DNCNet: Deep Radar Signal Denoising and Recognition

Mingyang Du, Ping Zhong, *Senior Member, IEEE,* Xiaohao Cai, and Daping Bi

***Abstract*—Deep learning with its rapid development and advancement has achieved unparalleled performance in many areas like computer vision as well as cognitive radio and signal recognition. However, the performance of most deep neural networks would suffer from degradation in the data mismatch scenario, e.g., the test dataset has a related but nonidentical distribution with the training dataset. Considering the noise corruption, a classifier's accuracy might drop sharply when it is tested on a dataset with much lower signal-to-noise ratio compared to its training dataset. To address this dilemma, in this work we propose an efficient DeNoising and Classification Network (DNCNet) for radar signals. DNCNet consists of denoising and classification subnetworks. Firstly, a radar signal detection and synthetic mechanism is designed to generate pairwise clean data and noisy data for the DNCNet to train its denoising subnetwork. Then a two-phase training procedure is proposed to train the denoising subnetwork in the first phase and strengthen the mapping between the denoising results and perceptual representation in the second. Experiments on synthetic and benchmark datasets validate the excellent performance of the proposed DNCNet against state-of-the-art methods in terms of both signal restoration quality and classification accuracy.**

***Index Terms*—Radio signal, deep learning, neural network, radar emitter recognition, denoising.**

I. Introduction

THE ability to instantly identify and label the interested radio spectrum in the crowded electromagnetic environment is an essential factor for spectrum monitoring, emitter recognition, mobile transmitter tracking and cognitive radio or radar applications [1]. In this work we focus on radar signal identification/recognition, which is extremely challenging when e.g. the data is highly noisy and the distributions across the training and test datasets are variant. The effective and unique signal features are the key in signal identification and verification. For many years, hand-crafted feature extractors based on the expertise have been utilized, e.g., the ones based on Wigner and Choi–Williams time-frequency distributions [2] and short-time Fourier transform [3]. The selected features from these techniques create wide separation in the spectrum domain and thus can successfully distinguish different signals or emitters. However, these approaches generally lack the capacity of automation and generalization.

M. Du is with the College of Electronic Engineering, National University of Defense Technology, Hefei, China, 230037. e-mail: (see dumingyang17@nudt.edu.cn).

D. Bi is with the College of Electronic Engineering, National University of Defense Technology, Hefei, 230037, China.

P. Zhong is with the National Key Laboratory of Science and Technology on ATR, National University of Defense Technology, Changsha, 410073, China.

X. Cai is with School of Electronics and Computer Science, University of Southampton, University Road, Southampton, SO17 1BJ, UK

Manuscript received xxx; revised xxx.

Deep learning greatly increases the capacity of feature extraction directly on raw data and realizes the end-to-end learning. This trend has not only been demonstrated in computer vision and natural language processing involving image, video, voice and text data, but also widely applied in the radio frequency device realm in recent years profiling mobile traffic data [4], communication [5] and radar signal. For example, in previous works such as [6], [7], [8], the raw time series radar signal was firstly transformed to two-dimensional (2D) images by some time-frequency analysis methods, and then used as inputs to the deep neural network (DNN) after some preprocessing like cropping. These image-based approaches have high prevalence but also some weaknesses. For example, i) the time-frequency analysis may cause the loss of amplitude information compared to the raw time series data; ii) it is inconsistent with the end-to-end trend in deep learning where the model learns all the steps between the initial input phase and the final output; and iii) it is time-consuming to transform all input raw signal samples to the time-frequency distribution images, which could increase the computation burden compared to the end-to-end learning.

In addition to utilizing the preprocessing time-frequency information in the aforementioned literature, the authors in e.g. [9], [10], [11] identified the radio signal directly from the raw time-series sequence. As for complex-valued inputs, their real and imaginary parts constituting a set of $2N$ vectors are input into a narrow 2D convolutional neural network (CNN), where N is the length of the signal sample. It proves the effectiveness of CNN on the high-level representation of radio signal data in addition to the vision data like images and videos. In these works, a small corpus of standard datasets, containing synthetic datasets simulated channel effects and over-the-air, was also created by the GNU Radio [12]. However, the performance of the pre-trained DNNs may suffer from degradation when the test dataset has a different distribution with the training dataset. For instance, since the environmental noise, intentional jamming and other distractions in the battle phase can damage the signal quality and lower the signal-to-noise ratio (SNR), the distribution of intercepted signals may probably be nonidentical to the training dataset. Moreover, numerical experiments illustrate that a plain CNN classifier, trained on a high SNR dataset, would experience abrupt accuracy drop when tested on a low SNR dataset. The problem and challenge raised by the distribution mismatch between the training and test datasets are therefore very common in the field of radio signal recognition and need to be settled urgently.

In this paper, we exploit the divide-and-conquer idea (e.g.

[13]) to address the above-mentioned problem/challenge, i.e., learning a denoising model to firstly restore signals followed by developing another classifier to identify them. According to the qualitative analysis, the denoising model can ensure robustness in handling noisy signals which cover a wide range of SNR, and thus the pre-trained DNN classifier will maintain the high accuracy on test datasets. There is a paucity of literature about signal denoising based on DNN, while we have witnessed the boom in image denoising. The U-Net architecture [14] has been widely utilized in model-based denoising techniques for images restoration, with a downsampling operation capturing context and a symmetric upsampling operation enabling precise localization. In [15], a residual learning strategy was utilized to embrace the denoising model in a very deep architecture and was able to handle Gaussian denoising with unknown noise level. The FFDNet proposed in [16] introduced a noise level estimation subnetwork to improve the flexibility of dealing with spatially variant noise. The work in [17] extended the denoising model to the true noise which was spatially/channel-correlated and spatially/channel-variant. In [18], a high-level vision task was combined with the denoising model to boost the perceptual quality of the denoised images by image semantic guidance.

The main contributions of this paper are threefold:

- We propose a radar signal detection and synthetic mechanism to construct datasets according to the intercepted signals and current threat database, considering the application background and the natural requirement of model-based denoising techniques which need the pairwise clean data and noisy data for learning.
- We adapt the 2D image denoising model to 1D to handle the radar signal restoration and propose the DeNoising and Classification Network (DNCNet) and a two-phase training procedure, enhancing the mapping between the denoising result and perceptual representation of the classifier and promoting the classification accuracy further.
- We thoroughly validate the performance of the proposed DNCNet and conduct comparison with the state-of-the-art methods. In particular, to enhance the validation and comparison, we also adopt the Gaussian colored noise and impulse noise which could ensure the experimental settings are much closer to the realistically received noise from the channel and environment, in addition to the widely used additive white Gaussian noise (AWGN) in literature, e.g., [6], [7], [8].

For the sake of readability, the variable notations used in the paper are summarized in Table I. A brief summary of the main attributes of the aforementioned related methods with the proposed method is provided in Table II. The structure of the main models utilized in the experiments is displayed in Appendix A. The remainder of this paper is organized as follows. Section II recalls the discrete radar signal model and three types of noise models. Sections III and IV present the proposed radar signal detection mechanism and the DNCNet. Section V reports the experimental results and detailed analysis. We conclude in Section VI.

TABLE I
LIST OF THE VARIABLE NOTATIONS USED IN THE PAPER

Variable	Description
$x(mT) \in \mathbb{C}^N$	m-th complex sample of the clean signal x with sampling period T and length N
n	additive noise
$\hat{x}$	received noisy signal
$\hat{y}$	the signal after denoising
a, θ	the amplitude and phase of the signal
$p(t)$	probability density function at time t
$\phi(t)$	characteristic function at time t
$P(\omega)$	power spectral density at frequency ω
R	autocorrelation function
$\mathcal{N}(\mu, \sigma)$	Gaussian distribution with the mean value μ and variance σ
$S(\alpha, \beta, \gamma, \delta)$	Alpha-stable distribution with characteristic exponent α, skewness β, scale γ and location δ
$\mathcal{F}$	the denoising model
$\mathcal{G}$	the classifier
$\lambda_1 L_r$, $\lambda_2 L_c$	reconstruction loss and classification loss with their corresponding weights
Θ	the trainable parameters in the network

II. RADAR SIGNAL AND NOISE MODEL

A. Signal model

In discrete setting, the received signal can be modeled as discrete time complex samples [20]. Let $m \in \mathbb{Z}$ and $T \in \mathbb{R}^+$ be the sample index and the sampling interval, respectively. Then the m-th sample of the received signal $\hat{x}(mT)$ with additive noise reads

$$\hat{x}(mT) = x(mT) + n(mT) = a(mT)\exp(j\theta(mT)) + n(mT), \quad (1)$$

where $x(mT) \in \mathbb{C}^N$ and $n(mT)$ denote the clean signal and noise, respectively; $a(mT)$ is the signal envelope within the pulse interval; and $\theta(mT)$ is the instantaneous phase. The noise signal types e.g. AWGN, Gaussian colored noise and impulse noise detailed below are all rather common in the received radar signals.

B. Noise model

In this subsection, we refer to two models, i.e., Gaussian distribution and Alpha-stable distributions, which are commonly utilized to form noise types including AWGN, colored Gaussian noise and impulse noise in research.

