IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 16, 2020, accepted January 12, 2021, date of publication January 18, 2021, date of current version February 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051906

Deep Learning-Based Bootstrap Detection Scheme for Digital Broadcasting System

TAE-HOON KANG[1,2], WON-SEOK LEE[1,2], MYUNG-SUN BAEK[3], (Member, IEEE), BYUNGJUN BAE[3], AND HYOUNG-KYU SONG[1,2]

[1]Department of Information and Communication Engineering, Sejong University, Seoul 05006, South Korea
[2]Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea
[3]Electronics and Telecommunication Research Institute (ETRI), Daejeon 34129, South Korea

Corresponding author: Hyoung-Kyu Song (songhk@sejong.ac.kr)

This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) funded by the Ministry of Science and ICT (MSIT), Korea Government, through the terrestrial UHD-based disaster broadcasting service for reducing disaster damage under Grant 2018-0-01364, and in part by the Institute for Information & Communications Technology Promotion (IITP) funded by MSIT, Korea Government, through the Development of immersive signage based on variable transparency and multiple layers under Grant 2017-0-00217.

ABSTRACT In the advanced television systems committee (ATSC) 3.0 system, the concept of flexibility is significant for supporting backward compatibility within the same ATSC 3.0 system. However, since the conventional bootstrap signal detection scheme is difficult to support the flexibility, the conventional bootstrap signal detection scheme should be newly designed according to the change of version. In this paper, a convolution neural network (CNN) model for bootstrap signal detection in ATSC 3.0 is proposed to maintain the flexibility of bootstrap. Additionally, for minimizing the loss of error performance of CNN-based bootstrap detection scheme, this paper proposes two dimensional alternate array to utilize the correlation of the adjacent bootstrap symbol and proposes the offline learning method using the bootstrap signal corrupted by noise to improve the error performance.

INDEX TERMS ATSC 3.0, bootstrap, deep learning, convolutional neural network, signal detection, broadcasting.

I. INTRODUCTION

The advanced television systems committee (ATSC) developed the standard known as ATSC 3.0 to support ultra-high definition(UHD) services. ATSC 3.0 is designed to provide flexibility and extensibility for backward compatibility within the same ATSC 3.0 systems [1]–[4]. The 'system discovery and signaling' (A/321) describes the system discovery and signaling architecture for the ATSC 3.0 physical layer. As broadcasters realize a need providing multiple wireless-based services in addition broadcast, ATSC 3.0 defines bootstrap identifying the type or form of transmitted signal [2].

The bootstrap signal provides a universal entry point into a digital transmission signal. The bootstrap is designed to use the concepts of versioning, scalability and extensibility to support backward compatibility which is called flexibility. The bootstrap version is expressed as a major and a minor version which are generated by Zadoff-Chu (ZC) and pseudo-noise (PN) sequence, respectively. Scalability means that the bootstrap can sustain backward compatibility when the number of bits signaled per bootstrap symbol is increased, up to a specified maximum. The bootstrap extensibility is obtained by additional bootstrap symbol and termination of bootstrap signal is signaled by a final symbol having 180° phase inversion relative to the preceding symbol.

The bootstrap signal is positioned in front of the frame. The bootstrap signal consists of a number of symbols, and the first symbol is enable to signal discovery, coarse synchronization, frequency offset estimation, and initial channel estimation [5]. The remainder of the bootstrap contains sufficient control signaling such as the minimum time interval to next frame and system bandwidth. For synchronization, channel estimation and signal detection, even if the version number evolves, ATSC 3.0 bootstrap configurations such as sampling rate, bandwidth, FFT size, and symbol length are unchanged [6]. Also, in time domain, the bootstrap sequence of the bootstrap symbols is generated by ZC sequence and PN sequence which are changed by the number of the bootstrap

The associate editor coordinating the review of this manuscript and approving it for publication was Jerry Chun-Wei Lin.

symbols and subcarriers. For the same version of bootstrap, since the bootstrap signal uses the fixed configuration, each symbol has different fixed sequence. The receiver can demodulate the received bootstrap signal by using the fixed sequence [7].

To demodulate the bootstrap signal successfully, many researchers have proposed various detection schemes [8], [9], [12], [13]. In [9], while the extensibility of the bootstrap is being maintained, a newly designed bootstrap sequence was proposed. The bootstrap of this scheme is signaled in frequency domain compared to the bootstrap of standardized scheme in time domain. In this way, the performance of the proposed scheme is improved compared to the performance of the standardized scheme. In many communication systems, the maximum likelihood (ML) detection method has been researched [10]–[12]. the ML detection method is known as the optimal scheme, and it also be researched in ATSC 3.0 system. In [12], an iterative detection scheme using maximum likelihood scheme was proposed. Also, this paper improves the reliability of the channel estimation by averaging the channel gains iteratively. Additionally, in [13], in order to improve detection performance for specific bits, characteristic of the fixed system parameters, such as the system bandwidth, sample rate, and etc was used. However, these conventional bootstrap detection schemes have no flexibility. When the standard is changed, conventional schemes is no longer practical. Thus, detection module should be newly designed and replaced with new one.

Recently, deep learning technology has received attention in communication engineering area [14]–[21]. In this aspect, this paper deals with convolutional neural network (CNN) among deep learning technologies for detection of bootstrap signal. However, since a detection scheme using simple CNN-based models is not enough to detect bootstrap symbols, this paper proposes two preprocess methods to improve reliability for detection of bootstrap signals. The conventional CNN model uses 2 dimensional array input data. Since the performance of CNN model defends on the method of array, it is significant to find proper array. First, a received bootstrap symbol is compensated by a bootstrap sequence, and then two adjacent compensated bootstrap symbols are arranged into two-dimensional array which is arranged alternately in each subcarrier. With application of the preprocess, training loss value is decreased and performance is improved. Additionally, to further improve the performance, this paper uses for training the alternately arranged input data corrupted by noise. By training the input data corrupted by noise, the threshold boundary of bootstrap symbol is trained and the actually received bootstrap symbol is detected more precisely. Consequently, in this paper, to help utilization of the CNN for ATSC 3.0 system and maintain the flexibility of evolving the major and minor version in the future, the structure of CNN and preprocessing method are presented.

In this paper, to verify the performance of proposed CNN-based detection model, the maximum likelihood (ML) detection method and the DNN-based detection model are introduced and compared with proposed CNN-based model.

