Received January 21, 2022, accepted February 3, 2022, date of publication February 15, 2022, date of current version February 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151980

Fully Complex Deep Learning Classifiers for Signal Modulation Recognition in Non-Cooperative Environment

SANGKYU KIM[1], HAE-YONG YANG[2], AND DAEYOUNG KIM[1]
[1]Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Republic of Korea
[2]The Affiliated Institute of Electronics and Telecommunications Research Institute (ETRI), Deajeon 34044, Republic of Korea

Corresponding author: Sangkyu Kim (skkim73@kaist.ac.kr)

This work was supported by the Smart City Research and Development Project of the Korea Agency for Infrastructure Technology Advancement (KAIA) Grant by the Ministry of Land, Infrastructure and Transport (MOLIT), Ministry of Science and ICT (MSIT), under Grant 22NSPS-C149388-05 and Grant 22NSPS-C152143-05.

ABSTRACT Deep learning (DL) classifiers have significantly outperformed traditional likelihood-based or feature-based classifiers for signal modulation recognition in non-cooperative environments. However, despite these recent improvements, the conventional DL classifiers still have an unintended problem in handling the received signal in which the in-phase and quadrature components are separated. Even though the two components seem to be individually uncorrelated to each other, they are definitely the theoretical real and imaginary parts of a signal sample in the complex domain. Thus, it may be helpful for a classifier to regard and treat the modulated signal as a complex data array representing beneficial mutual information between the two real data arrays. In this paper, we propose two types of fully complex convolutional neural network (CNN) and residual neural network (ResNet) classifiers that deal with the complex data instead of the two separated data. First, we organize and define the core complex operations for implementing the complex DL classifiers. Next, the architectures of the proposed classifiers are realized by applying the structural optimizations and the regularization technique. Then, the various aspects of the classifiers' performance are analyzed and explained for providing comprehensive and deep understandings: (a) the effectiveness of the complex signal handling, (b) the in-depth evaluation of classification accuracy, (c) the impact of the data size on performance, and (d) the computational complexity. The experimental results show that the proposed classifiers can provide the faster learning speed, the higher optimization, and the better generalization with sacrificing acceptable learning and classification costs. Especially, our approaches remarkably improve the performance on the mutually correlated modulation types (i.e., the phase-related modulations: PSK, APSK, and QAM) even in the less-scale datasets.

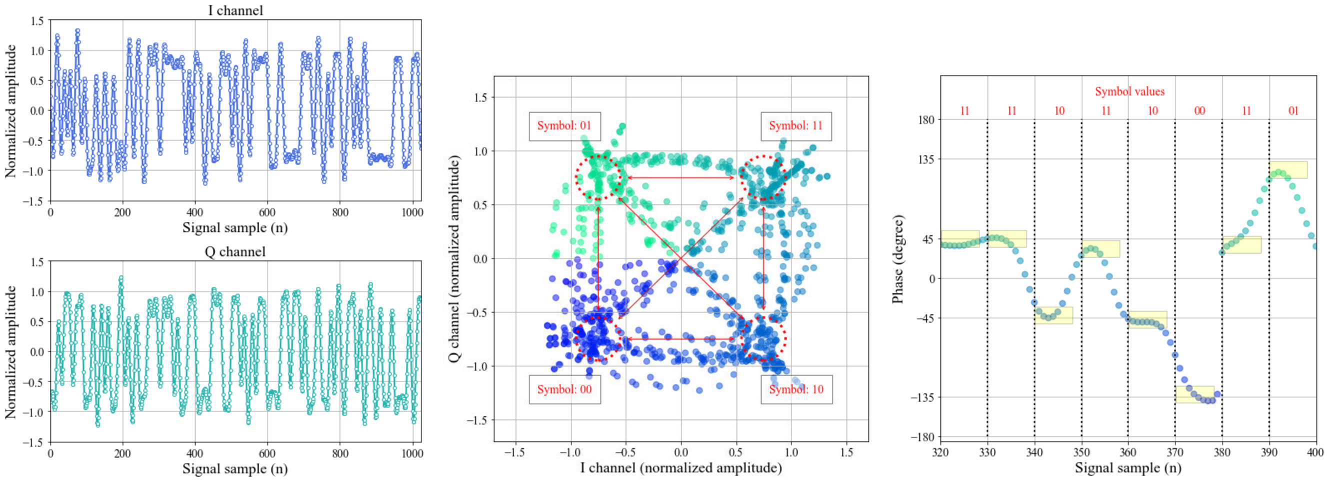
INDEX TERMS Complex deep learning, signal modulation recognition, non-cooperative environment.

I. INTRODUCTION

Signal modulation recognition (SMR) is a technique that can effectually classify the modulation type of an observed signal. This technique was initially developed for military purposes and has been still widely used in various applications, including the jamming attack and the radar signal reconnaissance [1], [2]. Also, in civilian applications, SMR has been perceived as one of the important technologies for transmission optimization and interference mitigation in cognitive radio systems, as well as for the spectrum detection of WiFi or drone signals [3]–[5].

Recently, the research importance of the SMR effectiveness in non-cooperative (or blind) environments has been emphasized. In these environments, any sensitive signal information, such as modulation type, transmit power, and frequency/phase deviation, cannot be readily obtained. For example, it is inherently impossible to acquire the information from the opponents beforehand in military applications. Even in civilian applications, it is also challenging to obtain the stationary prior information because the signal properties are dynamically changed to achieve the efficient management

The associate editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero.



(a) (b) (c)

FIGURE 1. Various representations of a QPSK signal (within the radioML2018.01a dataset [16]): (a) separated I and Q channels, (b) constellation diagram for I and Q channels, and (c) phase variation.

of the frequency spectrum resources [6], [7]. Moreover, modern wireless communication systems have become increasingly complicated by the techniques of anti-multipath fading, frequency selection, time-varying channel, etc. Accordingly, a more accurate and robust SMR method is required to counteract the upcoming harsh environments.

A few years ago, deep learning (DL) classifiers emerged as one of the new SMR alternatives. The several DL classifiers [8]–[10] showed positive feasibilities by applying the well-known DL architectures, whose performance has been validated in other fields (e.g., computer vision). Besides, the more advanced DL classifiers have outperformed both the traditional likelihood-based (LB) and feature-based (FB) classifiers in the noisy environments [11], [12]. Furthermore, the studies in [13]–[16] have proved that their classifiers could distinguish enough even when the modulation types are complicated and diverse.

However, despite their performance enhancement, the above-mentioned DL classifiers have an unexpected problem dealing with the dataset. In general, released datasets mostly consist of modulated signals with the separated in-phase (I) and quadrature (Q) channel data. Although the two channel data seem irrelevant to each other, these values are theoretically real and imaginary parts of complex signals [17]. Let's take a quadrature phase-shift keying (QPSK) modulation example. From Fig. 1 representing the separated I and Q channels of the QPSK modulated signal, we may identify only the straightforward information such as the signal's amplitude, shape, and period; as a result, it is not easy to determine at a glance whether the modulation type is QPSK or not. On the other hand, if the two separated channels are treated as the complex data, we can effectively recognize its modulation type as QPSK through the shape of the constellation diagram or the phase variation, as shown in Figs. 1b and 1c. Unfortunately, the conventional classifiers have overlooked the benefits of complex-valued information.

In this context, the major motivation is the valuable insight that the classifier's accuracy may be further enhanced if we regard two seemingly unrelated data as one mutually correlated meaningful data (i.e., a complex number). Also, the other encouraging motivation is the various positive effects on handling the complex number; the studies in [18], [19] have shown that the complex-valued data handling could provide faster learning, higher optimization, and better generalization. Although these results have been based on simple and shallow neural networks, it is sufficient to imply that the positive effects may be obtainable even in DL.

In this paper, we propose the two types of fully complex-valued classifiers based on convolutional neural network (CNN) and residual neural network (ResNet) architectures. At the same time, the aim of our study is to provide comprehensive experimental results and careful explanations regarding how the complex-valued data management delivers helpful benefits to the DL classifiers.

The main contributions of this study are summarized as follows:

- We newly define the complex-valued (a) max pooling and (b) softmax operations to actualize the proposed classifiers; also, (c) the complex-valued gradient-weighted class activation map (Grad-CAM) is introduced to verify the benefit of the complex data handling.
- We propose the architectures of the two complex-valued classifiers improving the SMR capability by using the structural optimization and regularization techniques.
- We provide thoughtful analyses and detailed descriptions about the performance of the suggested classifiers in various aspects, such as the effectiveness of the complex signal handling, the in-depth evaluation of the classification accuracy, the impact of the data size on performance, and the computational complexity.

The rest of this paper is organized as follows: Section II introduces recent research trends in SMR. In Section III, the core complex-valued operations are defined. Based on these operations, we propose the complex-valued CNN and ResNet classifiers in Section IV. Then, Section V presents the experimental results and discussions. Finally, the conclusion is given in Section VI.

II. RELATED WORK

Technically, SMR comprises pre-processing and classification processing steps. The purpose of the pre-processing is to make a received signal suitable (e.g., normalization) for the next step. Then, the classification processing tries to extract significant features and finally determine the best one among possible candidates. In general, the SMR algorithms are categorized in LB, FB, and DL approaches according to the inner classification processing way.

A. LIKELIHOOD-BASED SMR

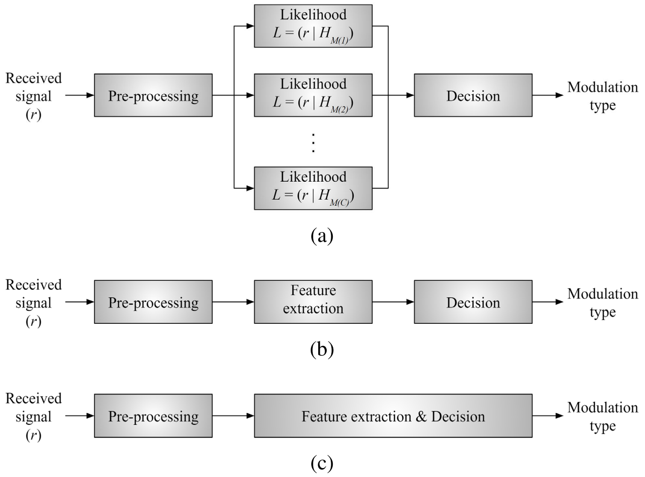
LB SMR is an algorithm conducting multiple hypothesis test problems [7] based on the likelihood function of the observed signal r, as shown in Fig. 2. The algorithm assumes a set $\mathfrak{M}$ in which C finite candidate modulation types are available. Then, the LB classifier estimates likelihoods $L(r|H_{M(c)})$ according to a hypothesis $H_{M(c)}$ for each modulation type $M(c)$, where $1 \leq c \leq C$. The next decision step involves comparing all the estimated likelihood values and making a classification decision. An intuitive and straightforward approach may be to choose the expected modulation type $\hat{M}$ representing the maximum likelihood as follow:

$$\hat{M} = \underset{M(c)\in\mathfrak{M}}{\operatorname{argmax}} L(r|H_{M(c)}). \tag{1}$$

The advantage of the LB classifier is that it can provide the optimal solution under the following ideal assumptions: (a) all modulation properties are already known, and (b) the estimation of the wireless channel model is perfect. However, employing the LB classifier in a realistic environment is challenging because the ideal conditions cannot be achievable in the non-cooperative environments. Moreover, such a classifier inevitably suffers from high computational complexity as the number of classification types increases.