For a Gaussian colored noise (or correlated noise) sequence, the amplitude of noise at any given time instant is correlated with the amplitude of noise occurring at other instants of time [21]. Thus, the noise Power Spectral Density (PSD) could completely describe the Gaussian colored noise. According to the Wiener-Khinchin theorem, the PSD is the (discrete time) Fourier transform of the autocorrelation function $R(p)$. Therefore, for colored noise n_1, its PSD $P_2(\omega)$ yields

$$P_2(\omega) = \sum_{p \in \mathbb{Z}} R(p) \exp(-j\omega p). \quad (2)$$

In the case of uncorrelated (noise) samples this would reduce to AWGN, which also refers to the idea of having uniform

TABLE II
SUMMARY OF THE MAIN ATTRIBUTES OF THE DIVERSE DENOISING APPROACHES RELATED TO OUR METHOD

Method	Residual learning	Noise level estimation	True noise influence	Cascade with high-level task
UNet [14]	×	×	×	×
DnCNN [15]	√	×	×	×
FFDNet [16]	×	√	√	×
PD denoising [17]	√	√	√	×
Deep denoising [18]	√	×	√	√
CBDNet [19]	×	√	√	×
Our method	√	√	√	√

power across the whole frequency band [22]. As a consequence, the PSD of AWGN is constant for all frequencies ranging from $-\infty$ to $+\infty$. Let $n_2 \sim \mathcal{N}(\mu, \sigma)$. We have the Probability Density Function of the Gaussian distribution $p_1(t)$, i.e.,

$$p_1(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{(t-\mu)^2}{2\sigma^2}]. \tag{3}$$

Alpha-stable distributions have been found applications in modeling impulse noise in radar and communication fields [23]. They arise naturally in the study of heavy-tailed distributions, and are usually denoted by $S(\alpha, \beta, \gamma, \delta)$. The parameter $\alpha \in (0, 2]$ is called the characteristic exponent describing the tail of the distribution; $\beta \in [-1, 1]$ is the skewness, where $\beta > 0$ and $\beta < 0$ represent the distribution is right-skewed and left-skewed, respectively; and $\gamma > 0$ and $\delta \in \mathbb{R}$ represent the scale and location, respectively. Let $n_3 \sim S(\alpha, \beta, \gamma, \delta)$. Its characteristic function $\phi(t)$ reads

$$\phi(t) = \begin{cases} \exp\{j\delta t - \gamma^\alpha |t|^\alpha [1 + j\beta \mathrm{sgn(t)} \tan(\frac{\alpha\pi}{2})]\}, & \alpha \neq 1, \\ \exp\{j\delta t - \gamma|t|[1 + j\beta \mathrm{sgn(t)} \frac{2}{\pi} \log|t|]\}, & \alpha = 1, \end{cases} \tag{4}$$

where $\mathrm{sgn}(\cdot)$ is the sign function.

III. Proposed radar signal detection and synthetic procedure

In a military application, we assume that a radar station has gathered tactical intelligence like technical parameters, emitter types and locations in the current threat database about specific noncooperative forces including sensing, communications, guidance, and navigation devices [24]. However, the electromagnetic environment in the battlefield is even more complex due to the barrage jamming, intense noise and unknown emitters existence. Therefore, there is a big gap about spatial electromagnetic spectrum information in the current threat database against the real-time battlefield.

To address this problem, we leverage DNN approaches to restore the corrupted received signals and lay a foundation for subsequent tasks like identifying hostile emitters, analyzing enemy actions and forecasting attack intentions [25], [26]. The learning-based DNN algorithm needs paired-samples for training to compare the discrepancy between clean signals and noisy signals so as to remove the interference in the received signals. To meet this requirement, we utilize parameters of the labeled true received signals to synthesize the corresponding clean signals through the signal generator.

We develop a radar system, which consists of a signal reception unit and a signal synthetic unit. It can serve the data preparation for learning-based DNN algorithms. More specifically, as shown in Fig. 1, the signal reception unit includes a deinterleaving and a parameter estimation procedure. Pulses of different radars are probably intersected because of the widespread usage of numerous radio devices or emitters over the air, and thus they need to be deinterleaved firstly to facilitate subsequent processes [27]. After that, the statistical feature such as carrier frequency, pulse width, angle-of-arrival and other parameters will be extracted through parameter estimation [28]. Combining with electronic intelligence (including emitter types, location, etc.), the received true signal will be labeled and recorded in the current threat database. Besides, the signal synthetic unit is designed to produce the ground-truth signal dataset, without noise and belonging to the identical category of the true signals. Among them, the signal generator produces signals according to the result of parameter estimation and electronic intelligence. The following oscilloscopes in the signal and spectrum analysis part visualize the waveforms' representation, testing and displaying the output of the signal generator. The two signal datasets (i.e., the labeled true received signal and the labeled clean synthetic signal) corresponding to these two units will thereby form a paired-dataset with one-to-one correspondence samples for the subsequent learning-based DNN algorithm.

It is worth remarking that the parameter estimation accuracy in the signal reception unit of the developed radar system is essential since this accuracy has a direct impact on the features of the generated clean synthetic signal, and thus as an example a low accuracy could lead to poor match/alignment between the paired-dataset which would then affect the performance of the subsequent learning-based DNN algorithm. Fortunately, the estimation accuracy of the parameters (i.e., regarding the inter-pulse and intra-pulse of the radar signals) could be well managed by existing techniques. For example, the standard deviation of the estimation about the time parameters (i.e., pulse width, pulse rise time and pulse fall time) and pulse amplitude can be lower than 0.2 for signals with SNRs from 5 dB to 25 dB, see [29]. Moreover, an even lower standard deviation (less than 0.1) can be achieved for calculating instantaneous phase, frequency and amplitude for signals in low SNR conditions of order 0 dB using the advanced pulse train measurement technique based on the moving autocorrelation for parameter extraction proposed in [30]. Therefore, by leveraging these

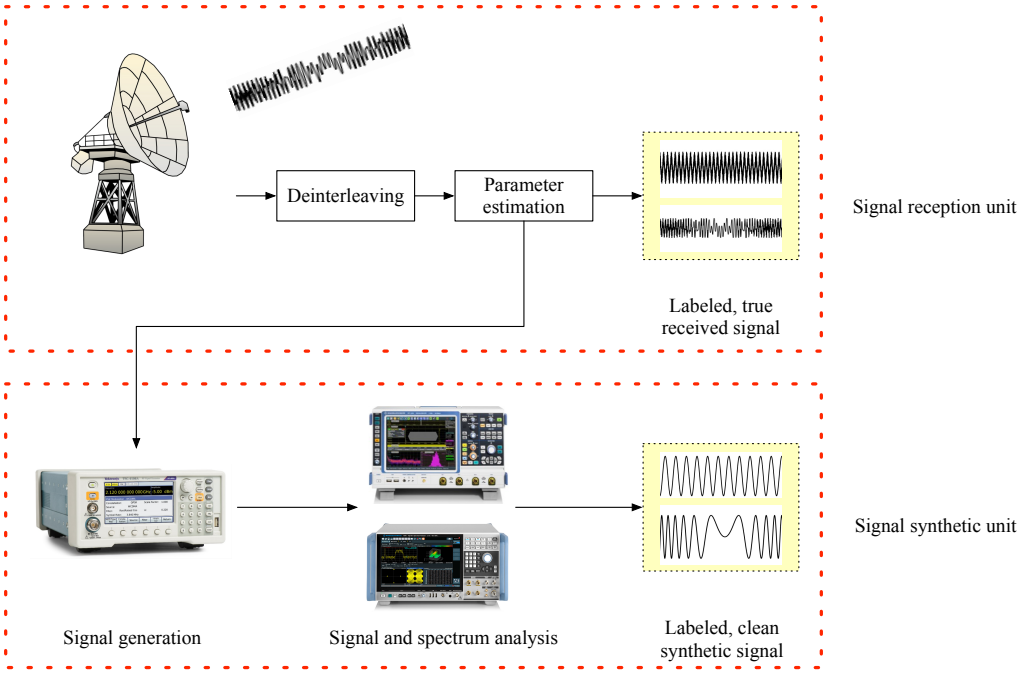


Fig. 1. The radar signal reception and synthetic procedure. The signal reception unit conducts the deinterleaving and parameter estimation for the over-the-air radar signals and outputs *the labeled true received signal*. This unit provides guidance for the signal generator in the signal synthetic unit to produce the same category as the received signal but without noise, i.e., *the labeled clean synthetic signal*. The outputs of both units constitute the paired-dataset for the subsequent DNN algorithm.

powerful parameter measurement and estimation methods, the expected essential fine-grained alignment between the paired-dataset can indeed be fulfilled for subsequent focused tasks.

IV. Proposed Denoising and Classification Framework

The overall structure of the proposed DNCNet is shown in Fig. 2. It contains two child models – a denoising model and a classification model – which are trained through two phases. We integrate the denoising task with the high-level vision task in a single manner so as to complete these two tasks concurrently by optimizing both losses jointly. The motivation is partially built upon the real military background, i.e., the received signal restoration is merely the intermediate process whereas the emitter recognition results are higher-value information for the commander.

