Modulation Recognition of Underwater Acoustic Signals Using Deep Hybrid Neural Networks

Weilong Zhang, Xinghai Yang, Changli Leng, Jingjing Wang, *Member, IEEE*, and Shiwen Mao, *Fellow, IEEE*

***Abstract*— It is a huge challenge for the receiver to correctly identify the modulation types due to the complex underwater channel environment and severe noise interference. Additionally, the real-time communications have strict requirements in terms of time. In order to solve this well-known issue, in this work, we combine the automatic feature extraction and learning ability of recurrent neural network (RNN) and convolutional neural network (CNN) for designing a modulation recognition model for underwater acoustic signals. The proposed model is based on deep hybrid neural networks called recurrent and convolutional neural network (R&CNN). As compared with the traditional modulation recognition techniques, this method achieves higher recognition accuracy without manual feature extraction. The experimental results show that the validation accuracy of the proposed R&CNN's on the Trestle data set is 98.21%. Similarly, the validation accuracy of the proposed R&CNN's on the South China Sea data set is 99.38%. The average recognition time is 7.164ms. As compared with the conventional deep learning methods, the proposed R&CNN not only has a higher recognition accuracy, but also greatly reduces the recognition time.**

***Index Terms*— Underwater acoustic signal, modulation recognition, deep hybrid neural network, R&CNN.**

I. Introduction

THE demand of data acquisition and processing in various marine applications, such as marine resources development [1]–[3], marine climate monitoring [4]–[6], national territorial sea security [7]–[9], has been quickly rising. The automatic modulation recognition (AMR) method for underwater acoustic communications has been increasingly adopted in civil and military applications. However, the marine environment is highly dynamic and difficult to model. The inhomogeneous underwater transmission medium [10], the temporal and spatial frequency variation in the propagation process [11], and the impulse noise and reverberation in the underwater environment [12] all cause considerable distortions in the received signal. Additionally, these phenomena also lead to inter symbol interference and Doppler frequency shift. All these make it very difficult to identify the modulation of underwater acoustic signals.

The existing AMR techniques can be roughly divided into two categories, namely, likelihood-based (LB) decision theory methods and feature-based (FB) pattern recognition methods. The LB algorithms identify the modulation type of the received signal by comparing the likelihood ratio of the received signal with the discriminant threshold. On the other hand, the FB algorithms extract the distinguishing features from the received signal. Afterwards, the modulation type of the received signal is identified by using a classifier. Among these two aforementioned categories, the FB algorithms are widely used due to their lower complexity and better performance [13]–[15]. The AMR systems that adopt FB algorithms usually comprise two steps, including feature extraction and recognition. In [16], the authors present a technique to extract the instantaneous statistical information from the received signal. This statistical information is then processed using a decision tree to identify the modulation type. In [17], the authors extract the statistical moment and then use a fuzzy classifier to identify the type of modulation. The authors in [18] use the wavelet transform to extract the features from the received signal. These features are then fed into a multilayer neural network to detect the modulation type. The researchers in [19] utilize a support vector machine (SVM) for identifying the type of modulation by extracting cyclo-stationary features and entropy of the signal. However, the accuracy of the FB algorithms largely depends on the effectiveness of feature extraction. It is notable that the marine environment varies significantly in different regions of the sea. Therefore, it is difficult to guarantee the generalization of the FB algorithms in different areas of the sea.

Recently, deep learning has been leveraged by AMR methods [20], [21]. The deep learning algorithms automatically extract the hidden features from the data for performing detection and recognition [22]. In radio modulation recognition, the inverse Fourier transform of interference spectrum is used to reconstruct the augmented signal. The convolutional neural networks (CNN) identify the modulation type of the signal on the basis of the original signal and the augmented signal. The effectiveness of this technique is verified based on the public data set RadioML 2016.10a [21],

Manuscript received 27 June 2021; revised 31 October 2021; accepted 13 January 2022. Date of publication 26 January 2022; date of current version 12 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant U1806201 and Grant 62171246 and in part by the Major Basic Research Project of Natural Science Foundation of Shandong Province under Grant ZR2021ZD12. The associate editor coordinating the review of this article and approving it for publication was Z. Cai. *(Corresponding authors: Jingjing Wang; Xinghai Yang.)*

Weilong Zhang, Xinghai Yang, and Jingjing Wang are with the College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, Shandong 266061, China (e-mail: zhangweilong2015@126.com; yangxh@qust.edu.cn; wangjingjing@qust.edu.cn).

Changli Leng is with the Qingdao Institute of Intelligent Navigation and Control, Qingdao, Shandong 266071, China (e-mail: lengchangli_1992@163.com).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.3144608.

Digital Object Identifier 10.1109/TWC.2022.3144608

[23]. In [24], the authors use the short-time discrete Fourier transform (STFT) to convert one-dimensional radio signal into a spectrum image and then use a CNN to learn its features. The modulation type recognition accuracy of this method, when the signal-to-noise ratio (SNR) is 0 dB, is about 80%. In [25], the authors extract the polar-transformed information of the received signal's *IQ* symbols (the received signal's symbols are real and complex values) and use a depth belief network (DBN) and spiking neural network (SNN) to accurately identify the modulation type under low SNR. In [26], the authors use dual stream structure consisting of CNN and long short-term memory network (LSTM) for estimating the type of modulation. The recognition accuracy of this method is approximately 90% under high SNR.

In short, the AMR methods based on deep learning are mostly designed based on the simulation data or simulated channels. However, the underwater environment is complex and dynamic. As a result, it is quite difficult to simulate the real underwater environment with high fidelity. So, the performance of an algorithm based on simulation data is not necessarily efficient on real-world data. Moreover, the number of layers in a neural network ranges from more than ten to dozens of layers. When there are too many layers, the model is too complex, and there is a high demand for computing power and energy supply, which is not conducive to deployment in underwater communication nodes, which have limited computing power and energy. Therefore, it is of great importance to design and test a fast deep learning-based AMR algorithm, which has a good performance on real-world underwater acoustic signal datasets.

In this paper, we propose a new neural network model, namely recurrent and convolutional neural network (R&CNN), for recognizing the modulation type of underwater acoustic signal. The model is developed to take advantage of the strengths of both recurrent neural network (RNN) and CNN on handling underwater acoustic signal data. We evaluate the proposed network by using two real world sea survey datasets. The results show that the proposed model achieves a higher recognition accuracy over three baseline models. In addition, the proposed model also has fewer parameters and a lower time complexity, making it highly suited for support real-time communication systems. The major contributions of this paper are summarized below.

1) We present a new hybrid architecture that integrates both the recurrent and convolutional layers. The recurrent layers have the capability to extract temporal information of time series data. We use recurrent layers as the shallow layers of R&CNN to extract signal features directly. The ability of spatial feature extraction of convolutional layers compensate the deficiency when the recurrent layer is directly used to identify signal modulation. We adopt the hybrid learning technique to automatically extract the most descriptive features of each type of modulation from underwater acoustic signal for better generalization performance.
2) The convolutional layers of the proposed R&CNN adopt the multi-branch architecture for improving the learning ability. This is accomplished by expanding the width of the proposed architecture, which leads to improved recognition accuracy.
3) We further improve the network architecture by removing the pooling layer in the CNN. This allows us to avoid the loss of features and increase the recognition accuracy. We also improve the design of the convolutional layer by using 1D convolutional kernel instead of 2D convolutional kernel to greatly reduce the computational complexity.

TABLE I
THE TERMS AND THEIR MEANINGS USED IN THIS PAPER

Term	Meaning
CNN	Convolutional neural network.
RNN	Recurrent neural network.
LSTM	Long short-term memory network, a recurrent neural network.
CNN-LSTM	A hybrid neural network. It adopts CNN as the shallow layer of the model and LSTM as the deep layer of the model.
SNR	Ratio of signal power to noise power.
Feature map	Output matrix of neural network layer.
GRU	A feature extraction unit of recurrent neural network.
Loss function	It is the reflection of the fitting degree of the model to the data. The smaller the loss, the better the performance of the neural network.
Optimizer	An algorithm that makes the value of the loss function as small as possible by updating the weights of the network.

The rest of the manuscript is organized as follows. In Section II, we present the underwater acoustic communications system design. In Section III, we present the architecture of the proposed network and the training method. In Section IV, with two underwater acoustic signal datasets obtained from real world sea measurements, we evaluate the effectiveness of the proposed network in identifying the modulation type of underwater acoustic signal. In addition, we also validate the advantages of the proposed network by comparing it with five conventional deep learning algorithms. Finally, in Section V, we conclude this work. Table I presents the terms and their meanings used in this paper.

