

## Research Article

# A New Generation Communication System Based on Deep Learning Methods for the Process of Modulation and Demodulation from the Modulated Images

Nihat Daldal <sup>1</sup>, Zeynel Abidin Sezer <sup>1</sup>, Majid Nour,<sup>2</sup> Adi Alhudhaif <sup>3</sup>, and Kemal Polat <sup>1</sup>

<sup>1</sup>Department of Electrical and Electronics Engineering, Bolu Abant Izzet Baysal University, Bolu 14280, Turkey

<sup>2</sup>Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

<sup>3</sup>Department of Computer Science, College of Computer Engineering and Sciences in Al-kharj, Prince Sattam Bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, Saudi Arabia

Correspondence should be addressed to Kemal Polat; [kpolat@ibu.edu.tr](mailto:kpolat@ibu.edu.tr)

Received 17 April 2022; Revised 8 May 2022; Accepted 12 May 2022; Published 28 May 2022

Academic Editor: Saeid Jafarzadeh Ghouschi

Copyright © 2022 Nihat Daldal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Demodulating the modulated signals used in digital communication on the receiver side is necessary in terms of communication. The currently used systems are systems with a variety of hardware. These systems are used separately for each type of communication signal. A single algorithm facilitates the classification and subsequent demodulation of signals without needing hardware instead of extra hardware cost and complex systems. This study, which aims to make modulation classification by using images of signals, provides this convenience. In this study, a classification and demodulation process is done by using images of digital modulation signals. Convolutional neural network (CNN), a deep learning algorithm, has been used for classification and recognition. Images of the signals of quadrature amplitude shift keying (QASK), quadrature frequency shift keying (QFSK), and quadrature phase shift keying (QPSK) digital modulation types at noise levels of 0 dB, 5 dB, 10 dB, and 15 dB were used. Thanks to this algorithm, which works without hardware, the success achieved is around 98%. Python programming language and libraries have been used in training and testing the algorithm. Demodulation processes of these signals have been performed for demodulation using the nonlinear autoregressive network with exogenous inputs (NARX) algorithm, an artificial neural network. As a result of using MATLAB, the NARX algorithm achieved approximately 94% success in obtaining the information signal. Thanks to the work done, it will be possible to classify and demodulate other communication signals without extra hardware.

## 1. Introduction

The communication systems used today mostly use digital communication methods. Additionally, the increase in digital devices has highlighted the need for fast data transmission. The receiver's distance and the transmitter, environmental reasons, and the devices used affect the communication quality [1]. The basic communication system given in Figure 1 includes the stages of transmitting an information signal. It is an important communication rule that this information can be transmitted quickly and without loss.

While the information signal is expressed as a baseband signal in digital communication, it is called passband

modulation with a high-frequency carrier [2]. The demodulation process must be performed following the modulation type of the transmitted signal [3]. However, in some cases, the modulation type of the signal obtained is unknown. In this case, the demodulation process is complicated, and extreme time loss occurs. In this study, it has been provided that the classification of the modulation signals is made by using the images, and thus, the demodulation processes will be easier. The convenience in this work, which is put forward for users such as security intelligence, where speed and time are valuable, will be a gain. Furthermore, developing a single software system that can receive and analyze different modulation signals will provide time and accuracy gains. Compared with the conventional

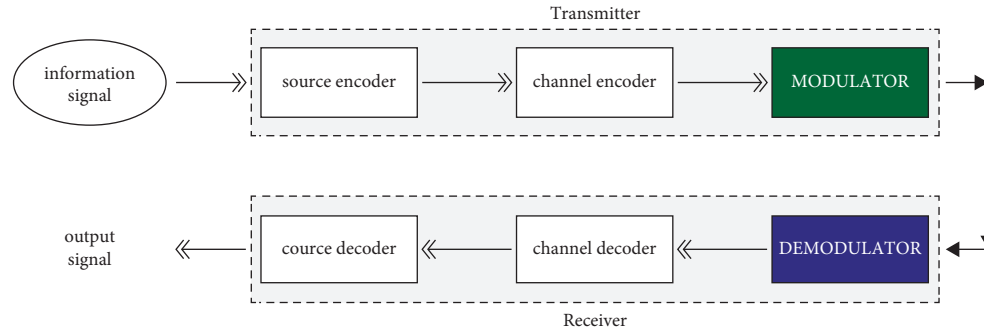


FIGURE 1: Basic communication system flowchart.

demodulator systems, a single software system that can also receive and analyze other modulation signals has been developed instead of systems that only receive a specific digital modulation signal.