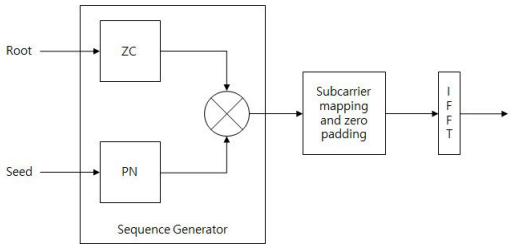


FIGURE 1. Frequency domain for bootstrap generation.

II. STRUCTURE OF BOOTSTRAP SIGNALS

Fig. 1 shows that the bootstrap sequence is generated by multiplying Zadoff-Chu (ZC) sequence and pseudo-noise (PN) sequence, in the frequency domain. The ZC sequence and PN sequence contain the information of the major and minor versions of the bootstrap, respectively. The ZC sequence is as follows,

$$z_q(k) = e^{-j\pi q \frac{k(k+1)}{N_{zc}}}, \quad k = 0, 1, 2, \ldots, N_{zc} - 1, \tag{1}$$

where $N_{ZC} = 1499$ is the largest length due to channel bandwidth 4.5MHz with a subcarrier spacing 3kHz and q is a root which is the major version number of the bootstrap.

The PN sequence $p(k)$ is generated by linear feedback shift register (LFSR) of length $l = 16$ where the generator polynomial is $p(x) = x^{16} + x^{15} + x^{14} + x + 1$ and the PN sequence is represented by 0 or 1. The initial state of generator polynomial r_{init} is called seed which means minor version of the bootstrap. The output of PN sequence $p(k)$ is converted as follows,

$$c(k) = 1 - 2 \times p(k), \tag{2}$$

where $c(k)$ is represented by -1 or 1 instead of 1 or 0. And then, $z_q(k)$ and $c(k)$ are mapped as follows,

$$S_n(k) = \begin{cases} z_q(k + N_H) \times c((n+1) \times N_H + k) & -N_H \le k \le -1 \\ z_q(k + N_H) \times c((n+1) \times N_H - k) & 1 \le k \le N_H, \\ 0 & otherwise \end{cases} \tag{3}$$

where $S_n(k)$ is the frequency domain bootstrap sequence of the n-th symbol and k-th subcarrier and $N_H = (N_{ZC} - 1)/2$. The power of the bootstrap sequence $S_n(k)$ is normalized to 1. To indicate the final symbol, phase of final symbol is inverted as follows,

$$\tilde{S}_n(k) = \begin{cases} S_n(k) & 0 \le n < N_S - 1 \\ -S_n(k) & n = N_S - 1, \end{cases} \tag{4}$$

where N_s is the number of bootstrap symbols. The frequency domain bootstrap sequence $\tilde{S}_n(k)$ is transformed to the time

domain sequence $\tilde{A}_n(t)$ through IFFT of size $N_{FFT} = 2048$.

$$\tilde{A}_n(t) = \frac{1}{\sqrt{N_{ZC}-1}}\left(\sum_{k=-(N_{ZC}-1)/2}^{-1} \tilde{s}_n(k) e^{j2\pi k f_\Delta t} + \sum_{k=1}^{(N_{ZC}-1)/2} \tilde{s}_n(k) e^{j2\pi k f_\Delta t} \right). \quad (5)$$

where $\tilde{A}_n(t)$ is the time domain bootstrap sequence of the n-th symbol at the time t. f_Δ has subcarrier space of 3kHz.

In current version, the bootstrap has four symbols for ATSC 3.0. To transmit the signal, the information is signaled via the bootstrap symbol using a cyclic shift in the time domain [2]. Since the bootstrap sequence has a length of N_{FFT}, 2048 distinct cyclic shifts are possible and it can be signaled up to $\log_2(2048) = 11$ bits. However, all of these bits actually are not used. When the number of used signaling bits for the n-th bootstrap symbol is N_b^n, the bit representation of the signaling information is given by $b_0^n, b_1^n, \ldots, b_{N_b^n-1}^n$ which have the value 0 or 1. The rest bits $b_{N_b^n}^n, \ldots, b_{10}^n$ are set to 0. And then, these bits are transformed to valid signaling bits using Gray code. When the Gray coded bits are represented by decimal $\tilde{M}_n$ which is relative cyclic shift, the absolute cyclic shift of the n-th bootstrap symbol is M_n and it is calculated as follows,

$$M_n = \begin{cases} 0 & n = 0 \\ (M_{n-1} + \tilde{M}_n) \bmod N_{FFT} & 1 \le n < N_S. \end{cases} \quad (6)$$

Since the first bootstrap symbol does not include the signaling information, the absolute cyclic shift value M_0 of the first bootstrap symbol is zero. The bootstrap signal $A_n(t)$ is finally generated by cyclically shifting the time domain sequence $\tilde{A}_n(t)$ using the absolute cyclic shift of the n-th bootstrap symbol. A_n is obtained as follows,

$$A_n(t) = \tilde{A}_n\left((t + M_n) \bmod N_{FFT}\right), \quad (7)$$

III. CONVENTIONAL MODELS FOR BOOTSTRAP DETECTION

A. ITERATIVE MAXIMUM LIKELIHOOD FOR BOOTSTRAP DETECTION

In this section, iterative ML detection method is represented by [12]. It is assumed that the synchronization between transmitter and receiver is perfect. The relative cyclic shift detection is performed and the signaling information bits are finally obtained by the gray code demapper. The relative cyclic shift detection is performed iteratively for the ML method. The received bootstrap signals can be represented as follows,

$$\begin{aligned} R_n(k) &= H_n(k) S_n(k) e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n(k) \\ &= H_{e,n}(k) S_n(k) + W_n(k), \end{aligned} \quad (8)$$

where $R_n(k)$, $H_n(k)$, $S_n(k)$ and $W_n(k)$ are the received bootstrap symbol, channel, the bootstrap sequence and AWGN of the n-th symbol and k-th subcarrier, respectively.

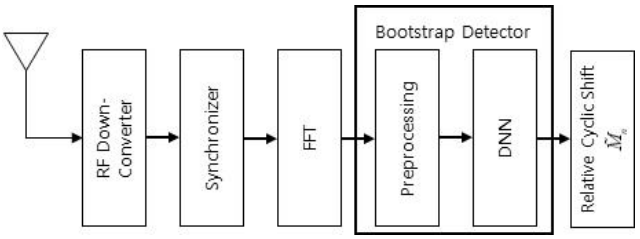


FIGURE 2. Block diagram to detect bootstrap signals based on DNN.