II. UNDERWATER ACOUSTIC COMMUNICATIONS SYSTEM DESIGN

As presented in Fig. 1, the underwater acoustic communications system consists of three parts, i.e., the transmitter, the underwater acoustic channel, and the receiver. First, the digital signals of different modulation types are converted into analog electrical signals by means of a digital-to-analog converter (DAC). Then, the analog electrical signal is amplified by the power amplifier and is transformed into an acoustic signal by using a transmitting transducer. The acoustic signal is then passed through the underwater acoustic channel where the signal is distorted by noise. The transducer at the receiver's end converts the received acoustic signal into an analog electrical signal. Afterwards, the analog electrical signal is filtered and

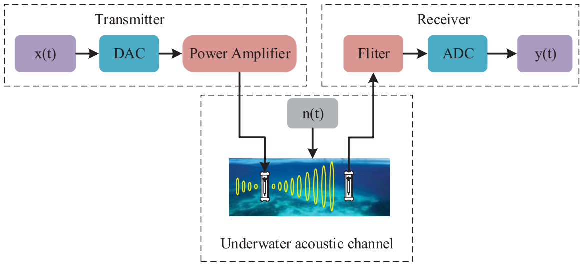

Fig. 1. The underwater communications system model.

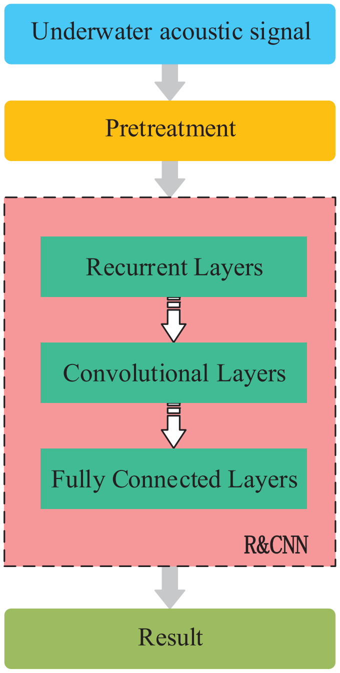

Fig. 2. The flow of information in the proposed R&CNN for identifying the modulation types of underwater acoustic signal.

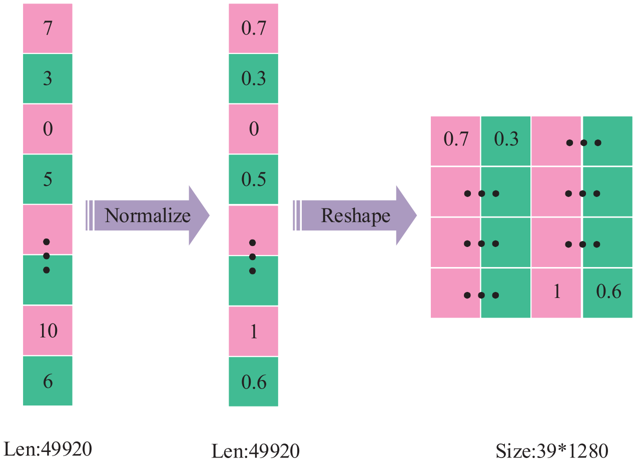

Fig. 3. The preprocessing of underwater acoustic signal.

sampled by an analog-to-digital converter (ADC). Finally, the receiver obtains the signal dataset of various modulation types.

The underwater acoustic communications system can be mathematically expressed as

$$y(t) = \zeta(x(t)) + n(t), \tag{1}$$

where $x(t)$ denotes the modulated signal sent by transmitter, $y(t)$ represents the received signal, $\zeta(\cdot)$ denotes the underwater acoustic channel, and $n(t)$ denotes additive noise at time t.

III. The Proposed R&CNN Model

On the basis of functionality, the network architecture can be divided into feature extraction structure and feature learning structure. It is notable that in the field of signal modulation recognition, the network with hybrid feature extraction structure performs better than the network with single feature extraction structure [27]. In this work, we propose a deep hybrid neural network, termed R&CNN, which is based on RNN and CNN. The proposed network consists of recurrent layers, convolutional layers, and fully connected layers, as shown in Fig. 2.

The process of identifying the modulation types using the proposed R&CNN consists of two major steps. The first step is preprocessing the underwater acoustic signal, and the second is using the preprocessed sample to train the proposed R&CNN model, which are discussed in the following.

A. Preprocessing of Underwater Acoustic Signal

It is necessary to preprocess the underwater acoustic signal before using it to train the network or for inference. The preprocessing procedure includes normalization and reshaping the signal sequence. The data samples are scaled down by using a normalization algorithm and the signal data is mapped to the interval [0, 1]. The normalization algorithm is expressed as

$$S' = F(S) = \frac{S - min}{max - min}, \tag{2}$$

where S denotes the original signal and its length is 49920, S' denotes the normalized signal, and max and min represent the maximum and minimum values of the original signal, respectively. After data normalization, the signal of length 1280 is used as the input of a moment in the R&CNN model. Note that each signal data sample has 39 moments. Based on this, we convert the 1D signal data sample into 2D data with a size of 39 × 1280. The underwater acoustic signal preprocessing is illustrated in Fig. 3.

B. Architecture of the Proposed R&CNN Model

Considering the ability of RNNs to extract temporal information [28], we adopt recurrent layers as the shallow layers of the R&CNN model. This design is to directly extract the features of underwater acoustic signal. Subsequently, the convolutional layers are used to combine the outputs of the second recurrent layer at each time index to extract features again. Finally, the fully connected layers of the proposed R&CNN model learn the features extracted by the convolutional layers and output the recognition results.

1) Recurrent Layers: There are two recurrent layers used in the proposed R&CNN model, which are stacked together. The structure of the recurrent layer is presented in Fig. 4.

In Fig. 4, **Q** denotes the weight matrix of the input layer, **P** denotes the weight matrix of the hidden layer, and **O** denotes

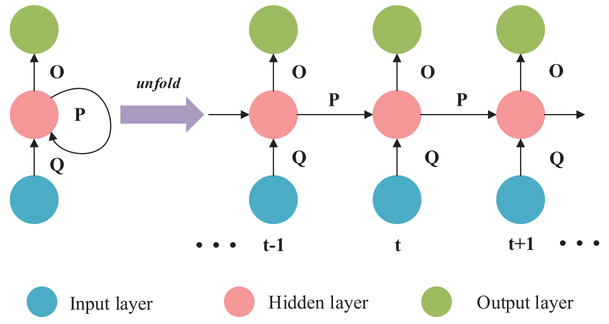

Fig. 4. The basic structure of the recurrent layer.

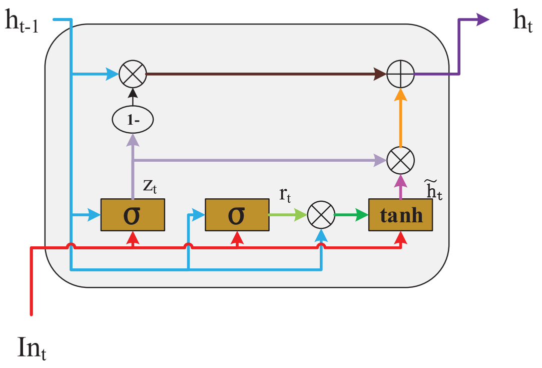

Fig. 5. The architecture of the GRU.

the weight matrix of the output layer. It is noteworthy that in the recurrent layer, the output of the hidden layer neuron is fed back to itself at the next time index. Therefore, as the iterations proceed, the input in the past affects the current output by using the information from the historical data. Therefore, the recurrent layers are suitable for extracting the information form the underwater acoustic signal as the input data is a time series. In underwater acoustic communications, due to the influence of Doppler effect, signal data sequences interfere with each other. Therefore, in this work, we use the gated recurrent unit (GRU) [29] for designing the recurrent layer. The GRU solves the aforementioned problem. The structure of the GRU is presented in Fig. 5.

The following expressions describe the functionality of the GRU.

$$Z_t = \sigma(W_z * [h_{t-1}, In_t] + b_z) \tag{3}$$
$$r_t = \sigma(W_r * [h_{t-1}, In_t] + b_r) \tag{4}$$
$$\tilde{h_t} = tanh(W_h * [r_t * h_{t-1}, In_t] + b_h) \tag{5}$$
$$h_t = (1 - Z_t) * h_{t-1} + Z_t * \tilde{h_t}, \tag{6}$$

where $*$ represents the element-wise multiplication; $tanh$ and σ represent the tangent activation function and sigmoid activation function, respectively; In_t denotes the input at time t; h_{t-1} denotes the output of hidden layer at time $t-1$ and h_t denotes the output of hidden layer at time t; W_z, W_r, and W_h represent the weight matrices; b_r, b_z, and b_h represent the bias terms; Z_t denotes the update gate, which determines the amount of information that should be retained from the past data samples; r_t represents the reset gate, which controls the combination of memory information and input at current time; and $\tilde{h_t}$ denotes the candidate state, which represents the state of the unit. The update gate and reset gate of GRU retain the information in long signal sequences to ensure that the effective information is not eliminated due to the time lapse or irrelevant prediction. In this work, the dimension of the output of the first GRU layer is 640, while the dimension of the output of the second GRU layer is 320. Note that each recurrent layer outputs a complete sequence.

