A Real-Time Modulation Recognition System Based on Software-Defined Radio and Multi-Skip Residual Neural Network

CHONG LIN1, WENJUN YAN1, LIMIN ZHANG1, AND WEN WANG2

1Department of Information Fusion, Naval Aviation University, Yantai 264001, China
2College of Aviation Foundation, Naval Aviation University, Yantai 264001, China
Corresponding author: Wenjun Yan (wj_yan@foxmail.com)

This work was supported by the Taishan Scholars Project of Shandong Province under Grant ts201511020.

ABSTRACT Communication signal modulation recognition has important research value in the fields of cognitive electronic warfare, communication countermeasures and non-collaborative communication. However, traditional signal recognition methods usually suffer some drawbacks, such as low accuracy, poor scalability, dependence on expert characteristics, and poor applicability to real-world environments. Therefore, in this article, a real-time modulation recognition system based on deep learning and software-defined radio (SDR) technology is designed. In the first step, an improved residual neural network is designed. A multi-skip residual stack (MRS) is designed to preserve more initial residuals information on the multi-scale feature map, which can simultaneously learn the deep and shallow characteristics of the signal. Then, a multi-skip residual network is designed with the MRS as the basic unit, and the network is trained using an adaptive moment estimation optimization algorithm. Finally, the network is tested on public datasets. In the second step, the network is embedded in a SDR platform composed of a GNU Radio and an universal software radio peripheral to realize real-time recognition of the input signal. Experiments show that this system has strong real-time capabilities, high recognition accuracy and considerable robustness.

INDEX TERMS Cognitive radio, modulation recognition, software-defined radio, multi-skip residual neural network, deep learning, non-collaborative communication.

I. INTRODUCTION

In order to achieve communication reconnaissance and electromagnetic dominance, dynamic spectrum monitoring and communication countermeasures as well as interference and anti-jamming technologies are needed. In recent years, rapid development of cognitive radio technology [1] has brought new solutions to anti-jamming technology using software-defined radio (SDR). Researchers can implement different functions due to the reconfigurable characteristics of SDR devices. GNU Radio is one of the most powerful SDR tools for Linux systems. Researchers in various fields have carried out many research studies (see [2]–[5]) on the combination of GNU Radio and universal software radio peripheral (USRP). With the rapid development of communication technology, communication systems often adopt a variety of modulation schemes. Automatic modulation classification (AMC) [6], [7] refers to the acquisition of signal samples before signal demodulation and automatic judgment of signal modulation type. Whether in the military or civil fields, AMC technology is of high research value.

At present, AMC approaches can be roughly categorized into maximum likelihood methods, statistical pattern recognition methods and deep learning (DL) recognition methods. In likelihood-based ratio recognition methods, the modulation recognition problem is regarded as a hypothesis test problem, and a likelihood function is used to make the decision to realize signal classification [8], [9]. The statistical pattern recognition method compares the modulation characteristics with the theoretical features [10], [11]. Compared with DL methods, the former two methods have some disadvantages, such as the need for prior probability information, high requirements for signal to noise ratio (SNR) [12], less recognition types, and poor robustness.

Lately, DL technology has made amazing achievements in the fields of action recognition [13], computer vision [14],
natural language processing [15], [16], character recognition [17] and modulation recognition (see [18]–[23]), among others. DL methods that deal with modulation recognition can be categorized into two types: a direct recognition method and an indirect recognition method. The direct recognition method is used to extract the features of the original baseband signal using a neural network to identify the modulation type. In [22], the DL technique is applied to the modulation recognition field for the first time. A convolutional neural network (CNN) is used to recognize up to 11 kinds of analog and digital modulations, with high accuracy even under low SNR. O’Shea et al. [23] recognizes 24 modulations based on residual neural network (ResNet) and the recognition accuracy is greatly improved. GNU Radio is used to generate the largest known public modulation recognition dataset, which has been uploaded to the Internet for researchers (from related fields) that are engaged in the combination of DL and AMC. An indirect recognition approach refers to the method of preprocessing an original baseband signal, and then combining it with a neural network for feature extraction. The preprocessing method includes calculating the spectrum [24], constellation diagram [25], higher order cumulants [11], [12], eye patterns and vector graphs [26]. However, with the increasing complexity of preprocessing methods, the deepening of network layers and the increasing modulation types, DL methods have lingering problems such as a large amount of training parameters, complicated parameter optimization, long training time and high requirements for hardware configuration.

Despite the rapid development of SDR and modulation recognition technology based on DL, there are few studies that combine the two technologies and observe real-time signal monitoring in real-world environments. Wang et al. [27] proposed a modulation recognition method based on a denoising auto encoder (DAE) and CNN (DAE-CNN). Wang et al. generate noiseless signal by GNU Radio and prove that the DAE-CNN classifier can effectively improve the recognition accuracy of CNN, but it does not test the model in a real-world environment. In [28], a blind demodulation system (BDS) based on AlexNet and SDR platform is designed, which can identify 3 modulations, and the real-time performance of the system is tested in a real-world environment.