$H_{e,n}(k) = H_n(k) e^{j2\pi \frac{kM_n}{N_{FFT}}}$ denotes the equivalent channel gain of the k-th subcarrier for the n-th symbol. The ML decision rule for the relative cyclic shift can be calculated using IFFT operation as follows,

$$\begin{aligned} \tilde{M}_n &= \arg \min_{\tilde{m}_n \in \chi_r} \sum_{k=0}^{N_{FFT}-1} \left| R_n(k) - H_{e,n-1}(k) S_n(k) e^{j2\pi \frac{k\tilde{m}_n}{N_{FFT}}} \right|^2 \\ &= \arg \min_{\tilde{m}_n \in \chi_r} \mathrm{RE} \left\{ \sum_{k=0}^{N_{FFT}-1} R_n^*(k) H_{e,n-1}(k) S_n(k) e^{j2\pi \frac{k\tilde{m}_n}{N_{FFT}}} \right\} \\ &= \arg \max \mathrm{RE} \left\{ \mathrm{IFFT} \left\{ R_n^*(k) H_{e,n-1}(k) S_n(k) \right\} \right\}, \end{aligned} \quad (9)$$

where χ_r denotes the set of all possible values of the relative cyclic shift. RE $\{\bullet\}$ and IFFT $\{\bullet\}$ are the real part of a complex value and IFFT operation, respectively. Consequently, by using the channel gain $H_{e,n-1}(k)$ of the previous bootstrap symbol and the received bootstrap signal $R_n(k)$, the relative cyclic shift value $\tilde{M}_n$ is obtained.

The ML method is known as an optimal detection scheme among the digital signal detection schemes. Since ML method determines the signal which is the highest possibility among all the signals that can be transmitted, the receiver needs to know the information about all the signals that can be transmitted. However, in the case of ATSC 3.0 bootstrap, the number of transmittable signals can be changed according to the request of broadcasters in the future. As a result, the ML method cannot adapt to the changing standards of ATSC 3.0 without changing the receiver.

B. DEEP NEURAL NETWORK FOR BOOTSTRAP DETECTION

In recent, deep learning technology has been applied in communication engineering area [18]–[21]. In [21], deep neural network model is applied in MIMO system. This section introduces a DNN-based detection model for ATSC 3.0 with reference to [21].

In order to detect relative cyclic shift $\tilde{M}_n$ with the DNN, preprocessing is performed on adjacent symbols and it is used for training in the deep neural network model. Fig. 2 shows the block diagram of the receiver to detect bootstrap signals based on DNN. The bootstrap detector consists of preprocessing and DNN module. In the off-line training step, the DNN-based model is trained without noise $W_n(k)$. M_n is the absolute cyclic shift of the n-th symbol. By (3), the power of the bootstrap sequence $S_n(k)$ is normalized to 1. When the received bootstrap symbol $R_n(k)$ is compensated by $S_n^*(k)$, the

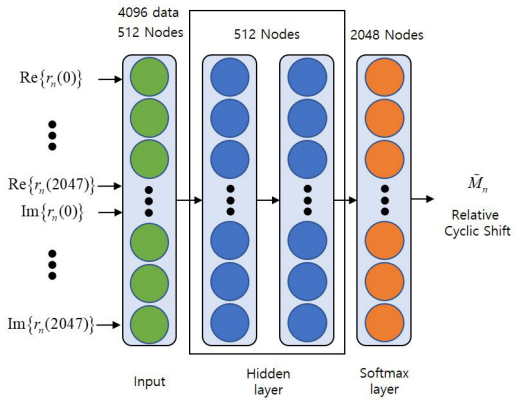


FIGURE 3. The proposed DNN model for ATSC 3.0.

result is as follows,

$$\begin{aligned} y_n(k) &= R_n(k)S_n^*(k) \\ &= H_n(k)S_n(k)S_n^*(k)e^{j2\pi\frac{kM_n}{N_{FFT}}} + W_n'(k) \\ &= H_n(k)e^{j2\pi\frac{kM_n}{N_{FFT}}} + W_n'(k), \end{aligned} \tag{10}$$

where $y_n(k)$ is the compensated bootstrap symbol and equivalent noise $W_n'(k)$ is $W_n(k) \times S_n^*(k)$. Since the power of $S_n(k)$ is normalized to 1, $S_n(k)S_n^*(k)$ is 1. The current bootstrap symbol is multiplied by the conjugate of the previous symbol and the result is obtained as follows,

$$\begin{aligned} r_n(k) &= y_n(k)y_{n-1}^*(k) \\ &= H_n(k)H_{n-1}^*(k)e^{j2\pi\frac{k(M_n-M_{n-1})}{N_{FFT}}} \\ &= H_n(k)H_{n-1}^*(k)e^{j2\pi\frac{k\tilde{M}_n}{N_{FFT}}}, \end{aligned} \tag{11}$$

where $\tilde{M}_n$ is the relative cyclic shift of the n-th symbol and $r_n(k)$ is preprocessed data of the n-th symbol and k-th subcarrier. And then, the real part and the imaginary part of $r_n(k)$ are arranged sequentially, for the deep neural network.

Fig. 3 shows the DNN-based detection model for ATSC 3.0 and this model consists of single input layer, single output layer and two hidden layers. Each layer except output layer has 512 nodes and this model uses tanh function as an activation function. Unlike other layers, output layer has 2048 nodes that perform classification of relative cyclic shift, and uses softmax function as an activation function. Also, it uses cross-entropy error as a loss function and adaptive moment estimation algorithm (Adam) which is an optimization algorithm.

In contrast to the ML method, the receiver does not know information for all transmitted signals in the DNN-based model. Therefore, since the receiver does not need to have information for all signals that are transmitted, the DNN-based detection model can adapt to the changing standard of ATSC 3.0. However, since the DNN-based model has very poor detection performance compared to the ML method, the method to improve detection performance is required.

TABLE 1. Architecture details of the CNN.

Layer Name	Input Dim.	Output Dim.	Filter
Convolutional Layer 1	$2 \times 4,096$	511×32	2×16
Convolutional Layer 2	511×32	62×32	1×16
Convolutional Layer 3	62×32	6×32	1×16
FC Layer 1	192	1,024	
FC Layer 2	1,024	1,024	
Classification Layer	1,024	2,048	

C. CNN FOR BOOTSTRAP DETECTION

In contrast to the CNN-based detection model, the demerit of DNN-based detection model is that the correlation of input data is ignored. For instance, image data has a two-dimensional shape (horizontal and vertical), and this shape has a spatial structure. Also, image data cells that are spatially close have similar or high correlation values. In the case of CNN, since the shape of the spatial structure of the input data is maintained and the correlation of data is transferred to the next layer, it can be said that there is a high possibility that can properly learn the shape data like image data. In the case of ATSC 3.0 bootstrap, since the duration of bootstrap symbol is shorter than the frame of entire bootstrap during bootstrap frame transmission, the channel information is considered to be invariant while the transmitter transmits the frame. In other words, it can be assumed that channel characteristics between adjacent symbols is similar. Also, bootstrap symbols are encoded according to the level of the relative cyclic shift of the current and previous symbols. For these two reasons, adjacent bootstrap signals have a high correlation. Therefore, the CNN-based detection technique is used to take advantage of the high correlation characteristics of these adjacent symbols.