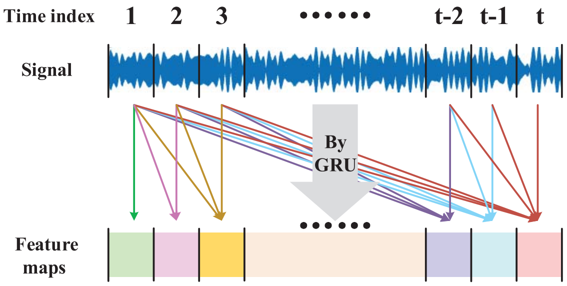

Fig. 6. The process of recurrent layer features extraction.

2) Convolutional Layers: The feature maps extracted by the recurrent layers are used as an input to the convolutional layers. As shown in Fig. 6, the feature map of the i-th time index of the recurrent layer is only related to the signals from the 1^{st} time index to the i-th time index, but not to the signals after the i-th time index. Therefore, only the output feature map of the last time index contains the complete signal, and the output feature map of other time indexes only contains incomplete signal. The conventional methods retain the output feature map of the last time index and discard the output feature map of other time indexes, which actually causes the information loss. In order to make full use of information, we retain the output feature maps of all time indexes and use a CNN to realize the cross-time index information interaction and integration, which helps to enhance the feature extraction ability of the network.

Moreover, due to the specific shape of the output feature map of the second recurrent layer and the temporal characteristics of signals, we improve the design of the convolutional layer. This is done by using 1D convolutional kernel instead of 2D convolutional kernel. On this basis, we integrate the Inception v1 network [30] to increase the network width for improving the learning ability. At the same time, based on less signal features (as compared with image features), we remove the pooling layers to avoid the loss of signal features, since the pooling layers perform dimensionality reduction, thus affecting the network accuracy. Although removing the pooling layer slightly increases the amount of computations, the simple 1D convolutional kernel is used to replace the relatively complex 2D convolutional kernel, which greatly reduces the computational complexity of the network.

There are two convolutional layers in the proposed R&CNN model. Each convolutional layer contains convolutional kernels of different sizes. The architecture of the convolutional layers is shown in Fig. 7. The first convolutional layer consists of three types of convolutional kernels. There are 160 convolutional kernels of each type. The sizes of the three types of convolutional kernels are 8 × 320, 16 × 320, and

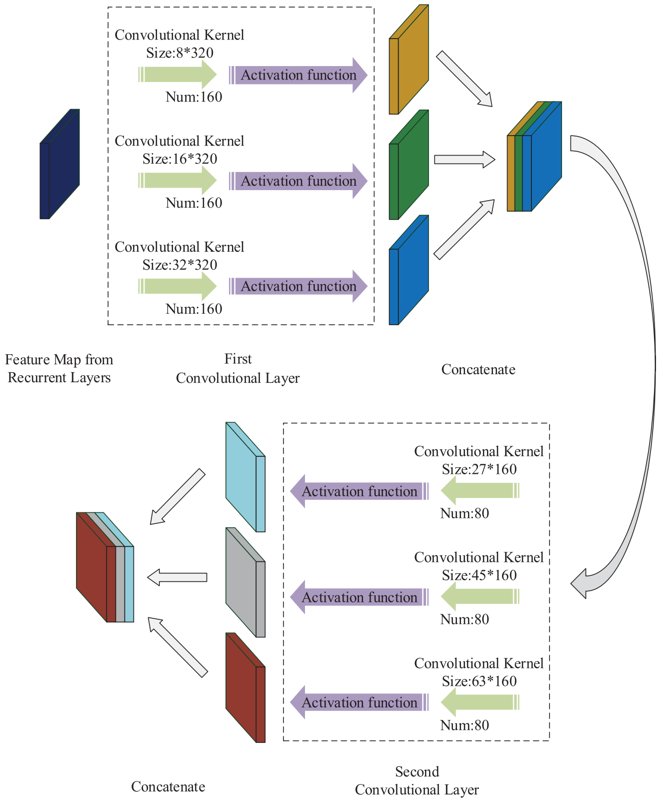

Fig. 7. The architecture of the convolutional layers in the proposed R&CNN model.

32 × 320. The output of the convolutional layers is subject to the activation function. The aforementioned process can be expressed as

$$C\_OUT_{1,j} = C\_A_1(C\_W_{1,j} \otimes C\_IN_1 + C\_B_{1,j}), \quad (7)$$

where C_IN_1 denotes the input of the first convolutional layer; $C\_W_{1,j}$ and $C\_B_{1,j}$ denote the weight matrix and the bias term of the convolutional kernel of type j in the first convolutional layer, respectively; $C\_OUT_{1,j}$ denotes the output feature map of convolutional kernel of type j; $C\_A_1(\cdot)$ denotes the activation function; and $\otimes$ denotes convolution operation. Finally, the output feature maps are concatenated as the input of the second convolutional layer.

Similarly, the second convolutional layer also contains three types of convolutional kernels, each of which has 80 kernels. The sizes of the three types of convolutional kernels are 27 × 160, 45 × 160, and 63 × 160. The process can be expressed as

$$C\_OUT_{2,j} = C\_A_2(C\_W_{2,j} \otimes C\_IN_2 + C\_B_{2,j}) \quad (8)$$

where C_IN_2 denotes the input of the second convolutional layer; $C\_W_{2,j}$ and $C\_B_{2,j}$ denote the weight matrix and the bias term of the convolutional kernel of type j in the second convolutional layer, respectively; $C\_OUT_{2,j}$ denotes the output feature map of the convolutional kernel of type j; $C\_A_2(\cdot)$ denotes the activation function; and $\otimes$ denotes convolution operation. The output feature maps of the second convolutional layer are concatenated and then flattened as input to the fully connected layers.

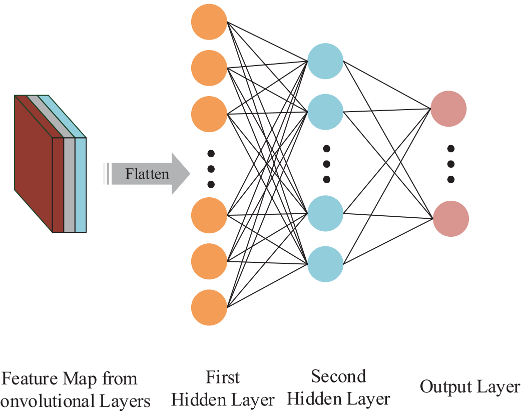

Fig. 8. The structure of the fully connected layers.

3) Fully Connected Layers: There are two fully connected layers in the proposed R&CNN model. The numbers of neurons in these fully connected layers are 120 and 84, respectively. There are N neurons in the output layer, which corresponds to the possible types of modulations, including BPSK, QPSK, BFSK, QFSK, 16QAM, 64QAM, OFDM, and DSSS. After flattening the feature maps obtained by the convolutional layers, they are used as input to the fully connected layers. The output layer provides the recognition results. The architecture of the fully connected layers is shown in Fig. 8.

The fully connected layers can be modeled as

$$F\_OUT = F\_A_2(F\_W_2 * F\_A_1(F\_W_1 * F\_IN_1 + F\_B_1) + F\_B_2), \quad (9)$$

where F_IN_1 denotes the input of the fully connected layers, F_W_1 denotes the weight matrix between the first hidden layer and the second hidden layer, F_W_2 denotes the weight matrix between the second hidden layer and the output layer, F_B_1 and F_B_2 denote the bias terms, and F_A_1 and F_A_2 denote the activation functions.

We use the LeakyReLU activation function in all the hidden layers and the convolutional layers, given by

$$Leaky\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{otherwise}, \end{cases} \quad (10)$$

where the coefficient a is in the interval [0, 1]. The LeakyReLU activation function includes a very small slope for negative value inputs. This mitigates the effect of dead neurons, as the derivative is always non-zero, thus allowing gradient based learning to occur.