B. FEATURE-BASED SMR

A typical FB classifier has the cascade structure of the feature extraction and decision steps, as shown in Fig. 2b. Various approaches have been studied to extract the following unique features for modulated signals: spectral features (e.g., amplitude and phase according to frequency.) [20], wavelet features (e.g., time-frequency spectrum) [21], high-order statistics (HoS) features (e.g., third or higher cumulants) [22], cyclostationary features (e.g., periodic spectrum properties) [23], and other features (e.g., entropy [24], constellation shape [25], and zero-crossing [26]), etc. The most straightforward approach in the decision step is the decision tree technique [7]. Furthermore, it has been emerged more advanced techniques such as K-nearest neighbors [27], support vector machines (SVMs) [28], or artificial neural networks (ANNs) [29] for better performance.



(a)

(b)

(c)

FIGURE 2. Flow diagrams of three different types for SMR: (a) LB classifier, (b) FB classifier, and (c) DL classifier.

The FB classifier typically shows a relatively low computational complexity than the LB classifier. However, its classification accuracy may be sub-optimal because the features of specific modulation types are manually handcrafted depending on the expert's experience. Besides, as the number of classification types enlarges, the increment in the system complexity is also unavoidable due to the demands for more simultaneous classification processings. Consequently, it may be challenging to apply the FB classifier in the real-world environment where various modulation types need to be expanded promptly and classified accurately.

C. DEEP LEARNING SMR

Recently, the classifiers based on DL techniques have overcome the limitations of the LB and FB classifiers. Unlike the two classifiers, the DL classifier merges the feature extraction (or likelihood) and decision steps in the single process, as shown in Fig. 2c. The fusion of the two steps in DL provides the following benefits: (a) multiple layer-based learning from low-level features to high-level abstracts features [30] and (b) capability of variation factors handling [31].

The early DL approaches in SMR tried to apply the famous architectural concepts that showed impressive performance in other fields. For example, some DL classifiers in [8]–[10] adapted various models, such as the visual geometry group (VGG), convolutional long short-term deep neural network (CLDNN), ResNet, densely connected network (DenseNet), and long short-term memory (LSTM). Although these classifiers were relatively simple, their performance results have been sufficient to show a positive prospect for SMR using DL.

Subsequently, several studies have demonstrated that the DL classifiers outperformed the traditional LB and FB classifiers in the sensitive environments. In [11], the suggested

LSTM-based classifier showed better accuracy than the LB classifier at the impairment channels (e.g., frequency offset). Also, the CNN-based classifier in [12] surpassed the HoS-based FB classifiers in the incoherent environment.

Meanwhile, the other researches have attempted to handle more signal modulation types accurately. In [13], it was proposed the serial or parallel fusion classifiers based on CNN and LSTM; these classifiers could handle 11 different modulation types. In addition, in [14], the CNN-based classifier, capable of recognizing 15 different modulation types, was proposed. Furthermore, [15] proposed the ResNet-based classifier, which discriminated among 19 different modulation types. Recently, the most remarkable radioML2018.01a dataset [16] was released. Using this dataset, the ResNet and CNN based classifiers suggested in [16] could recognize 24 different modulation types.

III. COMPLEX-VALUED OPERATIONS

This section mathematically represents the core complex-valued operations to realize the complex-valued classifiers. In Sub-sections III-B to F, the five pre-defined complex-valued operations are organized: the complex-valued full connection (CFC), convolution, batch normalization (CBN), ReLU (CReLU), and average pooling (CAP). After that, we propose the two new operations: complex-valued max pooling (CMP) and softmax (CSoftmax) in Sub-sections III-G and H.

A. COMPLEX DATA REPRESENTATION

In the subsequent sub-sections, an arbitrary complex data array (or matrix) $\boldsymbol{r}$ is represented by a real data array (or matrix) $\boldsymbol{h}$ with the real and imaginary parts of $\boldsymbol{r}$, as follow:

$$\boldsymbol{r} = \Re(\boldsymbol{r}) + j\Im(\boldsymbol{r}) \Leftrightarrow \boldsymbol{h} = [\Re(\boldsymbol{r}), \Im(\boldsymbol{r})]. \tag{2}$$

Using this representation, all complex-valued operations can be logically conducted with the real data and the already well-defined real functions (e.g., matrix multiplication, max, argmax, etc.).

B. COMPLEX-VALUED FULL CONNECTION

The full connection is one of the most important operations in a DL model. The concept of real-valued full connection (RFC), described as the product of an input and a trainable parameter (also known as a weight), can be extended to CFC.

Let z_i and $\tilde{z}_o$ be the i^{th} element of a complex input $\boldsymbol{z}$ with the length I and the o^{th} element of a complex output $\tilde{\boldsymbol{z}}$ with the length O, respectively, where $i = 1, 2, \ldots, I$ and $o = 1, 2, \ldots, O$. Depending on I and O, a weight $\boldsymbol{w}$ becomes the $I \times O$ complex matrix; an i^{th} row and o^{th} column element of $\boldsymbol{w}$ is notated as $w_{i,o}$. Then, $\tilde{z}_o$ can be defined by (3) [32].

$$\begin{aligned}
\tilde{z}_o &= \sum_{i=1}^{I} w_{i,o} z_i \\
&= \sum_{i=1}^{I} \left(\Re(w_{i,o}) + j\Im(w_{i,o})\right)\left(\Re(z_i) + j\Im(z_i)\right) \\
&= \left(\sum_{i=1}^{I} \Re(w_{i,o})\Re(z_i) - \sum_{i=1}^{I} \Im(w_{i,o})\Im(z_i)\right) \\
&\quad + j\left(\sum_{i=1}^{I} \Im(w_{i,o})\Re(z_i) + \sum_{i=1}^{I} \Re(w_{i,o})\Im(z_i)\right).
\end{aligned} \tag{3}$$

Equation(3) indicates that CFC can be carried out as the real-valued matrix multiplications for the real-valued components of $\boldsymbol{z}$ and $\boldsymbol{w}$.

Using the complex data representation in Sub-section III-A, $\tilde{\boldsymbol{z}}$ can be expressed as follow:

$$\begin{bmatrix} \Re(\tilde{\boldsymbol{z}}^T) \\ \Im(\tilde{\boldsymbol{z}}^T) \end{bmatrix} = \begin{bmatrix} \Re(\boldsymbol{w}^T) & -\Im(\boldsymbol{w}^T) \\ \Im(\boldsymbol{w}^T) & \Re(\boldsymbol{w}^T) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{z}^T) \\ \Im(\boldsymbol{z}^T) \end{bmatrix}, \tag{4}$$

where the symbol T denotes the transpose of a data array (or a matrix). The bias terms in (3) and (4) are omitted to simplify the equations without losing generality.

C. COMPLEX-VALUED CONVOLUTION

The real-valued convolution (RCONV) is another vital operation to extract the key features of the modulated signals. CCONV can be extended from the RCONV idea, which is explained as the behavior of the kernel scanning for the input.

Considering the number of channels C, it is assumed that $z_{i,c}$ is the i^{th} element at the c^{th} channel of an $I \times C$ complex input $\boldsymbol{z}$, where $i = 1, 2, \ldots, I$ and $c = 1, 2, \ldots, C$. Also, for the number of kernels N and the kernel length K, a convolutional kernel $\boldsymbol{w}$ is defined as the $N \times K \times C$ complex matrix; in $\boldsymbol{w}$, $w_{n,k,c}$ denotes the k^{th} element in the n^{th} kernel of the c^{th} channel, where $n = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$. Then, $\tilde{z}_{o,m}$, which is the o^{th} complex output element at the m^{th} channel, can be given as (5) [32], [33].

$$\begin{aligned}
\tilde{z}_{o,m} &= \sum_{c=1}^{C}\sum_{k=1}^{K} w_{m,k,c} z_{L+k,c}, \quad \text{where } L \triangleq o \cdot S - P - 1 \\
&= \sum_{c=1}^{C}\sum_{k=1}^{K} \left(\Re(w_{m,k,c}) + j\Im(w_{m,k,c})\right) \\
&\quad \times \left(\Re(z_{L+k,c}) + j\Im(z_{L+k,c})\right) \\
&= \left(\sum_{c=1}^{C}\sum_{k=1}^{K} \Re(w_{m,k,c})\Re(z_{L+k,c}) \right. \\
&\qquad \left. - \sum_{c=1}^{C}\sum_{k=1}^{K} \Im(w_{m,k,c})\Im(z_{L+k,c})\right) \\
&\quad + j\left(\sum_{c=1}^{C}\sum_{k=1}^{K} \Im(w_{m,k,c})\Re(z_{L+k,c}) \right. \\
&\qquad \left. + \sum_{c=1}^{C}\sum_{k=1}^{K} \Re(w_{m,k,c})\Im(z_{L+k,c})\right),
\end{aligned} \tag{5}$$

where $o = 1, 2, \ldots, O$ and $m = 1, 2, \ldots, N$; P and S indicate the padding and stride sizes, respectively; O can be determined from $\frac{1-K+2P}{S} + 1$. To simplify (5) without loss of generality, the bias components are ignored. Similar to CFC, CCONV can also be realized using the matrix multiplications and the real data.

D. COMPLEX-VALUED BATCH NORMALIZATION

The batch normalization allows a DL model to be trained stably by using the input normalization. Following [33], CBN can be developed from the notion of the real-valued batch normalization (RBN).

Suppose that $\mathbf{Z} = (z^1, \ldots, z^b, \ldots, z^B)$ is the input batch set with the batch size of B, where z^b is the complex input at the b^{th} batch. Then, $\tilde{z}^b$ (i.e., the complex output at the b^{th} batch) can be defined as follow [33]:

$$\begin{bmatrix}\Re(\tilde{z}^b)\\ \Im(\tilde{z}^b)\end{bmatrix} = \gamma V^{-\frac{1}{2}}\begin{bmatrix}\Re(z^b)-\Re(\mu)\\ \Im(z^b)-\Im(\mu)\end{bmatrix} + \begin{bmatrix}\Re(\beta)\\ \Im(\beta)\end{bmatrix}, \tag{6}$$

where $b = 1, 2, \ldots, B$. In (6), μ denotes the complex mean of $\mathbf{Z}$ and the covariance matrix V is defined by

$$V = \begin{bmatrix} Cov(\Re(\mathbf{Z}), \Re(\mathbf{Z})) & Cov(\Re(\mathbf{Z}), \Im(\mathbf{Z}))\\ Cov(\Im(\mathbf{Z}), \Re(\mathbf{Z})) & Cov(\Im(\mathbf{Z}), \Im(\mathbf{Z}))\end{bmatrix}. \tag{7}$$

β is the complex trainable shift parameter and γ is the 2 × 2 complex trainable scaling matrix as follow:

$$\gamma = \begin{bmatrix}\gamma_{re,re} & \gamma_{re,im}\\ \gamma_{im,re} & \gamma_{im,im}\end{bmatrix}. \tag{8}$$

For the complex normalized output with mean 0 and variance 1, γ and β are initialized to $\gamma_{re,re} = \gamma_{im,im} = \frac{1}{\sqrt{2}}$, $\gamma_{re,im} = \gamma_{im,re} = 0$, and $\Re(\beta) = \Im(\beta) = 0$.

E. COMPLEX-VALUED ReLU

One of the most successful activations is the rectified linear unit (ReLU), holding only the positive values in the input. This real-valued ReLU (RReLU) principle can be utilized to design CReLU.

Let us z_i be the i^{th} element of a complex input z with the length I, where $i = 1, 2, \ldots, I$. Then, for z_i, an o^{th} complex output element $\tilde{z}_o$ can be determined by (9) [32], [33].

$$\begin{aligned}\tilde{z}_o &= max(0, \Re(z_o)) + jmax(0, \Im(z_o))\\ &= \begin{cases} z_o, & \text{if } \Re(z_o) \geq 0,\ \Im(z_o) \geq 0\\ \Re(z_o), & \text{if } \Re(z_o) \geq 0,\ \Im(z_o) < 0\\ \Im(z_o), & \text{if } \Re(z_o) < 0,\ \Im(z_o) \geq 0\\ 0, & \text{otherwise,}\end{cases}\end{aligned} \tag{9}$$

where $o = 1, 2, \ldots, I$. To simplify (9) without losing generality, the input channel is assumed to be one.