Fig. 5 shows the CNN model for ATSC 3.0 bootstrap and TABLE 1 shows architecture details of CNN model. In this model, 3 convolution layer, 2 fully connected layer and 1 classification layer are used for bootstrap signal detection. 3 convolution layers are used for using the correlation of adjacent symbols. Since the ATSC 3.0 bootstrap is encoded via relative cyclic shift and the channel environment is static or slow fading, adjacent symbols are considered to be correlated. 2 fully connected layer and the number of neuros in each layer are determined by the empirical trials. The pooling layer is skipped to prevent the loss of information. As shown in Fig. 6, filter size of the first convolution layer is 2×16 and the number of filter is 32. Stride which is the distance between the positions that filters are applied is (2, 8). In order to prevent unnecessary information affecting on the output, padding which avoids the reduction of the output data size is set to 0. The output size of the convolution layer can be calculated as follows,

$$\begin{aligned} \text{RowSize} &= \left\lfloor \frac{r - F_r}{Stride_r} \right\rfloor + 1 \\ \text{ColumnSize} &= \left\lfloor \frac{c - F_c}{Stride_c} \right\rfloor + 1, \end{aligned} \tag{12}$$

Subcarrier

	$(n-1)$ -th Symbol	n -th Symbol	$(n-1)$ -th Symbol	n -th Symbol		$(n-1)$ -th Symbol	n -th Symbol
$\mathfrak{R}$	$\mathrm{Re}\{y_{n-1}(0)\}$	$\mathrm{Re}\{y_n(0)\}$	$\mathrm{Re}\{y_{n-1}(1)\}$	$\mathrm{Re}\{y_n(1)\}$	• • •	$\mathrm{Re}\{y_{n-1}(N_{FFT}-1)\}$	$\mathrm{Re}\{y_n(N_{FFT}-1)\}$
$\mathfrak{I}$	$\mathrm{Im}\{y_{n-1}(0)\}$	$\mathrm{Im}\{y_n(0)\}$	$\mathrm{Im}\{y_{n-1}(1)\}$	$\mathrm{Im}\{y_n(1)\}$	• • •	$\mathrm{Im}\{y_{n-1}(N_{FFT}-1)\}$	$\mathrm{Im}\{y_n(N_{FFT}-1)\}$

FIGURE 4. The two dimensional input data array of the CNN.

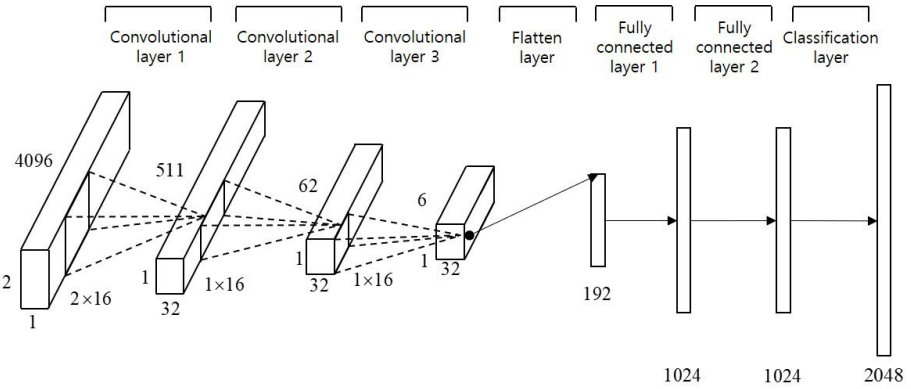


FIGURE 5. The structure of the CNN model for ATSC 3.0.

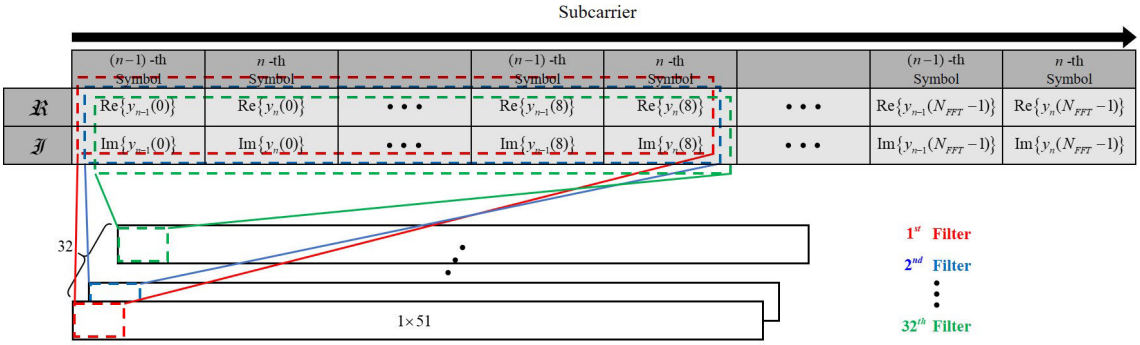


FIGURE 6. The first convolution layer architecture for bootstrap detection.

