electronics

MDPI

Article

Direction of Arrival Estimation Based on DNN and CNN

Wu Cao [1], Wen Ren [2], Zhenyu Zhang [3], Weiqiang Huang [4], Jun Zou [3,*] and Guangzu Liu [3]

[1] School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing 100081, China; 3220185055@bit.edu.cn
[2] Space Star Technology Co., Ltd., China Academy of Space Technology, Beijing 100095, China; renw@spacestar.com.cn
[3] School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; zzy2023@njust.edu.cn (Z.Z.); liuguangzu@njust.edu.cn (G.L.)
[4] Nanjing Panda Handa Technology Co., Ltd., China Electronics Technology Group Corporation, Nanjing 210001, China; 13851539419@163.com
* Correspondence: jun_zou@njust.edu.cn

Abstract: The accuracy of Direction of Arrival (DOA) estimation primarily depends on the precision of the data. When the receiver uses a low-precision analog-to-digital converter (ADC), traditional DOA estimation algorithms exhibit poor accuracy. To face the challenge of multi-target DOA estimation in scenarios with low-precision ADC quantized sampling, this paper proposes a novel DOA estimation algorithm for quantized signals based on classification problems. A deep learning network was constructed using Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), divided into the quantized signal recovery framework and the DOA estimation framework. The DNN network is utilized to recover signals that have undergone low-precision quantization, while the CNN network addresses the classification problem to estimate the DOA from received data with an unknown number of signal sources. A comprehensive analysis of the impact of signal-to-noise ratio (SNR), the number of array elements, and the number of quantization bits on the proposed algorithm was conducted. Simulation results indicate that the proposed algorithm exhibits superior DOA estimation performance in low-precision scenarios, characterized by reduced computational complexity, thereby facilitating real-time DOA estimation.

Keywords: DNN; CNN; DOA estimation; low-precision; classification

check for updates

Citation: Cao, W.; Ren, W.; Zhang, Z.; Huang, W.; Zou, J.; Liu, G. Direction of Arrival Estimation Based on DNN and CNN. *Electronics* **2024**, *13*, 3866. https://doi.org/10.3390/electronics13193866

Academic Editor: Djuradj Budimir

Received: 31 August 2024
Revised: 19 September 2024
Accepted: 23 September 2024
Published: 29 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The estimation of Direction of Arrival (DOA) is a central research focus within the domain of array signal processing [1–5]. Over several decades, numerous algorithms have been developed. These algorithms can be broadly categorized into two main types: beamforming and subspace fitting. Beamforming (BF) technology [6–11] is a type of spatial filtering that can receive signals through generated beams and reduce interference. DOA estimation algorithms based on Conventional Beamforming (CBF) generate multiple beams and then scan the entire space, detecting the target direction by the maximum output power. However, the accuracy of this algorithm is limited by the Rayleigh resolution limit, making it unable to distinguish between two closely spaced direction vectors [12]. Subspace fitting algorithms overcome the Rayleigh limit and achieve super-resolution, represented by the multiple signal classification (MUSIC) [13] and estimation of signal parameters via rotational invariance techniques (ESPRIT) [14] algorithms. However, the requirement for matrix eigen decomposition in the MUSIC algorithm results in high computational complexity, making it challenging to use in real-time estimation scenarios. In scenarios where the count of signal sources is unknown, the MUSIC algorithm first requires an estimation of the count of sources. Estimation errors can lead to non-orthogonality between the noise subspace and the steering vectors, resulting in an inaccurate spatial spectrum function and affecting angle estimation results [15]. The improved subspace fitting algorithms [15–20] are derived from

the array model of the receiving array and the mathematical model of the received signal. These algorithms exhibit high computational complexity. Due to severe information loss in low-precision quantized data and the reliance of traditional algorithms on physical array and signal mathematical models, traditional algorithms exhibit significant measurement errors in low-precision scenarios.

In recent times, machine learning has gained widespread application in DOA estimation problems as a novel array processing method [21], including techniques like neural networks (NNs) and Support Vector Regression (SVR). The SVR-based algorithms proposed in [21–26] approximate the input–output relationship by modeling the undefined mapping function of the array output correlation matrix in narrowband environments. Reference [26] analyzes broadband array outputs by decomposing them into narrowband components. However, a major limitation of the aforementioned supervised schemes is their performance degradation when configurations change, such as variations in the number of arrays or sources, potentially rendering the models inoperative.

NNs possess a significant degree of generalization ability, enabling them to handle noise, interference, and signals under non-ideal conditions to a certain extent. They exhibit a certain adaptability to variations in input data and changes in the environment [27–32]. Reference [27] applied Deep Neural Networks (DNNs) for DOA estimation and proposed an algorithm that effectively adapts to array imperfections. Reference [28] applied Convolutional Neural Networks (CNNs) to DOA estimation, utilizing CNN-1 to predict the count of signal sources and CNN-3 for DOA estimation of unknown signal sources. However, when the quantity of input signal sources requires modifications, it is essential to retrain the network. In [29], a deep learning-based spatial spectrum recovery algorithm is proposed. Reference [30] combined 1-D CNN, 2-D CNN networks, and DNN with physics-driven algorithms such as Delay-and-Sum Beamforming (DBF) and MUSIC, effectively enhancing the accuracy of traditional DOA estimation in multipath environments. Reference [31] proposes an algorithm that uses one CNN to partition the angular direction estimation into spatial subregions and another CNN to estimate the DOA. In [32], the deep learning algorithm achieved DOA estimation precision of one degree, performing well under a coherent source.

The NN in the algorithm proposed in [28] has output dimensions that vary with changes in the number of input signals. The algorithms proposed in [27,33–37] are based on multi-class DOA estimation problems. Although the output dimensions of the NNs in these algorithms do not change with variations in the number of input sources, a constant number of sources is utilized in the simulation experiments. To address this issue, this paper presents a classification-based NN whose output dimension remains unchanged regardless of the number of sources. Additionally, the network is capable of handling DOA estimation problems in scenarios where the number of sources is unknown. The input signals to the network consist of low-accuracy signals from a randomly varying number of targets, ranging from 1 to 8. This algorithm constructs a DOA estimation framework using NNs, proposing a hybrid model utilizing both DNN and CNN. The DNN framework is employed to process the quantized signals received from the antenna, which are contaminated with noise. The CNN framework divides the potential arrival angles into subintervals with a 0.1-degree resolution and performs DOA estimation on the processed signals.

The main points in this paper are:

(1) This structure enables accurate and efficient DOA estimation in scenarios with an ambiguous count of signal sources without the need for re-training when the quantity of sources changes.
(2) The algorithm is tested using multiple samples with a range of signal sources from 1 to 8, where the number of sources in different samples is independent.
(3) Simulation experiments display that this structure maintains stable performance across changes in quantization bits, number of antennas, and signal-to-noise ratio (SNR), making it suitable for low-precision quantization scenarios.