The output layer adopts the Softmax activation function, which is given by

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{n=1}^{N} e^{z_n}}, \quad (11)$$

where z_i denotes the input and N denotes the number of neurons in the output layer. The Softmax function enables the neurons in the output layer to output the prediction probabilities of modulation types. The modulation type with the largest prediction probability is chosen as the recognition result of the network.

C. Model Training

In this work, we use the mini-batch gradient descent method [31] to train the neural network. With this method, the training set is divided into several groups of data, and the neural network learns the information from one group of data in each iteration. We choose the cross entropy as the loss function of the proposed network. The adaptive motion estimation (Adam) optimizer [32] is used to update the network parameters.

1) Loss Function: In the training stage, the cross entropy is used as the loss function of neural network. The loss function describes the difference between the actual value and the predicted value, known as the loss. The neural network learns by adjusting its parameters, aiming to reduce the loss to 0. The loss function is described as

$$\mathcal{L}(Y,\hat{Y}) = -\sum_{i=1}^{N} Y_i \cdot \log(\hat{Y}_i), \tag{12}$$

where N denotes the number of neurons in the output layer, the ith neuron outputs the probability value $\hat{Y}_i$ of the corresponding classification in the output layer, and $\sum_{i=1}^{N}\hat{Y}_i = 1$. Y denote the real value, and if Y belongs to class j, then

$$Y_i = \begin{cases} 1, & i = j \\ 0, & i = other \end{cases} \tag{13}$$

Note that the smaller the loss, the closer the predicted value to the ground truth.

2) Optimizer: During the training process, the neural network calculates the gradient and uses it to update the weights of the network. In this work, the gradient of neural network is adjusted by using the Adam optimizer. The Adam optimizer combines the first and the second moments to correct the deviation and adjust the gradient of the neural network. The first moment m_t is mathematically expressed as

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t, \tag{14}$$

$$g_t = \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial w_t}, \tag{15}$$

where g_t denotes the gradient calculated at time t and m_t denotes the first moment at time t. w_t denotes the weights of the network at time t. The first moment is the exponential moving average of the gradient direction at each time index, which is approximately equal to the average of the sum of the gradient vectors over a period of time (time $t - 1/(1-\beta_1)$ to time t).

The second moment V_t is described as

$$V_t = \beta_2 \cdot V_{t-1} + (1-\beta_2) \cdot g_t^2, \tag{16}$$

where V_t represents the second moment at time t. Note that the second moment reflects the gradient change over a period of time.

The empirical values of β_1 and β_2 in this work are set to 0.9 and 0.999, respectively. The initial values of m_0 and V_0 are 0. Thus at the initial states the training process, the values of m_t and V_t are close to 0. The Adam optimizer modifies m_t and V_t to address this problem. The updating mathematical expressions are

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{17}$$

$$\hat{V}_t = \frac{V_t}{1-\beta_2^t}, \tag{18}$$

where $\hat{m}_t$ and $\hat{V}_t$ denote the first and second modified moments, respectively. The gradient of updating the network weights by using Adam optimizer is expressed as

$$\hat{\eta}_t = \frac{\hat{m}_t}{\sqrt{\hat{V}_t}}. \tag{19}$$

The weights of the network are updated as

$$w_{t+1} = w_t - \alpha \cdot \hat{\eta}_t, \tag{20}$$

where w_t denotes the weights of the network at time t and α denotes the learning rate. An appropriate learning rate allows the neural network to converge faster. With each iteration, the neural network updates the weights according to (20) for achieving accurate recognition. The training procedure of R&CNN is illustrated in **Algorithm 1**.

3) Time Complexity: The time complexity of Algorithm 1 is the sum of the time complexity of recurrent layers, convolutional layers and fully connected layers. The time complexity of recurrent layers is described as

$$Time\_r \sim O\left(\sum_{t\_index}\sum_{l\_num} Map\_L_{l\_num-1} Map\_L_{l\_num}\right), \tag{21}$$

where t_index is the time index, l_num is the l_numth layer of neural network, $Map\_L_{l\_num}$ is the length of the output map of l_numth layer.

The time complexity of convolutional layers is described as

$$Time\_c \sim O\Bigg(\sum_{l\_num}\sum_{class}\sum_{Map\_D_{l\_num,class}} s\_kernel_{l\_num,class} * Map\_D_{l\_num-1,class}\Bigg), \tag{22}$$

where $class$ is the type of convolutional kernel and each convolutional layer consists of three types of convolutional kernels, $Map\_D_{l\_num,class}$ is the depth of the output map of type $class$ convolution kernel at l_numth layer, $s\_kernel_{l\_num,class}$ is the size of type $class$ convolution kernel at l_numth layer, $Map\_D_{l\_num-1,class}$ is the length of output map of type $class$ convolution kernel at $l_num - 1$th layer.

The time complexity of fully connected layers is described as

$$Time\_f \sim O\left(\sum_{l\_num} D_{l\_num-1} * D_{l\_num}\right), \tag{23}$$

where D_{l_num} is the output dimension of l_numth layer. And the time complexity of Algorithm 1 is described as

$$Time = Time\_r + Time\_c + Time\_f. \tag{24}$$

Algorithm 1 R&CNN Training Algorithm

```
Input: $Ts = \{ts_i | i = 1, 2, \ldots, N\}$: the training set, the size of $ts_i$ is 39 × 1280   $Y = \{y_i | i = 1, 2, \ldots, N\}$: the labels of the training set
Output: $\hat{Y} = \{\hat{y_i} | i = 1, 2, \ldots, N\}$: the predicted result
for number of epochs to learn training sets do
  % Forward propagation
    % Set Recurrent Layers of the R&CNN.
    $OUT\_GRU_1 = GRU(output\_dimension = 640)(Ts)$
    $OUT\_GRU_2 = GRU(output\_dimension = 320)(OUT\_GRU_1)$
    % Set Convolutional Layers of the R&CNN.
    $OUT\_ConV_{1,1} = Conv1D(filters = 160, kernel\_size = 8, activation = LeakyReLU)(OUT\_GRU_2)$
    $OUT\_ConV_{1,2} = Conv1D(filters = 160, kernel\_size = 16, activation = LeakyReLU)(OUT\_GRU_2)$
    $OUT\_ConV_{1,3} = Conv1D(filters = 160, kernel\_size = 32, activation = LeakyReLU)(OUT\_GRU_2)$
    $OUT\_ConV_1 = concatenate(OUT\_ConV_{1,1}, OUT\_ConV_{1,2}, OUT\_ConV_{1,3})$
    $OUT\_ConV_{2,1} = Conv1D(filters = 80, kernel\_size = 27, activation = LeakyReLU)(OUT\_ConV_1)$
    $OUT\_ConV_{2,2} = Conv1D(filters = 80, kernel\_size = 45, activation = LeakyReLU)(OUT\_ConV_1)$
    $OUT\_ConV_{2,3} = Conv1D(filters = 80, kernel\_size = 63, activation = LeakyReLU)(OUT\_ConV_1)$
    $OUT\_ConV_2 = concatenate(OUT\_ConV_{2,1}, OUT\_ConV_{2,2}, OUT\_ConV_{2,3})$
    % Set Fully Connected Layers of the R&CNN.
    $OUT\_Flatten = Flatten(OUT\_ConV_2)$
    $OUT\_FC_1 = Dense(120, activation = LeakyReLU)(OUT\_Flatten)$
    $OUT\_FC_2 = Dense(84, activation = LeakyReLU)(OUT\_FC_1)$
  $\hat{Y} = Dense(8, activation = Softmax)(OUT\_FC_2)$
  % Back propagation
  % Adam optimizer minimizes $\mathcal{L}(Y, \hat{Y})$ to update the weights $w$ of the R&CNN with learning rate $\alpha$, $t$ is the current number of epochs.
    $\hat{\eta}_t = Adam(\mathcal{L}(Y, \hat{Y}))$
    $w_{t+1} = w_t - \alpha \cdot \hat{\eta}_t$
end for
return $\hat{Y}$
```

4) Training and Validation: In this work, we use tensorflow2.1 for implementing the proposed model. The neural network learns for 50 epochs. During the training process, we use two NVIDIA Titan XP GPUs for accelerating the training process. When evaluating the performance of the neural network, we use a computer with i5-8500 processor and 3.00GHz main frequency. Note that the GPU is not used during the model evaluation process.

IV. Experiment Study

A. Dataset Collection and Description

The underwater acoustic node of the underwater acoustic communication system we developed is shown in Fig. 9. The red part is the transducer, which is used to receive or send underwater acoustic signals. The green part is the data interface, which is used for the communication between the equipment and the host computer. The yellow part includes power supply, power amplifier, filter, etc. for signal processing.