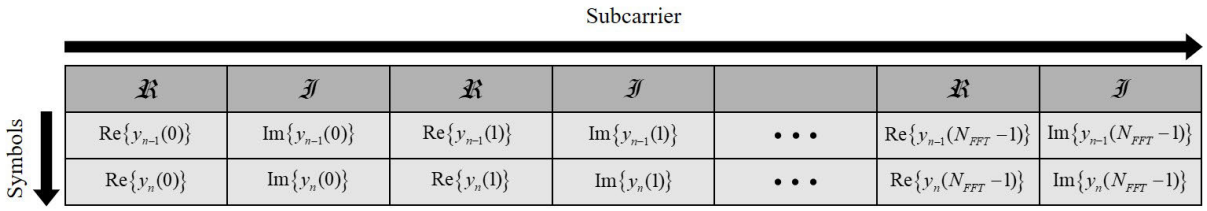


FIGURE 7. The alternately arranged input data array of the CNN.

where r and c are row and column size of the input data, F_r and F_c are row and column size of the filter, and $Stride_r$ and $Stride_c$ are the row and column stride of the filter, respectively. $\lfloor x \rfloor$ operator is floor function. With a consequence of (12), the output size of the first convolution layer is $1 \times 511 \times 32$ that row and column size and the number of filter are 1, 511 and 32, respectively. This output is used for the input of the second convolution layer. In the second and third convolution layer, filter size is 1×16, stride is $(1, 8)$ and the number of filters is 32. By equation (12), the

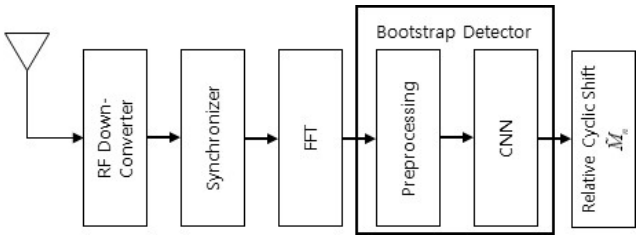


FIGURE 8. Block diagram to detect bootstrap signals based on CNN.

output size of the second and third convolution layer are $1 \times 62 \times 32$ and $1 \times 6 \times 32$, respectively. After performing the third convolution layer, a flatten layer is performed to convert $1 \times 6 \times 32$ data into 192 1-dimensional data. And then, two fully connected layers have 1024 nodes. Lastly the classification layer has 2048 class to estimate the relative cyclic shift.

IV. PROPOSED CNN FOR BOOTSTRAP DETECTION

Deep Learning have shown great performance in various fields such as pattern recognition [22], [23]. In this aspect, this paper deal with convolutional neural network (CNN) among deep learning technologies for detection of bootstrap signal. The why using CNN model is two. The first, While the major or minor version of ATSC 3.0 is evolved, hardware module using traditional detection scheme should be designed newly. But, If CNN-based detection model is used, the flexibility can be maintained without redesigning the hardware module by simply modifying the weights. The second is that the performance of DNN-based detection scheme is remarkably degraded compared to one of ML detection scheme. Since the bootstrap symbols has correlation with adjacent symbols, CNN-based model that can be utilized correlation property is used to improve the performance.

In this section, sequential array input is presented. And alternative array input data and corrupted by additive white Gaussian noise (AWGN) on purpose are proposed to improve the performance of CNN-based detection model using sequential array input.

A. STRUCTURE OF CNN MODEL

In this subsection, the structure of CNN model for detecting bootstrap symbol is described by the two-stage.

1) INPUT OF THE CNN

Fig. 8 shows the block diagram of the receiver to detect bootstrap signals based on CNN. The bootstrap detector consists of preprocessing and CNN module for detection of the bootstrap symbol. First, it is assumed that AWGN does not exist and synchronization is perfect. The received bootstrap symbol can be represented as follows,

$$R_n(k) = H_n(k)S_n(k)e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n(k), \tag{13}$$

where $R_n(k)$, $H_n(k)$, $S_n(k)$ and $W_n(k)$ are the received bootstrap symbol, channel, the bootstrap sequence and AWGN of the n-th symbol and k-th subcarrier, respectively. M_n is the absolute cyclic shift of the n-th symbol. When the received bootstrap symbol $R_n(k)$ is compensated by $S_n^*(k)$, the result is as follows,

$$\begin{aligned} y_n(k) &= R_n(k)S_n^*(k) \\ &= H_n(k)S_n(k)S_n^*(k)e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n'(k) \\ &= H_n(k)e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n'(k), \end{aligned} \tag{14}$$

where $y_n(k)$ is compensated bootstrap symbol and $W_n'(k)$ is $W_n(k) \times S_n^*(k)$. In order to make the received bootstrap signals into the input of CNN model, the received bootstrap signals are rearranged to a two-dimensional array. In the ATSC 3.0 system, channel information is considered invariant while the transmitter transmits the frame since the bootstrap symbol duration is short compared to the whole bootstrap frame duration to transmit the bootstrap frame. In other words, it can be assuming that the channel characteristic between adjacent symbols is similar. Additionally, the bootstrap symbol is encoded according to the relative cyclic shift of the current and previous symbol. For these two reasons, adjacent symbols are considered to be correlated. CNN-based detection scheme is used to utilize the correlation of the adjacent symbols.

Fig. 4 shows the two-dimensional input data array of the CNN. The row of Fig. 4 consists of real and imaginary value for compensated bootstrap symbol $y(k)$, and the column is arranged for the previous and current bootstrap symbol $y_{n-1}(k)$ and $y_n(k)$ of every subcarrier. Consequently, the size of input data is 2×4096.

2) ACTIVATION AND LOSS FUNCTION

Every layer except classification layer uses tanh function as activation function. The tanh function is as follows,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{15}$$

And classification layer uses softmax function performing 2048 classification. The softmax function $\sigma(x)$ is as follows,

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, \tag{16}$$

where n is the number of nodes and i is i-th index of nodes. Also, cross-entropy error function is used as a loss function and is as follows,

$$E = -\sum_k t_k \ln y_k, \tag{17}$$

where E is loss value, t_k is true label and y_k is the output of neural network.

V. PREPROCESSING METHOD FOR CONVOLUTIONAL NEURAL NETWORK

A. ALTERNATELY ARRANGED INPUT DATA

This subsection proposes an array structure for input data to improve the detection performance. Unlike the array structure

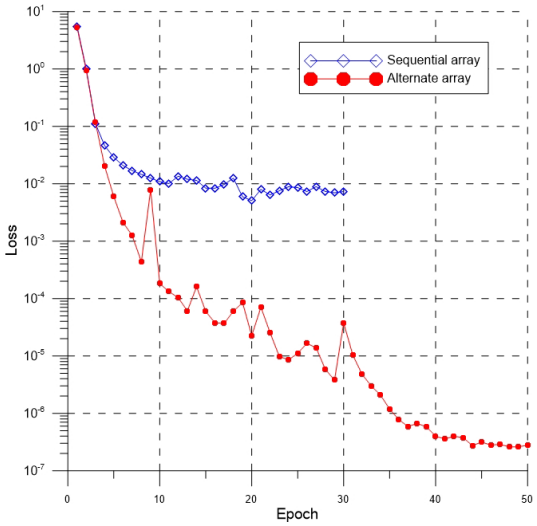


FIGURE 9. The loss using conventional and alternate arrange array.

of Fig. 4, the proposed array in Fig. 7 consists of rows with alternately arranged real and imaginary values of all subcarriers. The proposed array input data uses CNN model in Fig. 5. Since the same size of input data and CNN model is used, the sizes of output data for each layer are also same. Consequently, the output sizes of the each layer are $1 \times 511 \times 32$, $1 \times 62 \times 32$ and $1 \times 6 \times 32$, respectively.