A. Denoising model

Inspired by the literature in image restoration e.g. [16], [19], our proposed denoising model for time-series radar signals consists of a noise level estimation subnetwork and a U-Net architecture encoder-decoder, see Fig. 3 and Fig. 4, respectively. After fully trained to convergence, it will learn a mapping say $\mathcal{F}$ between the specific type noisy data and the clean data, defined as $\hat{y} = \mathcal{F}(x+n, \Theta)$, where Θ denotes the trainable parameters in the proposed denoising model.

The input of the denoising model is the pairwise of noisy true data and clean synthetic data. Firstly, we treat the complex-valued input or In-Phase and Quadrature (I/Q) samples as a set of $2N$ vectors (recall that N is the signal length). The real and imaginary parts or I/Q channels of the original input are separated and placed in each dimension [9]. We then adapt the convolution and pooling operations in the subnetworks to 1D to extract the feature of the time-series radar signals.

The network structure of the denoising model and the motivation is further explained below. The noise level map, denoted as ϵ, adopted to CNN based denoising methods was firstly proposed in [16]. The input noisy data and ϵ have the same dimension, i.e., $2 \times N$ for our dataset. All elements in ϵ are stretched by the noise variance σ in the earliest formation, i.e., $\begin{pmatrix} \sigma & \sigma & \cdots \\ \sigma & \sigma & \cdots \end{pmatrix}_{2\times N}$. The introduction of the noise level map, acting as a regularization term, enables the denoising algorithm trade-off between the noise reduction and data fidelity. It may unavoidably increase the difficulty to train the model, as well as experience performance degradation if the noise level map does not match the ground-truth well. Fortunately, this potential concern can be resolved by applying a plain five-layer fully convolutional network (without pooling operation) with the asymmetric loss proposed in [19]. Accordingly, the aforementioned mapping between noisy data and clean data can be rewritten as $\hat{y} = \mathcal{F}(x+n, \epsilon, \Theta)$. In the noise level estimation subnetwork, the shape of the input noisy samples is $L_{\text{bs}} \times 2 \times N$, where L_{bs} denotes the batch size. Given $L_{\text{bs}} = 64$, the number of the output channels respectively in the middle convolution layers and the last layer is 32 and 2, with filter size set to 3. Besides, there are zero-padding operations in each convolutional layer. All these settings keep the dimension identical between the input data and the output, see Fig. 3 for more details. The output will then be concatenated with the original input, forming a new sample with the size of

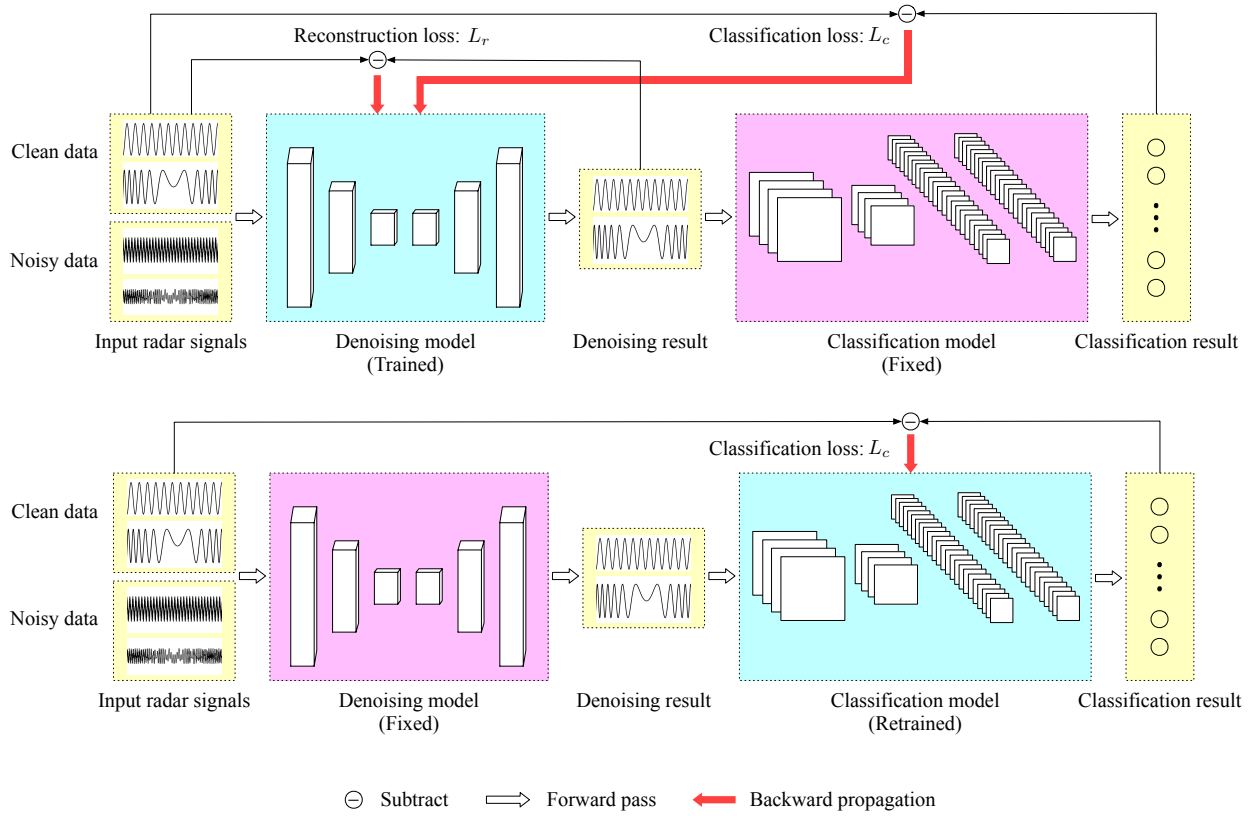


Fig. 2. The structure of the proposed DNCNet. The top subfigure is the first-training process which updates the denoising model (cyan part) by the reconstruction loss and classification loss jointly. The bottom subfigure is the retraining process which only optimizes parameters in the classification model (cyan part) after completing the training of the denoising model.

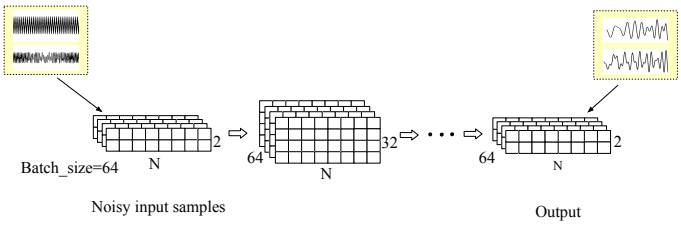


Fig. 3. The noise level estimation subnetwork in the proposed DNCNet. The noisy signals ($64\times2\times N$) are fed into this subnetwork first. Its corresponding output ($64\times2\times N$) will be concatenated with the original input signals, forming a new sample ($64\times4\times N$), which is then fed into the subsequent U-Net architecture subnetwork shown in Fig. 4.

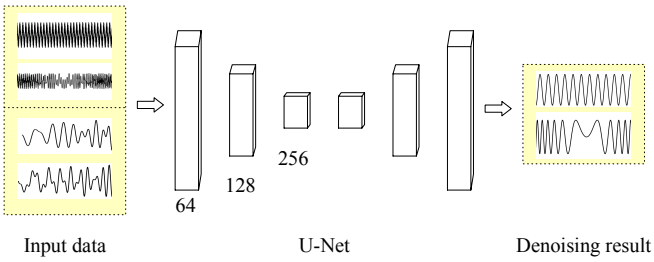


Fig. 4. The U-Net architecture subnetwork in the proposed DNCNet. Its input data is the concatenation of the original noisy data and the output of the noise level estimation subnetwork shown in Fig. 3.

$L_{\text{bs}}\times4\times N$, and fed into the subsequent U-Net architecture subnetwork which is described below.

The U-Net architecture subnetwork is a symmetric combination of downsampling and upsampling operations, see Fig. 4. This design could exploit multi-scale information as well as enlarge receptive field progressively. Conceptually, the downsampling procedure could be viewed as a feature extractor and the upsampling procedure is utilized to reconstruct information according to these extracted high-level features. This scheme can remove the random noise disturbance from the corrupted input data and enhance the robustness. As shown in Fig. 4, the downsampling operation is realized through convolution and average pooling, and the upsampling operation consists of transposed convolution and padding. Similarly, all the convolutional layers, pooling layers and transposed convolutional layers are adapted to 1D to handle the time-series data. There are 16 layers totally and all the filter size is 3. The ReLU nonlinearity is utilized after every convolutional layer except for the last one. The output channels of each downsampling operation is 64, 128 and 256, respectively. The denosing result will be obtained after the corresponding upsampling operations.

B. Classification model

Inspired by [18] and considering the fact that the emitter recognition is one of the main purposes of the received signals restoration, we cascade the denoising model and classification model. By undergoing this process, the denoising results

can be directly fed into a pre-trained classification model which will output the category of the received signal and emitter identity. Moreover, we optimize the parameters in the aforementioned denoising model by a joint loss, which consists of the reconstruction loss in the denoising model and the classification loss in the classification model. This way of combination can also enable the denoising model to produce high-quality perceptual and semantically faithful results.