2. Mathematical Formulation

2.1. Received Signal Model

The system employs a uniform linear array as the receiving antenna array, consisting of M antennas with a spacing of d_r between each element. There is a total of K far-field narrowband signal sources present, all operating at a central frequency of $\omega_0(\omega_0 = 2\pi f_0)$ with corresponding wavelengths of λ and speeds of c. The angles of incidence of the signals are denoted as $\theta_1, \theta_2, \ldots, \theta_K$. The diagram illustrating the multi-target receiving system model is depicted in Figure 1.

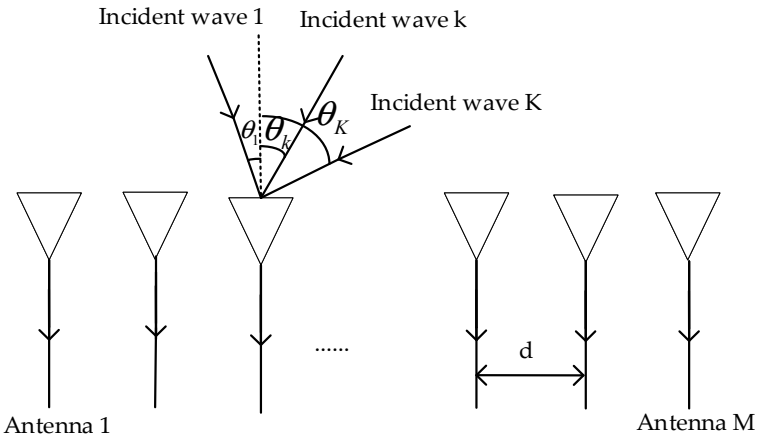

Figure 1. Received model.

The signal captured at the m-th element is denoted as

$$x_m(t) = \sum_{k=1}^{K} s_k(t) e^{-j\omega_0 \tau_m(\theta_k)} + n_m(t). \tag{1}$$

In this context, s_k represents the signal vector associated with the k-th target, while $\tau_m(\theta_k)$ signifies the signal vector received by the m-th element from the k-th target, considering the time delay relative to a chosen reference element (usually the first antenna element). Additionally, $n_m(t)$ represents Gaussian white noise. The time difference of arrival between element m and element 1 is denoted as $\tau_m(\theta_k)$,

$$\tau_m(\theta_k) = \frac{(m-1) d_r \sin\theta_k}{c}. \tag{2}$$

Since $c = \lambda f_0$, (1) can be expressed as follows:

$$x_m(t) = \sum_{k=1}^{K} s_k(t) e^{-j\frac{2\pi}{\lambda}(m-1) d_r \sin\theta_k} + n_m(t). \tag{3}$$

The received signal matrix is represented by the following expression

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\frac{2\pi}{\lambda} d_r \sin\theta_1} & e^{-j\frac{2\pi}{\lambda} d_r \sin\theta_2} & \cdots & e^{-j\frac{2\pi}{\lambda} d_r \sin\theta_K} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j\frac{2\pi}{\lambda}(M-1) d_r \sin\theta_1} & e^{-j\frac{2\pi}{\lambda}(M-1) d_r \sin\theta_2} & \cdots & e^{-j\frac{2\pi}{\lambda}(M-1) d_r \sin\theta_K} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_M(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{bmatrix}. \tag{4}$$

The aforementioned expression can be further transformed into vector form as

$$\mathbf{X}(t) = \mathbf{A}(\theta)\mathbf{S}(t) + \mathbf{N}(t). \tag{5}$$

In this context, $\mathbf{X}(t)$ signifies the $M \times 1$ dimensional received signal vector, $\mathbf{A}(\theta)$ stands for the $M \times K$ array steering matrix, $\mathbf{S}(t)$ denotes the signal vector transmitted by the signal source, and $\mathbf{N}(t)$ denotes the $M \times 1$ dimensional received noise vector. Notably, the steering matrix $\mathbf{A}(\theta)$ encompasses multiple steering vectors

$$\mathbf{A}(\theta) = \begin{bmatrix} a(\theta_1) & a(\theta_2) & \cdots & a(\theta_K) \end{bmatrix}. \tag{6}$$

Within the formula, $a(\theta_k)$ denotes the steering vector of the target impinging from direction θ_k onto the M elements antenna array, and $a(\theta_k)$ is written as

$$a(\theta_k) = \begin{bmatrix} 1 & \exp(-j\frac{2\pi}{\lambda} d_r \sin(\theta_k)) & \ldots & \exp(-j\frac{2\pi}{\lambda}(M-1) d_r \sin(\theta_k)) \end{bmatrix}^T. \tag{7}$$

Upon quantization and sampling by low-precision ADC, the expression for the received signal vector $\mathbf{Y}(t)$ is

$$\mathbf{Y}(t) = Q(\mathbf{X}(t)) = Q(\mathbf{A}(\theta)\mathbf{S}(t) + \mathbf{N}(t)). \tag{8}$$

In this context, the low-precision quantization function is denoted as $Q(\cdot)$ and can be represented as

$$Q(x) = \frac{2\left\lceil 2^b(1-\gamma)x \right\rceil - (1-\gamma)}{2^{b+1}}. \tag{9}$$

Here, γ stands for the reciprocal of quantization noise, b represents the number of quantization bits, and $\lceil \cdot \rceil$ denotes the ceiling operation. The quantization function $Q(\cdot)$ achieves uniform quantization by partitioning the real number line into 2^b quantization intervals, with each interval mapping specific input values to corresponding quantized outputs.

2.2. DNN Structure

The DNN consists of three fundamental network layers: the input layer, hidden layers, and output layer. The first layer serves as the input layer; all intermediary NNs are deemed hidden layers, while the final layer functions as the output layer. Each neuron undergoes both linear weighting operations and nonlinear transformations. Communication between network layers involves the utilization of bias values. Ultimately, the output signal from the higher layer is transmitted to the lower layer as input. Refer to Figure 2 for the structural diagram of the DNN network.

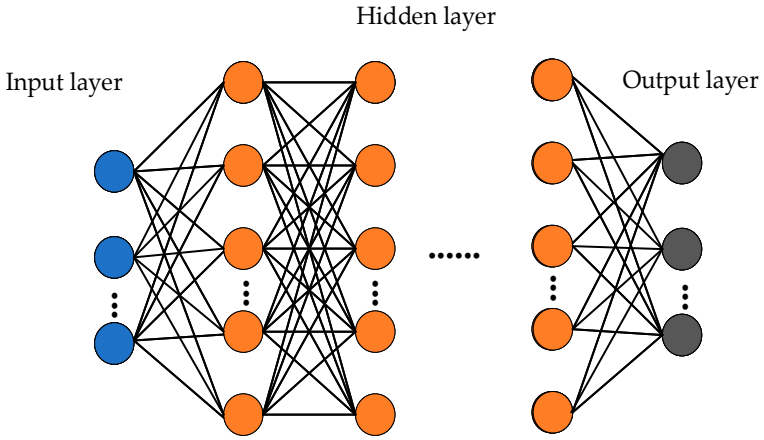

Figure 2. DNN structure.

