Joint Demodulation and Error Correcting Codes Recognition Using Convolutional Neural Network

JIAO WANG, HAO HUANG, JIAXU LIU, AND JIANQING LI, (Member, IEEE)
School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
Corresponding author: Jianqing Li (lijq@uestc.edu.cn)

ABSTRACT Demodulation of the communication signals and blind identification of error-correcting codes (ECC) are two essential purposes in adaptive modulation and coding (AMC) techniques and non-cooperative communication fields. The existing approaches treat them as two separate problems, namely the demodulation of signals with known a priori knowledge (such as channel state information (CSI) or channel noise) and the ECC type identification that depends on the demodulation results. In this paper, a novel one-stage ECC identification approach based on a multi-task deep convolutional neural network (MT-DCNN) is presented. With this architecture, the proposed method automatically recognizes the ECC types of the baseband in-phase and quadrature-phase (IQ) data without relying on any conventional demodulators. Precisely, the proposed MT-DCNN consists of three modules: the feature extraction module, the demodulation module, and the ECC types recognition module. Experiment results show that the proposed architecture can accurately identify the ECC types of the baseband IQ signals and is superior to those of the existing two-stage recognition approaches.

INDEX TERMS Convolutional neural network, demodulation, error-correcting codes identification, multitask learning.

I. INTRODUCTION

In wireless communication systems, adaptive modulation and coding is an essential technique to achieve spectrum efficient and robust link performance. AMC technique is a physical-link adaptive technology that can improve signal-to-noise performance by adjusting the modulation and coding of the transmitted data to compensate for the fading effects on the received signal due to channel variations. In real-world communication systems, the signal will generally be affected by fading, noise, interference, etc. In a cooperative scenario, the transmit side and the receive side have the knowledge of the channel state information and the ECC parameters. However, in a non-cooperative scenario, a third party needs a parameter analysis method to demodulate the baseband IQ signal and identify the error-correcting code types [1].

In general, the traditional demodulation methods of signals are usually categorized into two groups: coherent demodulation [2], [4] and non-coherent demodulation [3].

Non-coherent demodulation is an important demodulation method of communication principles. It has the advantage that channel estimation can be considered less or even omitted, processing complexity is reduced, and implementation is more straightforward, but performance degrades compared to the coherent demodulation method. Coherent demodulation, also called synchronous detector, is applicable to the demodulation of all linearly modulated signals. The key to achieving coherent demodulation is to recover a coherent carrier that is strictly synchronized with the modulating carrier at the receiver. However, when phase error and frequency offset exist in the syncing process, demodulation errors might appear. Hence when the carrier frequency offset and the sampling frequency error are inescapable, the performance of the receiver would drop.

As for error-correcting code recognition, there are two main approaches to identifying the encoder parameters. The first approach is to estimate the codeword length \( n \) and code dimension \( k \) given the type of the error-correcting code. In the existing literature, parameter estimation of several types of channel codes, such as convolutional codes [5],
cyclic codes [6], [7], [8], [9], [10], low-density parity-check (LDPC) [11], [12], and polar codes [13] are studied. For example, Swaminathan et al. [12] proposed a blind estimation algorithm that identifies code dimension and codeword length parameters of LDPC codes at the receiver for the noisy channel in a non-cooperative scenario. Arti et al. [6] investigated the problem of blind reconstruction of recurrent binary codes with unknown lengths when the received data isn’t synchronized. In addition, to improve the channel coding recognition performance, Wu et al. [8] proposed a novel blind recognition method for cyclic codes. This scheme uses a soft-decision sequence instead of a hard-decision sequence. It is based on the average cosine conformity (ACC) and is able to measure the reliability of the parity check relationship. For the second type, when the set of possible channel coding schemes and their check matrices is known to the receiver, they are applicable to identify which coding scheme is used by the transmit side using the received sequences. For instance, to tackle the blind identification of LDPC codes for BPSK signals. Xia et al. [11] proposed a blind identification system comprising three components: expectation-maximization (EM) estimator for signal amplitude and noise variance, log-likelihood ratio (LLR) estimator for syndrome a posteriori probability, and maximum average LLR detector.