Classical algorithms have a low digital signal modulation recognition rate at low signal-to-noise ratios [4]. Applications such as recognizing and classifying signals through machine learning are the important point. Machine learning, deep learning, and image processing methods achieve classification in a short time by eliminating extra hardware costs [5].

The automatic modulation classification and demodulation proposed in the studies in the literature show how critical this area is. The most important problem in modulation classification has been the noisy signals [6] because it is very difficult to recognize and classify a noisy signal. Thanks to the convolutional neural network (CNN) model developed to achieve swift and perfect accuracy results, the algorithm successfully has been performed for automatic modulation classification (AMC) [7]. Images of quadrature amplitude shift keying (QASK), quadrature frequency shift keying (QFSK), and quadrature phase-shift keying (QPSK) modulation signals are stored in three different folders. These images pertain to signals with 0 dB, 5 dB, 10 dB, and 15 dB noise ratios. Using images, folders belonging to individual signals were combined in a path, allowing them to be read with Python libraries. This model can understand whether any image loaded into the algorithm belongs to these three modulation types. If it is one of the QASK, QFSK, or QPSK, we can recognize it automatically. The created CNN model trained the images it reads from the folders according to the specified epoch number.

## 2. Related Work

Nandi et al. conducted experiments on modulation recognition using an artificial neural network in the literature. Using two modulation classification approaches, namely, the decision theory approach and the pattern recognition approach, they performed the recognition process with 100% accuracy for 0 dB, 95% for 3 dB, and 75% for 10 dB [6]. They used artificial neural networks in their studies by classifying radio communication signals to detect unknown communication signal types and achieved the success of over 85%. Wang et al. determined a 100% accuracy rate for images of

RZ-OOK, NRZ-OOK, RZ-DPSK, and PAM signals in the 10 dB to 25 dB noise range to select the modulation format and the estimation of the noise ratio using a convolutional neural network (CNN) [5]. In their study, Ali and Yangyu classified digital modulation using unsupervised pretraining and feed-forward deep neural network (DNN) for automatic modulation classification [8]. Ma et al., using CNN, DBN, and AdaBoost algorithms for demodulation of the signals, achieved a success of approximately 96% by demodulating these modulation types from the image set consisting of QAM QPSK, PPM, and OOK modulation signals [4]. Lee et al., in their studies, made a classification for BPSK (dual phase shift keying), QPSK, PSK, PAM, and QAM modulation types based on the images of these signals. They achieved 83.3% success by extracting features using the CNN algorithm and performing the classification process [9]. Daldal et al., in their study, used STFT and CNN-based hybrid models to classify the digital modulation type and achieved 99% success [10]. In Zhou et al.'s study, modulation recognition applications were carried out using deep learning. As they stated in the confusion matrix of the CNN algorithm, they found a success rate of 96.25% [11].

## 3. Material and Method

**3.1. Dataset and Structure of Quadrature Digital Modulation Signals.** Digital modulation means converting a baseband digital message signal into a bandpass signal at a carrier frequency [2]. Digital modulation is accomplished by changing the amplitude, frequency, or phase of the high-frequency sinusoidal analog carrier signal according to the incoming information carrying the digital baseband message signal [3, 12]. In quadrature or multilevel communications, multiple information with a single carrier is transmitted [10]. In multilevel modulation types, 2-bit can be transmitted simultaneously. These bit values consist of 00, 01, 10, and 11. Each symbol (00, 10, 01, and 11) corresponds to a phase state of the modulated carrier [12]. For QASK modulation, four amplitudes for each pair of bit values, four frequencies for each pair of bit values for QFSK modulation, and four phase values for each pair of bit values for QPSK modulation were determined [10].

The formulas of the quadrature modulation types are shown in Table 1. Modulated images were obtained by applying these formulas in MATLAB. Additive white

TABLE 1: Equations of QASK, QPSK, and QPSK modulation.