Assuming weighted operations are performed on the input values of the $(l-1)$-th layer of the DNN, we can obtain the output value $\mathbf{z}^{(l)}$ and activation value $\mathbf{a}^{(l)}$ for the l-th layer as follows:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \tag{10}$$

$$\mathbf{a}^{(l)} = f_l(\mathbf{z}^{(l)}) = f_l(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}). \tag{11}$$

Here, $f_l(\cdot)$ represents the activation function of the l-th layer. If the NN comprises a total of L layers, the output value of the output layer $\hat{y}$ is obtained as $\mathbf{a}^{(L)}$.

Therefore, to ensure that the output obtained from all training data closely approximates the label values, it is essential to obtain the optimal weight coefficient matrix **W** and bias vector **b** within the hidden and output layers. Typically, an appropriate loss function is selected to calculate the output error of the training samples. Subsequently, optimization is performed on this loss function to minimize its extreme value, resulting in a series of weight coefficient matrices **W** and bias vectors **b** as the final outcome. The commonly used loss function is the quadratic loss function. It can be represented as

$$Loss(y, \hat{y}) = \frac{1}{2}\|y - \hat{y}\|^2. \tag{12}$$

Then the error term in the output layer is

$$E = \frac{1}{2}\sum_{i=1}^{n}(T_i - O_i)^2. \tag{13}$$

Here, T_i expresses the target output of the i-th neuron, O_i denotes the output of the NN, and n is the number of neurons in the output layer.

The process of optimizing loss function by seeking extreme values is typically achieved through iterative gradient descent. The gradient of the output layer weights is

$$\frac{\partial E}{\partial w_{ij}} = -(T_i - O_i) \cdot f_i'(net_i) \cdot o_j. \tag{14}$$

Here, w_{ij} represents the weight that connects the j-th hidden layer neuron to the i-th output layer neuron, $f_i'(net_i)$ represents the differential of the activation function, and o_j denotes the output value of the j-th hidden layer neuron.

Then the error term in the hidden layer is

$$\delta_j = f_j'(net_j) \cdot \sum_k w_{kj}\delta_k, \tag{15}$$

δ_j denotes the error term of the j-th neuron in the hidden layer, while δ_k represents the error term of the k-th neuron in the output layer.

The gradient of the weights in the hidden layer is

$$\frac{\partial E}{\partial w_{ij}} = -\delta_i \cdot x_j. \tag{16}$$

Here, x_j represents the output of the j-th input layer neuron.

Through continuous iterations, the NN can eventually achieve a relatively ideal loss value.

2.3. CNN Structure

CNNs are a specialized form of multilayer feedforward NN, similar to other NNs, comprising input layers, output layers, and hidden layers. In CNNs, neurons are arranged in a three-dimensional array, organized by width, height, and depth. This unique arrangement allows CNNs to effectively process three-dimensional input data and transform them into corresponding output variables. The structure of a typical CNN is illustrated in Figure 3.

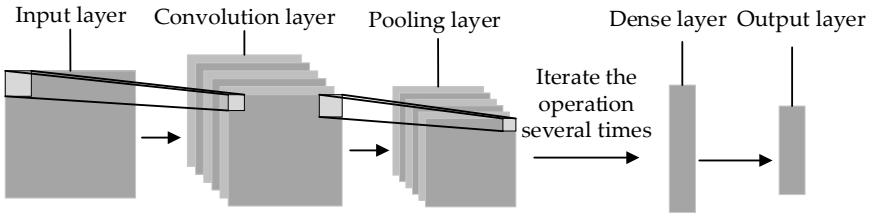

Figure 3. CNN structure.

The algorithm presented in this paper utilizes Conv1D as the convolutional layer. Assuming the input signal is x, $x = [x_1, x_2, \ldots, x_N]$, and the convolution kernel is w, $w = [w_1, w_2, \ldots, w_K]$, in this context, K represents the size of the filter. The calculation formula for the output of the 1D convolution is

$$y_i = \sum_{j=1}^{K} x_{i+j} \cdot w_j + b. \tag{17}$$

Here, i represents the index of the output signal, with a range of $0 \leq i \leq N - K$; b is defined as the bias term.

When the 1D convolutional layer has multiple filters, each filter corresponds to a convolution kernel. Assume there are F filters, each with a convolution kernel length of K. The input signal x has a shape of (N, C), where N is the signal length and C is the number of input channels. The convolution kernel w has a shape of (K, C, F), and the output signal y has a shape of (L, F),

$$y_{i,f} = \sum_{j=1}^{K} \sum_{c=1}^{C} x_{i \cdot s+j,c} \cdot w_{j,c,f} + b. \tag{18}$$

Here, i represents the index of the output signal, s is stride, the filter index is f, and c indicates the input channel index.

3. DOA System Model

Traditional DOA estimation algorithms typically rely on precise mathematical models of signal propagation and receiving arrays, whereas NNs do not require prior knowledge of the physical model. This characteristic makes NNs more flexible to some extent, enabling them to adapt to diverse scenarios and signal types. Trained NNs can exhibit high computational efficiency during the inference stage, especially for complex mathematical models or large-scale data, as NNs can achieve efficient parallel computing. Therefore, they may demonstrate better performance in practical implementations.

To address the problem of quantifying signal multi-target DOA estimation, DNNs and CNNs are applied in the research of quantifying signal DOA estimation algorithms.

3.1. Recovery Network

Initially, DNN is employed to reconstruct the received signal that has undergone low-precision quantization. The DNN comprises five layers, including one input layer, three hidden layers, and one output layer. The input data consist of preprocessed quantized signals $\mathbf{Y}'(t)$. The quantized signals $\mathbf{Y}(t)$ received by the antenna form a complex vector of dimensions $N \times M$, here N represents the number of snapshots, M represents the count of antennas, and can be written as

$$\mathbf{Y}(t) = \begin{bmatrix} y_1 & y_2 & \cdots & y_M \end{bmatrix}. \tag{19}$$

The DNN framework is shown in Table 1. To mitigate the computational complexity of NNs, the quantized signal is decomposed into real-valued vectors to facilitate its application in DNN models, which can be represented as

$$\mathbf{Y}'(t) = \begin{bmatrix} real(y_1) & real(y_2) & \cdots & real(y_M) & imag(y_1) & imag(y_2) & \cdots & imag(y_M) \end{bmatrix}. \quad (20)$$

Table 1. Parameter table of the DNN network.

Neural Layer	Size	Activation Function
Input layer	2M	-
Hidden layer 1	256	LeakyReLU
Hidden layer 2	1024	LeakyReLU
Hidden layer 3	256	LeakyReLU
Output layer	2M	Sigmoid

The real part of an element is represented by the function $real(\cdot)$, and the imaginary part is represented by the function $imag(\cdot)$. Therefore, the count of neurons in the input layer is $2M$. The hidden layers include three fully connected layers. The LeakyReLU is chosen to avoid the vanishing gradient problem, and the output layer's activation function is the Sigmoid. The output layer is used to output the ultimate training result of the DNN model, with the output data being the recovered quantized signal, and the count of neurons is aligned with that of the input layer. The expression for the activation function is as follows:

$$f_{\text{Leaky}\mathrm{ReLU}}(x) = \begin{cases} \alpha x, x \leq 0 \\ x, x > 0 \end{cases}, \quad (21)$$

$$f_{Sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (22)$$

3.2. Classification Network

The CNN network uses binary classification labels for DOA estimation of the recovered signal. A one-dimensional matrix of size 1800 is created as the machine learning label, with matrix values being 0 or 1. The angular range is evenly divided into discrete intervals with a step size of 0.1 degrees. Each position in the matrix corresponds to a specific angle value; if a target exists at the corresponding incident angle, the matrix element is set to 1; otherwise, it is set to 0. Based on this principle, training data samples are generated, followed by NN learning and training.