Fig. 9 shows the loss of the sequential and alternate array using CNN model in Fig. 5 for epochs. In this result, the training and validation sets contain $350,000$ and $35,000$ samples, respectively. The loss of the sequential array is saturated at about from 20 to 30 epochs and training terminates at 30 epochs. Meanwhile, the proposed alternate array is saturated at about 40 to 50 epochs and training terminates at 50 epochs. According to these results, the training of the sequential array ends earlier than the alternate array, but the training of the proposed alternate array is more accurate. When the compensated bootstrap signal used as the input to CNN-based model is rearranged, a loss of the CNN-based bootstrap signal detection model is improved.

B. INPUT DATA CORRUPTED BY NOISE

All the above input data used for training is pure data which is not corrupted by noise. However, in practice, since the received bootstrap symbols are corrupted by noise, corrupted symbol is needed to be considered. Fig. 10 shows the example of the received bootstrap symbols are corrupted by noise.

When the relative cyclic shift is assumed to be 2 bits, Fig. 10 shows a constellation of the bootstrap symbol for each subcarrier. In this figure, black dots indicate the position of the relative cyclic shift $p(k)$ which is transmitted for each bootstrap symbol. It is represented by $p(k) = e^{j2\pi \frac{kM_n}{N}}$, $M_n = \{0, 1, 2, 3\}$. Since the relative cyclic shift is affected by the AWGN at the receiver, colored circles surrounding the black dot indicate the possibility that a distorted signal can be detected. the pilot signal with AWGN is represented by

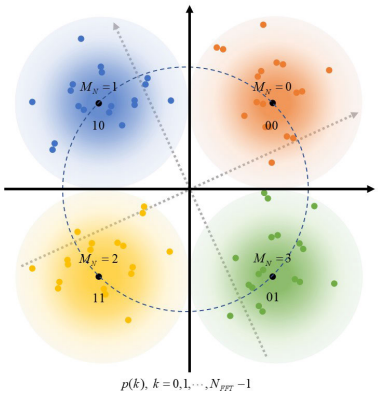


FIGURE 10. The example of constellation of the 2-bit relative cyclic shift.

$P'(k) = e^{j2\pi \frac{kM_n}{N}} + W(k)$. In this case, the relative cyclic shift is detected within the boundary which is black line axis as threshold. Additionally, when the subcarrier of the bootstrap symbols is considered, the boundary of threshold can be extended. Even if the bootstrap signal received for some subcarriers exceeds the threshold boundary, the bootstrap signal can be accurately detected when the bootstrap signal received for the most of subcarriers is within the threshold boundary. In order to train the threshold boundary, the input data corrupted by noise is used for offline learning.

For instance, if the proposed CNN-based model is trained by pure input data, the threshold boundary can be trained to gray dashed axes which are not able to provide proper separation. On the other hand, if the proposed CNN model is trained with the input data corrupted by noise (i.e. colored dots), it is able to train the proper threshold boundary such as black line axes. Consequently, through the proper threshold boundary, the practical received bootstrap signal is detected more precisely. On the contrary, if the noise power of the training data is too large, the size of the colored circle shown in Fig. 10 increases and is overlapped with each other. In this case, since the overlapping area is increased, it is difficult to find the proper threshold boundary.

In the off-line learning step, the received bootstrap signal $R_n(k)$ is used to pure data and it used for training. But, in practice, $R_n(k)$ is corrupted by thermal noise (i.e, AWGN). Therefore, $R_n(k)$ with AWGN is as follows,

$$R_n(k) = H_n(k)S_n(k)e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n(k), \tag{18}$$

where $R_n(k)$, $H_n(k)$, $S_n(k)$ and $W_n(k)$ are the received bootstrap symbol, channel, the bootstrap sequence and AWGN of the n-th symbol and k-th subcarrier, respectively. M_n is the absolute cyclic shift of the n-th symbol. When the received bootstrap symbol $R_n(k)$ is compensated by $S_n^*(k)$, the result is as follows,

$$\begin{aligned} y_n(k) &= R_n(k)S_n^*(k) \\ &= H_n(k)S_n(k)S_n^*(k)e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n'(k) \\ &= H_n(k)e^{j2\pi \frac{kM_n}{N_{FFT}}} + W_n'(k), \end{aligned} \tag{19}$$

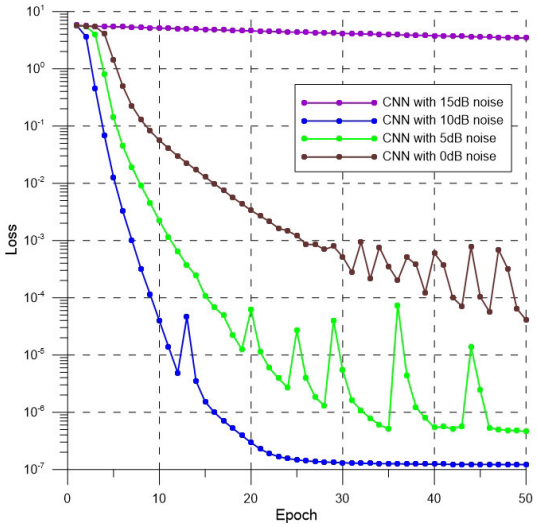


FIGURE 11. The loss of proposed CNN-based model trained by the noise data.

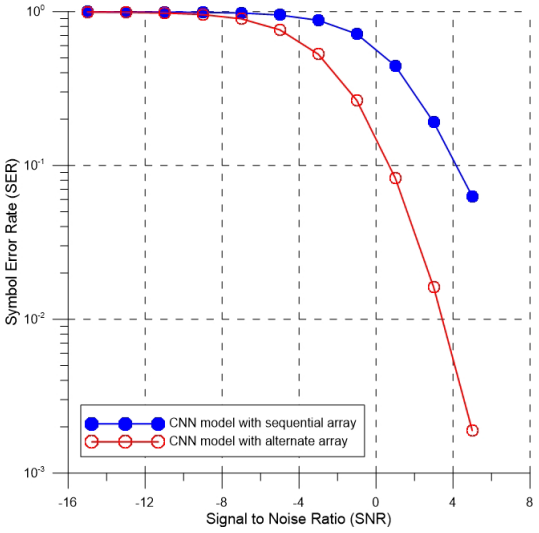


FIGURE 12. The SER performance of CNN model with sequential and alternate array.

where $y_n(k)$ is compensated bootstrap symbol and $W'_n(k)$ is $W_n(k) \times S^*_n(k)$. For off-line learning, $y_n(k)$ is arranged alternately, and then it is used to input of CNN in Fig. 5. Since the size of input data is same to alternately arranged input data and sequential arranged input data, the alternately arranged input data is trained in the same way as in section V-A.