	00	01	10	11
QASK	$A_1 \times \text{Cos}W_c t$	$A_2 \times \text{Cos}W_c t$	$A_3 \times \text{Cos}W_c t$	$A_4 \times \text{Cos}W_c t$
QFSK	$A \times \text{Cos}(2.\text{pi}.f_1 t)$	$A \times \text{Cos}(2.\text{pi}.f_2 t)$	$A \times \text{Cos}(2.\text{pi}.f_3 t)$	$A \times \text{Cos}(2.\text{pi}.f_4 t)$
QPSK	$A \times \text{Cos}(W_c t + \theta_1)$	$A \times \text{Cos}(W_c t + \theta_2)$	$A \times \text{Cos}(W_c t + \theta_3)$	$A \times \text{Cos}(W_c t + \theta_4)$

Gaussian noise (AWGN) has been added as noise here. As a result of using the procedure, the images given in Figure 2 are obtained. Signals of data between 1 and 255 were coded according to the quadrature modulation type.

In addition, 5 dB, 10 dB, and 15 dB as Gaussian noise have been added to these images. As shown in Table 1, QASK has the same frequency and phase but different amplitude for each logic level, while QFSK has the same amplitude and phase but different frequency. For QPSK, the stages are different for each logic level [2]. The signals obtained from changing the amplitude, frequency, or phase of the carrier between more than two different values are called multilevel transmission [1]. Thuswise, it becomes possible to transmit more information with a single carrier. These signals obtained were saved as images. Images of each class were stored in separate folders. The images acquired are read from the paths they are in using Python libraries. Classes are class 0 for QASK, class 1 for QFSK, and class 2 for QPSK.

**3.2. Convolutional Neural Network (CNN).** Convolutional neural networks (CNN) that are trainable are made up of multiple stages [13]. CNN is a type of algorithm consisting of an input, an exit, and many hidden layers [14].

It includes the hidden, convolution, pooling, flattened linear unit, fully connected, and classification sections. *The convolution layer and pooling layer* have the task of editing feature maps [15]. Convolution for one-dimensional data is mathematically the name given to the process that a function with a real value that runs through another part to generate a new function [16]. The convolution layer enables the adjustment of the neurons in the image matrix, which is defined as the input called feature map and facilitates the learning of the properties [17]. After the convolution layer is detected locally, the same properties are combined thanks to the pooling layer [15]. Otherwise, the size of the feature matrix resulting from the convolution layer will increase and affect the duration of the training [18]. Or it can lead to overlearning. Each neuron in the feature map is connected to its previous neighbor thanks to filters, in other words, trainable weights [15]. The equation for the convolution layer used for the images is as follows:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n), \quad (1)$$

used as [19]. The rectified linear unit (ReLU) layer performs the task of flattening the feature map that emerges after the convolution process [20]. Converting negative values to zero produces output between zero and positively infinite values [21]. There is no change in the size of the data in this layer. The ReLU activation function enables it to increase by affecting the nonlinear feature of the neural network. Other

activation functions, sigmoid, and tanh reduce the speed of the neural network algorithm and show lower performance than the ReLU activation function in terms of results [22] (Figure 3). Calculation of the ReLU function is simpler than other functions since it is not subjected to logarithmic operations [23].

Moreover, the gradient of the ReLU activation function is always 1. On the other hand, the different activation functions take the value 0. Therefore, if the input functions are not entered correctly, the gradient will always sub tend to zero in positive values. Such an undesirable situation prevents the training set from working effectively [19, 24]. The max-pooling layer performs the size reduction operation by performing the function operation defined as subsampling [25]. Additionally, thanks to this layer, excessive memorization is prevented. In this layer, the downsampling method is applied to the feature map created in the convolution layer, and the process of assigning the largest single value instead of large-sized values is performed [22].

In CNN architecture, the features produced by the final convolution layer correspond to a portion of the input image because the receiving area does not cover the entire spatial dimension of the image [24]. Therefore, the fully connected layer becomes mandatory [26]. Thanks to the fully connected layer, the properties that appear in the convolution and pooling layers become meaningful [17]. With each layer, the linear activation function in the previous layer is generated by a nonlinear activation function [25].

The neurons in the fully connected layer are interconnected with the neurons in the entire previous layer. Thus, the output of this layer consists of the labels belonging to the classes.