Although the aforementioned blind identification algorithms achieved impressive performance on traditional communication systems, they rely heavily on theoretical analysis and channel characteristics-based algorithms. When the data size of a task grows, the efficiency of these approaches reduces. With the rapid development of computer science and hardware, deep learning technology has achieved tremendous success for computer vision, data retrieval, natural language processing, etc. A variety of signal processing tasks have also benefited from deep learning technique, such as automatic modulation classification [14], [15], [16], [17], [18], [19], [20], demodulation [21], [22], blind recognition of channel code parameters over candidate sets [23], [24], [25], [26], [27], [28], channel decoding [29], [30], [31], channel estimation [32], [33], end-to-end wireless communication systems [34], etc. In recent years, digital signal modulation identification approaches based on deep learning have been extensively studied. O’Shea et al. [17] are the pioneers of the AMC field. They demonstrated that the performance of the CNN trained on baseband IQ data exceeds those trained on artificial features. Mendis et al. [20] presented the AMC method based on deep learning, which transforms signals into images through spectral correlation function (SCF), then extracts complex features from received signals through a deep belief network (DBN). Recently, Zhang et al. [22] presented a neural network demodulator based on a one-dimensional convolutional neural network (1-D CNN). The proposed structure puts forward a demodulation method that judges the occurrence of the phase shifts in the sampled data and reconstructs the transmitted data based on the locations of phase shifts. Therefore, the proposed 1-DCNN demodulator does not suffer from grouping the sampled data. Li et al. [23] proposed a TextCNN-based ECC identification model that classifies linear block codes, convolutional codes, and turbo codes using only noisy information streams. Dehdashtian et al. [28] proposed a deep learning-based parameters identification method that first decodes channel code with different parameters in the candidate set by a common decoding algorithm and then feeds the decoded data to the CNN model to detect the correct code parameters.

According to the above discussion, it can be seen that the current blind recognition of channel code approaches is divided into two stages. The first stage implements the demodulation of the baseband signals. The second stage recognizes the ECC types. In this paper, we present a CNN-based end-to-end multi-tasking signal process architecture capable of demodulating the received signals and identifying the channel coding types simultaneously. Here, we assume that the modulation type of a signal has been obtained with state-of-the-art AMC algorithms [14], [15], [16], [17], [18], [19], [20]. To the best of our knowledge, there is no CNN-based demodulation and identification scheme tailored for the tasks mentioned above. The proposed method can identify the ECC types of IQ data without relying on conventional demodulators. Extensive experiments show that the MT-DCNN architecture yields significant improvements over state-of-the-art methods.

The rest of this paper is organized as follows. Section II briefly introduces the background of the deep learning application, the structure of a typical CNN, and the system model. In Section III, the structure of the MT-DCNN and the loss function are explored in detail. Moreover, Section IV introduces the results of the experiment and performance analysis. Finally, this work is concluded in Section V.

II. BACKGROUND

A. DEEP LEARNING

In 2006, Hinton et al. [35] proposed artificial neural networks (ANN) with multiple hidden layers. This structure has excellent feature learning ability. Since then, deep learning has made outstanding speech recognition and image recognition achievements. At present, deep learning has a variety of structures, including deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN),...
and generative adversarial networks (GAN). Deep learning is an essential branch of machine learning. Compared to external neural networks, deep learning network has more hidden layers, which can improve the training performance of deep neural networks by adjusting the connection of neurons between different layers.