The ResNet18 [31] is adopted as the basic structure of the classification model with some adjustment made accordingly to match the signal dataset, denoted as *ResNet18-1D*. Specifically, we adapt the convolutional layer and average pooling layer to 1D counterpart to process the $2\times N$ signal data at first. Then, the 7×7 convolutional layer and the max-pooling layer at the top of the original ResNet18 are replaced by a single 3×3 convolutional layer. Moreover, the batch normalization (BN) layers are removed since we empirically find the BN layers help little on this occasion. The detailed structure of the ResNet18-1D is given in Appendix.

C. Two-phase training procedure

Different from the work in [18] which cascades the denoising network and high-level vision network (e.g., a classifier) and then conducts joint training from scratch until convergence, we suggest a two-phase training strategy to restore the noisy signals and recognize their categories in this paper. This strategy is based on the assumption that there exists a pre-trained classifier in the radar reconnaissance system, being trained on the high-SNR signal dataset. In this manner, rather than retraining the classifier from scratch, we attempt to fine-tune it by exploiting its high-level feature extraction capacity to preserve the details extracted from the raw clean data, and using which to assist and enhance the performance of the denoising model.

In preparation, we utilize the clean synthetic signal dataset to train the aforementioned ResNet18-1D from scratch, i.e., the pre-trained classifier. The choice of this type of training dataset stems from two reasons: 1) the classification loss between the restoration results and the clean data can facilitate the denoising model in removing more noise so that the restoration results could be closer to the original signal; and 2) there is not enough known labeled real radar signal data of hostile emitter for training in the realistic military background. Therefore, it is rather accessible and beneficial to synthesize abundant signals according to the received signals.

In the first training phase shown in the upper part of Fig. 2, we train the denoising model (cyan block) with the parameters in the classification model (purple block) fixed. The reconstruction loss L_r is the mean squared error between the output of the denoising model and the clean data, i.e.,

$$L_r(x,\hat{x}) = \frac{1}{2N}\sum_{i=1}^{2}\sum_{j=1}^{N}[\mathcal{F}(\hat{x})_{i,j} - x_{i,j}]^2, \tag{5}$$

where $\hat{x} = x+n$ represents the noisy data. Let $\mathcal{G}(\cdot)$ represent the classifier. The classification loss, defined as $L_c(\mathcal{G}[\mathcal{F}(\hat{x})], y)$ and abbreviated as L_c, is the cross-entropy between the classification result of the denoising data and the corresponding ground-truth labels y. The trainable parameters in the denoising model are jointly optimized according to the combination of L_r and L_c, i.e.,

$$L = \lambda_1 L_r + \lambda_2 L_c, \tag{6}$$

where λ_1 and λ_2 are the weights forming a trade-off between the two losses L_r and L_c. As for the loss function design, previous works have exploited more specific ones for their tasks, e.g., using a binary mask loss to integrate object detection and semantic segmentation [32], or considering edge details in gradient-domain [33] and/or statistical properties of the noise [34]. In this paper, we implement the classification loss and reconstruction loss in the denoising procedure. Other types which could involve e.g. the amplitude, phase and temporal dependency of the radar signals will be investigated as future work.

After the denoising model is fully trained, the classification model (cyan block) shown in the lower part of Fig. 2 is retrained in the second training phase. In this phase, the parameters in the denoising model (purple block) are fixed and only the classification loss L_c is utilized to update weights and biases of filters in the classification model. This process will not influence the denoising performance, but will further enhance the mapping between the denoising result and the clean data. Compared to the method in [18], for example, the classification accuracy can be clearly promoted utilising this retraining, see Section V for more details.

TABLE III
FREQUENCY MODULATION SIGNAL PARAMETERS

LFM/SFM	Pulse width (μs)	Band width (MHz)
1	1–16	1
2	1–16	5
3	10	20
4	1–20	180

TABLE IV
POLYPHASE CODE SIGNAL PARAMETERS

Frank/P1/P2/P3/P4	Time width (μs)	Bits
1	1	64
2	1.5	64
3	2	64

V. EXPERIMENTS

A. Datasets

Self-built synthetic dataset and publicly available dataset are used in the experiments. The self-built dataset is generated by Numpy, science computing package of python. This dataset, named "SIGNAL-8", contains eight different radar signal types and can be split into two categories, i.e., the frequency modulation (FM) and phase modulation (PM). The FM signals, compared to previous works in [35], [7] which used the positive slope LFM only, are augmented here by adding negative slope LFM and Sine FM. For PM signals, there are 64-bit polyphase code signals including Frank code, P1, P2,

P3 and P4 code. Eight different electromagnetic environments are formed with AWGN based on the SNR respectively set to −10dB, −8dB, −5dB, −1dB, 5dB, 8dB, 10dB and noise free. Each signal type in a single environment has 2,000 samples. The length of each signal sequence N is 1,000. The details of the signal parameters are given in Table III and Table IV.

The other dataset, called "RADIOML 2018.01A", is publicly available and was built in [11] by GNU radio tools, consisting of 24 different analog and digital modulators including OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK and OQPSK. This dataset includes both synthetic simulated channel effects and over-the-air propagation loss covering a wide range of SNR levels from −20dB to 30dB with a step of 2dB, which is commonly utilized in research experiments. The length of samples N is $1,024$. Except for the AWGN used in the original dataset, we also use the Gaussian colored noise and impulse noise introduced in section II to generate some extra datasets (see e.g. Section V-E) for more detailed methods validation and comparison.

B. Transfer performance of the conventional single classification models

To test the performance of the conventional single classification models, several conventional and well-known classifiers including the LeNet, AlexNet, VGG and ResNet are considered here. This test is to expose their weakness in dealing with the challenge raised by distribution mismatch between the training dataset and the test dataset in radar signals, which showcases the necessity of exploiting signal restoration before classification. To implement these classifiers, we first adjust their convolutional layers and pooling layers to 1D to handle time-series signal datasets.

For the SIGNAL-8 dataset, we firstly train the aforementioned classifiers by the noise free signal data, and evaluate their performance on other datasets with different SNR levels. Then these classifiers are trained in the environment with SNR equal to −5dB, and tested in the environments with all SNR levels used. For the RADIOML 2018.01A dataset, akin to the above settings, the classifiers are trained in the environments with SNR equal to 30dB and −10dB, respectively, and tested in the environments with all SNR levels used.

The experimental results are given in Fig. 5 and Fig.6. It is evident that, when the test data has similar distribution to the training data, these classifiers can perform excellently on 1D radar signal data given their achieved over 90% test accuracy. However, when the test data has a related but not identical distribution to the training data, these classifiers perform weakly at once, and even nearly fail when the distribution mismatch becomes severe. This level of performance reveals that single CNN classifiers indeed suffer from the circumstance that the test dataset has different distribution with the training dataset. In practice, when a classifier is trained with the received signals in the current threat database, it may suffer if the battlefield electromagnetic environments change which lead to e.g. a lower SNR.

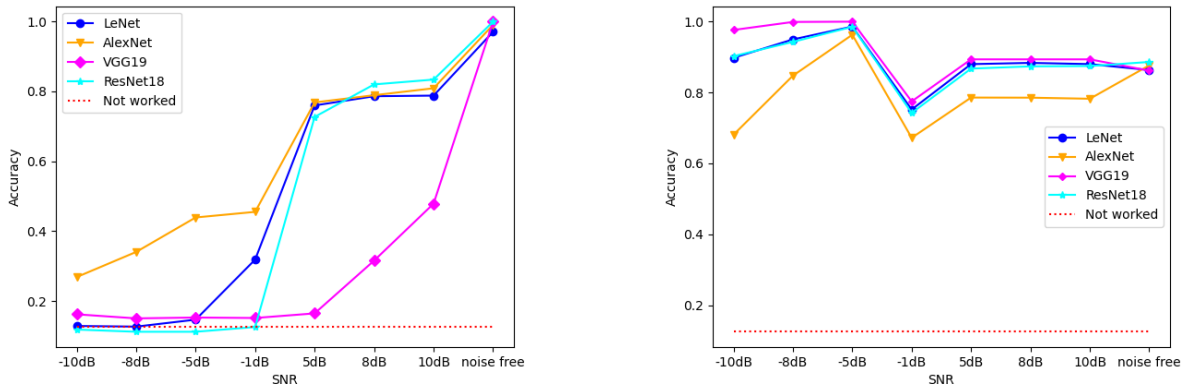
C. Performance comparison between the single denoising model and its counterpart cascading with classification model

We in this subsection compare the performance in signal restoration and classification between the single denoising model and its counterpart which cascades with classification model proposed in [18], coined as "cascading model" for simplicity. (Recall that the working principle of the model in [18] cascades the denoising network and high-level vision network [e.g., a classifier] and then conducts joint training from scratch until convergence). As for the single denoising model, except for that proposed in our method, we also implement the DnCNN [15] as well as FFDNet [16], adapting them to match the signal dataset. Specifically, for the DnCNN, we set the number of output channels to 2 (which is 3 for RGB image data in the original work). Simultaneously, we adjust the convolutional layers and BN layers to 1D. For the FFDNet, except for the aforementioned adjustment used in DnCNN, we omit the downsampling layer and upsampling layer which are originally designed for image patches generation and high-resolution realization. The newly adapted DnCNN and FFDNet for signals here are denoted as *DnCNN-1D* and *FFDNet-1D*, respectively; please refer to their detailed structures in Appendix. Recall that the difference between the single denoising model and the cascading model is that the former is trained only by using the reconstruction loss L_r, while the latter is trained jointly by using the reconstruction loss L_r and the classification loss L_c.