To solve these problems, this work combines the improved deep ResNet with SDR, and proposes a real-time modulation recognition system:

1) A multi-skip residual neural network (MRNN) is proposed. Firstly, a multi-skip residual stack (MRS) with variable feature extraction capability is designed, which can learn more initial residual information from multi-scale feature map and extract deep and shallow features at the same time. Then, the MRNN is designed with the MRS as the basic unit, and the adaptive moment estimation optimization algorithm is used to train the network to obtain the optimal model. Finally, the performance of the model is tested with a common dataset. Compared with the traditional algorithm, the method avoids the complex theoretical derivation process of the traditional algorithm and does not need any prior knowledge. It can directly recognize baseband signal without any preprocessing compared with the existing DL algorithms, with low complexity and strong real-time performance, which is convenient for engineering applications. Furthermore, the overall recognition accuracy is high.

2) A real-time modulation recognition system based on SDR and MRNN is proposed. Firstly, a communication system is built using GNU Radio and USRP, and a dataset containing 6 modulations is made. The MRNN is trained with this dataset. Secondly, a real-time recognition module is designed based on GNU Radio’s custom block. This module can load a trained MRNN model to recognize the signals received by USRP in real-time. Finally, the performance of the system is tested. The experimental results show that the system has the advantages of real-time performance and high accuracy. The feasibility of combining SDR technology with DL technology is verified.

II. MRNN

A. MRS

With the development of DL, more and more researchers have applied it to the field of communication and achieved excellent results. The essential premise of neural network is feature extraction of input data. Through training and continuous fitting, a complex function mapping model is finally obtained.

ResNet, as a kind of CNN, has the advantage of a layer number backoff mechanism by adding skip connections between input and output of different convolutional layers. While the number of neural network layers is deepening, the performance of the network is not inferior to that before deepening. This has led to amazing improvements in computer vision performance. Fig. 1a is the basic unit of ResNet, called residual stack (RS). Each RS consists of a convolutional layer, two residual units and a max-pooling layer.
To achieve different degrees of feature extraction, retain more initial residual information on the multi-scale feature maps, learn the deep and shallow features of the signal, and avoid the complexity of network structure, the RS is improved, labelled MRS, as shown in Fig. 1b. The original residual unit is labelled ‘residual unit A’. After removing a convolutional layer from ‘residual unit A’, the remaining layer is labelled ‘residual unit B’. Next, the input of ‘residual unit A’ is short circuited with the output of ‘residual unit B‘ and different multi-skip connections are set. After many experiments, we found that this design really improves the method.

In MRS, ‘residual unit A’ is composed of two convolutional layers and a \( \xi \)-times skip connection; ‘residual unit B’ is composed of a convolutional layer and a \( \eta \)-times skip connection; and a \( \lambda \)-times skip connection is connected between ‘residual unit A’ and ‘residual unit B’.

For a single signal sample, the MRS’s input is \( x \) and its output is \( y \). For simplicity of expression, the convolutional layer bias weight \( b \) is omitted. Thus,

\[
\begin{align*}
  u &= f (x, W_0) \quad (1) \\
  v &= g (u, W_1, W_2) + \xi u \quad (2) \\
  w &= h (v, W_3) + \eta v \quad (3) \\
  y &= \text{Maxp} (w + \lambda u) \quad (4)
\end{align*}
\]

where \( W_i \) stand for the weight of the convolutional layer, \( f (x, W_i) \), \( g (x, W_i) \) and \( h (x, W_i) \) refer to the function to be learned, \( \xi, \eta \) and \( \lambda \) are multiplier coefficients of skip connection, and \( \text{Maxp} (\cdot) \) represents max-pooling operation.

Therefore, the MRS is equivalent to the fitting function with a single convolutional layer when \( g(\cdot) \) and \( h(\cdot) \) are both zero. The MRS is equivalent to the fitting function with 2 or 3 convolutional layers when \( g(\cdot) \) or \( h(\cdot) \) is zero, respectively. When neither \( g(\cdot) \) nor \( h(\cdot) \) is zero, the MRS is equivalent to the fitting function with 4 convolutional layers. Therefore, MRS can always fit the features of input \( x \) in different degrees.

B. MRNN

Due to the limited feature extraction capability of the MRS, in order to extract the deep features of modulated signals and prevent the occurrence of gradient diffusion and gradient explosion, an MRNN is designed by connecting several MRSs in series, as shown in Fig. 2. The network is composed of an input layer, 6 MRSs, 2 fully connected (FC) layers and 1 softmax layer. Thus, there are 24 convolutional layers, 6 max-pooling layers and 3 FC layers (softmax layer is FC layer in essence). After training, MRNN can realize the feature expression of at least 6 and up to 24 convolutional layers.