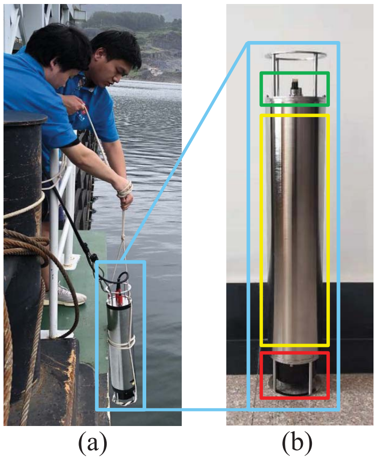
(a) (b)

Fig. 9. The underwater acoustic node of the underwater acoustic communication system.

TABLE II
Parameters of the Communication System and the Environment for the Trestle Dataset

Parameter name	**Parameter value**
Water depth	4.2m
Distance between transmitting and receiving transducers	7m
Bandwidth of transducer	10kHz-20kHz
Distance between transducer and sea surface	2m
Distance between transducer and seabed	2m
DAC sampling rate	336kHz
ADC sampling rate	336kHz
Modulation types of transmission signal	BFSK, QFSK, BPSK, QPSK, 16QAM, 64QAM, OFDM
Sample size	200 signals for each type of modulation, 1,400 signals in total.

The setting and configuration of the two experiments are shown in Tables II-III.

The Trestle underwater acoustic signal dataset was collected on August 13, 2020, in the shallow coastal area of the Trestle scenic spot in the Shinan District, Qingdao City, Shandong Province, China, located at 120°32′7.016″E and 36°6′4.331″N. This Trestle dataset is obtained by using the underwater acoustic communication systems we developed. The modulation types of underwater acoustic signals are identified based on the Trestle dataset. The parameters of the communication system and the environment selected during the collection of the Trestle dataset are presented in Table II. There are seven modulation types of signals in the Trestle data set. In this work, we select 160 signals in each modulation type for training the network. The remaining 40 signals are used as the validation set. There are 1,120 training samples in the training set and 280 data samples in the validation set.

The South China Sea underwater acoustic signal dataset was collected on December 14, 2020, in the South China Sea, located at 109°40′59.117″E and 18°6′37.765″N. The South China Sea dataset was obtained through the same underwater acoustic communication system we developed. The modulation types of underwater acoustic signal are identified based on the South China Sea dataset. The parameters of

TABLE III

PARAMETERS OF THE COMMUNICATION SYSTEM AND THE ENVIRONMENT OF THE SOUTH CHINA SEA DATASET

Parameter name	Parameter value
Water depth	80m
Wind power	Grade 6
Wave height	2-3m
Distance between transmitting and receiving transducers	1km
Bandwidth of transducer	10kHz-20kHz
Distance between the receiving transducer and sea surface	5m
Distance between the transmitting transducer and sea surface	3m
DAC sampling rate	336kHz
ADC sampling rateh	336kHz
Modulation types of transmission signal	BFSK, QFSK, BPSK, QPSK, 16QAM, 64QAM, OFDM, DSSS
Sample size	200 signals of each type of modulation, 1,600 signals in total.

TABLE IV

EXPERIMENTAL RESULTS OF THE SIX NETWORKS WITH THE TRESTLE DATASET

	R&CNN	R&CNN-2D	CNN-LSTM
Validation accuracy	98.21%	94.29%	41.43%
Training accuracy	99.11%	90.09%	30.36%
Validation loss	0.0056	0.4842	1.9808
Training loss	0.0255	0.4045	1.854
	LSTM	**AlexNet8**	**LeNet5**
Validation accuracy	93.57%	92.14%	17.86%
Training accuracy	92.77%	98.75%	11.79%
Validation loss	0.1813	0.2811	1.9459
Training loss	0.1669	0.047	1.9462

the communication system and the environment during the collection of the South China Sea dataset are summarized in Table III. There are eight modulation types of signals in the South China Sea dataset. In this work, we select 160 signals in each modulation type for training the network. The remaining 40 signals are used as the validation set. There are 1,280 training samples in the training set and 320 data samples in the validation set.

B. Results and Discussions

In this paper, we use some popular neural network models as baseline schemes, including LeNet5 [33], AlexNet8 [34], LSTM [35], CNN-LSTM [26] and R&CNN-2D(R&CNN containing two-dimensional convolution kernels). The experimental results of the six neural network models with the Trestle dataset and the South China Sea dataset are presented in Tables IV-VI. Note that the loss is calculated by using the aforementioned entropy based loss function. The loss measures the learning of the neural network corresponding to the input data. The smaller the loss, the better the performance of the neural network.

As presented in Table IV and Table V, the validation accuracy and training accuracy of LeNet5 are the lowest and its losses are the highest. This is because the architecture of the LeNet5 is too simple, due to which it lacks the ability to effectively learn from the underwater acoustic signal dataset. The training and validation losses of LeNet5 with the Trestle and the South China Sea datasets are shown in Figs. 10 and 11, respectively. As shown in Figs. 8 and 10, the validation loss and training loss are both quite stable, indicating that LeNet5 has reached the limit of its learning ability.

TABLE V

EXPERIMENTAL RESULTS OF THE SIX NETWORKS WITH THE SOUTH CHINA SEA DATASET

	R&CNN	R&CNN-2D	CNN-LSTM
Validation accuracy	99.38%	99.54%	96.56%
Training accuracy	99.45%	99.39%	93.28%
Validation loss	1.7105e-05	1.6204e-04	0.1308
Training loss	1.2307e-05	8.5043e-05	0.2102
	LSTM	**AlexNet8**	**LeNet5**
Validation accuracy	96.88%	98.12%	12.81%
Training accuracy	93.83%	88.98%	12.03%
Validation loss	0.1352	0.1893	2.0795
Training loss	0.1923	0.2138	2.0799

TABLE VI

AVERAGE EXECUTION TIME OF EACH NETWORK FOR PERFORMING A SINGLE SIGNAL RECOGNITION TASK

	R&CNN	R&CNN-2D	CNN-LSTM
Ave. execution time	7.164ms	653.321ms	688.161ms
	LSTM	**AlexNet8**	**LeNet5**
Ave. execution time	7.167ms	46.596ms	3.58ms

TABLE VII

TRAINING PARAMETERS OF THE PROPOSED R&CNN MODEL WITH THE TRESTLE AND SOUTH CHINA SEA DATASETS

Parameter	Trestle	South China Sea
Learning rate	0.0001	0.0001
No. of neurons in the output layer	7	8
No. of epochs to learn training sets	50	50
No. of signals divided into a group	14	16

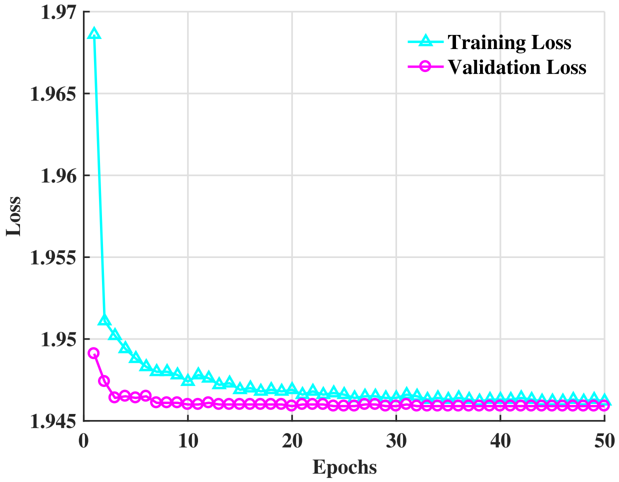

Fig. 10. The training and validation losses of LeNet5 with the Trestle dataset.

With the Trestle dataset, the training accuracy and validation accuracy of AlexNet8 are 98.75% and 92.14%, respectively. The difference between the training accuracy and the validation accuracy of AlexNet8 is 6.61%. With the South China Sea dataset, the training accuracy and the validation accuracy of AlexNet8 are 88.98% and 98.12%, respectively. The difference between the training accuracy and the validation accuracy

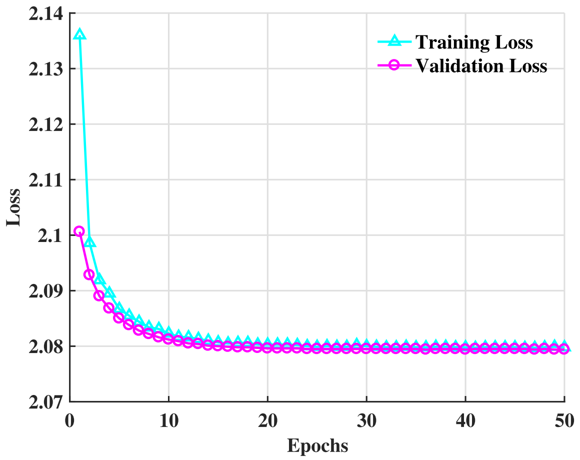

Fig. 11. The training and validation losses of LeNet5 with the South China Sea dataset.