The CNN framework presented in Table 2 comprises 12 layers, encompassing convolutional, pooling, Flatten, and fully connected layers. The input layer utilizes a convolutional layer. The input is decomposed into a real-valued vector of $(N \times 2M)$, with the imaginary part follows the real part. The input signal for the CNN framework is the received signal after DNN restoration processing. Layers 2 to 9 comprise alternating convolutional and pooling layers aimed at extracting intricate features and alleviating overfitting issues. The 10th layer is the Flatten layer, responsible for converting multidimensional data into one-dimensional form to interface with fully connected layers. Layer 11 is the dense layer with 320 neurons. The final layer, the output layer, employs fully connected layers comprising 1800 neurons that partition the space into 1800 intervals; the activation function is Sigmoid, yielding a probability between 0 and 1, indicating the likelihood of a positive class. The detailed structure of the CNN is provided in the table below. The network is optimized using the Adam optimizer, with binary cross-entropy serving as its loss function, described as

$$Loss(y, \hat{y}) = -\sum_{i=1}^{n} (y \log \hat{y} + (1 - y) \log(1 - \hat{y})). \quad (23)$$

Table 2. Architecture of The Proposed CNN Framework.

Neural Layer	Enter Dimension	Output Dimensions
Input layer	(None, 32, 2M)	(None, 32, 32)
Convolutional layer	(None, 32, 32)	(None, 32, 64)
Pooling layer	(None, 32, 64)	(None, 16, 64)
Convolutional layer	(None, 16, 64)	(None, 16, 128)
Pooling layer	(None, 16, 128)	(None, 8, 128)
Convolutional layer	(None, 8, 128)	(None, 8, 64)
Pooling layer	(None, 8, 64)	(None, 4, 64)
Convolutional layer	(None, 4, 64)	(None, 4, 32)
Pooling layer	(None, 4, 32)	(None, 2, 32)
Flatten layer	(None, 2, 32)	(None, 64)
Dense layer	(None, 64)	(None, 320)
Output layer	(None, 320)	(None, 1800)

3.3. Data Processing

In order to filter out interference in the NN output, an adaptive false alarm threshold is employed to differentiate between real targets and interference. The adaptive threshold formula is

$$T = \mu + k * \sigma, \tag{24}$$

μ represents the mean of sample, σ is the standard deviation. The selection of the coefficient k depends on the desired false alarm rate (FAR). For different implementation contexts, the optimal value of k can be determined experimentally. In this algorithm, a value of 2.5 is selected.

$$\mu = \frac{1}{L}\sum_{i=1}^{L} x_i, \tag{25}$$

$$\sigma = \sqrt{\frac{1}{L-1}\sum_{i=1}^{L} (x_i - \mu)^2}, \tag{26}$$

L expresses the count of samples, while x_i denotes the value of the i-th sample.

The adaptive thresholding technique is employed to dynamically adjust the threshold based on local image statistics, thereby more effectively extracting features.

In order to obtain precise incident angle values, the peaks of NN output values are traversed. As adjacent closer positions may produce multiple peaks, the resolution of the algorithm is set to three degrees. Subsequently, other output results inside the range of plus or minus three degrees from the traversed peaks are set to zero.

4. Simulation Results and Analyses

4.1. Simulation Setup

This section conducts simulation validation of the proposed NN-based quantized signal DOA estimation algorithm, examining the DOA estimation accuracy in relation to quantization bits, SNR, and the quantity of antenna arrays. The NN is implemented using the Tensorflow framework, and simulations are performed in a Python environment.

Figure 4 illustrates the block diagram of the entire processing flow, which includes both the data processing and angle estimation components.

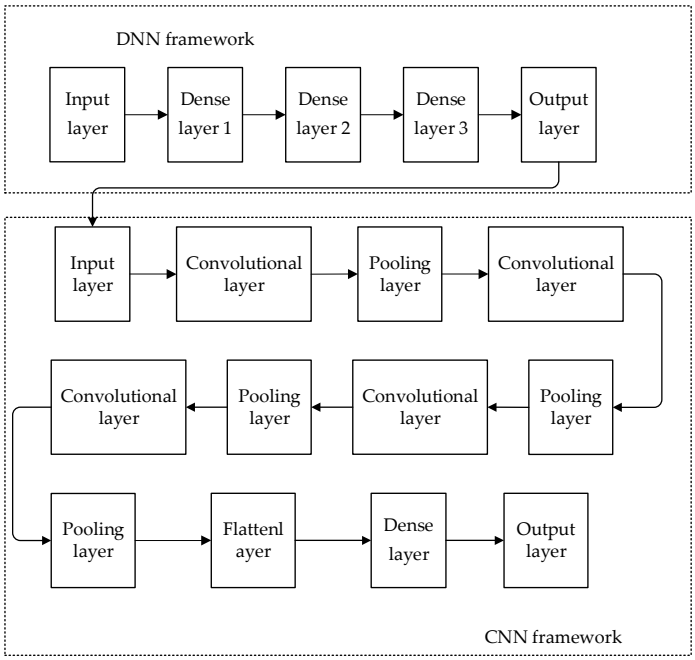

Figure 4. Overall framework.

4.1.1. DNN Framework

To study the impact of the DNN architecture on noisy quantized signals, the network was trained using ideal, noise-free received signals as labels for supervised learning. The selection of hyperparameters will have an impact on the performance of the NN. The performance of hyperparameters is evaluated using validation sets [38]. To select an appropriate learning rate, Figure 5 compares the validation losses of models with different learning rates. Configure the learning rates of the network to 0.03, 0.01, and 0.001, respectively. It was observed that with a learning rate of 0.01, the validation loss was lower than at 0.001, and the curve's decline was more stable compared to 0.03. Hence, a learning rate of 0.01 was chosen for the network.

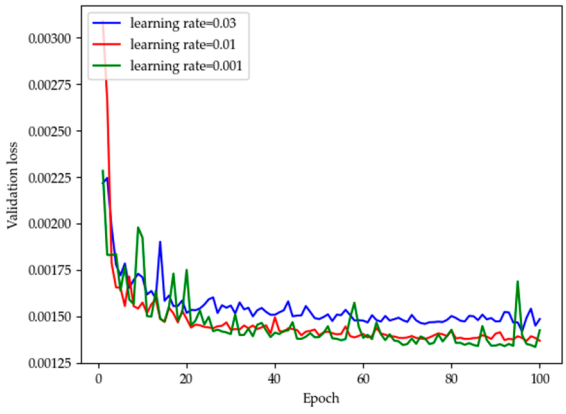

Figure 5. The comparison of validation loss for different learning rates in the DNN network.