B. CONVOLUTIONAL NEURAL NETWORK
A convolutional neural network is a feed-forward neural network. It usually consists of an input layer, multiple convolutional layers, multiple activation layers, multiple pooling layers, fully connected layers, and an output layer, as shown in Fig. 1. The data input layer mainly preprocesses the original data. The convolutional computation layer is the core layer of CNN. The role of the activation layer is to introduce a nonlinear function to make a nonlinear mapping of the result of the convolutional layer. The pooling layer is often used to reduce the size of the model and increase the computational speed while improving the robustness of the extracted features. The pooling layer is sandwiched between successive convolutional layers. It can be very effective in reducing the size of the parameter matrix, thus reducing the number of parameters in the final fully-connected layer. The fully-connected layer acts as a classifier in the convolutional neural network, and each of its nodes is connected to all neurons in the previous layer, combining the previous features. In summary, convolutional neural networks have a significant role in computer vision (CV), natural language processing (NLP), etc. With the development of hardware and the advancement of tuning methods, CNN is also developing rapidly.

C. SYSTEM MODEL
Fig. 2 shows a typical wireless communication system consisting of a transmitter and a receiver. At the transmitter, the information is represented as a sequence of binary bits. The encoder then encoded the binary bits and modulated them to an analog signal before transmission. Due to additive white Gaussian noise (AWGN) and interference, errors may occur during transmission. At the receiver, the received analog signal is demodulated back to a soft-decision sequence which contains the bit information and its reliability value. The bit sequence extracted from the soft-decision sequence is referred to as the hard-decision sequence. Then, the decoder checks the incoming soft-decision or hard-decision sequence and performs the error-correction process to retrieve the original data.

In this work, we process and identify signals with a CNN-aided deep learning receiver. Specifically, the receiver employs neural networks to demodulate noisy signals and identify their channel coding schemes, eliminating the need for conventional signal processing methods. During training, by learning the relationship between the received analog signal and its original information sequence, we retrieve the information from non-ideal signals as reliable as possible to improve the adaptability of the receiver.

III. THE PROPOSED APPROACH
In this section, we will introduce MT-DCNN in detail. Fig. 3 shows that MT-DCNN is divided into three parts: a feature extractor, a demodulator, and an identifier. The following subsections are devoted to explaining MT-DCNN in terms of its main building blocks, overall architecture, and loss function for training.

A. OVERALL ARCHITECTURE
1) FEATURE EXTRACTOR
First, the input of the feature extractor is the IQ data of the signal time-domain waveform. Then, the feature extraction module extracts local features of modulation information from the IQ samples through convolution operation. Such operations can decrease the computational load during the feature abstraction process, making the network less susceptible to distortions of the input samples. The multi-kernel convolution property of CNNs ensures that the network extracts various features in parallel. Thus, the demodulation function of the modulated signals is better deduced. In detail, the feature extraction module comprises three convolutional blocks. Each convolutional block consists of two convolutional layers and a max-pooling layer, as shown in Fig. 3. The max-pooling layer with filters of $2 \times 1$ applied with a stride of 1 decreases their spatial resolution to one-fourth of the original value.
After a max-pooling operation, the number of the input feature maps remains unchanged.

2) DEMODULATOR
The output of the feature extraction module is fed into the demodulator to obtain the predicted symbols. The demodulator uses $M$ filters of size $3 \times 1$ followed by Softmax activation. $M$ is the modulation order.

3) CLASSIFIER
The channel code classifier consists of four convolutional layers, a global average pooling (GAP) layer, and two fully connected (FC) layers. The first two convolutional layers contain 64 filters, and the final two convolutional layers contain 128 filters. Every convolutional layer uses the $3 \times 1$ sized 1-D convolutional kernels. The GAP layer is added after the last convolutional layer to reduce the dimension of the feature maps. After convolution and pooling operations, features are output into the FC layers to get a probability distribution using the Softmax function.

In addition, we adopt batch normalization [36] after each convolutional layer and before activation, which allows for a higher learning rate and faster network convergence while preventing the network from overfitting. We use the rectified linear unit (ReLU) [37] for the entire network as the activation function.

B. LOSS FUNCTION
Loss functions are of central importance in deep learning techniques, and a loss function quantifies the error between the output and the target value. The choice of the loss function is directly related to the network’s performance. In this work, we utilize the combination of demodulation loss and recognition loss to train MT-DCNN.