F. COMPLEX-VALUED AVERAGE POOLING

The average and max poolings are the most representative operations in DL. For fulfilling CAP in the complex domain, the real-valued average pooling (RAP) concept, which averages all entries within the pooling size, can be applied.

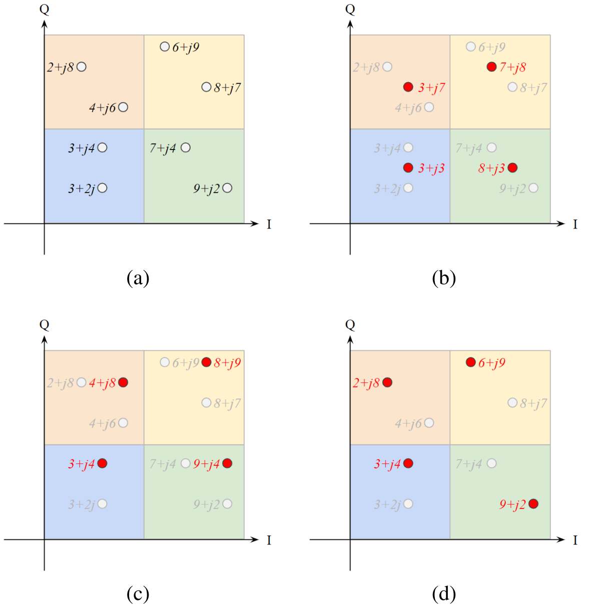


FIGURE 3. Various complex-valued pooling results with $G = 2$ and $S = 2$: (a) pooling input, (b) CAP results, (c) CMP results with separate max operations, and (d) CMP results with magnitude.

Like in CReLU, it is regarded that z_i is the i^{th} element of a complex input z of the length I, where $i = 1, 2, \ldots, I$; also, the input channel size is supposed to be one. Then, given the pooling size G and the stride size S, an o^{th} complex output element $\tilde{z}_o$ can be defined as (10) [32].

$$\begin{aligned}\tilde{z}_o &= \frac{1}{G}\sum_{g=1}^{G} z_{o\cdot S+g}\\ &= \frac{1}{G}\sum_{g=1}^{G}\Re(z_{o\cdot S+g}) + j\frac{1}{G}\sum_{g=1}^{G}\Im(z_{o\cdot S+g}),\end{aligned} \tag{10}$$

where $o = 1, 2, \ldots, O$, and O can be derived from $\frac{I-G}{S} + 1$. Equation (10) permits CAP to derive the final output from the separate real-valued averages. For example, given the input in Fig. 3a, the CAP result becomes the red dots in Fig. 3b.

G. PROPOSED: COMPLEX-VALUED MAX POOLING

The real-valued max pooling (RMP) chooses only the maximum value among the pooling entries. When considering the complex extension of RMP, we may imagine a simple way of separately applying the max functions to a complex data's real and imaginary parts, respectively. However, this CMP with the separated max functions results in unavoidable data distortion, as shown in Fig. 3c; i.e., it does not seem that the outputs have statistical meanings. To achieve a stable CMP, we utilize the physically meaningful quantity of the complex data: *the magnitude* of a complex number.

Assume the complex input, the pooling size, and the stride size are identical to those of CAP. Then, a complex output element $\tilde{z}_o$ can be represented as follow:

$$\begin{aligned}\tilde{z}_o &= \underset{g=1,2,\ldots,G}{\text{argmax}}\ |z_{o\cdot S+g}|\\ &= \underset{g=1,2,\ldots,G}{\text{argmax}}\ \sqrt{\Re(z_{o\cdot S+g})^2 + \Im(z_{o\cdot S+g})^2},\end{aligned} \tag{11}$$

where $o = 1, 2, \ldots, O$. As shown in Fig. 3d, the CMP output with the magnitude does not introduce the signal distortion; also, it corresponds to the original intention of RMP.

H. PROPOSED: COMPLEX-VALUED SOFTMAX

Typically, the softmax is the last step of a DL model to normalize the prediction results into the probability distribution; consequently, each output entry of the softmax has a probability between 0 and 1, and the sum of all entries is 1. We extend the real-valued softmax (RSoftmax) to CSoftmax by applying the magnitude of the complex data as in CMP.

Suppose z_i is defined as the i^{th} element of a complex input $\boldsymbol{z}$ with the length I, where $i = 1, 2, \ldots, I$. Then, an o^{th} complex output element $\tilde{z}_o$ can be given by

$$\tilde{z}_o = \frac{exp(|z_o|)}{\sum_{i=1}^{I} exp(|z_i|)} = \frac{exp(\sqrt{\Re(z_o)^2 + \Im(z_o)^2})}{\sum_{i=1}^{I} exp(\sqrt{\Re(z_i)^2 + \Im(z_i)^2})}, \quad (12)$$

where $o = 1, 2, \ldots, I$.

IV. FULLY COMPLEX-VALUED DL CLASSIFIERS

The architectures of the new DL classifiers are described in this section. First, the baseline classifiers are introduced: (a) real-valued CNN (R-CNN) and (b) real-valued ResNet (R-ResNet). Next, we suggest our fully complex-valued classifiers: (c) complex-valued CNN (C-CNN) and (d) complex-valued ResNet (C-ResNet).

A. REAL-VALUED BASELINE CLASSIFIERS

We adopt R-CNN and R-ResNet suggested in [16] as the baseline classifiers based upon the preliminary comparison results of the recent real-valued classifiers in Appendix VI. The architecture of R-CNN is based on the VGG principle, which consists of the convolutional and fully connected parts as shown in Fig. 4a. The convolutional part extracts features from low-level elements to high-level abstractions of the input signals. After then, the fully connected part tries to map the extracted features into the modulation types. Another baseline, R-ResNet, is shown in Fig. 4b. According to the ResNet principle, shortcuts are included in the convolution part; they can mitigate gradient vanishing or explosion problems that may occur as the number of layers increases [34].

Meanwhile, it should be noted that R-CNN and R-ResNet are slightly modified versions of their original structures; in the fully connected parts, RReLUs are employed instead of the real-valued self exponential linear units (RSELUs) with alpha dropouts. The reason for the substitution is that it is still challenging to extend the concept of RSELU into the complex domain; development and evaluation on the complex version of RSELU are not investigated clearly yet, and further studies are needed. On the other hand, CReLU can be defined as described in Sub-section III-E.

As a result of compiling the baseline classifiers, R-CNN and R-ResNet have 159,832 and 165,144 parameters, respectively.

B. COMPLEX-VALUED CNN CLASSIFIER

Next, we propose the C-CNN classifier that more efficiently utilizes the mutually correlated information of the dataset. As shown in Fig. 4c, C-CNN basically shares the R-CNN's fundamental architecture. However, it is significantly noted that the data handling ways between the two classifiers are totally different; i.e., the complex-valued operations described in Section III are employed instead of all the real-valued operations.

Also, we adopt two additional structural changes to improve SMR capability. First, we optimize C-CNN by replacing the max poolings with the average poolings (i.e., CAPs) in the convolutional part. In general, the pooling operation provides computational efficiency by reducing the intermediate data sizes as the statistical summary. However, since the pooling obviously accompanies the information loss, this loss may affect the performance of a DL model variously depending on the complicated relationships among the dataset characteristics, the extracted features, and the architecture [35]. To choose the suitable pooling, we conducted the preliminary experiment comparing the classification performance between CMP and CAP with the radioML2018a dataset. As a result, C-CNN with CAPs showed slightly better performance; accordingly, CAPs are finally adopted. Second, we apply the regularization technique to C-CNN through the batch normalization. The batch normalization provides not only the training stability but also the regularization effect [36]; it is because the stochastic jittering (i.e., the small changes in variance and mean) of each batch acts as a regularizer, which helps mitigate the overfitting. We place CBNs before CReLUs providing the non-linearity (i.e., between CONV (or CFC) and CReLU). From the newly joined CBNs, we can expect better regularization.

In architecture compilation, the number of parameters for C-CNN is 322,800, which is about twice that of R-CNN.

C. COMPLEX-VALUED ResNet CLASSIFIER

This sub-section proposes the other complex-valued classifier: C-ResNet. The detailed architecture of C-ResNet is given in Fig. 4d. C-ResNet also follows the basic architecture of R-ResNet; however, like in C-CNN, all the real-valued operations are replaced by the complex-valued operations.

Moreover, there are four meaningful structural modifications in C-ResNet to enhance SMR performance. First, as in C-CNN, the pooling optimization is adopted: CAPs. Second, the regularization technique is also applied: CBNs. In third, we further optimize C-ResNet architecture by modifying the complex-valued residual block (CRB). To find a more effective structure than the real-valued residual block (RRB) in Fig. 4b, we conducted another preliminary experiments on the five different types of CRBs, which are the complex versions of the RRB structures illustrated in [37]. According to the comparison result, the CRB structure in Fig. 4d (i.e., CCONV → CBN → CReLU → CCONV → CBN → Addtion → CReLU) showed the most stable performance.

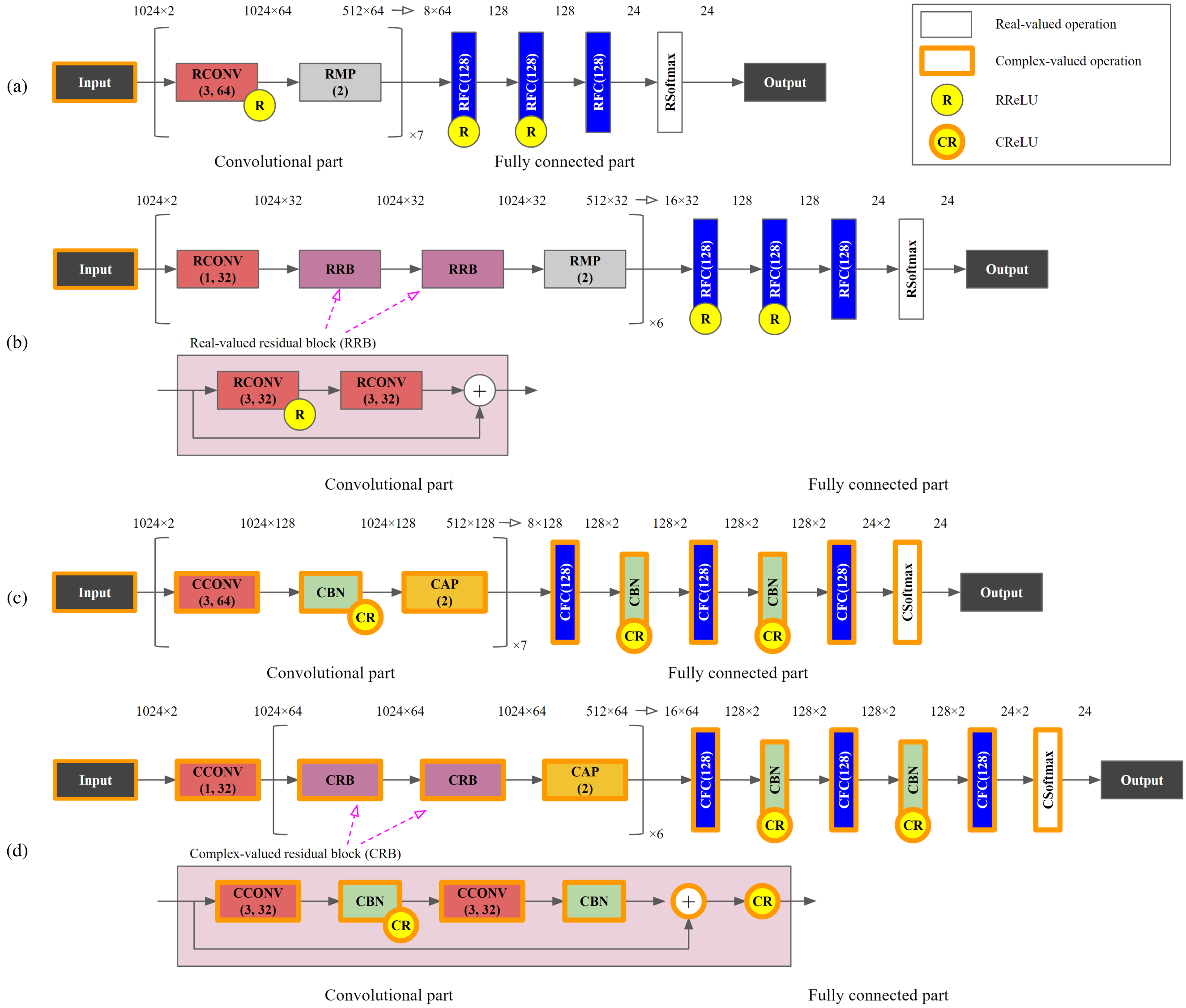


FIGURE 4. Architectures of the baseline and proposed classifiers: (a) R-CNN, (b) R-ResNet, (c) C-CNN, and (d) C-ResNet. The values inside RCONVs or CCONVs show the filter size and number of filters, respectively. And, the numbers within RMPs and CAPs indicate the pooling sizes. Also, the values included in RFCs and CFCs denote the output lengths. Lastly, the output size of each operation is represented above the arrows.