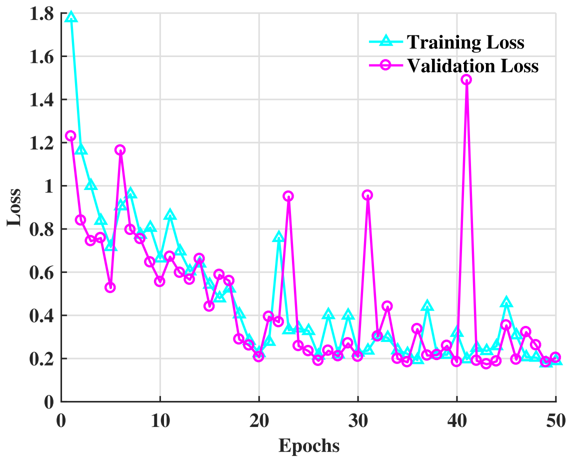

Fig. 13. The training and validation losses of AlexNet8 with the South China Sea dataset.

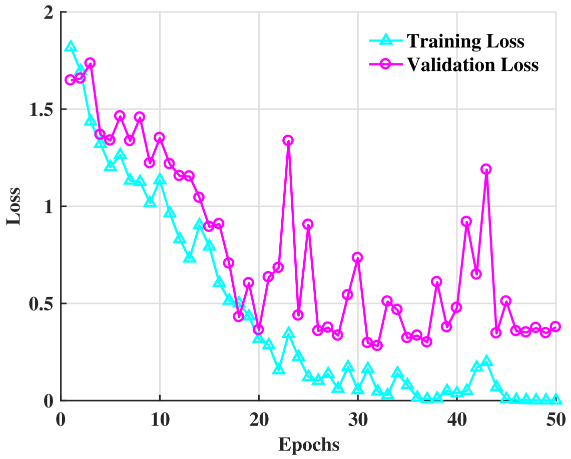

Fig. 12. The training and validation losses of AlexNet8 with the Trestle dataset.

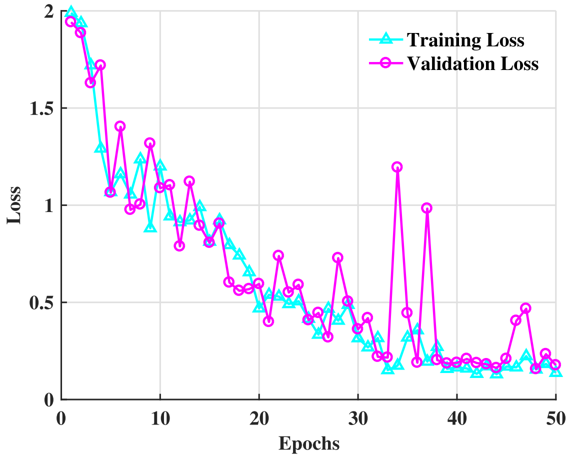

Fig. 14. The training and validation losses of LSTM with the Trestle dataset.

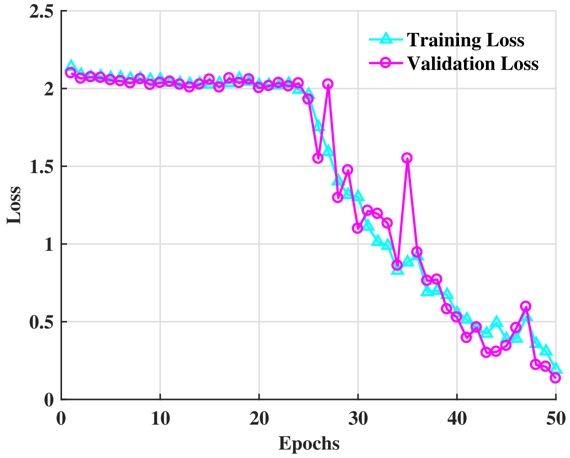

Fig. 15. The training and validation losses of LSTM with the South China Sea dataset.

of AlexNet8 is 9.22%. These results show that AlexNet8 learns the signal data more effectively as compared to LeNet5. However, the architecture of AlexNet8 makes it only be able to learn the local features of data (i.e., the spatial learning ability), such as images, but unable to learn the temporal features of the signals, unlike LSTM and R&CNN. The temporal features of underwater acoustic signal represent the signal characteristics in a better way than the local features. Therefore, the recognition accuracies for the training and validation sets of AlexNet8 are not consistent. Based on the Trestle and the South China Sea datasets, the evolution of loss through epochs are presented in Figs. 12 and 13, respectively.

As shown in Figs. 12 and 13, with the increase in the number of epochs, the training losses of AlexNet8 for Trestle and South China Sea datasets converge to stable values. However, the validation loss is relatively more unstable with large jitters. The results show that the local features of underwater acoustic signal learned by AlexNet8 are not sufficient to describe the modulation characteristics of the signal. Therefore, the training loss and validation loss of AlexNet8 are not very consistent. At the same time, due to the deep network architecture, the average time for AlexNet8 to process one signal is 46.596ms, which is the third highest time complexity among all the six networks. In comparison, the training loss and validation loss of LSTM decline in a relatively more stable manner. Finally, the two losses tend to be consistent, and the difference between the training accuracy and the validation accuracy of LSTM is no more than 3.05%.

These results are shown in Figs. 14 and 15, which show that the signal features extracted by LSTM are more descriptive

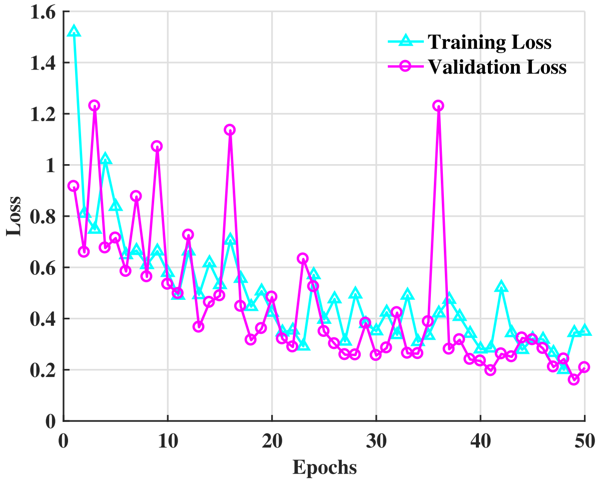

Fig. 16. The training and validation losses of R&CNN with the Trestle dataset.

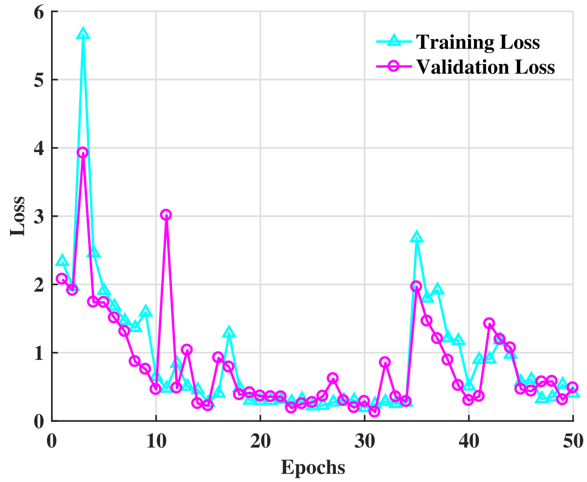

Fig. 18. The training and validation losses of R&CNN-2D with the Trestle dataset.

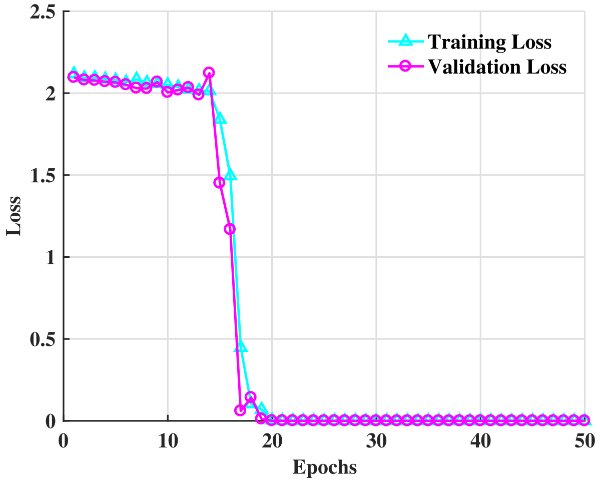

Fig. 17. The training and validation losses of R&CNN with the South China Sea dataset.