1) NETWORK PARAMETERS CONFIGURATION

In the deep neural network, the number of network parameters is very large, and the parameter setting is particularly important for the learning capability of the network. In order to reduce the computational complexity of the network and improve the training speed of the network, the same parameters configuration is used as much as possible in MRNN. Table 1 shows the layer parameters configuration of each MRS. The convolutional layer parameters are convolutional kernel size and activation function. Max-pooling layer parameters are receptive field window size and step size. Set the skip connection parameters \( \xi = 1, \eta = 1 \) and \( \lambda = 1 \) for MRS \( 1 \sim 6 \).

For the first two FC layers, the number of neurons is 128, the activation function is S-shaped Rectified Linear Activation Unit (SReLU), and there is a dropout layer behind each FC layer, and the drop probability is set to 0.3.

For the last FC layer, the number of neurons is 24 (the number of modulation types to be recognition), and the activation function is softmax function. Xavier initialization method is used to initialize the weights of all convolutional layers and FC layers, which has been proved to be superior to the random initialization method in [29].

2) NETWORK TRAINING

Neural network training is divided into forward propagation and back propagation. Through repeated iterations, the network parameters are continuously optimized, and finally the input characteristics are fitted, and the optimization model is obtained.

Suppose that the number of training samples in each batch of networks is \( m \), the input is sample data \( \{ x^{(1)}, x^{(2)}, \ldots, x^{(m)} \} \) and the corresponding modulation label is \( \{ y^{(1)}, y^{(2)}, \ldots, y^{(m)} \} \), and the output is \( \{ \hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(m)} \} \). Then,

**TABLE 1. Layer Parameters Configuration of MRS.**

| Layer   | MRS 1  | MRS 2–6 |
|---------|--------|---------|
| Conv    | 1×1/Liner | 1×1/Liner |
| Conv 2–4| 3×2/ReLU | 3×2/ReLU |
| Pooling | 2×2/2×2 | 2×1/2×1 |

**FIGURE 2. MRNN.**
\( \hat{y}^{(i)} \) is represented as
\[
\hat{y}^{(i)} = F \left( W, b; x^{(i)} \right)
\]
where \( W \) and \( b \) are the weight matrix of the network, \( i = 1, \ldots, m \).

In forward propagation, the cross-entropy loss function \( \Gamma \) is used as the evaluating indicator to measure the difference between the true label and the predicted label in each batch of samples
\[
\Gamma (W, b) = -\frac{1}{m} \sum_{i=1}^{m} \left( y^{(i)} \right) \log \left[ \hat{y}^{(i)} \right] + \mu \sum \| W \|^2 \tag{6}
\]
where \( \mu \sum \| W \|^2 \) is the regularization term of network weights, \( \mu \) is the regularization coefficient, and \( \sum \| W \|^2 \) is the sum of squares of ownership values.

In back propagation, an adaptive moment estimation optimization algorithm is used to optimize MRNN parameters. This algorithm is superior to the stochastic gradient descent algorithm and is suitable for large datasets. Taking the weight \( w^l \) to be optimized as an example, the updating process in iteration \( t \) is as follows:

1. Calculate the gradient of loss function \( \Gamma \) with respect to \( w^l \).
\[
g^l_t = \frac{\partial \Gamma (W, b)}{\partial w^l}
\tag{7}
\]
2. Update the biased first moment estimate and second raw moment estimate.
\[
\begin{align*}
    m^l_t &= \beta_1 \cdot m^l_{t-1} + (1 - \beta_1) \cdot g^l_t \\
    V^l_t &= \beta_2 \cdot V^l_{t-1} + (1 - \beta_2) \cdot (g^l_t)^2
\end{align*}
\tag{8}
\]
3. Update the parameters.
\[
w^l_{t+1} = w^l_t - lr \cdot m^l_t / \sqrt{V^l_t}
\tag{9}
\]
where \( l \) is the number of layers where the weight \( w^l \) is located, the learning rate \( lr = 0.001 \), and the exponential decay rates are \( \beta_1 = 0.9, \beta_2 = 0.999 \).

3) PERFORMANCE TESTING
After training, the trained model is tested to verify the performance of MRNN. The accuracy rate (AR) is used to measure the performance of the model
\[
AR = \frac{k}{n} \times 100\%
\tag{10}
\]
where \( k \) is the number of correctly recognized samples, and \( n \) is the total number of test samples.

According to the analysis, the modulation recognition algorithm based on MRNN can be expressed as follows.