The loss function utilized by the NN is the Mean Square Error (MSE) function. The MSE function has a fast convergence speed and low complexity, and its expression is

$$Loss(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2. \tag{27}$$

The Adaptive Moment Estimation (Adam) optimizer is chosen for NN optimization. The Adam optimizer adjusts the learning rate for each parameter by utilizing the gradients derived from the first-order and second-order moment estimates. This approach enables more efficient updates of network weight parameters.

The batch size of the DNN is assigned to 32, the learning rate is 0.01, and the total of training epochs is 100, and both the number of training and testing samples are set to 100,000. The effectiveness of the DNN framework is evaluated using the root mean square error (RMSE) function between the output signal and the unquantized noiseless signal, as given by the following formula

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - f(x_i))^2}. \tag{28}$$

Here, N denotes the quantity of experiments, $f(x_i)$ expresses the output value of the DNN framework, and Y_i denotes the ideal received signal. For this study, N is set to 100,000. Figure 6 illustrates the RMSE between the recovered quantized signals at different SNRs and the ideal signal.

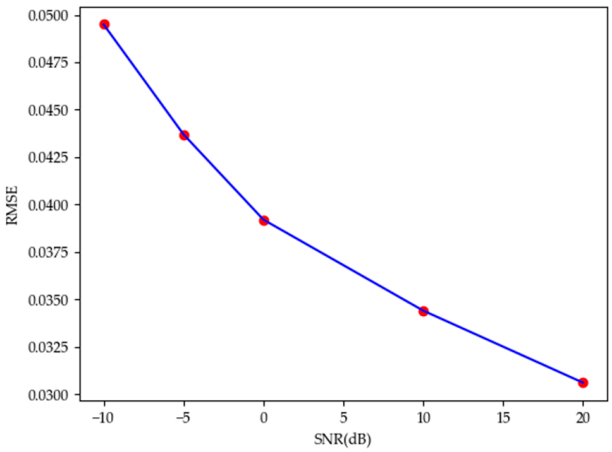

Figure 6. The comparison of the RMSE with different SNRs.

The MUSIC algorithm was applied to perform DOA estimation on various signals, including 3-bit quantized noisy signals processed by the DNN, 3-bit quantized signals with SNR = 20 dB, noise-free 3-bit quantized signals, noisy unquantized signals, and noise-free unquantized signals. By comparing the DOA estimation results of these different signals, the impact of quantization and noise on the received signals and the effectiveness of the NN in signal recovery were assessed. The number of targets was fixed at 1, with a snapshot count of 1, four antennas, and a sample size of 100,000. As shown in Figure 7, both quantization and noise impact the accuracy of DOA estimation. However, processing the received signals with a DNN network significantly enhances the angle estimation performance of signals subjected to low-precision quantization

The performance of DOA estimation is represented by the cumulative distribution function (CDF), as follows:

$$F(x) = P(X \leq x), \ for - \infty < x < +\infty. \tag{29}$$

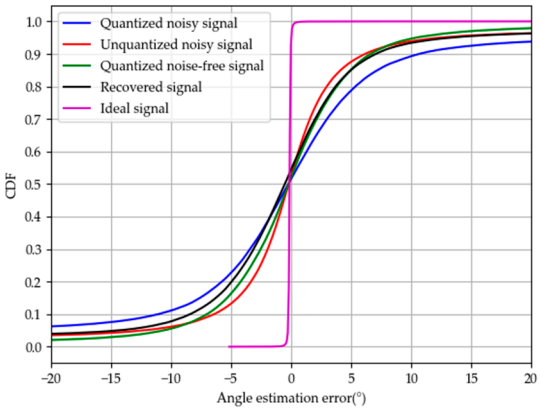

Figure 7. DOA estimation results under different signals.

4.1.2. CNN Framework

Figure 8 shows the validation loss curves of the CNN network at different learning rates. It can be observed that the model achieves the lowest loss and the best performance when the learning rate is configured to 0.01.

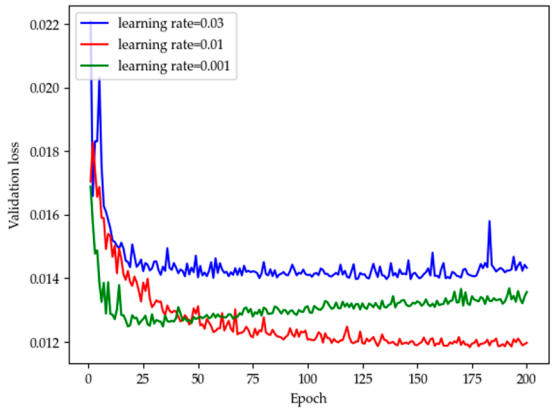

Figure 8. Comparison of validation loss for different learning rates in the CNN network.

The batch size of the CNN network is 32, the learning rate is 0.01, and the total training epochs is 200, and both the number of training and testing samples is 100,000.

4.2. Simulation Result

In multi-target scenarios, the number of targets ranges from 1 to 8 randomly. Due to higher testing errors at edge angles, the target angles are randomly chosen within a range of 20 to 160 degrees. The count of snapshots is set to 32, SNR is 20 dB, the number of receiving antenna arrays to 32, and the quantization bits are set to 3.

Figure 9 shows the network output results using the introduced algorithm. Both the number of targets and the incident angles were randomly chosen. In this example, there are 4 targets, with DOA estimation results and true results shown in Table 3.

Table 3. Actual angle and estimated angle.

True Angles (°)	Estimated Angle (°)
32.1	32.3
90.4	90.6
94.4	94.2
133.2	133.1

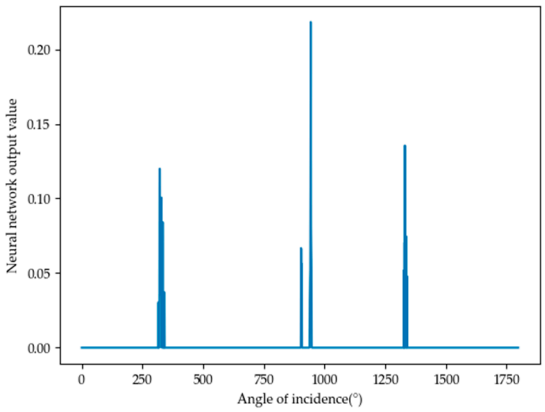

Figure 9. CNN network output value.

In scenarios with a large number of targets, the proposed algorithm may exhibit instances of missed detections. One primary reason is that closely spaced peaks can be obscured during peak traversal. Additionally, the decision threshold significantly impacts the estimation results. A high decision threshold increases the probability of missed detections, while a low decision threshold raises the false alarm rate. Statistical analysis reveals that the proposed algorithm has a missed detection probability of approximately 4.01%. Missed detections mainly occur in samples with more than four targets or where multiple targets have similar angles of arrival. However, in practical DOA estimation scenarios, the quantity of targets is typically not large. Therefore, the proposed algorithm can fulfill the system's requirements for high accuracy and low cost.