Lastly, the iterative RCONV(1, 32)s in the convolutional part of R-ResNet are replaced by a CCONV(1, 32) in C-ResNet. Except for the first CCONV(1, 32) for dimension matching, the repeated others are no longer required since the two inputs for the summation within CRBs already have equal dimensions. These removals of the unnecessary operations can help reduce the learnable parameters.

As an architecture compilation result, C-ResNet requires 324,784 parameters, almost twice that of R-ResNet.

D. COMPLEX-VALUED TRAINABILITY ANALYSIS

The training of a DL model generally consists of two steps: the back-propagation and the gradient descent. In the back-propagation step, the gradient of the loss function for a weight is determined via the chain rule. Then, the weight is updated in the gradient descent step to minimize the loss by using the computed gradient. This real-valued training rule can be extended to the complex domain by employing the complex-valued chain rule and derivation in the *Wirtinger Calculus* [38]. A detailed analysis of complex-valued trainability is described in Appendix VI. From the analysis results, we confirm that the proposed classifiers are trainable.

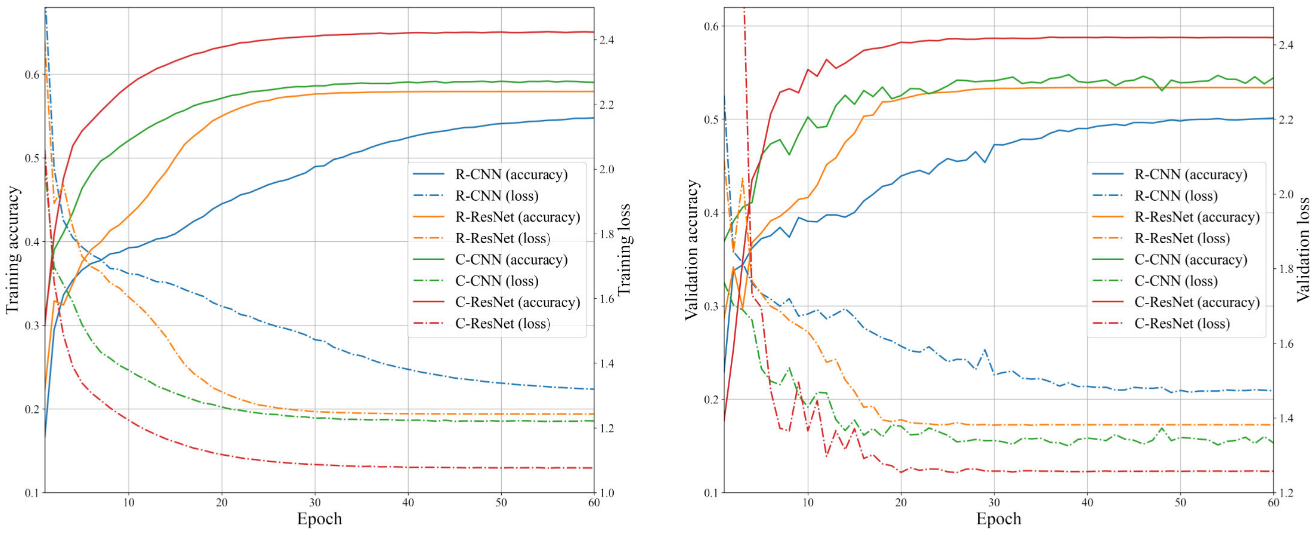
V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we analyze the performance improvements of the proposed classifiers in various aspects and provide in-depth explanations. Also, thoughtful discussions about the issues of our approaches are presented.

A. EXPERIMENTAL ENVIRONMENT

1) IMPLEMENTATION TOOLS

The software are based on Python [39], Tensorflow [40], Keras [41], and NVIDIA Cuda toolkit packages. And, the



(a)

(b)

FIGURE 5. Training and validation results of the proposed and baseline classifiers trained on 240 K: (a) training accuracies/losses and (b) validation accuracies/losses.

TABLE 1. Attributes of the radioML2018.01a dataset.

Attributes	Details
No. of samples	2,555,904 signals with I and Q channels
Signal length	1,024 signal elements per channel
Modulation type	24 types: OOK, $M_{4,8}$-ASK, $M_{2,4,8,16,32}$-PSK, $M_{16,32,64,128}$-APSK, $M_{16,32,64,128,256}$-QAM, FM, GMSK, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, OQPSK
SNR range	-20 dB to 30 dB (26 intervals with 2 dB step)

TABLE 2. Training and validation result summary of the proposed and baseline classifiers.

Classifiers	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
R-CNN	0.5401	1.3407	0.4993	1.4681
R-ResNet	0.5780	1.2463	0.5337	1.3805
C-CNN	0.5919	1.2206	0.5500	1.3222
C-ResNet	0.6427	1.0945	0.5863	1.2539

hardware is configured with Intel Xeon Gold 6152 CPU, NVIDIA Titan RTX, and 64 GB SDRAM.

2) TRAINING SETUP

For training the baseline and proposed classifiers, the Adam optimizer [42] with a default learning rate of 0.001 is used. In addition, the default batch size and epoch are set to 1,024 samples and 100, respectively.

3) DATASET

As an experimental dataset, the radioML2018.01a is employed; this dataset is based on a more practical propagation model with channel impairments such as additive white Gaussian noise (AWGN), carrier frequency offset, symbol rate offset, multipath fading (e.g., Rayleigh fading), and thermal noise [16]. Table 1 shows the attributes of this dataset; to the best of our knowledge, it contains the most diverse signal modulation types, the most number of signals, and the widest signal-to-noise ratio (SNR) ranges so far. In this study, we divide the dataset into the following three different sub-datasets: (a) training dataset, (b) validation dataset, and (c) test dataset. We randomly choose signal samples for each dataset to make even distributions for all modulation types and SNR values. And the three sub-datasets do not overlap any signal samples. By default, each sub-dataset includes 240,000 (240 K) signal samples.

B. EFFECTIVENESS OF COMPLEX SIGNAL HANDLING

We experimentally analyze the effectiveness of the complex signal handling of the complex-valued classifiers, including convergence speed, optimization, and generalization performance. Fig. 5 and Table 2 show the training and validation results of the baseline and proposed classifiers on the default training and validation datasets.

First, the proposed classifiers have faster learning convergence speeds than the baseline classifiers. As shown in Fig. 5, the validation losses of C-CNN and C-ResNet decrease rapidly and already achieve the best loss values of the R-CNN and R-ResNet around the ten epochs. Second, the proposed classifiers provide higher optimization results. In general, a DL model with a low loss value tends toward the more optimized solution while avoiding the local minima. From Table 2, we can confirm that C-CNN and C-ResNet show the lower train and validation loss values. Lastly, the proposed classifiers also offer better generalization properties for new and unseen data. According to the higher validation accuracies in Table 2, it can be seen that C-CNN and C-ResNet

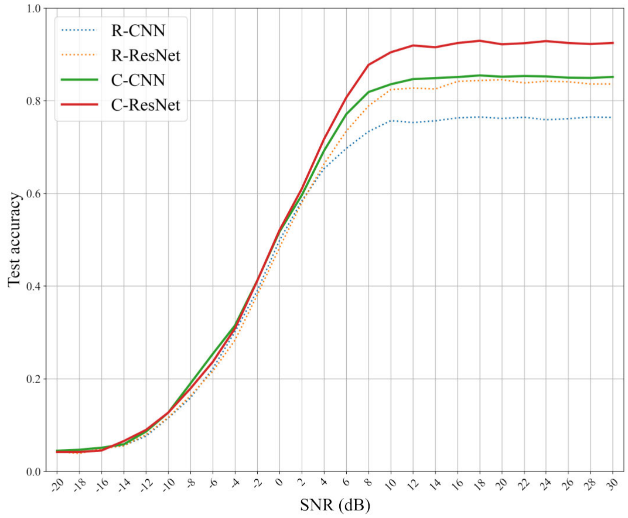


FIGURE 6. Test results for SNR of the proposed and baseline classifiers trained on 240 K.

TABLE 3. Test result summary of the proposed and baseline classifiers.

Classifier	Accuracy > 0 dB SNR	Accuracy for all SNR	Top3 accuracy for all SNR	F1 score for all SNR
R-CNN	0.7210	0.4999	0.6836	0.4852
R-ResNet	0.7782	0.5337	0.7027	0.5292
C-CNN	0.7962	0.5510	0.7138	0.5437
C-ResNet	0.8545	0.5854	0.7213	0.5820

outperform each baseline for unknown signals, which have never been observable during the training process.

C. IN-DEPTH ANALYSES OF SMR ACCURACY

This sub-section presents the detailed SMR performance on the default test dataset, according to SNR and modulation type variations.

Fig. 6 shows the test accuracy results of each classifier by SNR; also, Table 3 summarizes these results. First of all, the proposed classifiers show meaningful improvements in the SMR accuracies of more than 5 % on average at all SNRs compared to the baseline classifiers; the accuracy improvements are more noticeable with gains of about 7 % on average, at the relatively high-quality environment above 0 dB SNR. In addition, C-CNN and C-ResNet outperform the baseline classifiers even in the Top3 accuracies, demonstrating the potential to enhance the classifying accuracy. Furthermore, the higher F1 scores indicate that the proposed classifiers statistically provide the more balanced performance in multiple signal classification problems.

Next, looking at the test results for each modulation type in Fig. 7, the proposed classifiers show the improved performance significantly in M-PSK, M-APSK, and M-QAM types, which have sensitive to complex-valued information: *the phase*. The detailed SMR improvements of C-ResNet for the phase-related modulation types are suggested in Table 4; refer to Appendix VI for the detailed accuracy plots according to modulation type and receiver operating characteristic (ROC) curves; also, due to the space limitation, we only suggest the representative test results of C-ResNet and R-ResNet. When looking at the accuracy results for all SNR, it can be seen C-ResNet shows the more robust capabilities at the relatively sophisticated modulation types with large M. For example, in QPSK, C-ResNet improves the R-ResNet's accuracy by 2.26 %. On the other hand, it improves surprisingly by 12.65 % in 32-PSK. Similarly, this trend can be confirmed in the APSK and QAM types with $M > 32$. In addition, the C-ResNet's higher F1 scores imply that it obtains more stable accuracies; also, the larger area under curve (AUC) values within the ROC plots for all phase-related modulation types show better distinguishability in multiple classification problems.