### III. REAL-TIME MODULATION RECOGNITION WITH GNU RADIO AND USRP

#### A. GNU RADIO + USRP

In recent years, the SDR platform composed of GNU Radio + USRP has developed rapidly and has been widely used. GNU Radio provides a large number of libraries needed by general software radio, including various modulation and demodulation modules, error correction coding modules, signal processing modules. Researchers can design and build communication systems as a whole by calling various libraries and designing effective connections. USRP is the default hardware front-end of GNU Radio, which has the functions of transmitting and receiving RF signal, converting between RF signal and intermediate frequency signal, and converting between intermediate frequency signal and baseband signal.

Based on GNU Radio Companion 3.7.14.0 and Ettus USRP B210, this article proposes a real-time modulation recognition system embedded with MRNN (RMRS-MRNN). As shown in Fig. 3, the right part is the proposed system model and the left part is a transmitter that is used to test the system performance and can simulate the signal. We use two computers equipped with Ubuntu 18.04 to connect one USRP B210 respectively to build a communication system. The realization of real-time modulation recognition is divided.
into two stages. The first stage is off-line training. The transmitter transmits signals of different modulation types, and the receiver collects baseband signals in time domain and saves them as a dataset. Then the MRNN is trained with the dataset and the trained model is saved. The training was carried out on a computer equipped with NVIDIA 2080Ti and Windows operating system. The second stage is the online testing, as shown in Fig. 3. The transmitter transmits signals of different modulation types, and the receiver obtains baseband signal after down conversion. One channel of time domain baseband signal is sent to the trained MRNN for real-time recognition (see Section III-B), and the other is sent to demodulation module. The demodulation module demodulates according to the recognition result. Experimental results show that the real-time recognition time is less than 5 ms.

B. REAL-TIME MODULATION RECOGNITION WITH MRNN

GNU Radio contains many general signal processing modules. When these modules cannot meet the application requirements, the user-defined signal processing function can also be realized through an embedded python block (EPB) [30].

The bold text box in Fig. 3 is the modulation recognition module implemented by an EPB, which is embedded in the MRNN and its structure is shown in Fig. 4. In the data preprocessing part, the baseband data from USRP B210 is segmented, where each segment has a length of 1024, and the in-phase and quadrature components of the signal are separated to generate data with dimension $1024 \times 2$. The Keras DL framework loads the trained MRNN model, identifies the input data, and finally sends the output results to the demodulation module.

IV. EXPERIMENT

The experiment is divided into two parts. The first part optimizes the performance of the MRNN based on the largest known common dataset Radio ML2018.01A [22] and compares MRNN with several existing neural network algorithms. In the second part, the neural network is embedded into the SDR platform composed of the GNU radio and USRP to verify the feasibility of real-time modulation recognition. We have 3 computers: one is equipped with Windows 10 and NVIDIA 2080Ti for training MRNN in Section IV-A; the other two computers are equipped with Ubuntu 18.04 in Section IV-B, and each computer is connected with a USRP B210 to build a communication system.

A. PERFORMANCE OF MRNN

1) EXPERIMENTAL ENVIRONMENT

The dataset used in this section is one of the most popular datasets, RadioML2018.01A, and has a size of about 20 GB. Its structure is shown in Fig. 5.

The dataset contains 24 modulated signals (5 analog modulations and 19 digital modulations). The SNR range of each signals is $-20$ dB to $+30$ dB, and the step size is 2 dB. Each modulated signal includes 4096 vectors at a specific SNR. Each vector has 1024 samples. Each sample contains data of X, Y and Z dimensions. The X dimension is the in-phase and quadrature components of the signal. The Y dimension is the modulation label of the signal. The Z dimension is the SNR label. In order to simulate the real-world environment as much as possible, improve the engineering application value, and make the neural network trained by the dataset better applied in the wireless channel, the dataset is generated by GNU Radio, and the channel is multi-path fading channel.

The hardware configuration of the experiment for training MRNN is Windows 10, Intel (R) core (TM) i7-9700 CPU, 32 GB RAM and GeForce RTX 2080Ti. The DL environment is a Keras DL framework based on TensorFlow. Because the RadioML dataset is too large, only 1/2 of the dataset (about 1.27 million samples) is used so as to shorten the training time. The experimental setup are as follows: the number of MRS stacks $= 6$, batch size $= 1024$, epochs $= 500$. 
2) PERFORMANCE OPTIMIZATION OF MRNN

The performance of a neural network is not only related to the network structure, but also to the training parameters. The experimental steps of optimizing MRNN are as follows:

Step 1: Reads the dataset and randomly divides the dataset into training set and test set according to a 7:3 ratio.

Step 2: Sets different batch sizes, observes the result and gets the optimal batch size.

Step 3: Sets a different data length, observes the results and gets the optimal data length. If the optimal effect is not obtained, selects the suboptimal batch size and repeats this step.