1) DEMODULATION LOSS
The demodulation loss function $\ell_{\text{demod}}$ calculates the demodulation error.

$$\ell_{\text{demod}} = -\frac{1}{N_B} \sum_{i=1}^{N_B} \sum_{c=1}^{C} \sum_{m=1}^{M} t_{icm} \log(p_{icm})$$  \hspace{1cm} (1)

where $N_B$ is the number of samples for each mini-batch, $C$ indicates that the demodulation output contains $C$ Softmax classifiers, and each Softmax contains $M$ (modulation order) neurons. $t_{icm}$ is the actually transmitted symbol, $p_{icm}$ is the predicted modulated symbol for the demodulator.

2) RECOGNITION LOSS
By denoting the actual ECC type of input IQ signal as $y_i$ and the MT-DCNN predicted type as $\hat{y}_i$, the ECC types recognition loss $\ell_{\text{cls}}$ is given as

$$\ell_{\text{cls}} = -\sum_{i} y_i \log \hat{y}_i$$  \hspace{1cm} (2)

3) TOTAL LOSS
To perform the training, we define the total training loss as:

$$\ell_{\text{Total}} = \alpha \ell_{\text{demod}} + \beta \ell_{\text{cls}}$$  \hspace{1cm} (3)
where scaling factors $\alpha$ and $\beta$ are hyperparameters used to control the weight balance of $\ell_{\text{demod}}$ and $\ell_{\text{cls}}$ in the total loss. In this work, we use $\alpha + \beta = 1$ during training.

IV. SIMULATION RESULTS

In this section, we first describe the experimental datasets and the training details. Then we evaluate the effect of different loss hyperparameters $\alpha$ and $\beta$ values. Finally, we provide comparisons with several state-of-the-art coding recognition models. All experiments are implemented on a single NVIDIA GeForce RTX 3090 with the TensorFlow [38] deep learning framework.

A. EXPERIMENT SETTINGS

The simulation setup is shown in Fig 2. In this paper, the dataset needed for the simulation is generated by Matlab2020b. Two forms of modulation included in the simulation are binary phase-shift keying (BPSK) and quadrature phase-shift keying (QPSK). The error-correcting codes used in the datasets are convolutional codes, Bose–Chaudhuri–Hocquenghem (BCH) codes, turbo codes, and turbo product codes (TPCs). Table 1 lists the extended parameters of the used dataset. Specifically, the generator polynomials of convolutional codes are given in octal representation (23, 35), (53, 75), and (133, 171), and constraint lengths are 5, 6, and 7, respectively. For BCH codes, block lengths $N \in \{7, 12, 15\}$ and information lengths $K \in \{4, 8, 11\}$ have been considered. The candidate set consists of three turbo codes with code rates $r = 1/2, 1/3,$ and $1/4$. Furthermore, the forward generating polynomials are $(33, 33), (33, 33), (25, 37, 33)$, respectively. The feedback polynomial is 23, and the constraint length is 5. Finally, we use three Hamming codes $(7, 4), (15, 11),$ and $(31, 26)$ as the subcodes of the TPC.

For the wireless channel models, the carrier frequency $f_c$, the symbol rate $f_d$, and the sampling rate $f_s$ are 1 Mega Hertz (MHz), 1 Mega Baud (MBd), and 8 million samples per second (Msps), respectively. The carrier frequency offset (normalized to the symbol rate) is randomly chosen from -0.01 to 0.01. The raised-cosine filter is used for pulse-shaping, and the roll-off factor varies between 0.1-0.5.

For the training sets, validation sets, and test sets concerned in this work, the $E_b/N_0$ range of the included signal samples is 0 dB to 10 dB with a step of 1 dB. Sixty-six thousand samples are generated for each set of parameters. Therefore, for each modulation scheme, 792000 samples are generated. The generated samples are then divided into training sets, validation sets, and test sets in a ratio of 8:1:1. Each signal contains 512 symbols, and the oversampling rate is 8. That is, each signal has 4096 sampling points. Adam [39] optimizer is used to train the networks. Settings for the Adam optimizer are initial learning rate $\beta_1 = 0.9, \beta_2 = 0.999$. For training, the number of epochs is 100, and the batch size is 256.