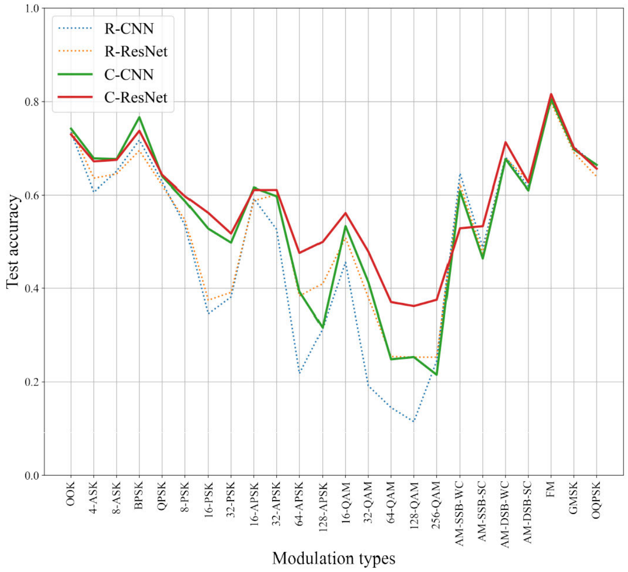


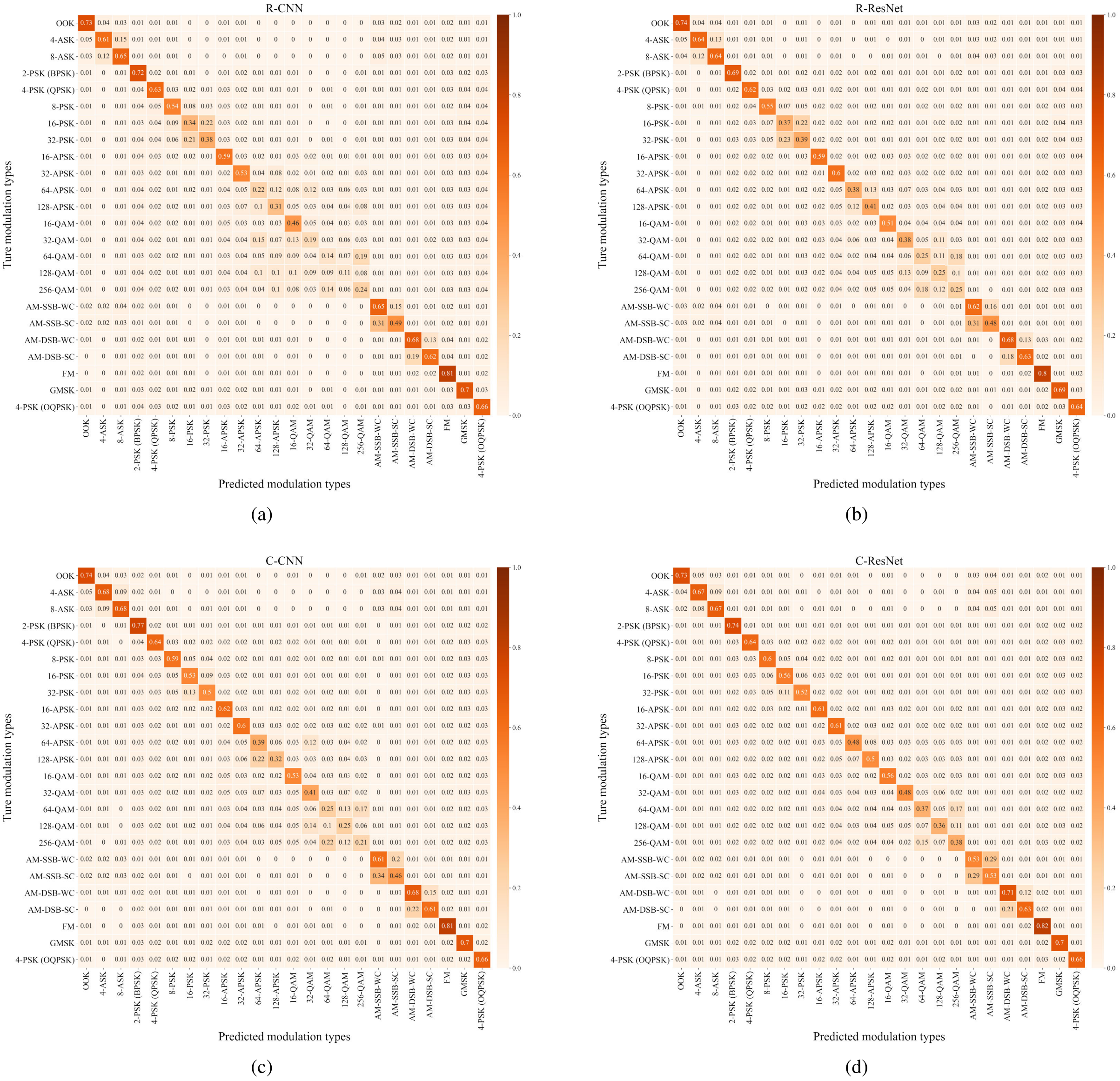
FIGURE 7. Test results for each modulation type of the proposed and baseline classifiers trained on 240 K.

To further prove how the complex-valued information provides beneficial effects, we newly introduce the gradient-weighted class activation map (Grad-CAM) method in the complex domain: the complex-valued Grad-CAM (C-Grad-CAM). The detailed derivation and analyses of C-Grad-CAM are described in Appendix VI. From the C-Grad-CAM results, we can intuitively confirm that the phase-related elements of the complex signal affect the final decision more significantly, since the newly introduced information is adequately propagated within C-ResNet.

The SMR improvement trends of proposed classifiers also can be seen in the confusion matrix, as shown in Fig. 8. We can see that the confusion matrix diagonals of C-CNN and C-ResNet are more vivid than those of R-CNN and R-ResNet in phase-related types. It implies that the proposed classifiers more accurately distinguish those classes with less confusion.

TABLE 4. Detailed test results for *M*-PSK, *M*-APSK, and *M*-QAM of C-ResNet and R-ResNet trained on 240 K.

Modulation types	R-ResNet	C-ResNet	R-ResNet	C-ResNet	R-ResNet	C-ResNet	R-ResNet	C-ResNet
	Accuracy > 0 dB SNR		Accuracy for all SNR		F1 score		AUC	
2-PSK (BPSK)	0.9987	1.0000	0.6947	0.7372	0.6482	0.6659	0.9557	0.9634
4-PSK (OQPSK)	0.9688	0.9945	0.6385	0.6556	0.5748	0.6357	0.9417	0.9489
4-PSK (QPSK)	0.9641	0.9917	0.6200	0.6426	0.6271	0.6341	0.9426	0.9474
8-PSK	0.8516	0.9239	0.5467	0.5974	0.5681	0.5808	0.9369	0.9421
16-PSK	0.5826	0.8745	0.3742	0.5619	0.4026	0.5622	0.9228	0.9413
32-PSK	0.5903	0.8164	0.3910	0.5175	0.3779	0.5847	0.9246	0.9412
16-APSK	0.9310	0.9503	0.5883	0.6107	0.6309	0.6076	0.9336	0.9388
32-APSK	0.9211	0.9445	0.6008	0.6109	0.5700	0.5969	0.9357	0.9396
64-APSK	0.6000	0.7482	0.3831	0.4757	0.4177	0.5126	0.9044	0.9227
128-APSK	0.6325	0.7830	0.4095	0.4994	0.4210	0.5235	0.9077	0.9240
16-QAM	0.7998	0.8779	0.5085	0.5613	0.5256	0.5626	0.9183	0.9309
32-QAM	0.5930	0.7612	0.3800	0.4787	0.3893	0.5380	0.8949	0.9226
64-QAM	0.3856	0.5756	0.2543	0.3701	0.2748	0.4011	0.8882	0.9112
128-QAM	0.3825	0.5631	0.2528	0.3615	0.2648	0.4044	0.8758	0.9039
256-QAM	0.3781	0.5807	0.2527	0.3750	0.2668	0.3946	0.8893	0.9080



(a)

(b)

(c)

(d)

FIGURE 8. Confusion matrices of the proposed and baseline classifiers trained on 240 K: (a) R-CNN, (b) R-ResNet, (c) C-CNN, (d) C-ResNet.

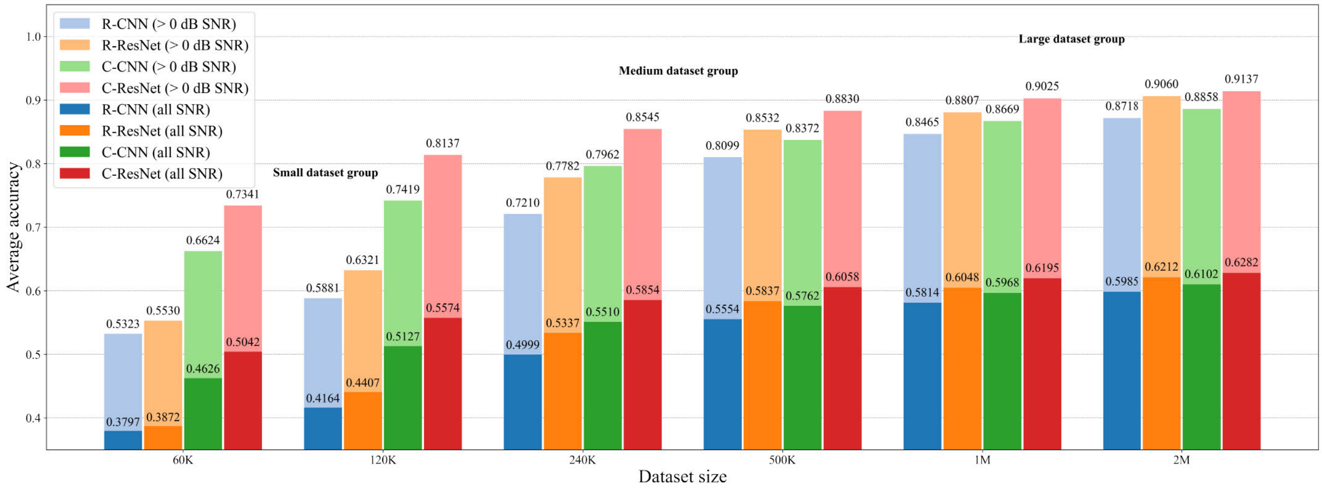


FIGURE 9. Test results for the diverse dataset sizes of the proposed and baseline classifiers.

D. IMPACT OF DATASET SIZE ON PERFORMANCE

The dataset size is a primary factor affecting the performance of a DL model. We would probably expect the DL model to perform better when more diverse and extensive data are available. However, since it is not easy to obtain enough samples in the real world, the performance analysis according to the dataset size can provide helpful insight in a practical situation. For this analysis, we train the proposed and baseline classifiers with three different dataset groups: (a) small, (b) medium, and (c) large groups. And each group consists of two datasets; 60 K and 120 K datasets in the small group; 240 K and 500 K datasets in the medium group; lastly, 1,000,000 (1 M) and 2 M datasets in the large group.

Fig. 9 shows the test results of the proposed and baseline classifiers for the various dataset sizes. C-CNN and C-ResNet represent better accuracies than R-CNN and R-ResNet for all dataset sizes. Especially, the performance improvement on relatively insufficient datasets is impressive. In the small and medium dataset groups, the proposed classifiers increase the accuracies for all SNR by 10.32 % and 3.64 % on average, respectively; when considering only above 0 dB SNR, the averaging improvements are 16.17 % and 5.22 %, respectively.