In the following experiments, if not particularly specified, for the SIGNAL-8 dataset, we train the single denoising model and the cascading model on the pairwise noise free dataset and the one with SNR equal to 5dB. For the RADIOML 2018.01A dataset, we train them on the pairwise datasets with SNR equal to 30dB and 5dB.

For the signal restoration ability, the performance of the single denoising model and the cascading model on the SIGNAL-8 dataset and the RADIOML 2018.01A dataset is shown in Fig. 7. It reveals that both methods have prominent denoising functions and they could restore the basic signal features like rising edges, failing edges and other obvious fluctuations. Regarding the signal amplitude, the restoration results of the cascading model are much closer to the clean signal compared to those obtained by the single denoising model.

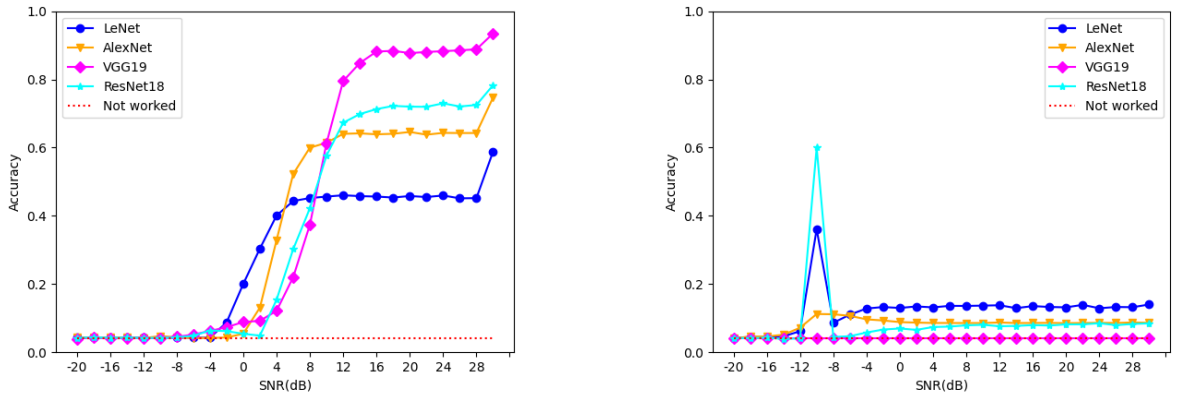
For the classification accuracy, we adopt the same pretrained classifier as the cascading model for the single denoising model to recognize its denoising results. Both training processes are conducted in the environment with SNR equal to 5dB. The numerical results in Table V show that the classification accuracy of the cascading model on both datasets has a significant promotion compared to the proposed denoising subnetwork, i.e., by 44.04% and 18.31% increase, respectively. However, when the noise intensity is intensified (i.e., SNR is lowered), the cascading model also suffers significantly from degradation, i.e., its accuracy was reduced by nearly a half. Since the average accuracy of the cascading model in all environments outperforms that of the the proposed denoising subnetwork (i.e., 67.22% vs. 52.26% and 44.59%



(a) Models trained on noise free dataset

(b) Models trained on dataset with SNR = −5dB

Fig. 5. Performance of conventional CNNs (i.e, LeNet, AlexNet, VGG and ResNet) on the SIGNAL-8 dataset. (a)–(b): Test accuracies when models are trained on datasets with noise free and SNR = −5dB, respectively.



(a) Models trained on dataset with SNR = 30dB

(b) Models trained on dataset with SNR = −10dB

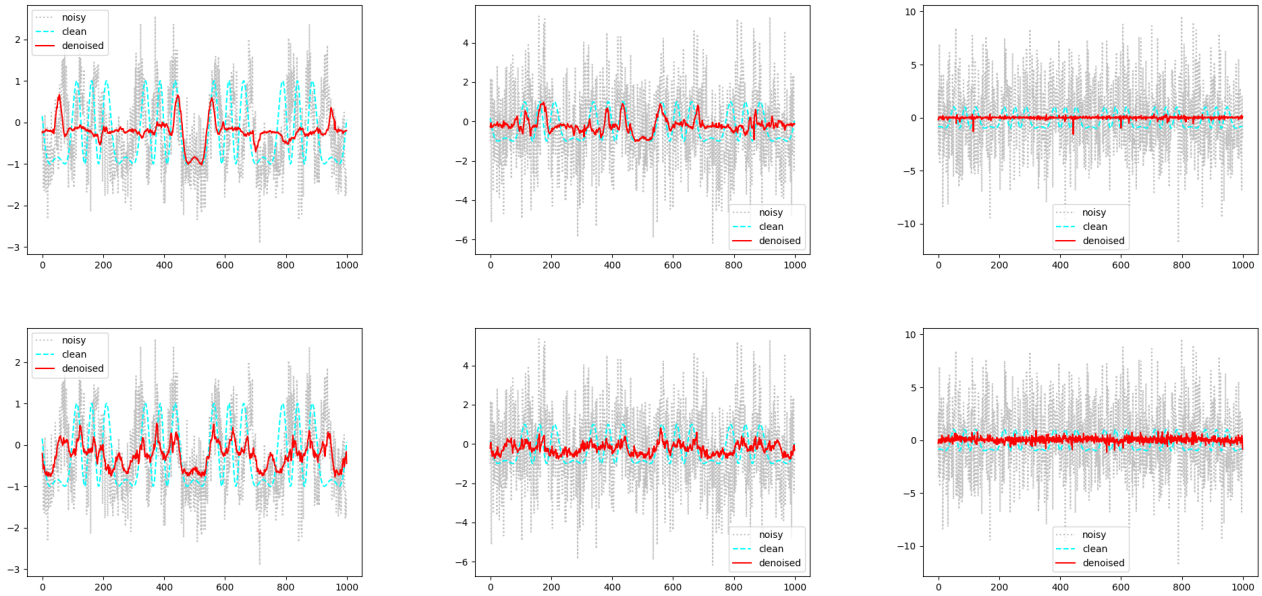
Fig. 6. Performance of conventional CNNs (i.e, LeNet, AlexNet, VGG and ResNet) on the RADIOML 2018.01A dataset. (a)–(b): Test accuracies when models are trained on datasets with SNR = 30dB and SNR = −10dB, respectively.

vs. 42.63%), the cascading model in the following is used for further comparison with our proposed method. Incidentally, the performance of the baseline methods, i.e., FFDNet-1D and DnCNN-1D, in Table V shows that they both suffer more serious degradation than the proposed denoising subnetwork when SNR decreases in both datasets; moreover, compared to DnCNN-1D, FFDNet-1D cannot converge on the RADIOML 2018.01A dataset.

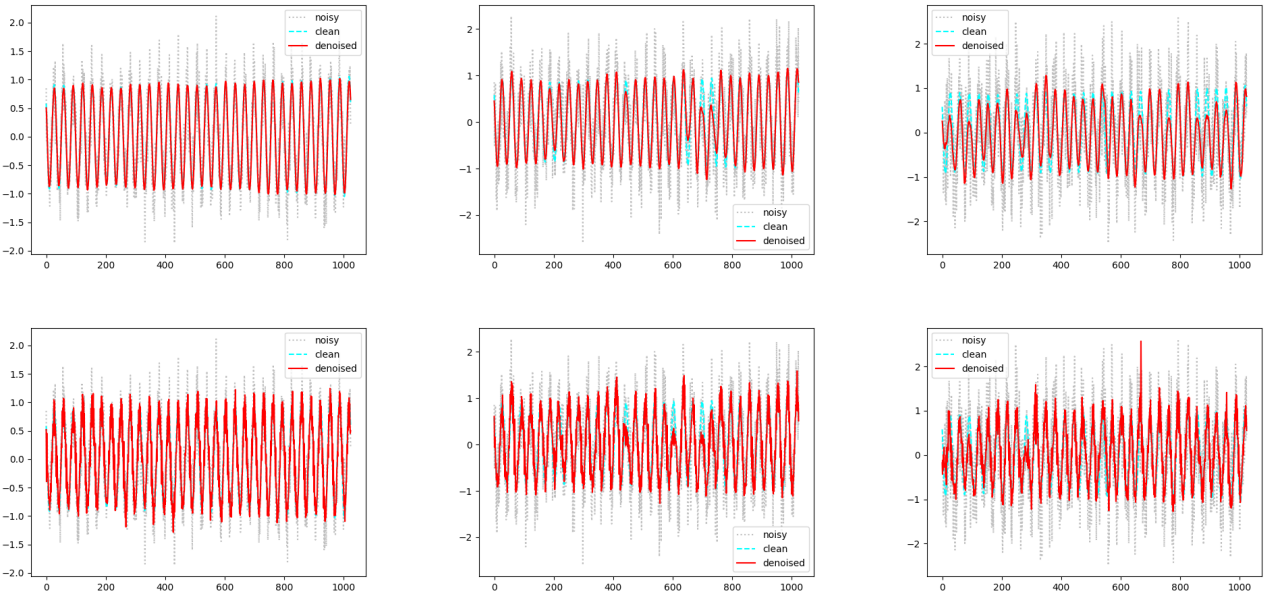
Finally, we investigate the effect of the weights λ_1 and λ_2 used in Eq. (6) to train our denoising model in cascade manner. Different ratio of λ_1/λ_2 is applied on both the SIGNAL-8 dataset (trained on the pairwise noise free signal data and the one with SNR equal to 5dB and tested on the one with SNR equal to 5dB) and RADIOML 2018.01A dataset (trained on the pairwise signal data with SNR equal to 30dB and 5dB and tested on the one with SNR equal to 5dB). The classification accuracy of our cascading model without retraining for the ratio of λ_1/λ_2 set to e.g. 0.2, 0.5, 1.0 and 5.0 is illustrated in Fig. 8. The excellent result shows the significant improvement compared to the single denoising subnetwork. For example, even the lowest accuracy (i.e., 68.6%) obtained (by $\lambda_1/\lambda_2 = 0.2$) has a near 15.5% improvement against the single denoising subnetwork on the RADIOML 2018.01A dataset. Moreover, we also see that a larger ratio of λ_1/λ_2 leads to a higher classification accuracy on both datasets. This also implies that a higher weight (i.e., a larger ratio of λ_1/λ_2) for the denoising subnetwork could provide higher restoration quality of the noisy signals which could then help the classifier perform more accurately (cf. the next subsection for more experiments with retraining). A caveat is that the ratio of λ_1/λ_2 is not recommended to be too large since in this case the effect of the classification loss will be weakened.