B. INFLUENCE OF LOSS HYPERPARAMETERS

To find the optimal values of $\alpha$ and $\beta$, we evaluate the network with the values of $\alpha$ as 0.5, 0.9, and 0.1 ($\alpha + \beta = 1$). Under the same condition and network architecture, we compare the recognition loss and demodulation loss of different $\alpha$ and $\beta$ values, as shown in Fig. 4. Experiments show that when the values of $\alpha$ and $\beta$ are 0.9, 0.1, recognition loss and demodulation loss have the minimum values, indicating that the model is best fitted.

C. THE PROPOSED MT-DCNN VERSUS OTHER METHODS

To evaluate the superiority of MT-DCNN, we calculate the average recognition accuracy obtained by four state-of-the-art methods using traditional demodulators (i.e. a typical three-layer MLP with 128, 64, 32 neurons, TextCNN [23], 1-D CNN [27], and CNN-bidirectional long short-term memory (BLSTM) [24]) and a two-step approach based on CNN (CNN demodulator + CNN classifier [27]). All models are compared using the same training and test data. To the best of our knowledge, there is no CNN-based demodulation and identification scheme tailored for the above tasks. Therefore, the four state-of-the-art methods first use conventional demodulators to obtain the encoded bitstream, then
they identify the encoding type of the bitstream. It is well known that the soft-decision (SD) sequence-based channel coding recognition methods outperform the hard-decision (HD) sequence-based recognition methods. The HD demodulator needs to explicitly decide whether a bit 0 or a bit 1 is transmitted, and it determines the mapping relationship between the accepted symbols and the bits. Compared to SD demodulators, this approach loses the information about the statistical characteristics of channel interference contained in the received signal. It does not take full advantage of the output of the demodulator’s matched filter. Hence, we provide the recognition performance comparisons of HD sequence and SD sequence over CNN, which achieves the best results on the HD sequence.

Figs. 5 and 6 demonstrate the ECC types identification accuracy of different algorithms on the BPSK and QPSK datasets, where the SD result of CNN is given for comparison. All methods based on CNN are able to obtain close to 80% of accuracy at high SNR, while MLP could reach only around 40%. It can be clearly noticed that the MT-DCNN gives a more reliable performance in BPSK and QPSK datasets compared to the existing counterpart, meaning that we can achieve performance gain by implementing intelligent learning. Meanwhile, the two-step approach performs similarly to the CNN (HD) classifier, and they are both inferior to MT-DCNN. This justifies that joint training is a viable method to improve demodulation and recognition performance. Moreover, we see that the SD-based method can attain more accurate identification results than the HD-based method. This result also further illustrates the advantage of the SD demodulator over the HD modulator.

### D. CLASSIFICATION PERFORMANCE BY ECC TYPE

Table 2 lists the recognition accuracy of CNN (HD), CNN (SD), and MT-DCNN for different modulation and coding types at various $E_b/N_0$. It can be seen that the proposed MT-DCNN method attains better results when compared to previous works, especially for turbo codes and TPC codes. For instance, at 0 dB of $E_b/N_0$, the proposed method improves the identification probability by 1% to 15% in BPSK and QPSK cases compared to the best existing methods (CNN (SD)), except for the BPSK-modulated convolutional codes. The results show that the MT-DCNN structure even helps to improve the recognition accuracy of the ECC types. This may be because the bit error rate (BER) performance of the MT-DCNN is superior to that of the traditional demodulator. Table 3 shows the BER performance of a conventional demodulator and the MT-DCNN demodulators. As shown in this table, the MT-DCNN demodulator achieves better BER.