Simulations were conducted on 100,000 samples with zero targets, to analyze the false alarm probability of the proposed algorithm under the constant false alarm rate (CFAR) method. If the DOA estimation result of the proposed algorithm indicates several targets greater than zero, it is considered a false alarm. Statistical analysis reveals that the false alarm probability of the proposed algorithm is 0.72%.

To investigate the outcome of the proposed DNN-CNN framework under lower precision and to study the impact of quantization bits on the DOA estimation accuracy, DOA estimation was conducted in scenarios with quantization bits of 3, 2, and 1. Figure 10 presents a performance comparison of the introduced algorithm under different quantization bit scenarios. It can be observed that the impact of quantization bits on the outcome of the introduced algorithm is minimal. As the quantity of quantization bits decreases, the DOA estimation accuracy of the framework declines only slightly, allowing effective DOA estimation even in 1-bit quantization scenarios.

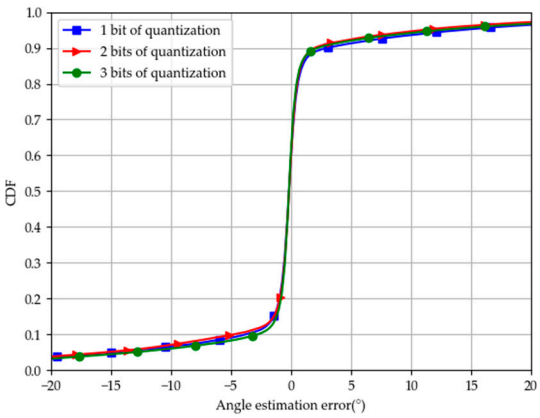

Figure 10. DOA estimation error under different quantization numbers.

To investigate the impact of SNR on the estimation accuracy of the introduced algorithm, DOA estimation was performed in scenarios with SNR values of 20 dB, 10 dB, and

0 dB. Figure 11 presents the outcome comparison of the introduced algorithm under different SNR. The results indicate that as the SNR decreases, the DOA estimation accuracy of the proposed algorithm is minimally affected, demonstrating the excellent noise resistance of the proposed algorithm.

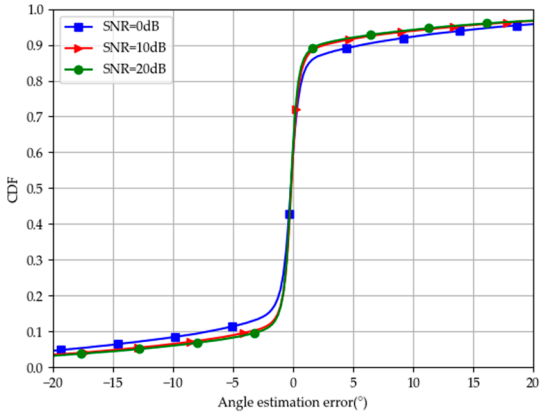

Figure 11. DOA estimation error under different SNRs.

The number of array elements is a crucial parameter influencing the accuracy of DOA estimation. The number of antenna arrays is proportional to the input dimensions of the deep learning network framework. To further study the impact of the quantity of antenna arrays on the DOA estimation accuracy of the proposed algorithm, simulations were conducted with 8, 16, and 32 antenna elements. The simulation results are shown in Figure 12. It is evident that as the quantity of antenna elements rises, the accuracy of DOA estimation improves, primarily due to the increase in useful information in the received signal data. The number of antenna elements has a relatively limited impact on the DOA estimation algorithm proposed in this paper; for instance, with $M = 16$ and $M = 32$, the angle estimation errors are concentrated within a range of five degrees.

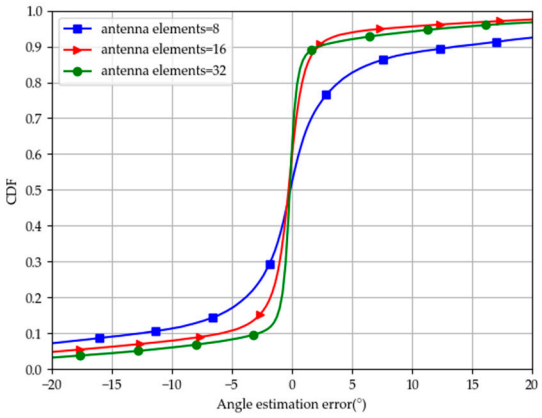

Figure 12. DOA estimation error under different numbers of receive antennas.

Figure 13 presents the DOA estimation results for received signals with an SNR of 20 and a quantization bit width of 3 bits using various algorithms. Due to the inability to separate the noise subspace and signal subspace when the quantity of sources is unknown, traditional MUSIC algorithms struggle to achieve precise DOA estimation in scenarios with unknown target counts. By employing the improved MUSIC-based Signal Subspace Suppress Algorithm (SSSA) (with k = 5 as recommended in [39]) and the improved MUSIC algorithm from [15], the deep learning method proposed by [32], and the DNN-CNN framework in this paper, angle estimation was performed for the precision scenario. Experimental results reveal that in scenarios with a large number of targets, both the SSSA algorithm and the algorithm proposed by [32] perform poorly. Consequently, the number of

sources was randomly set between 1 and 3, with a sample size of 100,000. This demonstrates that the DOA estimation outcome of both the SSSA algorithm and the algorithm proposed by [32] is inferior to that of the algorithm presented in this paper, while the DOA estimation accuracy of the improved MUSIC algorithm in [15] surpasses that of proposed algorithm. In scenarios with fewer sources, the NN framework proposed in this paper achieves higher DOA estimation accuracy compared to the 1–8 target scenario, with errors concentrated within a two-degree range.

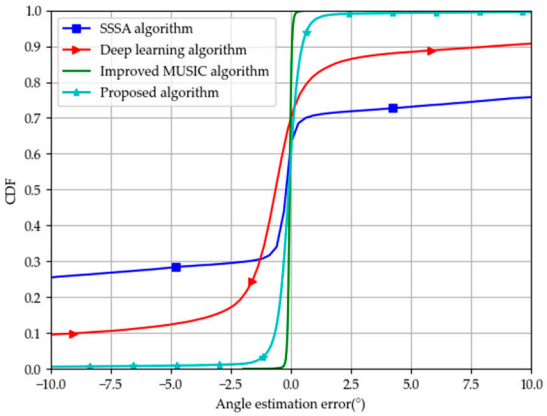

Figure 13. DOA estimation error under different algorithms.

4.3. Comparison of Runtimes of Different Methods

The amount of information received varies with the quantity of antenna elements, and thus the computation time of the algorithms differs. Table 4 presents the computation times for estimating DOA using four different algorithms, each with varying numbers of antennas, across 100,000 datasets and 32 snapshots. The central processing unit (CPU) of the computer used in the laboratory is an Intel Core i7-10700F. The data demonstrate that the computation time of the proposed DOA estimation method is only 0.01 times that of the SSSA algorithm and 10^{-4} times that of the algorithm proposed by [15], significantly enhancing computational efficiency and enabling real-time DOA estimation in practical applications.