D. Performance of the proposed two-phase training

Compared to the cascading model in [18], our proposed two-phase training procedure aims to reinforce the mapping between the denoising result and the perceptual respresentation. Note that after completing the training of the cascading model,



(a) Denoising results on the SIGNAL-8 dataset



(b) Denoising results on the RADIOML 2018.01A dataset

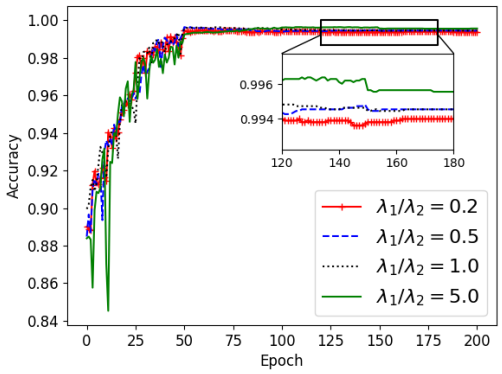
Fig. 7. Performance comparison in terms of signal restoration. (a)–(b): Denoising results on the SIGNAL-8 dataset and RADIOML 2018.01A dataset, respectively. The first and second rows in each panel are the results of the single denoising model and the cascading model, respectively.

our proposed method will retrain the classifier by exploiting the denoising result. Moreover, another way jointly training the denoising subnetwork and classification subnetwork until convergence in a single procedure rather than in the sequential manner, coined as "one-round training" for simplicity, is also implemented for comparison.

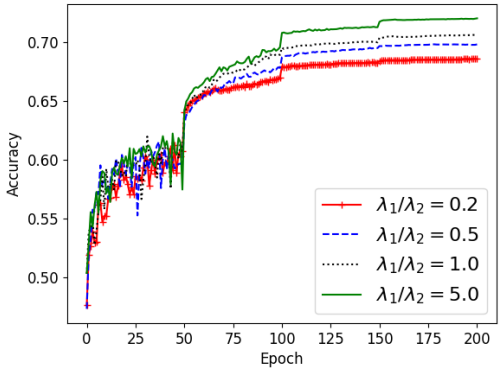
The numerical results of the cascading model and our proposed method are given in Table VI. It is clear that our method brings a dramatic enhancement in terms of the classification accuracy in all the test environments with different SNR compared to the cascading model, i.e., $\sim$ 22% maximum increase and 11.9% and 15.8% average increases respectively on these two datasets. Note that the performance of the signal restoration is not improved since the denoising model here is fixed in the retraining process. Our proposed method ensures the classifier fine-tunes according to the output of the denoising model, and thus tightens the mapping relationship between the signal restoration results and perceptual representation of the classifier. This strategy as shown above can indeed bring a dramatic enhancement in terms of the classification accuracy,

TABLE V
CLASSIFICATION ACCURACY COMPARISON ON THE SIGNAL-8 DATASET AND RADIOML 2018.01A DATASET

Dataset	SNR (dB)	FFDNet-1D	DnCNN-1D	Our denoising subnetwork	Cascading model
SIGNAL-8	5	83.54%	59.94%	55.55%	**99.59%**
	−5	19.86%	46.18%	53.53%	**58.94%**
	−10	18.67%	26.09%	**47.69%**	43.14%
	Average	40.69%	44.07%	52.26%	**67.22%**
RADIOML 2018.01A	5	8.85%	45.81%	53.19%	**71.50%**
	1	4.19%	33.56%	**41.60%**	39.27%
	−1	4.38%	25.58%	**33.13%**	23.00%
	Average	5.81%	34.98%	42.64%	**44.59%**



(a) Classification result on the SIGNAL-8 dataset



(b) Classification result on the RADIOML 2018.01A dataset

Fig. 8. Classification performance of our cascading model without retraining for different ratio of λ_1/λ_2 in Eq. (6). (a) and (b): Results on the SIGNAL-8 dataset and RADIOML 2018.01A dataset, respectively (see the main text for the detailed setting).

TABLE VI
CLASSIFICATION ACCURACY COMPARISON AFTER RETRAINED ON THE SIGNAL DATASET AND RADIOML 2018.01A DATASET

Dataset	SNR (dB)	Cascading model	Ours
SIGNAL-8	5	99.59%	**99.79%**
	−5	58.94%	**80.72%**
	−10	43.14%	**56.86%**
	Average	67.22%	**79.12%**
RADIOML 2018.01A	5	71.50%	**83.83%**
	1	39.27%	**56.49%**
	−1	23.00%	**40.84%**
	Average	44.59%	**60.39%**

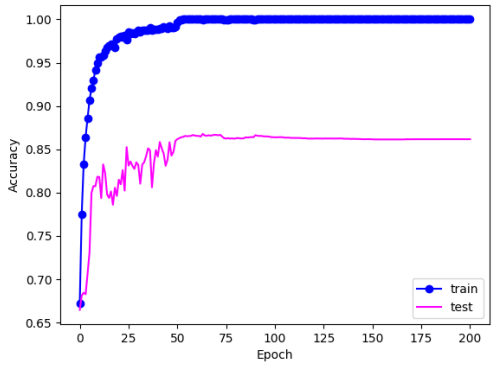


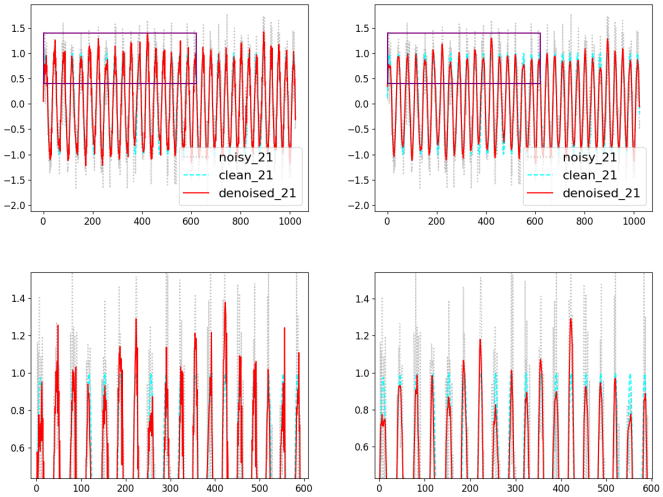
Fig. 9. Training and test classification accuracy of the one-round training strategy, i.e., simultaneously updating the denoising subnetwork and classification subnetwork. The test accuracy of the one-round training strategy is close to the proposed two-phase training strategy; however, it suffers a severe overfitting given the big accuracy gap between the training and test.

which provides a new pathway of improving the classification accuracy of noisy data except for directly strengthening the ability of DNNs in noise removal.

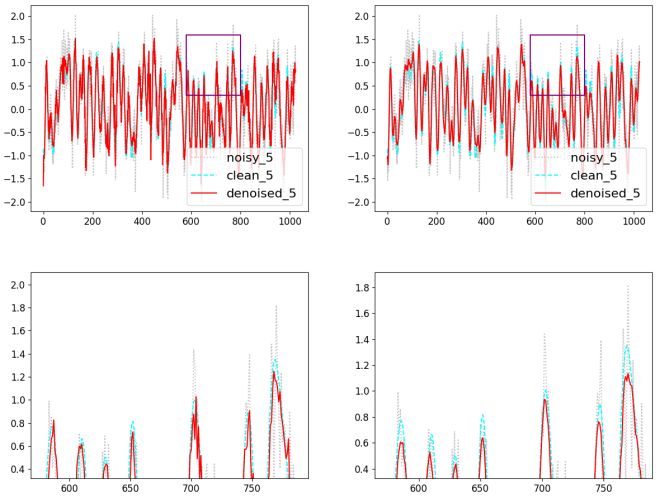
We finally conduct performance comparison between the one-round training strategy and the two-phase training on the RADIOML 2018.01A dataset in term of restoration and classification. The training and test classification accuracy about the one-round training is presented in Fig. 9, revealing that this strategy may suffer severe overfitting in that the accuracy gap between the training and test exceeds 15%, even though the test accuracy is as good as that of the two-phase training strategy (cf. Table VI regarding SNR = 5dB on the RADIOML 2018.01A dataset). For the restoration performance, the results and zoomed-in details of both strategies are given in Fig. 10. It shows that the proposed two-phase training can achieve more accurate restoration regarding the signal amplitude against the one-round training, even though the results of the one-round training seem smoother with less burrs.