Furthermore, we can confirm that the proposed classifiers only require about half the dataset size to attain the similar performance of the baseline classifiers. For example, the accuracies of C-CNN and C-ResNet at 240 K are comparable to those of R-CNN and R-ResNet at 500 K. We carefully deliberate these results in terms of the feature space. According to [43], it is well known that the required data size in a DL model is closely related to the dimensionality of the feature space; as the feature space's dimension increases, the space volume naturally rises rapidly; thus, since the available data becomes sparse, the DL model will require more data to achieve the acceptable performance. From the data processing point of view, the proposed classifiers may have a smaller dimension of the feature space than the baseline classifiers because they extract the features at the complex domain instead of each real and imaginary part of the complex data. In consequence, we believe that the smaller feature space makes the suggested classifiers demand fewer datasets. Undoubtedly, this effect can offer a significant benefit to the real-world environment where sufficient data is challenging to obtain.

E. COMPUTATIONAL COMPLEXITY

In terms of the learning and classification costs, we analyze the computational complexity of the proposed and baseline classifiers. Table 5 shows the measured learning costs, including peta floating-point operations (PFLOPs) and time required to train the classifiers. Looking into PFLOPs, it can be seen that the proposed classifiers inevitably require about four times more computations than the baselines for all datasets; When considering the primary complex-valued operations such as CFC or CCONV demanding the four real multiplications for a complex multiplication, these values seem to be reasonable. Also, in the learning time, our classifiers need approximately six-fold more time on average. Nevertheless, we do not regard these increments as a severe drawback, since the learning cost is mainly required during the initial training process.

On the other hand, the classification costs can be a critical burden because it directly affects the signal recognition interval in the practical applications. Table 6 shows the classification costs: mega floating-point operations (MFLOPs) and time per a randomly selected signal. As expected, we can see the classification FLOPs of the proposed classifiers is about four times higher than those of the baseline classifiers. Similar to the learning cost, we can not avoid increasing the classification cost due to the rise of complex-valued operations.

However, the measured classification time needs to be assessed carefully from the viewpoint of the proposed classifiers' applicability to the real-worlds. We believe that a

TABLE 5. **Learning costs of the proposed and baseline classifiers.**

Classifier	Learning PFLOPs (time (hours))					
	Small group		Medium group		Large group	
	60 K	120 K	240 K	500 K	1 M	2 M
R-CNN	0.47 (0.35)	0.93 (0.45)	1.87 (0.67)	3.90 (1.15)	7.79 (2.08)	15.59 (3.92)
R-ResNet	0.95 (0.55)	1.88 (0.72)	3.77 (1.89)	7.85 (1.80)	15.71 (3.30)	31.43 (6.30)
C-CNN	1.94 (1.92)	3.86 (2.53)	7.71 (5.58)	16.08 (6.37)	32.16 (11.52)	64.35 (21.73)
C-ResNet	3.82 (3.58)	7.57 (4.72)	15.14 (6.58)	31.57 (11.82)	63.15 (21.72)	126.36 (41.10)

TABLE 6. **Classification costs of the proposed and the baseline classifiers.**

Classifiers	Classification MFLOPs (time (milliseconds))
R-CNN	25.99 (0.12)
R-ResNet	52.39 (0.23)
C-CNN	107.26 (0.54)
C-ResNet	210.62 (1.06)

few milliseconds is enough time to recognize the modulation types in real-time for the following basis: (a) the minimum allowable measurement period of the wireless link quality (WLQ) is as long as 2 milliseconds in the long term evolution (LTE) [44], which is one of the fastest and most sophisticated modern communications, and (b) at least in LTE, a few milliseconds of SMR time may be enough to recognize dynamically changed modulation types without omission. Consequently, it can be said that the proposed classifiers are suitable for most real-time SMR applications.

F. DISCUSSIONS

So far, we have described that the suggested data handling way gives us the following significant benefits: (a) faster learning speed due to the improved optimization, (b) higher recognition accuracy along with better generalization, and (c) less dataset demand. However, despite these useful benefits, there are several issues worthwhile to discuss:

1) NO ACCURACY GAIN AT LOW SNR

As shown in Fig. 6, the proposed classifiers show no profit compared to the baseline classifiers at low SNR (i.e., approximately below 0 dB). However, it should be noted that this limitation is not the unique phenomenon to our case but one of the main challenges of SMR. The signals below 0 dB SNR depict that the noise strength is larger than the pure modulated signal; undoubtedly, these corrupted data have no choice but to disrupt the proper training process of the classifier. In the same context, the proposed classifiers cannot easily overcome the unsolved problem of low accuracy in the noisy environment, even using the complex numbers. The possible improvement approach may be to block the inflowing noise in the pre-processing stage (as shown in Fig. 2c) by adopting techniques like frequency filtering or common-mode time noise reduction.

2) NO ACCURACY GAIN IN AM TYPES

The newly suggested classifiers do not show the noticeable performance in the case of amplitude modulation (AM) types (i.e., AM-DSB-WC, AM-DSB-SC, AM-SSB-WC, and AM-SSB-SC), as shown in Fig. 7. It seems that the root cause of this issue is the underlying nature of AM. In detail, only the amplitude (not frequency or phase) of the carrier wave is varied in proportion to that of the message signal [45]; as a result, a pure modulated AM signal is composed of only real parts without any imaginary parts. Consequently, it is no wonder that the real-value only modulation types do not stand to the benefits of the suggested complex data handling. In order to solve this limitation, we need to consider a different perspective: the hybrid learning of both time and frequency domains. In other words, a complex time/frequency-based classifier may be a helpful alternative to overcome this issue. We plan to address it as a future research topic.

3) ACCURACY GAIN SATURATION AT LARGE DATASET GROUP

In the average accuracy comparison result (in Fig. 9), the performance gap between the proposed and baseline classifiers gradually decreases as the dataset size increases. This result can be explained by the DL model's performance convergence effect [46] that the continuous growth of dataset size accompanies the accuracy convergence. Observing the accuracy results of the baseline classifiers, it can be discovered that the performance is getting better as the dataset size increases up to about 500 K. On the other hand, at the large dataset group (1 M and 2 M), the accuracies for above 0 dB SNR are roughly saturated into about 0.9. Following the baseline classifiers' trends, we can confirm that the similar saturations also occur in the suggested complex-valued classifiers, which share the fundamental frames of the baselines. However, it should be noted that our classifiers outperform the baseline classifiers on all the dataset sizes, although the accuracy gains become smaller. Furthermore, we would like to note that the purpose of presenting the performance saturation outcomes is to provide more in-depth analysis results considering diverse user circumstances. In practice, it may be much more challenging or nearly impossible to collect large enough datasets that reflect the real-worlds. Once again, we emphasize that our classifiers show the better enhancement in normal-scale datasets in the practical circumstance.

VI. CONCLUSION

In this study, we demonstrated that proper complex-valued data handling of the wireless signal could provide an opportunity to improve the SMR performance. Especially, the proposed classifiers could significantly improve the recognition accuracies even in relatively fewer datasets within the acceptable classification time. By providing diverse experimental analysis results and careful explainable causes, we believe that the proposed classifiers can play an important role in non-cooperative SMR applications, such as the recently sophisticated wireless environments. Finally, our study re-emphasizes the well-known insight that the considerate and deep comprehension of handling data is significantly important and may be the prerequisite for the optimal solution.

APPENDIX A
PERFORMANCE COMPARISON OF RECENT DL CLASSIFIERS

Based on only the given accuracy results in [11]–[16], it is hard to directly compare the conventional DL classifiers, since their performance is tested in different experimental environments (e.g., signal modulation types, signal length, signal quality, etc.). In order to achieve overall consistency, we suggest the performance comparison result of the following four representatives among the conventional DL classifiers' structures: LSTM-based [11], fusion-based [13], CNN-based [16] (R-CNN), and ResNet-based [16] (R-ResNet) classifiers. During training and testing of each classifier, the identical train, validation, and test datasets are employed; these sub-datasets are selected from the radioML2018.01a dataset. According to the accuracy comparison result in Fig. 10, the ResNet-based classifier shows the best performance; the CNN-based classifier follows it.

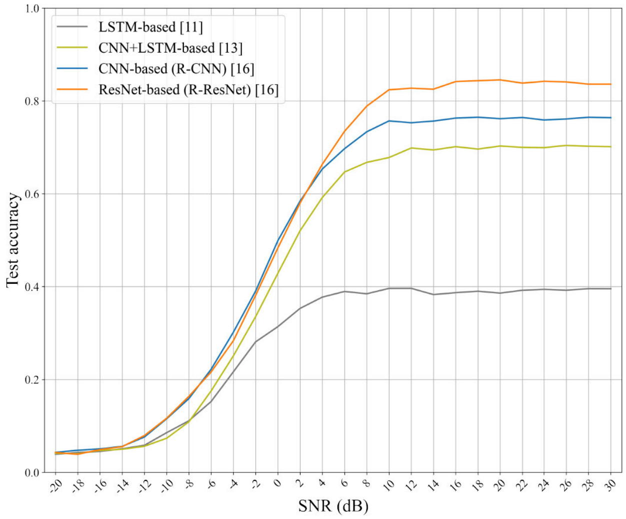


FIGURE 10. Performance comparison results of the conventional DL classifiers trained on 240 K.

APPENDIX B
COMPLEX-VALUED BACK-PROPAGATION AND GRADIENT DESCENT

The Wirtinger Calculus provides the general partial derivative method and chain rule for a differentiable or non-differentiable complex-valued function. Suppose a complex variable $z = x + jy$ and its complex conjugate z^*. Then, the differentiation of a complex-valued function f is defined as the partial derivative pair (i.e., *R-derivative* and *conjugate R-derivative*) with respect to z and z^* as in (B-1) [38].

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - j\frac{\partial f}{\partial y}\right), \quad \frac{\partial f}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right). \quad \text{(B-1)}$$

Also, the complex-valued chain rule for a composite function $f(g)$, where f and g are complex-valued functions, is given as follows [38]:

$$\frac{\partial f(g)}{\partial z} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial z} + \frac{\partial f}{\partial g^*}\frac{\partial g^*}{\partial z}, \quad \frac{\partial f(g)}{\partial z^*} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial z^*} + \frac{\partial f}{\partial g^*}\frac{\partial g^*}{\partial z^*}. \quad \text{(B-2)}$$

In the training stage of the proposed complex-valued classifiers, we employ a real-valued loss function such as the cross-entropy. The use of the real-valued loss function is rational for the following reasons: (a) the last layer of the proposed classifiers is Csoftmax (i.e., its final prediction outputs are the real-valued probabilities whose distributions are between 0 and 1); (b) the true labels of the dataset are the one-hot encoded real-valued data; thus, (c) the loss function using these prediction and truth data also becomes real-valued.