Fig. 11 shows the loss performance of proposed CNN-based model according to the noise level of training data in the off-line training step. This result is simulated with 350, 000 training sets and 35, 000 validation sets. When the training data is too damaged by noise, the proposed CNN-based model is hard to train the threshold boundary. In this result, the proposed CNN-based model with training 15dB noise is too damaged to train the threshold boundary.

VI. SIMULATION RESULTS

For evaluating the proposed alternate array input data and with noise data, symbol error rate is measured. The simulations are based 1×1 SISO system and assumed that the synchronization is perfect. The major and minor version are set to 0, in other words, root q is 147 and initial state r_{init} is 413.

Fig. 12 shows the comparison of symbol error rate (SER) performance between the sequential array and the alternate array under pure data. Fig. 13 shows the SER performance of the replacement array according to the level of distortion caused by noise. For evaluating the performance of CNN-based model, Fig. 14 shows iterative ML detection scheme and DNN-based. Fig. 12, 13 and 14 are simulated under 20-path Rayleigh channel.

For CNN model using sequential array in Fig. 12, since the training loss value which shows in Fig. 9 is lower than the case using alternate array, the SER performance of sequential array is degraded compared to alternate array. Consequently, this simulation result shows that the structure of the proposed alternate array is more efficient than sequential array.

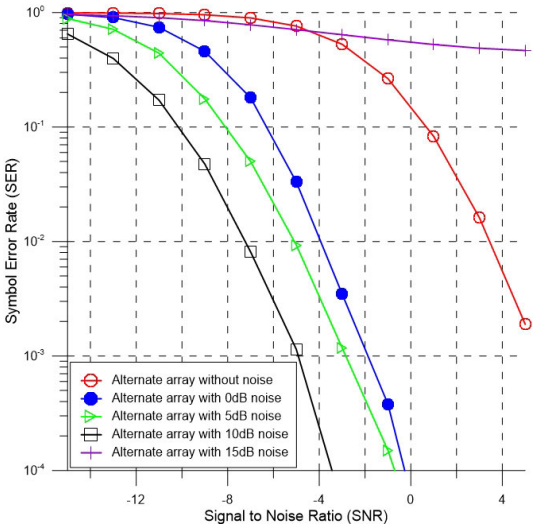


FIGURE 13. The SER performance of CNN model with alternate array according to the level of noise.

Fig. 13 shows the SER performance according to the level of noise of training data using alternate array. For comparison, the performance of alternate data without noise is inserted. The degree of noise is expressed as the signal to noise power ratio. According to this result, by increasing the SNR of proposed training data, the SER performance generally increases. However, the SER performance with low SNR of the proposed training data is poor. Because the alternate array that is distorted too much by noise is hard to train the threshold boundary among the different symbols.

Fig. 14 shows the SER performance of iterative ML method, DNN-based model and proposed CNN-based model with training 10dB noise data according to SNR. Since the CNN-based model utilizes the correlation between adjacent symbols, the SER performance of proposed CNN-based model trained with -10dB data is better than DNN-based

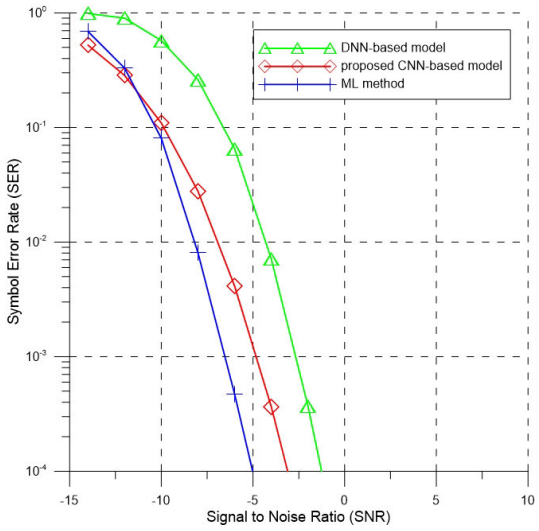


FIGURE 14. The SER performance comparison of DNN-based model, ML method and proposed CNN-based model.

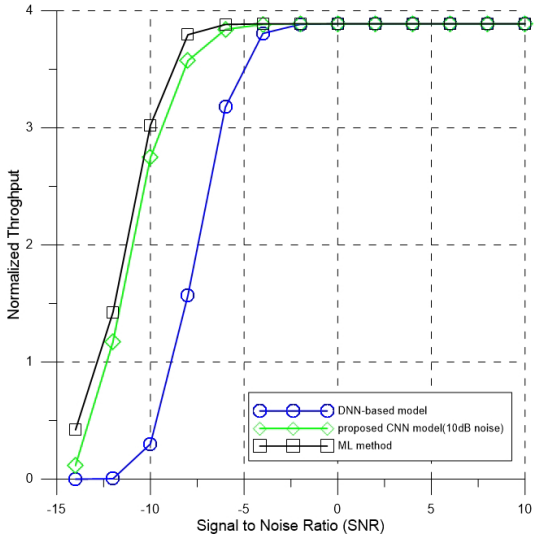


FIGURE 15. The normalized throughput performance comparison of DNN-based model, ML method and proposed CNN-based model.

model. On the other hand, the SER performance of proposed CNN-based model is degraded about 1.5dB compared to ML. However, the proposed CNN-based model is flexible for the change of the standard. By software update for the model weights, the receiver can adapt the change of the standard with the similar performance to the ML method. On the other hand, the ML method needs new scheme when the standard is changed.

Fig. 15 shows the normalized throughput performance for DNN-based model, iterative ML method and proposed CNN-based model with 10dB noise data according to SNR. To show that the throughput of the CNN-based method has similar performance with the ML method, throughput performance for the proposed CNN-based model with training 10dB noise is compared with ML method, which is well known as the optimal method in the detection of digital signal [10]–[12]. On the other hand, the throughput performance of proposed CNN-based model provides improved throughput performance compared to the DNN-based model at SNR under −2dB.