TABLE VII
CLASSIFICATION ACCURACY COMPARISON ON THE GAUSSIAN COLORED NOISE DATASET AND IMPULSE NOISE DATASET

Noise type	SNR (dB)	Before denoising	Single denoising model	Cascading model	DNCNet
Gaussian colored noise	5	6.79%	55.58%	67.21%	**83.85%**
	1	4.34%	41.92%	32.34%	**51.55%**
	−1	4.64%	33.73%	14.76%	**33.96%**
Impulse noise	5	4.54%	16.88%	43.27%	**77.25%**
	1	4.49%	5.72%	13.89%	**24.24%**
	−1	4.16%	6.82%	**7. 29%**	5.92%



(a) Example 1 of the restoration performance



(b) Example 2 of the restoration performance

Fig. 10. Restoration performance comparison between the proposed two-phase training strategy (left column) and the one-round training (right column). The 1st row in (a) and (b) gives two examples of the restoration result on the RADIOML 2018.01A dataset; and the 2nd row of each subfigure presents the zoomed-in details of the purple rectangular areas in the 1st row.

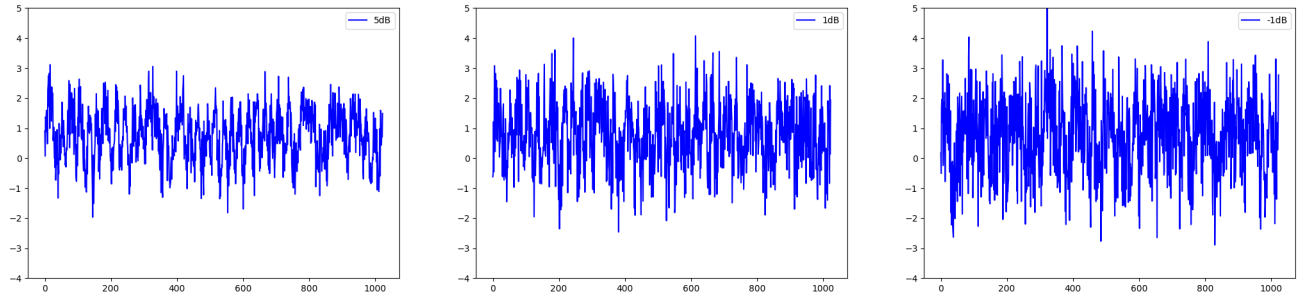
E. Performance comparison on other types of noise model

To further validate the excellent performance of the proposed method and complete the comparison, we conduct further experiments using the RADIOML 2018.01A dataset, which is now corrupted by Gaussian colored noise and impulse noise. Fig. 11 shows some noisy samples which contain Gaussian colored noise and impulse noise. It indicates that the amplitude of the signal samples with impulse noise is larger than that of Gaussian colored noise with the same SNR. Fig. 12 showcases some restoration results of the proposed DNCNet on the dataset with impulse noise and SNR = 5dB.

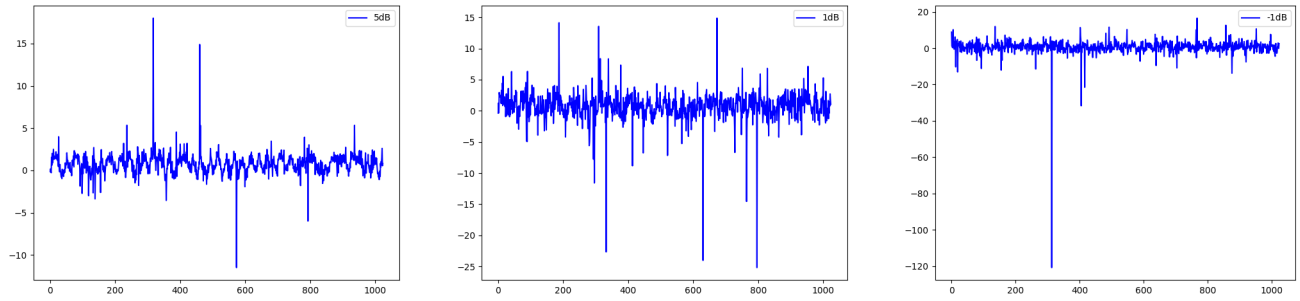
The quantitative comparison in terms of the classification accuracy between the single denoising model, cascading model and the proposed DNCNet is given in Table VII, regarding Gaussian colored noise and impulse noise. We see that the proposed DNCNet outperforms all the competing approaches by a large margin across different noise levels, except for the impulse noise situation with SNR equal to −1dB. It is worth mentioning that the impulse noise situation with SNR equal to −1dB is quite challenging for all the methods, which all achieved accuracy less than 10%. The possible reason is that the impulse noise model has many peak values in the signal amplitude which is non-successive, and the low SNR (e.g. −1dB) makes the situation even more challenging for DNNs. In contrast, the Gaussian white noise and Gaussian colored noise are handled quite well, given the evidence that much higher classification accuracy is achieved on the datasets with Gaussian white noise and Gaussian colored noise against the impulse noise at the same noise level. Moreover, the experimental results also reveal that the cascading model is not that stable as the single denoising model and ours since its classification accuracy experiences dramatic fluctuations across different noise levels.

VI. CONCLUSION

In this paper, we proposed a DNCNet to restore and recognize noisy radar signal data, motivated by the challenge raised by the distribution mismatch between the training and test datasets which is ubiquitous in the field of radio signal recognition. Moreover, a radar signal detection and synthetic mechanism was designed based on the intercepted signal and signal generator to prepare the training dataset for the DNCNet. The combination of the denoising model and classification model was experimentally validated by the highly promoted restoration quality. Most importantly, the classification accuracy can also be significantly improved



(a) Signal samples with additive Gaussian colored noise.



(b) Signal samples with additive impulse noise.

Fig. 11. Examples of the noisy signals corrupted by additive Gaussian colored noise (a) and impulse noise (b) with different SNR on the RADIOML 2018.01A dataset. From columns left to right, the SNR used is 5dB, 1dB and -1dB, respectively.

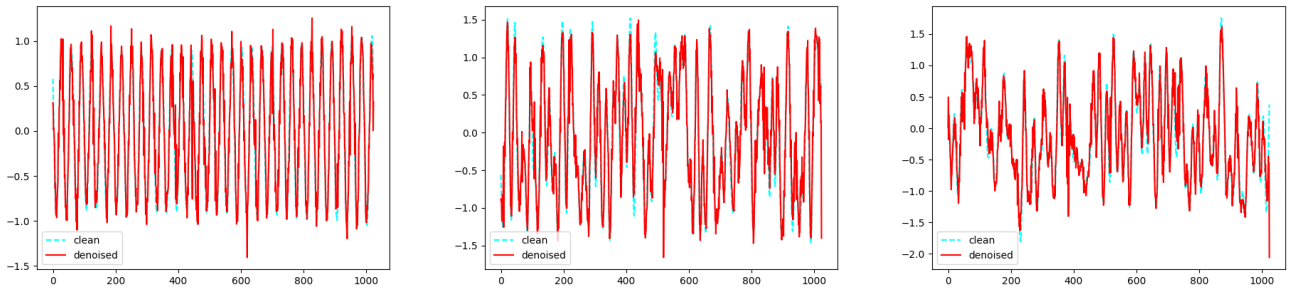


Fig. 12. Restoration results of the proposed DNCNet on the RADIOML 2018.01A dataset with impulse noise and SNR = 5dB.

with the usage of the proposed two-phase training procedure. Extensive experiments including those on different types of noise model demonstrated the excellent performance of our proposed DNCNet compared to the state-of-the-art techniques. The impulse noise situation with low SNR might be of interest for future investigation. Another avenue for future work might be exploiting interpretability techniques in understanding the black-box behavior of the DNN, e.g., the work in [36], which might shed light on developing networks to further enhance the performance in signal restoration.

APPENDIX

Fig. 13 gives the structures of the classifier *ResNet18-1D* (see the first column of Fig. 13) utilized in the proposed method, and the classifiers *DnCNN-1D* and *FFDNet-1D* (see the middle and right columns of Fig. 13) adapted from DnCNN and FFDNet, respectively, utilized for comparison.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant no. 61671453).

REFERENCES

[1] S. Liu, X. Yan, P. Li, X. Hao, and K. Wang, "Radar emitter recognition based on sift position and scale features," *IEEE Trans. Circuits Syst. II*, vol. 65, no. 12, pp. 2062–2066, 2018.