Step 4: Sets different MRS numbers, observe the result and get the optimal number of MRS. If the optimal result is not obtained, selects the suboptimal data length for training, and repeats this step.

Step 5: Saves the model.

Fig. 6 gives an experimental comparison of batch size, data length and network depth (MRS number).

In Fig. 6a, the recognition accuracy gradually increases as the batch size increases from 128 to 1024 but decreases as the batch size increases to 2048. On the one hand, batch size determines the direction in which the gradient decreases. The larger the batch size, the more accurate the model will be. On the other hand, when the number of batch size reaches a certain value, hardware performance reaches a bottleneck, parameter learning becomes slow, and accuracy decreases. Thus, the optimal batch size is 1024.

As Fig. 6b shows, with the increase of the data length of the training samples, the more signal characteristics fed into the network, and the performance of network gradually improves. The optimal data length in this case is 1024.

Fig. 6c shows that the network feature extraction capability is limited when the number of stacks is 2, and the recognition accuracy only reaches 80% at high SNR. When the number of stacks is 4 or 6, the recognition accuracy reaches about 95% at high SNR, and the latter’s effect is slightly better. When the number of stacks is 8, the network is too large to fit the input features, and the recognition accuracy is about 91% at high SNR. Thus, the optimal number of MRS stacks is 6.

Therefore, MRNNs have the best performance when the batch size is 1024, the data length is 1024, and the number of MRS stacks is 6. As Fig. 6c (MRSs = 6) shows, the overall recognition accuracy of MRNN is very high for distinguishing 24 modulations. Although the performance is poor at low SNR, it can reach 50% when SNR = 0 dB (low SNR) and 94% when SNR = 10 dB (high SNR). When SNR exceeds 14 dB, the accuracy rate is about 96%.

3) ANALYSIS OF MRNN RECOGNITION ACCURACY

Fig. 7 shows the confusion matrix for MRNN at SNR = 10 dB. The vertical axis is the true label and the horizontal axis is the predicted label.

FIGURE 6. Comparison of different network parameters. (a) Performance regarding to various batch size. (b) Performance regarding to various data length. (c) Performance regarding to various MRSs.
From the modulation classification point of view, MRNN has a higher recognition accuracy for all classes except the amplitude modulation (AM) class, which has a recognition accuracy more than 82%. However, MRNN has a lower recognition accuracy 68% for AM class. The probability of identifying double-sideband suppressed-carrier modulation (DSB-SC) errors as double-sideband modulation with carrier (DSB-WC) is 14%, and that of identifying single-sideband suppressed-carrier modulation (SSB-SC) errors as single-sideband modulation with carrier (SSB-WC) is 32%. It may be that the signal contains a small carrier component, which makes it difficult to distinguish whether or not to suppress the carrier.

From the order of digital modulation, the recognition accuracy of MRNN of high-order modulation (16PSK, 32PSK, 64APSK, 128APSK, 64QAM, 128QAM and 256QAM) is not 100%. This is to a large extent expected, because it is very difficult to distinguish the high-order modulation completely in any way when the signal is observed for a short time in a noisy background. Therefore, the following confusion blocks appear in Fig. 7: 16PSK confusion with 32PSK, 64APSK confusion with 128APSK, 64QAM confusion with 128QAM, 256QAM confusion.

4) COMPARISON OF DIFFERENT ALGORITHMS
Comparing MRNN with existing several DL algorithms based on dataset RadioML, the experimental results are shown in Fig. 8. The baseline classification method [23] uses a classifier based on the extreme gradient boosted trees (XGBoost) to extract the statistical characteristics of signals for classification. A neural network based on Visual Geometry Group (VGG) [23] is similar to the method in this article. It does not extract expert features or preprocess serial time series, but learns and classifies them directly. A CNN-AMC method [27] uses a time series and SNR as inputs for CNN. The modulation classification network (McNet) method [31] designs several specific convolutional blocks to learn the spatial-temporal signal correlations of different asymmetric convolution kernels in parallel for classification.
Fig. 8 shows the baseline classification method with advanced decision tree classifier that has the worst accuracy. Although VGG and CNN-AMC have the same backbone architecture, VGG is slightly better than CNN-AMC because it deploys more $3 \times 3$ convolutional kernels on each convolutional layer to extract more features. Because ResNet obtains the highly correlated features of input sequences through residual structure, the recognition accuracy of ResNet is about 9% higher than that of the baseline classification method, CNN-AMC and VGG in the case of high SNR.

Because MRS learns both deep and shallow features of signals on multi-scale feature maps at the same time, and one MRS can be fitted as a function with 1 ~ 4 convolutional layers, MRNN is more capable of extracting features than ResNet: the recognition accuracy reaches 50% when SNR $\geq 0$ dB, and the performance increases from 74% to 96% when SNR $= 14$ dB.