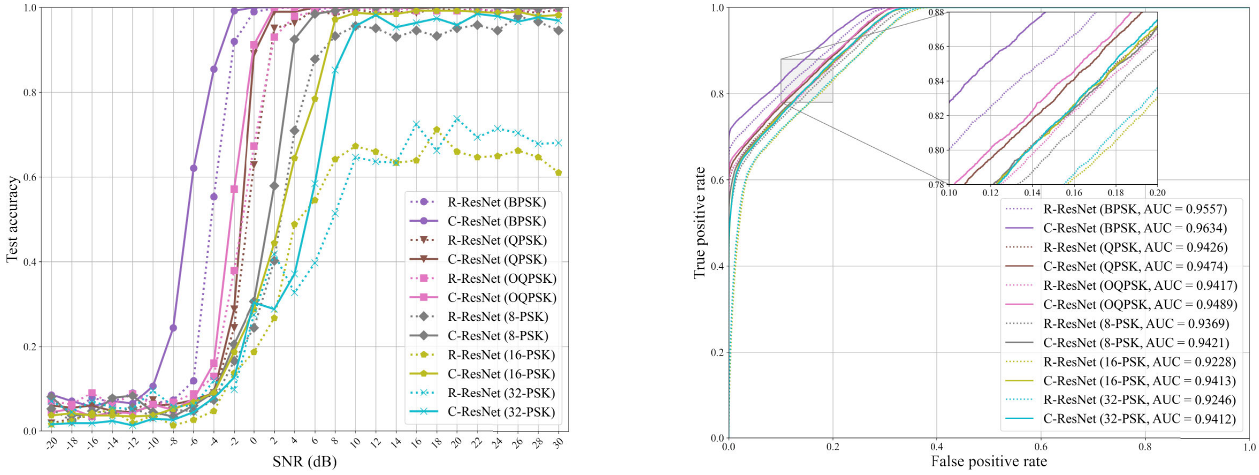
According to [47], the complex-valued gradient of the real-valued loss function L for z can be defined as

$$\nabla_z L = 2\frac{\partial L}{\partial z^*}. \quad \text{(B-3)}$$

In (B-3), it is significant to recognize that the gradient of L is related only to *the conjugate R-derivative* instead of the partial derivative pair. In other words, this property indicates that the multi-layer back-propagation for the complex-valued training can be achieved in the form of the one-side derivative (i.e., *the conjugate R-derivative*).

In detail, the complex-valued back-propagation can be derived from the complex-valued chain rule of (B-2). Assume that a complex-valued function with a complex weight w is $\tilde{z} = g(z; w) = r(x, y; \Re(w), \Im(w)) + js(x, y; \Re(w), \Im(w))$, where r and s are the real-valued functions. Then, the partial derivative of L with respect to z^* can be determined as:

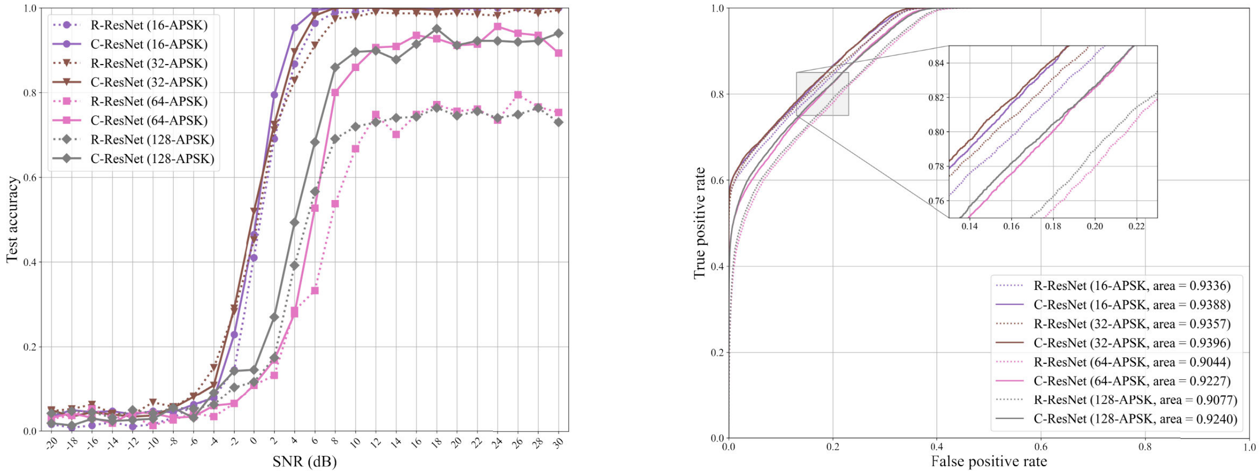
$$\begin{aligned}
\frac{\partial L}{\partial z^*} &= \frac{\partial L}{\partial g}\frac{\partial g}{\partial z^*} + \frac{\partial L}{\partial g^*}\frac{\partial g^*}{\partial z^*} \\
&= \frac{\partial L}{\partial \tilde{z}}\frac{\partial \tilde{z}}{\partial z^*} + \frac{\partial L}{\partial \tilde{z}^*}\left(\frac{\partial \tilde{z}}{\partial z}\right)^* \\
&= \frac{1}{2}\left(\frac{\partial L}{\partial r} - j\frac{\partial L}{\partial s}\right) \times \frac{1}{2}\left(\frac{\partial r}{\partial x} + j\frac{\partial s}{\partial x} + j\frac{\partial r}{\partial y} - \frac{\partial s}{\partial y}\right) \\
&\quad + \frac{1}{2}\left(\frac{\partial L}{\partial r} + j\frac{\partial L}{\partial s}\right) \times \frac{1}{2}\left(\frac{\partial r}{\partial x} - j\frac{\partial s}{\partial x} + j\frac{\partial r}{\partial y} + \frac{\partial s}{\partial y}\right) \\
&= \frac{1}{2}\frac{\partial L}{\partial r}\frac{\partial r}{\partial x} + \frac{1}{2}\frac{\partial L}{\partial s}\frac{\partial s}{\partial x} + j\frac{1}{2}\frac{\partial L}{\partial r}\frac{\partial r}{\partial y} + j\frac{1}{2}\frac{\partial L}{\partial s}\frac{\partial s}{\partial y} \\
&= \left[\Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial x} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial x}\right] \\
&\quad + j\left[\Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial y} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial y}\right] \qquad \text{(B-4)}
\end{aligned}$$



(a)

(b)

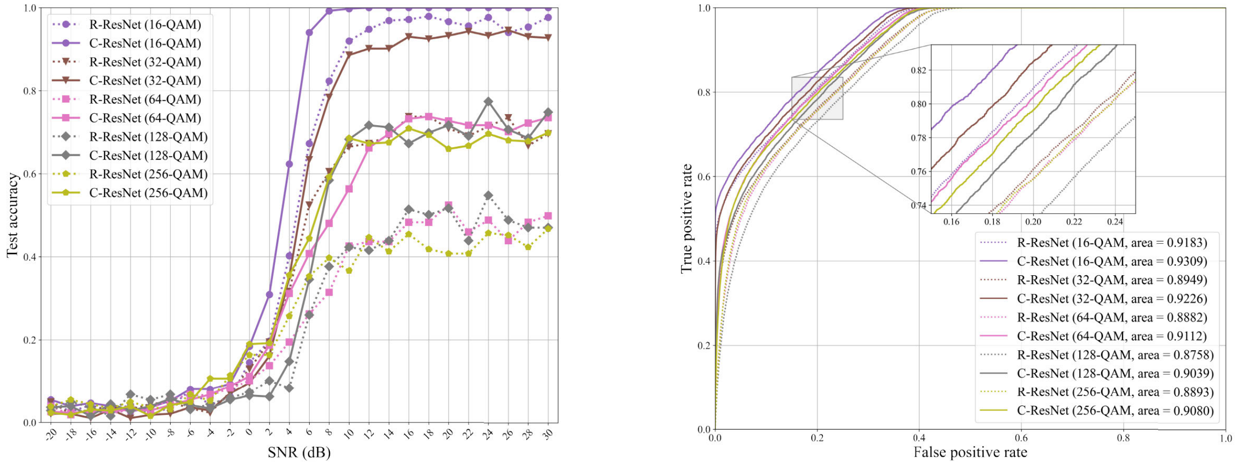
FIGURE 11. Detailed test results for *M*-PSK of C-ResNet and R-ResNet classifiers trained on 240 K: (a) test accuracies, (b) ROC curves.



(a)

(b)

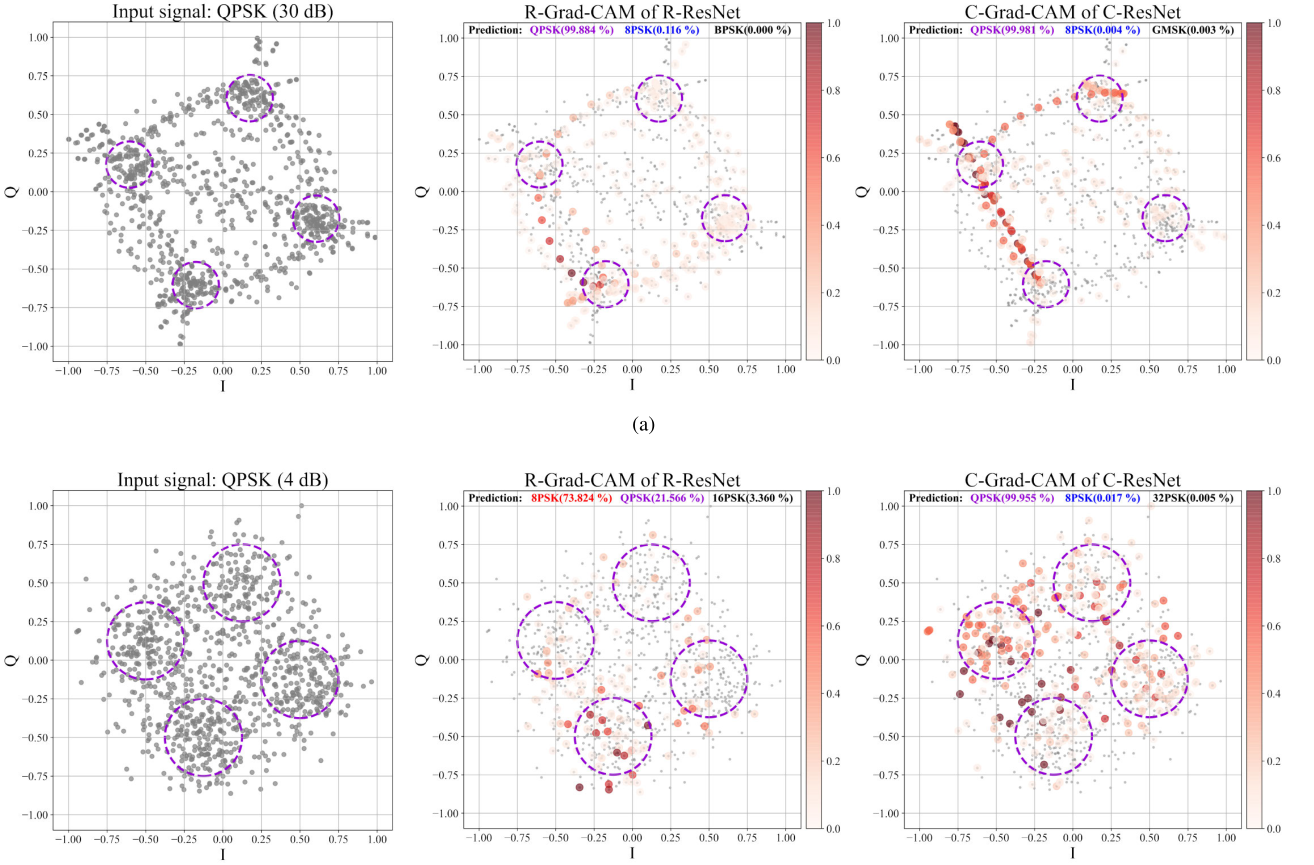
FIGURE 12. Detailed test results for *M*-APSK of C-ResNet and R-ResNet classifiers trained on 240 K: (a) test accuracies, (b) ROC curves.



(a)

(b)

FIGURE 13. Detailed test results for *M*-QAM of C-ResNet and R-ResNet classifiers trained on 240 K: (a) test accuracies, (b) ROC curves.