Table 4. The computation time of different algorithms under varying numbers of antennas.

Algorithm	Antenna Numbers	Computation Time per 100,000 Times (s)
DNN-CNN framework	16	11.267
	32	25.304
Deep learning algorithm in [32]	16	3.791
	32	4.976
SSSA algorithm in [39]	16	2306.997
	32	2677.327
Improved MUSIC algorithm in [15]	16	80,206.856
	32	90,758.921

From the aforementioned simulations, it can be observed that in terms of estimation performance, the DNN-CNN framework for DOA estimation outperforms the SSSA and [32] algorithms, and is slightly inferior to the algorithm in [15]. In terms of time complexity, the proposed DOA estimation method significantly outperforms the SSSA and [15] algorithms, while being slightly less efficient than the [32] algorithm. The number of model parameters (space complexity) of the proposed algorithm is 1,214,472 parameters. The number of parameters in the NN proposed in [32] is 362,849. It can be concluded that the space complexity of the algorithm proposed in [32] is approximately 0.3 times that of the space complexity of the algorithm presented in this paper.

5. Discussion

The improved MUSIC algorithm proposed in [15] demonstrates higher accuracy than the algorithm presented in this paper when the number of targets ranges from one to three; concurrently, our proposed algorithm outperforms the one in [32]. When the number of targets is unknown, it is not possible to directly distinguish between the signal subspace and the noise subspace. Therefore, Reference [15] constructs all possible covariance feature matrices and sequentially performs spectral peak searching on each matrix. This approach, while yielding high accuracy, is time-consuming and not suitable for real-time estimation. Both the algorithm presented in [32] and the one proposed in this paper utilize NNs based on classification tasks; however, the network structure in [32] is simpler compared to ours, resulting in shorter execution times. However, under low-precision scenarios (such as the low-precision block sampling scenarios assumed in this paper), the DOA estimation accuracy of Reference [32]'s algorithm is inferior to that of our proposed method. Therefore, after weighing the trade-offs between DOA estimation time and accuracy, our proposed algorithm better addresses the real-time estimation requirements in low-precision scenarios.

By comparing the CDF curves of the proposed algorithm in Figures 12 and 13, it can be observed that under identical received signal quality conditions (with 32 antennas, 32 snapshots, 3-bit quantization, and SNR of 20 dB), the DOA estimation accuracy for a random target count ranging from 1 to 3 is higher than that for a range from 1 to 8. This is due to the fact that with a larger quantity of targets, the incident angles tend to become more concentrated, thereby increasing the difficulty of DOA estimation. Furthermore, when multiple signals coexist, the effective signal strength of each individual signal may diminish, leading to a decrease in the overall SNR, which further affects the accuracy of DOA estimation. In practical DOA estimation problems, the quantity of targets is typically not large; therefore, the proposed algorithm can satisfy the system's requirements for high accuracy and low cost.

6. Conclusions

The current paper introduces a DOA estimation algorithm based on a classification problem by constructing a hybrid framework of DNN and CNN to estimate the DOA of received signals with an unknown quantity of signal sources. The DOA estimation algorithm based on classification leverages deep learning techniques to learn and identify signal sources corresponding to different DOAs from the received signals. By uniformly dividing the angle range of the incoming waves into discrete angles, a supervised learning label with dimensions of 1×1800 was generated. The algorithm achieves a precision of 0.1° and a resolution of 3°. This algorithm enables precise estimation and classification of the angles of arrival of signal sources. Simulation experiments were conducted in scenarios with varying SNR, antenna numbers, and quantization bits. The data indicate that the proposed algorithm has significant noise resistance and performs well in low-precision quantization scenarios. Additionally, in multi-target scenarios, changes in the quantity of signal sources do not affect the network's DOA estimation performance, significantly reducing the number of training sessions required and improving training efficiency in practical applications.

The algorithm proposed in this paper is currently still in the theoretical stage, and learning how to implement it through hardware is the most important goal for future research. At the same time, the scenarios assumed in this paper only consider the quantity of antennas, signal-to-noise ratio, and quantization bits, which are more ideal compared to real-world application scenarios. In the hardware implementation of the algorithm, we are considering using HLS tools to convert the neural network into hardware description language. During the process of hardware implementation, we will consider more interferences from real-world environments, such as DOA estimation under multipath effects and the interference of Doppler frequency shift on DOA estimation.

Author Contributions: Writing—original draft preparation, W.C., W.R. and Z.Z.; writing—review and editing, W.C. and Z.Z.; conceptualization, W.H.; supervision, J.Z.; Formal analysis, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the Editor and the anonymous Referees for their instructive comments.

Conflicts of Interest: Author Wen Ren was employed by the company Space Star Technology Co., Ltd., China Academy of Space Technology; Author Weiqiang Huang was employed by the company Nanjing Panda Handa Technology Co., Ltd., China Electronics Technology Group Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Jinde, H.; Hong, L. Estimation of Direction of Arrival(DOA) Based on the Spatial Kurtosis Spectrum. In Proceedings of the 2021 International Conference on Information Science, Parallel and Distributed Systems (ISPDS), Hangzhou, China, 13–15 August 2021; pp. 47–51. [CrossRef]
2. Rajani, A.; Kora, P. Direction of Arrival Estimation by Using Artificial Neural Networks. In Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2020), Tirunelveli, India, 4–6 February 2021; pp. 1360–1363. [CrossRef]
3. Fuchs, J.; Gardill, M.; Lübke, M.; Dubey, A.; Lurz, F. A Machine Learning Perspective on Automotive Radar Direction of Arrival Estimation. *IEEE Access* **2022**, *10*, 6775–6797. [CrossRef]
4. Ge, S.; Li, K.; Rum, S.N.B.M. Deep learning approach in DOA estimation: A systematic literature review. *Mob. Inf. Syst.* **2021**, *2021*, 6392875. [CrossRef]
5. Zhao, F.; Hu, G.; Zhan, C.; Zhang, Y. DOA estimation method based on improved deep convolutional neural network. *Sensors* **2022**, *22*, 1305. [CrossRef]
6. Cox, H.; Zeskind, R.; Owen, M. Robust adaptive beamforming. *IEEE Trans. Acoust. Speech Signal Process.* **1987**, *35*, 1365–1376. [CrossRef]
7. Elbir, A.M.; Mishra, K.V.; Vorobyov, S.A.; Heath, R.W. Twenty-Five Years of Advances in Beamforming: From convex and nonconvex optimization to learning techniques. *IEEE Signal Process. Mag.* **2023**, *40*, 118–131. [CrossRef]
8. Huang, H.; Peng, Y.; Yang, J.; Xia, W.; Gui, G. Fast beamforming design via deep learning. *IEEE Trans. Veh. Technol.* **2019**, *69*, 1065–1069. [CrossRef]
9. Mucci, R. A comparison of efficient beamforming algorithms. *IEEE Trans. Acoust. Speech Signal Process.* **1984**, *32*, 548–558. [CrossRef]
10. Zhou, C.W.; Gu, Y.J.; Shi, Z.G.; Haardt, M. Structured Nyquist Correlation Reconstruction for DOA Estimation with Sparse Arrays. *IEEE Trans. Signal Process.* **2023**, *71*, 1849–1862. [CrossRef]
11. Benesty, J.; Chen, J.; Huang, Y. Conventional beamforming techniques. In *Microphone Array Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 39–65.
12. Peng, J.Y.; Nie, W.H.; Li, T.; Xu, J. An end-to-end DOA estimation method based on deep learning for underwater acoustic array. In Proceedings of the OCEANS Hampton Roads Conference, Hampton Roads, VA, USA, 17–20 October 2022.
13. Schmidt, R.O. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280. [CrossRef]
14. Roy, R.; Kailath, T. ESPRITestimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 984–995. [CrossRef]
15. Chen, F.; Yang, D.S.; Wu, D.; Gui, C.Y.; Mo, S.Q. A Novel Direction-of-Arrival Estimation Algorithm without Knowing the Source Number. *IEEE Commun. Lett.* **2021**, *25*, 985–989. [CrossRef]
16. Debasis, K. Modified MUSIC algorithm for estimating DOA of signals. *Signal Process.* **1996**, *48*, 85–90. [CrossRef]
17. Pillai, S.U.; Kwon, B.H. Forward/backward spatial smoothing techniques for coherent signal identification. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 8–15. [CrossRef]
18. Viberg, M.; Ottersten, B.; Kailath, T. Detection and estimation in sensor arrays using weighted subspace fitting. *IEEE Trans. Signal Process.* **1991**, *39*, 2436–2449. [CrossRef]
19. Wax, M.; Kailath, T. Detection of signals by information theoretic criteria. *IEEE Trans. Acoust. Speech Signal Process.* **1985**, *33*, 387–392. [CrossRef]
20. Zhang, Y.; Ng, B.P. MUSIC-Like DOA Estimation Without Estimating the Number of Sources. *IEEE Trans. Signal Process.* **2010**, *58*, 1668–1676. [CrossRef]
21. Pastorino, M.; Randazzo, A. A smart antenna system for direction of arrival estimation based on a support vector regression. *IEEE Trans. Antennas Propag.* **2005**, *53*, 2161–2168. [CrossRef]