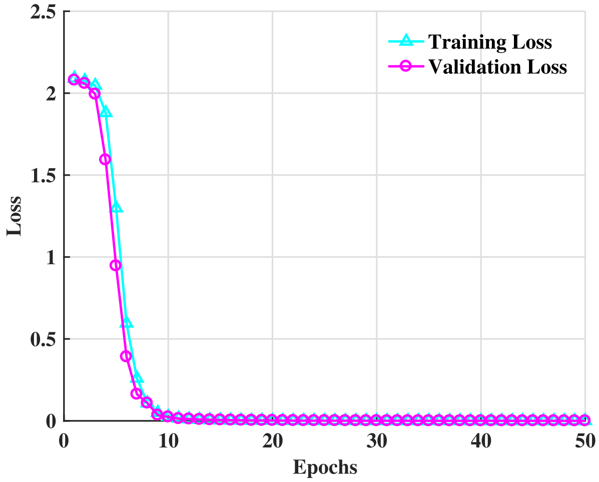

Fig. 19. The training and validation losses of R&CNN-2D with the South China Sea dataset.

and more effectively reflect the characteristics of the underwater acoustic signal modulation. In addition, the average computation time for LSTM to recognize one signal is around 7.167ms, which is second only to R&CNN. This demonstrates the superiority of neural networks which incorporate recurrent layers.

Finally, the training and validation accuracies of R&CNN for the Trestle and the South China Sea datasets are the highest among all the six models. Especially, the training and validation accuracies for the South China Sea dataset are higher than 99%. Figs 16 and 17 show the evolution of the training and validation losses against the number of epochs for R&CNN for the two datasets.

Similar to LSTM, the training and validation losses of R&CNN gradually decrease and approximately converge. At the same time, the training loss of R&CNN is smaller than the other five networks, while the accuracy is the highest. This indicates that the proposed R&CNN has the best learning ability for underwater acoustic signals. In addition, the difference in the recognition accuracies of the validation and the training sets of R&CNN is not more than 0.9%. Note that this is significantly smaller than the 3.05% of LSTM and the 9.22% of AlexNet8. Such high consistency shows that R&CNN has the best generalization ability, and the model can effectively learn the characteristics of underwater acoustic signals for performing accurate recognition. In terms of time complexity, the average time consumed by R&CNN to identify one signal is 7.164ms.

R&CNN-2D is R&CNN containing two-dimensional convolution kernels. Figs. 18 and 19 show the evolution of the training and validation losses against the number of epochs of R&CNN-2D for the two datasets. Similar to R&CNN, the training and validation losses of R&CNN-2D gradually decrease and approximately converge. In addition, the difference in the recognition accuracies of the validation and the training sets of R&CNN is little, and the recognition accuracy for the South China Sea dataset is comparable to that of R&CNN. However, the 2D convolution kernel used in this model greatly increases its computational complexity. The average time consumed by the R&CNN-2D to identify one signal is 653.321ms, which is about 91 times that of R&CNN.

The CNN-LSTM is also a hybrid model. As compared to the proposed R&CNN, it adopts CNN as the shallow layer of the model and LSTM as the deep layer of the model.

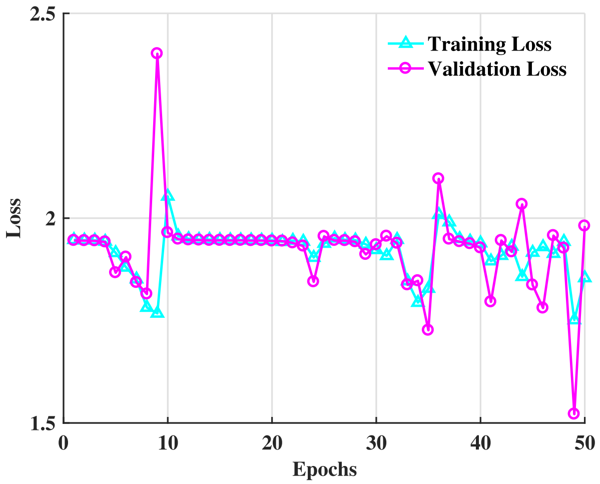

Fig. 20. The training and validation losses of CNN-LSTM with the Trestle dataset.

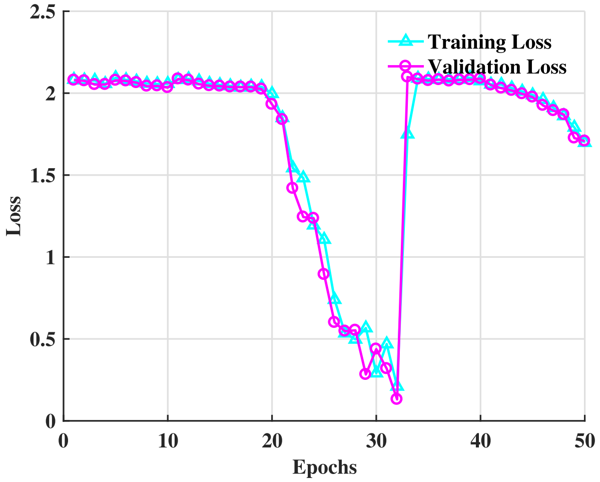

Fig. 21. The training and validation losses of CNN-LSTM with the South China Sea dataset.

Figs. 20 and 21 show the evolution of the training and validation losses against the number of epochs of CNN-LSTM for the two datasets. The training and validation losses of CNN-LSTM for the Trestle dataset steadily decline. The training and validation losses of CNN-LSTM for the South China Sea dataset fall and then suddenly rise. This shows that CNN-LSTM, which adopts CNN as the shallow layer, is unstable and does not have the ability to effectively extract features from the actual underwater acoustic signals. In addition, for the Trestle dataset, the training and validation accuracies of the CNN-LSTM are 30.36% and 41.43%, respectively. For the South China Sea dataset, the training and validation accuracies of the CNN-LSTM are 93.28% and 96.56%, respectively. The CNN-LSTM has great differences in the performance of the two data sets, and the generalization ability of the model is poor. In terms of time complexity, the average time consumed by the CNN-LSTM to identify one signal is 688.161ms, which is the highest time complexity among the compared algorithms.

The results show that the proposed R&CNN has higher recognition accuracy than other single feature extraction models (LeNet5, AlexNet8 and LSTM). As compared with CNN-LSTM, which uses the hybrid feature extraction structure, it has low time complexity and strong generalization. As compared with the R&CNN-2D, its time complexity is much lower and the recognition accuracy is roughly the same. So, the proposed model not only ensures high recognition accuracy, but also has the lowest computational complexity, which meets the real-time requirements of underwater communication systems.

V. Conclusion

In this work, we presented a neural network model termed R&CNN for effectively and accurately identifying the modulation type of underwater acoustic signals. Compared with the traditional automatic modulation recognition methods, the proposed R&CNN does not need to extract the signal features in advance, thus avoiding the problem that the FB algorithms based on simulation face. As compared with the conventional deep learning algorithms, the R&CNN neural network combines the great advantages of recurrent layers in processing the time series data and the spatial learning ability of convolutional layers to mitigate the shortcomings of recurrent layers in feature extraction. The experimental results showed that the proposed R&CNN achieves higher recognition accuracy and better generalization than the traditional LeNet5, AlexNet8, LSTM, and CNN-LSTM models. Additionally, it effectively identified 8 kinds of modulated signals, including BFSK, QFSK, BPSK, QPSK, 16QAM, 64QAM, OFDM, and DSSS. The improvements in the network structure effectively reduce the complexity of the model, which meets the real-time requirements of underwater acoustic communications.

References

[1] J. Xi, S. Yan, L. Xu, and C. Hou, “Sparsity-aware adaptive turbo equalization for underwater acoustic communications in the Mariana Trench,” *IEEE J. Ocean. Eng.*, vol. 46, no. 1, pp. 338–351, Jan. 2021.

[2] N. Xia, Y. Ou, S. Wang, R. Zheng, H. Du, and C. Xu, “Localizability judgment in UWSNs based on skeleton and rigidity theory,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 980–989, Apr. 2017.

[3] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb, “Void-handling techniques for routing protocols in underwater sensor networks: Survey and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 800–827, 2nd Quart. 2017.