[2] J. Lundén and V. Koivunen, "Automatic radar waveform recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 1, pp. 124–136, 2007.

[3] G. López-Risueño, J. Grajal, and A. Sanz-Osorio, "Digital channelized receiver based on time-frequency analysis for signal interception," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 879–898, 2005.

[4] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 445–458, 2019.

[5] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019.

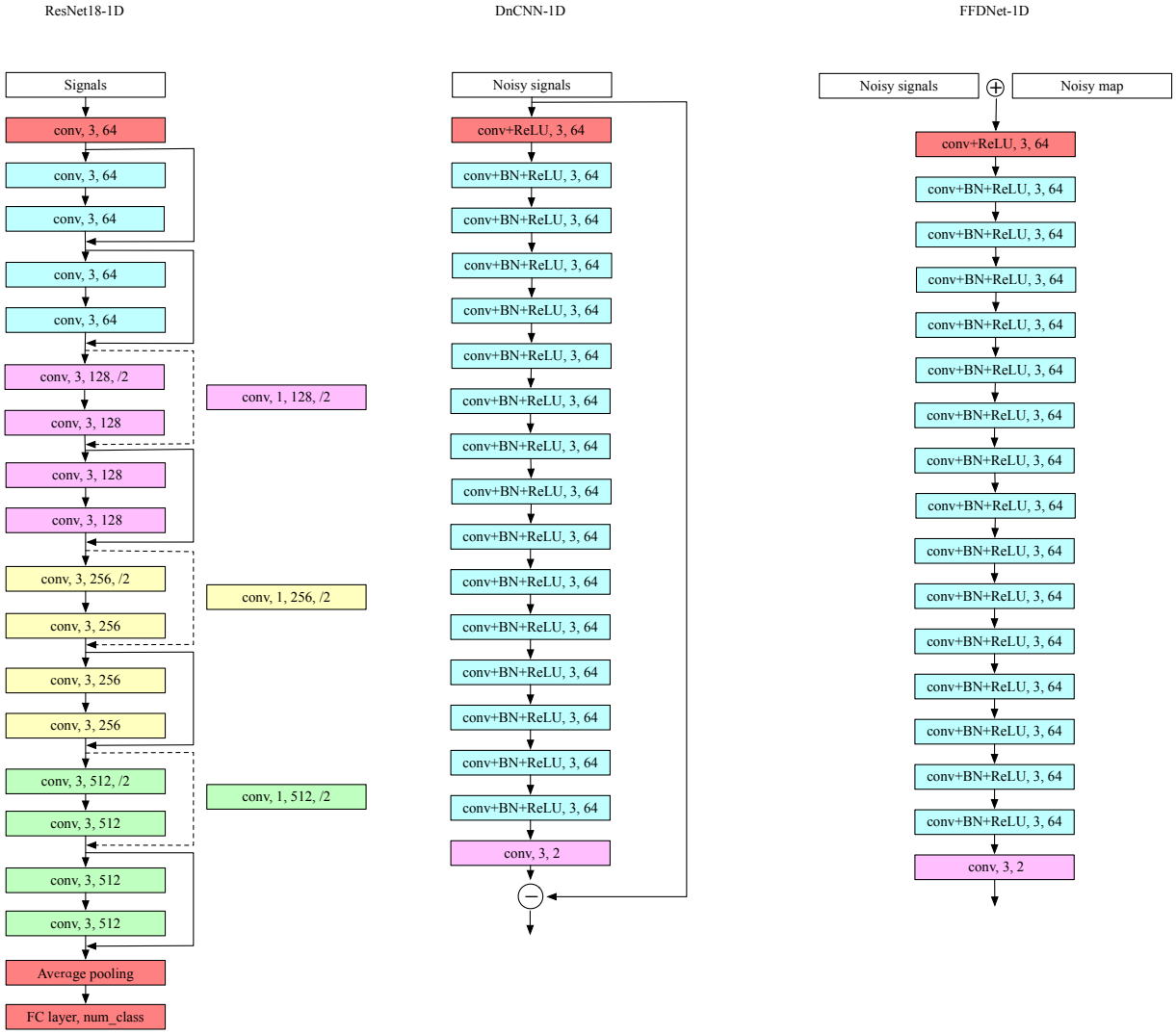


Fig. 13. The structures of the ResNet18-1D, DnCNN-1D and FFDNet-1D from left to right, respectively. The values in the convolution rectangle (e.g., [conv, 3, 128, /2]) refer to the kernel size, the number of output channels and the stride, respectively. The default value of the stride is 1. The "num_class" in FC layers of ResNet18-1D represents the number of units which is equal to the number of the classes in datasets, i.e., num_class is 8 for the SIGNAL-8 dataset and 24 for the RADIOML 2018.01A dataset. The ReLU activations in ResNet18-1D are not shown in this figure for abbreviation, while those are displayed in the other two denosiers. There is a long-path residual connection in the DnCNN-1D that performs a minus operation between the output of the final convolution layer and the input noisy observation to reconstruct the latent clean data. Moreover, the $\oplus$ symbol in FFDNet-1D means the concatenation between the noisy data and the noise map.

[6] C. Wang, J. Wang, and X. Zhang, "Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network," in *IEEE Int. Conf. Acoust. Speech Signal Process Proc.*, 2017, pp. 2437–2441.

[7] S. Kong, M. Kim, L. M. Hoang, and E. Kim, "Automatic lpi radar waveform recognition using cnn," *IEEE Access*, vol. 6, pp. 4207–4219, 2018.

[8] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of spwvd and bjd," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 469–478, 2019.

[9] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Springer, 2016, pp. 213–226.

[10] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[11] 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, 2018.

[12] E. Blossom, "Gnu radio: tools for exploring the radio frequency spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.

[13] P. Wei, Z. Xie, H. Lu, Z. Zhan, Q. Ye, W. Zuo, and L. Lin, "Component divide-and-conquer for real-world image super-resolution," in *Lect. Notes Comput. Sci.* Springer, 2020, pp. 101–117.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[15] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.

[16] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2018.

[17] Y. Zhou, J. Jiao, H. Huang, Y. Wang, J. Wang, H. Shi, and T. Huang, "When awgn-based denoiser meets real noises," in *AAAI - AAAI Conf. Artif. Intell.*, vol. 34, no. 07, 2020, pp. 13 074–13 081.

[18] D. Liu, B. Wen, J. Jiao, X. Liu, Z. Wang, and T. S. Huang, "Connecting image denoising and high-level vision tasks via deep learning," *IEEE Trans. Image Process.*, vol. 29, pp. 3695–3706, 2020.

[19] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional

blind denoising of real photographs," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 1712–1722.
[20] K. Cui, W. Wu, J. Huang, X. Chen, and N. Yuan, "Doa estimation of lfm signals based on stft and multiple invariance esprit," *Int. J. Electron. Commun.*, vol. 77, pp. 10–17, 2017.
[21] F. Wen, X. Xiong, J. Su, and Z. Zhang, "Angle estimation for bistatic mimo radar in the presence of spatial colored noise," *Signal processing*, vol. 134, pp. 261–267, 2017.
[22] A. Thangaraj, G. Kramer, and G. Böcherer, "Capacity bounds for discrete-time, amplitude-constrained, additive white gaussian noise channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4172–4182, 2017.
[23] M. L. De Freitas, M. Egan, L. Clavier, A. Goupil, G. W. Peters, and N. Azzaoui, "Capacity bounds for additive symmetric alpha-stable noise channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 5115–5123, 2017.
[24] A. Spezio, "Electronic warfare systems," *IEEE Trans. Microw. Theory Techn.*, vol. 50, pp. 633–644, Mar. 2002. [Online]. Available: http://ieeexplore.ieee.org/document/989948/
[25] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, 2019.
[26] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, 2018.
[27] Z. Liu, "Online pulse deinterleaving with finite automata," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1139–1147, 2019.
[28] Z. Liu and S. Y. Philip, "Classification, denoising, and deinterleaving of pulse streams with recurrent neural networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 1624–1639, 2018.
[29] M. A. Zanina, A. A. Belov, and S. V. Volvenko, "Estimation of accuracy of algorithm for measuring radiofrequency pulse parameters," in *IEEE Int. Conf. Electr. Eng. Photonics, EExPolytech*. IEEE, 2018, pp. 98–102.
[30] R. Niranjan, A. Singh, and C. Rao, "High accuracy parameter estimation for advanced radar identification of electronic intelligence system." *Def. Sci. J.*, vol. 70, no. 3, pp. 278–284, 2020.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020.
[33] L. Xu, J. S. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *Int. Conf. Mach. Learn.*, vol. 2, Lile, France, 2015, pp. 1669–1678.
[34] S. Vitale, G. Ferraioli, and V. Pascazio, "Multi-objective cnn-based algorithm for sar despeckling," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9336–9349, 2021.
[35] M. Zhang, L. Liu, and M. Diao, "Lpi radar waveform recognition based on time-frequency distribution," *Sensors*, vol. 16, no. 10, pp. 1682–1701, 2016.
[36] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "Xai meets mobile traffic classification: Understanding and improving multimodal deep learning architectures," *IEEE Trans. Netw. Service Manag.*, pp. 1–1, 2021.