(a)

(b)

FIGURE 14. Grad-CAM results of the C-ResNet and R-ResNet classifiers trained on 240 K: (a) R-Grad-CAM and C-Grad-CAM for 30 dB SNR QPSK signal input and (b) R-Grad-CAM and C-Grad-CAM for 4 dB SNR QPSK signal input.

where $\frac{\partial L}{\partial \tilde{z}^*}$ is the back-propagated complex-valued partial derivative from the subsequent function of g. In the same way, the partial derivative of L for w^* is given by

$$\frac{\partial L}{\partial w^*} = \left[\Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial \Re(w)} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial \Re(w)}\right] + j\left[\Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial \Im(w)} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial \Im(w)}\right]. \tag{B-5}$$

Finally, we can minimize the loss by extending the real-valued gradient descent to the complex domain. According to (B-3) and (B-5), w is updated as follow:

$$w \leftarrow w - \alpha\nabla_w L = w - 2\alpha\frac{\partial L}{\partial w^*}, \tag{B-6}$$

where α is the learning rate.

The complex data representation in Sub-section III-A is still available to realize the complex-valued training. Through this representation, the complex-valued partial derivatives in (B-4) and (B-5) can be carried out logically by using the real-valued partial derivatives as follow:

$$\Re\left(\frac{\partial L}{\partial z^*}\right) = \Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial x} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial x},$$

$$\Im\left(\frac{\partial L}{\partial z^*}\right) = \Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial y} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial y}, \tag{B-7}$$

$$\Re\left(\frac{\partial L}{\partial w^*}\right) = \Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial \Re(w)} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial \Re(w)},$$

$$\Im\left(\frac{\partial L}{\partial w^*}\right) = \Re\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial r}{\partial \Im(w)} + \Im\left(\frac{\partial L}{\partial \tilde{z}^*}\right)\frac{\partial s}{\partial \Im(w)}. \tag{B-8}$$

Likewise, the weight in (B-6) can be updated by

$$\Re(w) \leftarrow \Re(w) - 2\alpha\Re\left(\frac{\partial L}{\partial w^*}\right),$$

$$\Im(w) \leftarrow \Im(w) - 2\alpha\Im\left(\frac{\partial L}{\partial w^*}\right). \tag{B-9}$$

APPENDIX C
DETAILED TEST RESULTS OF PHASE-RELATED MODULATION TYPES

Figs. 11, 12, and 13 show the detailed test accuracies and ROC curves of C-ResNet and R-ResNet for the M-PSK, M-APSK, and M-QAM modulation types.

APPENDIX D
COMPLEX-VALUED GRAD-CAM

Generally, Grad-CAM can intuitively visualize the input impact on the final decision by using the feature map of the convolutional layer and the gradient from the feature map to the last class prediction. We can extend the concept of real-valued Grad-CAM in [48] to the complex domain.

For any class c, we can define C-Grad-CAM $L^c_{C-Grad-CAM}$ as follow:

$$L^c_{C-Grad-CAM} = \left| CReLU\left(\sum_n \alpha^c_n \boldsymbol{A}^n\right)\right|, \qquad \text{(D-1)}$$

where $\boldsymbol{A}^n$ is the complex-valued output at the n^{th} channel of the target convolutional layer. Also, α^c_n denotes the gradient mean of the complex-valued softmax input magnitude $|y^c|$ with respect to $\boldsymbol{A}^n$ as in (D-2).

$$\alpha^c_n = \frac{1}{Z}\sum_i \frac{\partial |y^c|}{\partial (A^n_i)^*}, \qquad \text{(D-2)}$$

where A^n_i and Z are an i^{th} element and the length of $\boldsymbol{A}^n$, respectively.

Fig. 14 shows the Grad-CAM results on the last convolutional layers of R-ResNet and C-ResNet under QPSK input signals of 30 dB and 4 dB SNRs, respectively. The four dotted circles (with the purple color) in the input constellation diagram indicate the QPSK symbols related to phase information. In the Grad-CAM results, we highlight the elements of the input signal affecting the final decision with red color; the stronger the influence, the deeper the intensity of the color.

As shown in Fig. 14a, we can see that R-ResNet and C-ResNet find out the QPSK type correctly with the great accuracies (99.88 % and 99.98 %, respectively) in the high-quality (i.e., 30 dB SNR) signal. However, in the relatively low-quality (i.e., 4 dB SNR) signals, as shown in Fig. 14b, it is not easy to interpret that R-ResNet trains the input signals' features sufficiently since it misunderstands the QPSK input signal as 8-PSK with the 73.82 % accuracy. On the other hand, the input elements related to the phase information of QPSK are more deeply emphasized in C-ResNet; that is, it means that they influence on the final decision more powerfully.

REFERENCES

[1] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[2] P. Li, "Research on radar signal recognition based on automatic machine learning," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 1959–1969, Apr. 2020.

[3] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, and E. P. Simon, "On classifiers for blind feature-based automatic modulation classification over multiple-input–multiple-output channels," *IET Commun.*, vol. 10, no. 7, pp. 790–795, May 2016.

[4] I. Bisio, C. Garibotto, A. Sciarrone, and F. Lavagetto, "Blind detection: Advanced techniques for WiFi-based drone surveillance," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 938–946, Jan. 2019.

[5] I. Bisio, C. Garibotto, F. Lavagetto, and A. Sciarrone, "Outdoor places of interest recognition using WiFi fingerprints," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5076–5086, May 2019.

[6] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, Apr. 2007.

[7] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*. Hoboken, NJ, USA: Wiley, 2015.

[8] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Mar. 2017, pp. 1–6.

[9] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 915–919.

[10] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El Gamal, and Y. C. Eldar, "Fast deep learning for automatic modulation classification," 2019, *arXiv:1901.05850*.

[11] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, "Deep neural network for robust modulation classification under uncertain noise conditions," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 564–577, Jan. 2020.

[12] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10760–10772, Nov. 2018.

[13] D. Zhang, W. Ding, B. Zhang, C. Xie, H. Li, C. Liu, and J. Han, "Automatic modulation classification based on deep learning for unmanned aerial vehicles," *Sensors*, vol. 18, no. 3, p. 924, 2018.

[14] S. Zhou, Z. Yin, Z. Wu, Y. Chen, N. Zhao, and Z. Yang, "A robust modulation classification method using convolutional neural networks," *EURASIP J. Adv. Signal Process.*, vol. 2019, no. 1, p. 21, 2019.

[15] R. Yin, J. Huang, and Z. Fei, "Short-time modulation classification of complex wireless communication signal based on deep neural network," in *Proc. 24th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2018, pp. 520–524.

[16] 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.

[17] S. Haykin, *Communication Systems*. Hoboken, NJ, USA: Wiley, 2008.

[18] T. Nitta, "On the critical points of the complex-valued neural network," in *Proc. 9th Int. Conf. Neural Inf. Process. (ICONIP)*, vol. 3, Nov. 2002, pp. 1099–1103.

[19] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.

[20] E. Azzouz and A. K. Nandi, *Automatic Modulation Recognition of Communication Signals*. Boston, MA, USA: Springer, 2013.

[21] K. Hassan, I. Dayoub, W. Hamouda, and M. Berbineau, "Automatic modulation recognition using wavelet transform and neural networks in wireless systems," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, pp. 1–13, Dec. 2010.

[22] O. A. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants for high order modulation classification," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, vol. 1, Oct. 2003, pp. 112–117.

[23] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *IEEE Circuits Syst. Mag.*, vol. 9, no. 2, pp. 27–45, Jun. 2009.

[24] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[25] B. G. Mobasseri, "Constellation shape as a robust signature for digital modulation recognition," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, vol. 1, Oct./Nov. 1999, pp. 442–446.

[26] S.-Z. Hsue and S. S. Soliman, "Automatic modulation classification using zero crossing," *IEE Proc. F (Radar Signal Process.)*, vol. 137, no. 6, pp. 459–464, 1990.

[27] M. W. Aslam, S. Member, Z. Zhu, and S. Member, "Automatic modulation classification using combination of genetic programming and KNN," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2742–2750, Jun. 2012.

[28] X. Sun, S. Su, Z. Zuo, X. Guo, and X. Tan, "Modulation classification using compressed sensing and decision tree–support vector machine in cognitive radio system," *Sensors*, vol. 20, no. 5, p. 1438, 2020.

[29] M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm," *Signal Process.*, vol. 84, no. 2, pp. 351–365, 2004.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[31] Y. Bengio, I. Goodfellow, and A. Courville, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2017.

[32] C.-A. Popa and C. Cernăzanu-Glăvan, "Fourier transform-based image classification using complex-valued convolutional neural networks," in *Advances in Neural Networks—ISNN 2018*, T. Huang, J. Lv, C. Sun, and A. V. Tuzikov, Eds. Cham, Switzerland: Springer, 2018, pp. 300–309.

[33] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Felipe Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," 2017, *arXiv:1705.09792*.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[35] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 111–118.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 630–645.

[38] K. Kreutz-Delgado, "The complex gradient operator and the CR-calculus," 2009, *arXiv:0906.4835*.

[39] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

[40] M. Abadi, P. Barham, J. Chen, Z. Chen, and A. Davis, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, Savannah, GA, USA, Nov. 2016, pp. 265–283.

[41] A. Géron, *Hands-on Machine learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[43] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[44] 3GPP, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures (3GPP TS 36.213 Version 12.4.0 Release 12)*, document ETSI TS 136 213 V12.4.0., Technical Specification, 2015.

[45] J. Proakis and M. Salehi, *Communication Systems Engineering*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002. [Online]. Available: https://books.google.co.kr/books?id=8WqfQgAACAAJ

[46] X. Zhu, C. Vondrick, C. C. Fowlkes, and D. Ramanan, "Do we need more training data?" *Int. J. Comput. Vis.*, vol. 119, pp. 76–92, Aug. 2015.

[47] A. Hirose, "Learning algorithms in complex-valued neural networks using Wirtinger calculus," in *Complex-Valued Neural Networks: Advances and Applications*. Piscataway, NJ, USA: IEEE, 2013, pp. 75–102, doi: 10.1002/9781118590072.ch4.

[48] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

SANGKYU KIM received the B.S. and M.S. degrees in electronics and computer engineering from Hanyang University, Republic of Korea, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree in computer science from KAIST, Republic of Korea.

Since 2010, he has been working with the Affiliated Institute of Electronics and Telecommunications Research Institute (ETRI), Republic of Korea, where he is currently a Senior Researcher. His research interests include deep learning for signal processing and embedded/Internet of Things security.

HAE-YONG YANG received the B.S. and M.S. degrees in electrical engineering from Pusan National University, Republic of Korea, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Korea University, Republic of Korea, in 2012.

Since 2001, he has been working with the Affiliated Institute of Electronics and Telecommunications Research Institute (ETRI), Republic of Korea, where he is currently a Principal Researcher. Before joining ETRI, he was with Hyundai Electronics, Republic of Korea. His research interests include audio/video signal processing and multimedia/Internet of Things security.

DAEYOUNG KIM received the B.S. and M.S. degrees in computer science from Pusan National University, Republic of Korea, in 1990 and 1992, respectively, and the Ph.D. degree in computer engineering from the University of Florida, in August 2001.

Since March 2014, he has been a Professor with the Department of School of Computing, KAIST, Republic of Korea. He was an Assistant/Associate Professor with the Department of Computer Science, KAIST, from 2002 to 2014. From September 2001 to 2002, he was a Research Assistant Professor with Arizona State University. He worked as a Research Staff Member with the Electronics and Telecommunications Research Institute (ETRI), Republic of Korea, from 1992 to 1997. Currently, he is the Director of the Auto-ID Lab Korea (www.autoidlabs.org) and the Global USN National Research Laboratory. His research interests include sensor networks, real-time and embedded systems, and the Internet of Things.