MRNN is slightly better than McNet. When the SNR is less than 3 dB, the recognition accuracy of MRNN is lower than McNet, 7% lower than McNet at -4 dB. However, MRNN is better than that of McNet when the SNR is more than 4 dB, which is 9% higher at 8 dB.

Computational complexity analysis including both spatial and temporal complexity is only compared to algorithms that are available. Spatial complexity refers to the number of parameters to be trained in the model, which is mainly determined by the convolutional layers and the FC layers. The spatial complexity of MRNN compared with VGG, CNN-AMC, and ResNet is shown in Table 2. MRNN has about 73% fewer parameters than CNN-AMC, 40% fewer parameters than VGG, 34% fewer parameters than ResNet. Therefore, MRNN has low spatial complexity.

Time complexity is measured by the training time per iteration and the inference time of the test set. The time complexity comparison between MRNN and ResNet is shown in Table 3. Because of fewer convolutional operations in each RS, MRNN performs more quickly than ResNet, which is about 17% shorter than ResNet under the same experimental conditions, even though the number of MRNN skip connections increases. The inference time of the MRNN for 383,400 samples in the test set is about 15.8 s, which is about 14% shorter than ResNet. Thus, MRNN has low time complexity.

Based on this analysis, and compared with several existing DL algorithms, MRS can learn the deep and shallow characteristics of signals on multi-scale feature maps simultaneously, which greatly improves the recognition accuracy of MRNN without increasing the computational complexity.

### Comparison of Space Complexity

| Network   | Trainable parameters |
|-----------|----------------------|
| VGG       | 257K                 |
| CNN-AMC   | 575K                 |
| ResNet    | 236K                 |
| MRNN      | 155K                 |

### Comparison of Time Complexity

| Network    | Iteration time(s) | Inference time(s) |
|------------|-------------------|-------------------|
| ResNet     | 156               | 18.4              |
| MRNN       | 129               | 15.8              |

### B. PERFORMANCE OF RMRS-MRNN

In Section IV-A, MRNN has shown excellent performance on public dataset. Now, we embed it in an SDR platform consisting of a GNU Radio and a USRP to implement and test the real-time modulation recognition system shown in Fig. 3. This article does not discuss the specific process of signal modulation and demodulation, instead it focuses on the feasibility of real-time modulation systems embedded with MRNN. Because of the limited transmission power of USRP B210, we only conduct the indoor experiments. Therefore, we choose the indoor wireless communication frequency 2.4 GHz as the working frequency of the system. Bluetooth, ZigBee, wireless USB and 802.11 all work in this frequency band. The modulation schemes of Bluetooth are GFSK, $\pi/4$-DQPSK, and 8DPSK. When ZigBee works in 2.4 GHz frequency band, OQPSK modulation is adopted. Wireless USB usually adopts GFSK modulation. In the signal modulation level, 64QAM is used in 802.11n, and 256QAM is achieved in 802.11ac. We regenerated the dataset with only 6 modulations, including BPSK, QPSK, 8PSK, GMSK, GFSK and 16QAM. Two computers with Ubuntu 18.04 are used. Computer A’s GNU Radio controls a USRP B210 as a transmitter to manually control the various transmitted modulated signals to simulate interference signals in a real-world environment. Computer B’s GNU Radio controls another USRP B210 as a receiver to collect different received signals and save them as a dataset. Because it is very difficult to measure the SNR of the received signal in real-world environment, we change the SNR by changing the distance.

We use 2.4 GHz in the public frequency band which is free of license as the communication frequency of the system. The sample frequency is set to 1 MHz. The bandwidth is set to 5 MHz. Bits per sample is set to 4. The transmission gain type is set to ‘Normalized’ and the gain value is set to 0.8. The receive gain type is set to ‘Normalized’ and the gain value is set to 0.6. Next, the received signal data is saved and the downsampled dataset size is $6 \times 10 \times 10000 \times 1024$. The number of modulated signals is 6. The communication distance range is 20 m, i.e., from 2 m to 20 m with 2 m step size. Each signal collects 10,000 data samples of 1024 sample length.

1) **RECOGNITION ACCURACY ANALYSIS**

The MRNN network is trained using a regenerated dataset and the experimental results are shown in Fig. 9. Fig. 9a shows the average recognition accuracy of the model at different distances is greater than 93.9%. There are confusion blocks between the MPSK signals. The recognition accuracy for 8PSK is the lowest, with 5% probability of being recognized as BPSK and 1.1% probability of being
recognized as QPSK, which may be due to strong interference when collecting 8PSK data. Because MPSK, GMSK, GFSK, and 16QAM belong to different modulation types, their constellation maps are significantly different, so the recognition accuracy is above 98.4\%.