#### TABLE 2. Recognition accuracy of different ECC types versus $E_b/N_0$.

| Type    | $E_b/N_0$ | MT-DCNN | CNN (HD) | CNN (SD) |
|---------|-----------|---------|----------|----------|
|         |           | Conv | BCH | Turbo | TPC | Conv | BCH | Turbo | TPC | Conv | BCH | Turbo | TPC |
| BPSK    | 0 dB      | 0.62  | 0.90 | 0.86  | 0.65 | 0.43 | 0.67 | 0.46  | 0.66 | 0.65 | 0.87 | 0.72  | 0.618 |
|         | 4 dB      | 0.99  | 0.99 | 0.89  | 0.94 | 0.94 | 1.00 | 0.80  | 0.84 | 0.95 | 1.00 | 0.87  | 0.923 |
|         | 8 dB      | 1.00  | 1.00 | 0.98  | 0.98 | 1.00 | 1.00 | 0.98  | 0.81 | 1.00 | 1.00 | 0.97  | 0.968 |
| QPSK    | 0 dB      | 0.74  | 0.95 | 0.90  | 0.80 | 0.37 | 0.64 | 0.32  | 0.73 | 0.58 | 0.87 | 0.77  | 0.678 |
|         | 4 dB      | 0.99  | 1.00 | 0.94  | 0.99 | 0.98 | 1.00 | 0.86  | 0.89 | 0.98 | 1.00 | 0.87  | 0.915 |
|         | 8 dB      | 1.00  | 1.00 | 1.00  | 0.99 | 1.00 | 1.00 | 0.98  | 0.83 | 1.00 | 1.00 | 0.98  | 0.97  |
performance than the conventional demodulator when the SNR of the test sets is higher than 0 dB.

Furthermore, we provide two confusion matrices of average accuracy with the $E_b/N_0$ ranging from 0 dB to 10 dB, facilitating detailed analysis of the performance of our algorithm. As illustrated in Fig. 7, identification accuracy for BPSK and QPSK signals is relatively high. The primary error is that turbo codes and convolutional codes are confused, and TPC codes are misrecognized turbo codes. The main reason is that the component encoder of Turbo code uses a recursive system convolutional encoder (RSC), while both Turbo and TPC codes are cascaded codes.

V. CONCLUSION

In this paper, we proposed a multi-task deep convolutional neural network (MT-DCNN)-based one-stage ECC types recognition method. Interestingly, unlike the traditional two-stage blind recognition of channel coding methods, the proposed method utilizes an end-to-end multi-task architecture capable of demodulating the received IQ signals and identifying the channel coding types simultaneously. The proposed MT-DCNN architecture consists of three modules: the feature extraction module, the demodulation module, and the ECC types recognition module. Experiments show that the proposed one-stage approach can accurately identify coding types of modulated signals, and its performance is superior to that of existing two-stage methods.

In the future, we plan to add a variety of new signal simulation scenarios and demonstrate the applicability of the proposed approach. Also, using simulated data to train deep learning models is not ideal, we intend to train and test the proposed MT-DCNN models using actual signals from real environments. Besides, we will consider more target modulations and ECCs to prove the suitability of the presented method.