[4] Y. Marcon *et al.*, “A rotary sonar for long-term acoustic monitoring of deep-sea gas emissions,” in *Proc. OCEANS*, Marseille, France, Jun. 2019, pp. 1–8.

[5] L. Jing, C. He, J. Huang, and Z. Ding, “Energy management and power allocation for underwater acoustic sensor network,” *IEEE Sensors J.*, vol. 17, no. 19, pp. 6451–6462, Oct. 2017.

[6] G. Han, Z. Tang, Y. He, J. Jiang, and J. A. Ansere, “District partition-based data collection algorithm with event dynamic competition in underwater acoustic sensor networks,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5755–5764, Oct. 2019.

[7] Y. Song, “Underwater acoustic sensor networks with cost efficiency for Internet of Underwater Things,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1707–1716, Feb. 2021.

[8] S. Jiang, “On securing underwater acoustic networks: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 729–752, 1st Quart. 2019.

[9] Y. Noh *et al.*, “HydroCast: Pressure routing for underwater sensor networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 333–347, Jan. 2016.

[10] M. F. Ali, D. N. K. Jayakody, Y. Chursin, S. Affes, and S. Dmitry, “Recent advances and future directions on underwater wireless communications,” *Arch. Comput. Methods Eng.*, vol. 26, no. 100, pp. 1–34, Nov. 2019.

[11] K. Kim, G. Shevlyakov, J. S. Kim, M. Soufian, and L. Statsenko, “Editorial: Underwater acoustics, communications, and information processing,” *MDPI Appl. Sci.*, vol. 9, no. 22, p. 4873, Nov. 2019.

[12] H. Khan, S. A. Hassan, and H. Jung, "On underwater wireless sensor networks routing protocols: A review," *IEEE Sensors J.*, vol. 20, no. 18, pp. 10371–10386, Sep. 2020.

[13] D. Boutte and B. Santhanam, "A hybrid ICA-SVM approach to continuous phase modulation recognition," *IEEE Signal Process. Lett.*, vol. 16, no. 5, pp. 402–405, May 2009.

[14] S. Norouzi, A. Jamshidi, and A. R. Zolghadrasli, "Adaptive modulation recognition based on the evolutionary algorithms," *Appl. Soft Comput.*, vol. 43, pp. 312–319, Jun. 2016.

[15] H. C. Wu, M. Saquib, and Z. Yun, "Novel automatic modulation classification using cumulant features for communications via multipath channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3098–3105, Aug. 2008.

[16] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, Apr. 1998.

[17] J. Lopatka and M. Pedzisz, "Automatic modulation classification using statistical moments and a fuzzy classifier," in *Proc. 5th Int. Conf. Signal Process. (WCCC-ICSP)*, vol. 3, Beijing, China, Aug. 2000, pp. 1500–1506.

[18] M. Walenczykowska and A. Kawalec, "Type of modulation identification using wavelet transform and neural network," *Bull. Polish Acad. Sci.-Tech. Sci.*, vol. 64, no. 1, pp. 257–261, Mar. 2016.

[19] Y. Wei, S. Fang, and X. Wang, "Automatic modulation classification of digital communication signals using SVM based on hybrid features, cyclostationary, and information entropy," *Entropy*, vol. 21, no. 8, p. 745, Jul. 2019.

[20] Y. Tu, Y. Lin, S. Wang, Z. Dou, and S. Mao, "Complex-valued networks for automatic modulation classification," *IEEE Trans. Veh. Technol.*, vol. 89, no. 9, pp. 10085–10089, Sep. 2020.

[21] M. Patel, X. Wang, and S. Mao, "Data augmentation with conditional GAN for automatic modulation classification," in *Proc. 2nd ACM Workshop Wireless Secur. Mach. Learn.*, Linz, Austria, Jul. 2020, pp. 31–36.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[23] Q. Zheng, P. Zhao, Y. Li, H. Wang, and Y. Yang, "Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification," in *Neural Computing and Applications*. Berlin, Germany: Springer-Verlag, Nov. 2020, pp. 1–23.

[24] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 929–932, Jun. 2019.

[25] P. Ghasemzadeh, S. Banerjee, M. Hempel, and H. Sharif, "A novel deep learning and polar transformation framework for an adaptive automatic modulation classification," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13243–13258, Nov. 2020.

[26] Z. Zhang, H. Luo, C. Wang, C. Gan, and Y. Xiang, "Automatic modulation classification using CNN-LSTM based dual-stream structure," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13521–13531, Oct. 2020.

[27] Y. Sang and L. Li, "Application of novel architectures for modulation recognition," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Chengdu, China, Oct. 2018, pp. 159–162.

[28] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*.

[29] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[30] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[31] Q. Qian, R. Jin, J. Yi, L. Zhang, and S. Zhu, "Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (SGD)," in *Machine Learning*. Berlin, Germany: Springer-Verlag, Jul. 2015, vol. 99, no. 3, pp. 353–372.

[32] D. Yi, J. Ahn, and S. Ji, "An effective optimization method for machine learning based on ADAM," *MDPI Appl. Sci.*, vol. 10, no. 3, p. 1073, Feb. 2020.

[33] G. Zhao, B. Pang, Z. Xu, D. Peng, and D. Zuo, "Urban flood susceptibility assessment based on convolutional neural networks," *J. Hydrol.*, vol. 590, Nov. 2020, Art. no. 125235.

[34] S. Lu, Z. Lu, and Y.-D. Zhang, "Pathological brain detection based on AlexNet and transfer learning," *J. Comput. Sci.*, vol. 30, pp. 41–47, Jan. 2019.

[35] G. Van Houdt, C. Mosquera, and G. Napoles, "A review on the long short-term memory model," in *Artificial Intelligence Review*, vol. 53. Berlin, Germany: Springer-Verlag, May 2020, pp. 5929–5955.

Weilong Zhang received the bachelor's degree from Qingdao Technological University, Qingdao, China, in 2019. He is currently pursuing the M.S. degree with the School of Information Science and Technology, Qingdao University of Science and Technology. His current research interests include underwater acoustic communications and artificial intelligence.

Xinghai Yang received the B.S. degree in electronic information science and technology, the M.D. degree in communication and information system, and the Ph.D. degree in communication and information system from Shandong University, China, in 2000, 2003, and 2011, respectively. From 2015 to 2016, he worked as a Visiting Professor at The University of British Columbia, Canada. He is currently an Associate Professor with the School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China. His current research interests include signal processing, deep learning, wireless communications, and the Internet of Things.

Changli Leng received the B.S. degree in communication engineering from the China University of Petroleum (East China), Qingdao, China, in 2015, and the M.Sc. degree in acoustics from Harbin Engineering University, Harbin, China, in 2018. From 2018 to 2020, she worked as an Hydroacoustic Engineer with Zhongke Great Wall Marine Information System Company Ltd. She is currently an Acoustic Engineer with the Qingdao Institute of Intelligent Navigation and Control. Her research interests include underwater acoustic communications, underwater acoustic navigation and positioning, sonar array design, and vibration and noise reduction of underwater structures.

Jingjing Wang (Member, IEEE) received the B.S. degree in industrial automation from Shandong University, Jinan, China, in 1997, the M.Sc. degree in control theory and control engineering from the Qingdao University of Science and Technology, Qingdao, China, in 2002, and the Ph.D. degree in computer application technology, Ocean University of China, Qingdao, in 2012. From 2014 to 2015, she was a Visiting Professor with The University of British Columbia. She is currently a Professor with the School of Information Science and Technology, Qingdao University of Science and Technology. Her research interests include underwater wireless sensor networks, acoustic communications, ultra-wideband radio systems, and MIMO wireless communications.

Shiwen Mao (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Polytechnic University, Brooklyn, NY, USA, in 2004. He is currently a Professor and an Earle C. Williams Eminent Scholar Chair in electrical and computer engineering at Auburn University, Auburn, AL, USA. His research interests include wireless networks, multimedia communications, and smart grid. He is a member of the ACM. He was a co-recipient of the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE ComSoc MMTC 2018 Best Journal Paper Award, the 2017 Best Conference Paper Award, the Best Demo Award from IEEE SECON 2017, the Best Paper Awards from IEEE GLOBECOM 2019, 2016, and 2015, the IEEE WCNC 2015, the IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is on the Editorial Board of the IEEE/CIC China Communications, the IEEE Transactions on Wireless Communications, the IEEE Internet of Things Journal, the IEEE Open Journal of the Communications Society, *ACM GetMobile*, the IEEE Transactions on Cognitive Communications and Networking, the IEEE Transactions on Network Science and Engineering, the IEEE Transactions on Mobile Computing, IEEE Multimedia, IEEE Network, and the IEEE Networking Letters.