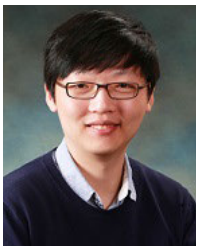
VII. CONCLUSION

This paper proposes CNN-based bootstrap detection model with two preprocessing methods for ATSC 3.0 systems. By using the proposed CNN-based model, this paper maintains the flexibility of the bootstrap signal and improve the performance of CNN-based detection model through two proposed preprocessing methods. First, by rearranging the array of the received bootstrap symbol which is used for the input of the CNN model, the CNN-based signal detection has better performance than the sequential array. Second, in the offline learning, by using the received bootstrap symbol corrupted by noise, the boundary of threshold is expanded and the SER performance of the CNN model is improved. Simulation results show that that the performance improvement for the CNN-based bootstrap signal detection is remarkable in ATSC 3.0 system.

REFERENCES

[1] *ATSC Standard:ATSC 3.0 System*, document A/300, Advanced TV System Committee, Washington, DC, USA, Sep. 2019.
[2] *System Discovery and Signaling*, document A/321, Advanced TV System Committee, Washington, DC, USA, Mar. 2016.
[3] *Physical Layer Protocol*, document A/322, Advanced TV System Committee, Washington, DC, USA, Jan. 2019.
[4] *Signaling Delivery Synchronization and Error Protection*, document A/331, Advanced TV System Committee, Washington, DC, USA, Dec. 2017.
[5] M. Earnshaw, K. Shelby, H. Lee, Y. Oh, and M. Simon, "Physical layer framing for ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 263–270, Mar. 2016.
[6] D. He, K. Shelby, M. Earnshaw, Y. Huang, H. Xu, and S.-I. Park, "System discovery and signaling transmission using bootstrap in ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 172–180, Mar. 2016.
[7] L. Fay, L. Michael, D. Gomez-Barquero, N. Ammar, and M. W. Caldwell, "An overview of the ATSC 3.0 physical layer specification," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 159–171, Mar. 2016.
[8] Y. Huang, D. He, W. Zhang, Y. Xu, and Y. Guan, "Adaptive bootstrap design for hybrid terrestrial broadcast and mobile communication networks," *IEEE Trans. Broadcast.*, vol. 65, no. 4, pp. 755–769, Dec. 2019.
[9] Y. Huang, D. He, Y. Xu, Y. Wang, W. Zhang, M. Wang, "Improved bootstrap design for frequency-domain signaling transmission," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 172–180, Mar. 2016.
[10] M. Rotoloni, S. Tomasin, and L. Vangelista, "Maximum likelihood estimation of time and carrier frequency offset for DVB-T2," *IEEE Trans. Broadcast.*, vol. 58, no. 1, pp. 77–86, Mar. 2012.
[11] B. Tan, X. Li, S. Ding, Y. Li, S. Akaho, and H. Asoh, "A novel maximum-likelihood detection for the binary MIMO system using DC programming," in *Proc. IEEE 10th Int. Conf. Awareness Sci. Technol. (iCAST)*, Oct. 2019, pp. 1–6.
[12] H. Kim, J. Kim, S.-I. Park, N. Hur, M. Simon, and M. Aitken, "A novel iterative detection scheme of bootstrap signals for ATSC 3.0 system," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 211–219, Jun. 2019.
[13] H. Kim, J. Kim, S.-I. Park, J.-Y. Lee, N. Hur, M. Simon, M. Aitken, and K. Gage, "An improved decoding scheme for emergency alert wake-up bits in ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 66, no. 1, pp. 1–8, Mar. 2020, doi: 10.1109/TBC.2019.2933798.
[14] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[15] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless network," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[16] K. Kim, J. Lee, and J. Choi, "Deep learning based pilot allocation scheme (DL-PAS) for 5G massive MIMO system," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 828–831, Apr. 2018.
[17] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
[18] T. Lin and Y. Zhu, "Beamforming design for large-scale antenna arrays using deep learning," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 103–107, Jan. 2020.
[19] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep learning based channel estimation for MIMO systems with received SNR feedback," *IEEE Access*, vol. 8, pp. 121162–121181, 2020.
[20] Q. Chen, S. Zhang, S. Xu, and S. Cao, "Efficient MIMO detection with imperfect channel knowledge—A deep learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.
[21] M. Baek, S. Kwak, J. Jung, H. M. Kim, and D. Choi, "Implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters," *IEEE Trans. Broadcast.*, vol. 65, no. 3, pp. 636–642, Sep. 2019.
[22] R. Mu and X. Zeng, "A review of deep learning research," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 4, pp. 1738–1764, 2019, doi: 10.3837/tiis.2019.04.001.
[23] A. Garg and A. Negi, "A survey on content aware image resizing methods," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 7, pp. 2997–3017, 2020, doi: 10.3837/tiis.2020.07.015.



TAE-HOON KANG received the B.S. degree in electronic information and communications engineering from Sejong University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S. degree with the Department of Information and Communications Engineering and the Department of Convergence Engineering for Intelligent Drone. His research interests include the areas of broadcasting communication system design and MIMO signal processing.

WON-SEOK LEE received the B.S. and M.S. degrees in information and communication engineering from Sejong University, Seoul, South Korea, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information and Communications Engineering and the Department of Convergence Engineering for Intelligent Drone. His research interests include the areas of signal processing for wireless communication systems and ambient RF communication systems.

MYUNG-SUN BAEK (Member, IEEE) received the Ph.D. degree in information and communication engineering from Sejong University, Seoul, South Korea, in 2009. Since 2009, he has been with the Electronics and Telecommunication Research Institute, where he is currently a Senior Researcher. His research interests include the areas of digital broadcasting systems, MIMO signal processing, FTN signaling, deep learning-based physical layer design, and full duplex.

BYUNGJUN BAE received the B.S., M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University, South Korea, in 1995, 1997, and 2006, respectively. From 1997 to 2000, he was a Researcher with LG Electronics Inc., where he was involved in digital signal processing in digital television. Since 2000, he has been with the Media Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. He has also been a Professor with the Department of Communication and Media Technology, University of Science and Technology (UST), Daejeon. His current research interests include next-generation broadcasting protocols and systems, emergency information signal processing, and intelligent media processing.

HYOUNG-KYU SONG received the B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, South Korea, in 1990, 1992, and 1996, respectively. From 1996 to 2000, he was a Managerial Engineer with the Korea Electronics Technology Institute (KETI), South Korea. Since 2000, he has been a Professor of the Department of Information and Communication Engineering and the Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul. His research interests include digital and data communications, information theory, and their applications with an emphasis on mobile communications.