REFERENCES

[1] R. Swaminathan and A. S. Madhukumar, “Classification of error correcting codes and estimation of interleaver parameters in a noisy transmission environment,” IEEE Trans. Broadcast., vol. 63, no. 3, pp. 463–478, Sep. 2017.
[2] P. Shachi, R. Mishra, and R. K. Jatoth, “Coherent BPSK demodulator using Costas loop and early-late gate synchronizer,” in Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2013, pp. 1–6.
[3] M. K. Simon and J. C. Springett, “The performance of a noncoherent FSK receiver preceded by a bandpass limiter,” IEEE Trans. Commun., vol. COM-20, no. 6, pp. 1128–1136, Dec. 1972.
[4] D. Ly-Gagnon, S. Tsukamoto, K. Katoh, and K. Kikuchi, “Coherent detection of optical quadrature phase-shift keying signals with carrier phase estimation,” J. Lightw. Technol., vol. 24, no. 1, pp. 12–21, Jan. 2006.
[5] P. Yu, H. Peng, and J. Li, “On blind recognition of channel codes within a candidate set,” IEEE Commun. Lett., vol. 20, no. 4, pp. 736–739, Apr. 2016.
[6] A. D. Yardi, S. Vijayakumaran, and A. Kumar, “Blind reconstruction of binary cyclic codes from unsynchronized bitstream,” IEEE Trans. Commun., vol. 64, no. 7, pp. 2693–2706, Jul. 2016.
[7] S. Ramabadran, A. S. Madhukumar, N. W. Teck, and C. M. S. See, “Parameter estimation of convolutional and helical interleavers in a noisy environment,” IEEE Access, vol. 5, pp. 6151–6167, 2017.
[8] Z. Wu, Z. Zhong, and L. Zhang, “Blind recognition of cyclic codes based on average cosine conformity,” IEEE Trans. Signal Process., vol. 68, pp. 2328–2339, 2020.
[9] R. Swaminathan, A. S. Madhukumar, and G. Wang, “Blind estimation of code parameters for product codes over noisy channel conditions,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 2, pp. 1460–1473, Apr. 2020.
[10] P. Liu, Z. Pan, and J. Lei, “Parameter identification of Reed-Solomon codes based on probability statistics and Galois field Fourier transform,” IEEE Access, vol. 7, pp. 33619–33630, 2019.

[11] T. Xia and H.-C. Wu, “Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability,” IEEE Trans. Signal Process., vol. 62, no. 3, pp. 532–540, Feb. 2014.

[12] S. Ramabadran, A. S. Madhukumar, W. Guoshua, and T. S. Kee, “Blind recognition of LDPC code parameters over erroneous channel conditions,” IET Signal Process., vol. 13, no. 1, pp. 86–95, Feb. 2019.

[13] C. Condo, S. A. Hashemi, and W. J. Gross, “Blind detection with polar codes,” IEEE Commun. Lett., vol. 21, no. 12, pp. 2550–2553, Dec. 2017.

[14] 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.

[15] A. Ali and F. Yangyu, “Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints,” IEEE Signal Process. Lett., vol. 24, no. 11, pp. 1626–1630, Nov. 2017.

[16] T. Huynh-The, Q.-V. Pham, T.-V. Nguyen, T. T. Nguyen, R. Ruby, M. Zeng, and D.-S. Kim, “Automatic modulation classification: A deep architecture survey,” IEEE Access, vol. 9, pp. 142950–142971, 2021.

[17] Q. Zheng, F. Zhao, and H. Wang, “Fine-grained modulation classification using multi-scale radio transformer with dual-channel representation,” IEEE Commun. Lett., vol. 26, no. 6, pp. 1298–1302, Jun. 2022.

[18] J. Cai, F. Gan, X. Cao, and W. Liu, “Signal modulation classification based on the transformer network,” IEEE Trans. Cogn. Commun. Netw., early access, May 20, 2022, doi: 10.1109/TCOCN.2022.3176640.

[19] T. Huynh-The, Q.-V. Pham, T.-V. Nguyen, T. T. Nguyen, D. B. D. Costa, and D.-S. Kim, “RanNet: Learning residual-attention structure in CNNs for automatic modulation classification,” IEEE Wireless Commun. Lett., vol. 11, no. 6, pp. 1243–1247, Jun. 2022.

[20] G. J. Mendis, J. Wei, and A. Madanayake, “Deep learning-based automated modulation classification for cognitive radio,” in Proc. IEEE Int. Conf. Commun. Syst. (ICCS), Dec. 2016, pp. 1–6.

[21] L. Zhang, H. Zhang, Y. Jiang, and Z. Wu, “Intelligent and reliable deep learning LSTM neural networks-based OFDM-DCSK demodulation design,” IEEE Trans. Veh. Technol., vol. 69, no. 12, pp. 16163–16167, Dec. 2020.

[22] M. Zhang, Z. Liu, L. Li, and H. Wang, “Enhanced efficiency BPSK demodulator based on one-dimensional convolutional neural network,” IEEE Access, vol. 6, pp. 26939–26948, 2018.

