ph.D. degree in Instrumentation Science and Technology from Harbin Institute of Technology, China, in 2013. He is currently an associate Professor and Deputy Dean of School of Measurement-control Technology & Communication Engineering with Harbin University of Science and Technology. His research interests include sparse signal processing, sub-Nyquist sampling, spectrum sensing, and machine learning. He has authored or coauthored more than 40 scientific articles. As principal investigator, he has received many supports of research foundation from national natural scientific foundation of China, provincial natural scientific foundation of Heilongjiang, science and technology research plan project of Heilongjiang Department of Education and so on. He won the second prize for scientific and technological progress in 2014.

XIANFENG XUE received the B.S. degree in Communication Engineering from Harbin University of Science and Technology, Harbin, China, in 2019, where he is currently pursuing the M.S. degree in Instrumentation Science and Technology. His research interests include spectrum sensing, cognitive radios, and machine learning.

ZHEN LI received the B.S. degree in Harbin University of Science and Technology, Harbin, China, in 2004, the M.S. degree in Harbin Institute of Technology, Harbin, China, in 2006, and the Ph.D. degree in Nanyang Technological University, Singapore in 2013. He is currently an Adjunct Professor in College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From 2012 to 2016, he was a Scientist with the Agency for Science, Technology and Research (A*STAR), Singapore. His research interest includes the development of computer vision techniques using deep learning, signal processing systems, and information security systems. From 2016, Dr. Zhen Li has been applying machine learning technologies in various industries such as intelligent video surveillance, cognitive radio, ad targeting, product recommendation and computer security, and got Best Award of AI Startup Team in Jiangsu Province, China, in 2019.

LINGHUI ZHANG received the B.S. degree in Measurement-control Technology and Instrumentation from Harbin University of Science and Technology, Harbin, China, in 2020, where she is currently pursuing the M.S. degree in Instrumentation Engineering. Her research interests include spectrum sensing, cognitive radios, and machine learning.