22. Hasan, M.I.; Saquib, M. Low Complexity Single Source 2-D DOA Estimation Based on Reduced Dimension SVR. In Proceedings of the 2022 IEEE 22nd Annual Wireless and Microwave Technology Conference (WAMICON), Clearwater, FL, USA, 27–28 April 2022; pp. 1–4. [CrossRef]
23. Liu, S.B.; Wang, D.Q.; Xing, R.Z.; Ren, J.L.; Lu, W.K. Research on error correction model of surface acoustic wave yarn tension transducer based on DOA-SVR model. *Measurement* **2024**, *226*, 114126. [CrossRef]
24. Randazzo, A.; Abou-Khousa, M.A.; Pastorino, M.; Zoughi, R. Direction of arrival estimation based on support vector regression: Experimental validation and comparison with MUSIC. *IEEE Antennas Wirel. Propag. Lett.* **2007**, *6*, 379–382. [CrossRef]
25. Wajid, M.; Alam, F.; Yadav, S.; Khan, M.A.; Usman, M. Support Vector Regression based Direction of Arrival Estimation of an Acoustic Source. In Proceedings of the 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Sakheer, Bahrain, 20–21 December 2020; pp. 1–6. [CrossRef]
26. Wu, L.L.; Huang, Z.T. Coherent SVR Learning for Wideband Direction-of-Arrival Estimation. *IEEE Signal Process. Lett.* **2019**, *26*, 642–646. [CrossRef]
27. Liu, Z.M.; Zhang, C.W.; Yu, P.S. Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections. *IEEE Trans. Antennas Propag.* **2018**, *66*, 7315–7327. [CrossRef]
28. Yu, J.R.; Wang, Y.F. Deep Learning-Based Multipath DoAs Estimation Method for mmWave Massive MIMO Systems in Low SNR. *IEEE Trans. Veh. Technol.* **2023**, *72*, 7480–7490. [CrossRef]
29. Wu, L.L.; Liu, Z.M.; Huang, Z.T. Deep Convolution Network for Direction of Arrival Estimation with Sparse Prior. *IEEE Signal Process. Lett.* **2019**, *26*, 1688–1692. [CrossRef]
30. Xiang, H.H.; Chen, B.X.; Yang, T.; Liu, D. Improved De-Multipath Neural Network Models with Self-Paced Feature-to-Feature Learning for DOA Estimation in Multipath Environment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5068–5078. [CrossRef]
31. Zhao, Z.; Li, J. DoA Estimation based on Deep Learning in Low SNR. In Proceedings of the 2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, China, 14–16 June 2023; pp. 1–4. [CrossRef]
32. Ge, X.; Hu, X.; Dai, X. DOA Estimation for Coherent Sources Using Deep Learning Method. *J. Signal Process.* **2019**, *35*, 1376–1384.
33. Barthelme, A.; Utschick, W. A Machine Learning Approach to DoA Estimation and Model Order Selection for Antenna Arrays With Subarray Sampling. *IEEE Trans. Signal Process.* **2021**, *69*, 3075–3087. [CrossRef]
34. Kassir, H.A.; Rekanos, I.T.; Lazaridis, P.I.; Yioultsis, T.V.; Kantartzis, N.V.; Antonopoulos, C.S.; Karagiannidis, G.K.; Zaharis, Z.D. DOA Estimation for 6G Communication Systems. In Proceedings of the 2023 12th International Conference on Modern Circuits and Systems Technologies (MOCAST), Athens, Greece, 28–30 June 2023; pp. 1–4. [CrossRef]
35. Kassir, H.A.; Kantartzis, N.V.; Lazaridis, P.I.; Sarigiannidis, P.; Goudos, S.K.; Christodoulou, C.G.; Zaharis, Z.D. Improving DOA Estimation via an Optimal Deep Residual Neural Network Classifier on Uniform Linear Arrays. *IEEE Open J. Antennas Propag.* **2024**, *5*, 460–473. [CrossRef]
36. Ch, L.D.T.; Nagaraju, L.; Puli, K.K. Classification based DOA estimation using ANN and CNN Models. In Proceedings of the IEEE Microwaves, Antennas, and Propagation Conference (MAPCON), Bangalore, India, 12–16 December 2022; pp. 1470–1473.
37. Hu, J.; Mo, Q.; Liu, Z.; Li, H. Multi-Source Classification: A DOA-Based Deep Learning Approach. In Proceedings of the 2020 International Conference on Computer Engineering and Application (ICCEA), Guangzhou, China, 18–20 March 2020; pp. 463–467. [CrossRef]
38. Weidman, S. *Deep Learning from Scratch: Building with Python from First Principles*; O'Reilly Media: Sebastopol, CA, USA, 2019.
39. LI, X.; Zhang, L.; Sheng, X.; Wei, X. DOA Estimation without Knowing the Number of Sources. *Commun. Technol.* **2023**, *56*, 135139.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.