[23] S. Li, J. Zhou, Z. Huang, and X. Hu, “Recognition of error correcting codes based on CNN with block matrix and embedding,” Digit. Signal Process., vol. 111, Apr. 2021, Art. no. 102986.

[24] S. Zhang, L. Zhou, Y. Tang, and L. Wang, “Blind recognition of channel coding based on CNN-BLSTM,” in Proc. IEEE Int. Conf. Netw. Sens. Control (ICNSC), Dec. 2021, pp. 1–5.

[25] A. Bonvard, S. Houcke, R. Gautier, and M. Marazin, “Classification based on Euclidean distance distribution for blind identification of error correcting codes in noncooperative contexts,” IEEE Trans. Signal Process., vol. 66, no. 10, pp. 2572–2583, May 2018.

[26] J. Wang, J. Li, H. Huang, and H. Wang, “Fine-grained recognition of error correcting codes based on 1-D convolutional neural network,” Digit. Signal Process., vol. 99, Apr. 2020, Art. no. 102668.

[27] L. Li, Z. Huang, C. Liu, and J. Zhou, “Blind recognition of LDPC codes using convolutional neural networks,” in Proc. IEEE 4th Int. Conf. Electron. Technol. (iCET), May 2021, pp. 696–700.

[28] S. Dehodashi, M. Hashemi, and S. Salehkaleybar, “Deep-learning-based channel recognition of channel code parameters over candidate sets under AWGN and multi-path fading conditions,” IEEE Wireless Commun. Lett., vol. 10, no. 5, pp. 1041–1045, May 2021.

[29] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, “Deep learning-based sphere decoding,” IEEE Trans. Wireless Commun., vol. 18, no. 9, pp. 4368–4378, Sep. 2019.

[30] F. Liang, C. Shen, and F. Wu, “An iterative BP-CNN architecture for channel decoding,” IEEE J. Sel. Topics Signal Process., vol. 12, no. 1, pp. 144–159, Feb. 2018.

[31] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be’ery, “Deep learning methods for improved decoding of linear codes,” IEEE J. Sel. Topics Signal Process., vol. 12, no. 1, pp. 119–131, Feb. 2018.

[32] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, “Deep learning-based channel estimation,” IEEE Commun. Lett., vol. 23, no. 4, pp. 652–655, Apr. 2019.

[33] Y. Yang, F. Gao, X. Ma, and S. Zhang, “Deep learning-based channel estimation for doubly selective fading channels,” IEEE Access, vol. 7, pp. 36579–36589, 2019.

[34] H. Ye, L. Liang, G. Y. Li, and B.-H. F. Juang, “Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3133–3143, May 2020.

[35] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, Aug. 2006, doi: 10.1162/neco.2006.18.7.1527.

[36] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 448–456.

[37] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 807–814.

[38] M. Abadi, “TensorFlow: A system for large-scale machine learning,” in Proc. 12th USENIX Symp. Operating Syst. Design Implement., 2016, pp. 265–283.

[39] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent., vol. 12, 2015, pp. 1–9.

Jiao Wang received the B.S. degree in electronic information science and technology from China West Normal University, Nanchong, Sichuan, China, in 2016. She is currently pursuing the Ph.D. degree with the College of Electronic Science and Technology, University of Electronic Science and Technology of China (UESTC). Her research interests include wireless communications, signal processing, deep learning, and channel codes.

Jianqing Li Member, IEEE received the B.S. degree in electronics and information systems from Northeast Normal University, Changchun, Jilin, China, in 1998, and the Ph.D. degree in physical electronics from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2003. He is currently a Professor with UESTC. His research interests include deep learning, computer vision, and signal processing.

Jiaxu Liu received the B.S. degree in integrated circuit design and integrated systems from the University of Electronic Science and Technology of China (UESTC), in 2019, where he is currently pursuing the Ph.D. degree with the College of Electronic Science and Technology. His research interests include wireless communications, signal processing, deep learning, and specific emitter identification.