Fig. 9b shows the model recognition accuracy of different signals at different distances. Overall, the recognition accuracy is high. To demodulate the signal successfully at the receiver, experiments have been performed at a limited distance; so, the experimental results do not show that the recognition accuracy deteriorates significantly with the distance. When the communication distance exceeds 10 m, the recognition accuracy decreases gradually.

Table 4 gives the comparison results of network performance with [28]. In [25], Peng et al. collect data from a specific distance to draw a constellation diagram (CD) and uses the CD to train the model. The test results of the network model trained by the method in [25] are also shown in Table 4.

The recognition accuracy of the algorithm proposed in this work is higher than that of the algorithm in [25]. Although the recognition accuracy of this work is approximately the same as [28], the modulated signal types are increased from 3 to 6, the system complexity is increased, and the accuracy is still above 93.9\%.

2) REAL-TIME PERFORMANCE ANALYSIS

After training the MRNN model with our own dataset, we use a general office notebook to load the model through GNU Radio’s EPB and recognize the received signal in real-time. Fig. 10 shows an example screenshot of the output of the signal recognition method when the communication distance is 2 m. Because there are few papers on signal recognition that use neural networks in real-world environments, this subsection is only compared with the current available paper [28]. Table 5 shows the comparison of average recognition times. The average recognition time was shortened from 39.5 ms in [28] to 4.1 ms in this work. Recognition speed will be further accelerated if a better-performing computer is used.

V. CONCLUSION

In order to solve the problem of signal modulation recognition, this article proposes a modulation recognition method based on MRNN. A real-time modulation recognition system based on SDR and MRNN is designed. The system has the following advantages: 1) Compared with traditional algorithms, it does not need to extract signal features manually, and has many recognition types, which is suitable for non-collaborative communication and communication.
confrontation. 2) Compared with existing DL algorithms, it can recognize baseband signal directly without any pre-processing. The algorithm has low complexity, strong real-time performance and is easy to deploy in many engineering scenarios. 3) The recognition accuracy is high. For 24 modulations, it can reach 50% when SNR = 0 dB (low SNR) and 96% when SNR = 14 dB (high SNR).

REFERENCES

[1] I. Kakalou, K. E. Psannis, P. Krawiec, and R. Badea, “Cognitive radio network and network service chaining toward 5G: Challenges and requirements,” IEEE Commun. Mag., vol. 55, no. 11, pp. 145–151, Nov. 2017.

[2] A. Mate, K.-H. Lee, and I.-T. Lu, “Spectrum sensing based on time covariance matrix using GNU radio and USRP for cognitive radio,” in Proc. IEEE Long Island Syst., Appl. Technol. Conf., Farmingdale, NY, USA, May 2011, pp. 1–6.

[3] A. Marwanto, M. Adib Sarijani, N. Fisal, S. Kamihal Syed Yusof, and R. A. Rashid, “Experimental study of OFDM implementation utilizing GNU radio and USRP-SDR,” in Proc. IEEE 9th Malaysia Int. Conf. Commun. (MICC), Kuala Lumpur, Malaysia, Dec. 2009, pp. 132–135.

[4] Y. P. Saputra, D. Herdiana, H. Madinawati, A. B. Sukumono, and A. Munir, “Linear frequency modulated continuous wave radar using GNU radio and USRP,” in Proc. 1st Int. Conf. Wireless Telematics (ICWT), Manado, Indonesia, Nov. 2015, pp. 1–6.

[5] B. I. Supriyatno, T. Hidayat, A. B. Sukumono, and A. Munir, “Development of radio telescope receiver based on GNU radio and USRP” in Proc. 1st Int. Conf. Wireless Telematics (ICWT), Manado, Indonesia, Nov. 2015, pp. 1–4.

[6] C. H. Liao, S. L. Tu, and J. Wan, “An anti-frequency-offset algorithm for modulation recognition of satellite amplitude-phase modulated signals,” J. Electr. Inf. Technol., vol. 36, no. 2, pp. 346–352, 2014.

[7] M. Liu, B. Li, and C. Cao, “Recognition method of digital modulation signals over non-Gaussian noise in cognitive radio,” J. Commun., vol. 35, no. 1, pp. 82–88, 2014.

[8] L. Wang, Q. Gao, K. Zhang, Y. Zhang, and Z. Feng, “Modulation classification of mixed signals using fast independent component analysis,” in Proc. IEEE Wireless Commun. Netw. Conf., Doha, Qatar, Apr. 2016, pp. 1–5.

[9] D. Zhou, J. Dai, K. Niu, C. Dong, J. Sun, Y. Zhang, and H. Guan, “Polar-coded modulation based on the amplitude phase shift keying constellations,” China Commun., vol. 14, no. 9, pp. 166–177, Sep. 2017.

[10] L. M. Zhang, Q. Ling, and W. J. Yan, “Research on blind recognition algorithm of space-time block code based on higher-order cumulant,” J. Commun., vol. 37, no. 5, pp. 1–8, 2016.

[11] X. H. Tan, “Modulation recognition algorithm based on higher order cumulant and wavelet transform,” Syst. Eng. Electron., vol. 40, no. 1, pp. 171–177, 2018.

[12] X. Zhao, C. Guo, and J. Li, “Mixed recognition algorithm for signal modulation schemes by high-order cumulants and cyclic spectrum,” J. Electr. Inforn. Technol., vol. 38, no. 3, pp. 674–680, Mar. 2016.

[13] T. Huynh-The, C.-H. Hua, and D.-S. Kim, “Encoding pose features to images with data augmentation for 3-D action recognition,” IEEE Trans. Ind. Inform., vol. 16, no. 5, pp. 3100–3111, May 2020.

[14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[15] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, “Deep learning for audio signal processing,” IEEE J. Sel. Topics Signal Process., vol. 13, no. 2, pp. 206–219, May 2019.

[16] E. Cho and S. Kumar, “A conversational neural language model for speech recognition in digital assistants,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Calgary, AB, Canada, Apr. 2018, pp. 5784–5788.

[17] L. Lei, J. Lu, and S. Ruan, “Hierarchical recurrent and convolutional neural network based on attention for chinese document classification,” in Proc. Chin. Control Decis. Conf. (CCDC), Beijing, China, Jun. 2019, pp. 809–814.

[18] S. Huang, Y. Yao, Z. Wei, Z. Feng, and P. Zhang, “Automatic modulation classification of overlapped sources using multiple cumulants,” IEEE Trans. Veh. Technol., vol. 66, no. 7, pp. 6089–6101, Jul. 2017.

[19] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, “Robust modulation classification under uncertain noise condition using recurrent neural network,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[20] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, “Deep learning models for wireless signal classification with distributed low-cost spectrum sensors,” IEEE Trans. Cognit. Commun. Netw., vol. 4, no. 3, pp. 433–445, Sep. 2018.

[21] S. Huang, L. Chai, Z. Li, D. Zhang, Y. Yao, Y. Zhang, and Z. Feng, “Automatic modulation classification using compressive convolutional neural network,” IEEE Access, vol. 7, pp. 79636–79643, 2019.

[22] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” in Proc. Int. Conf. Eng. Appl. Neural Netw. Cham, Switzerland: Springer, 2016, pp. 213–226.

[23] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-Air deep learning based radio signal classification,” IEEE J. Sel. Topics Signal Process., vol. 12, no. 1, pp. 168–179, Feb. 2018.

[24] M. Kutlin, T. Kazaz, I. Moerman, and E. De Poorter, “End-to-End learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications,” IEEE Access, vol. 6, pp. 18484–18501, 2018.

[25] S. Peng, H. Jiang, H. Wang, H. Alwakeed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, “Modulation classification based on signal constellation diagrams and deep learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 3, pp. 718–727, Mar. 2019.

[26] X. Zha, “Modulation recognition method based on multi-inputs convolution neural network,” J. Commun., vol. 40, no. 11, pp. 30–37, 2019.

[27] J. Wang, W. Wang, F. Luo, and S. Wei, “Modulation classification based on denoising autoencoder and convolutional neural network with GNU radio,” J. Eng., vol. 2019, no. 19, pp. 6188–6191, Oct. 2019.

[28] Z. Chen, “Research on blind demodulation system based on deep learning and software defined radio,” J. Signal Process., vol. 35, no. 4, pp. 649–655, 2019.

[29] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” J. Mach. Learn. Res., vol. 9, pp. 249–256, May 2010.

[30] Embedded Python Block. Accessed: Jun. 17, 2020. [Online]. Available: https://wiki.gnuradio.org/index.php/Embedded_Python_Block

[31] T. Huynh-The, C. Hua, Q. Pham, and D. Kim, “MCNet: An efficient CNN architecture for robust automatic modulation classification,” IEEE Commun. Lett., vol. 24, no. 4, pp. 811–815, Apr. 2020.
LIMIN ZHANG received the Ph.D. degree from Tanjin University, Tanjin, China, in 2005. He is currently a Professor and a Ph.D. Supervisor with Naval Aviation University, Yantai, China. His current research interests include signal processing, deep learning, the blind identification of channel coding, and satellite reconnaissance.

WEN WANG received the M.S. and Ph.D. degrees from Naval Aviation University, Yantai, China, in 2002 and 2006, respectively. She is currently a Professor with Naval Aviation University. Her current research interests include signal processing, deep learning, and satellite reconnaissance.