Regularization in the training phase in the convolutional neural network is an essential element for data augmentation, regularization of weights, and batch normalization [15]. Hereby, the method called dropout is used. Its main use is to prevent overfitting [27]. If the subtilization number is too high, its use in some cases causes learning problems since it dilutes the fully connected layer and filters in neural networks. The subtilization process used in the pooling layer instead of the convolution layers reduces the error rates in the test data [28].

**3.3. Nonlinear Autoregressive Network with Exogenous Inputs (NARX).** Nonlinear autoregressive network with exogenous inputs (NARX) is the name given to neural networks with feedback structure to reach target values [29]. It is used for nonlinear modeling data. Autoregression is defined as a concept that shows the relationship between the previous value and other values in the network [30].

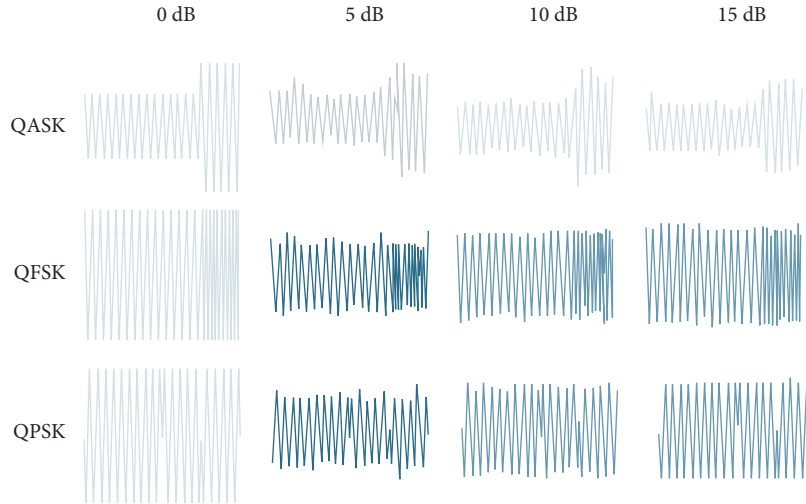


FIGURE 2: Images of QASK, QFSK, and QPSK modulation.

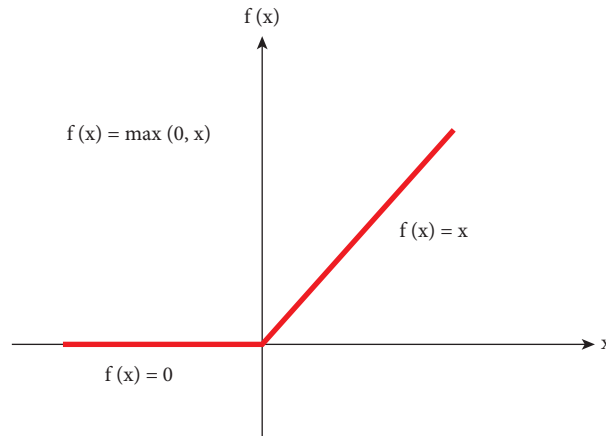


FIGURE 3: ReLU activation function.

Artificial neural networks are used in classification, clustering, object recognition, and prediction, which have a wide range of uses. Since it is a feedback neural network, it has a structure that learns errors and takes advantage of them. NARX model is a model that depends on nonlinear dynamic variables [31]. It has many layers due to the feedback it has. The weights and feedback of the NARX model can be randomly selected.

$$y(t) = f[y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)]. \quad (2)$$

As shown in (1), the nonlinear function  $f$  represents the NARX output  $y(t)$ , the input values  $u(t)$ ,  $n_u$  and  $n_y$  the input-output layers [32]. The output value  $y(t)$  improves the duration of the network by converging to the previous values of independent or self-linked input signals. The NARX model is divided into two types. These are as follows.

*Series-Parallel Model.* In this structure, the next values of  $y(t-1)$  make predictions based on the current and past values of  $x(t)$ , as well as the actual past values of  $y(t)$ .

*Parallel Model.* It estimates by gaining experience from the values of  $x(t)$  and the predicted values of  $y(t)$  in the past.

The NARX neural network has a structure with feedback, that is, progressing according to the error. Since the NARX structure is a type of artificial neural network, it has many hidden layers, convolution layers, pooling, and full connection layers. The flowchart of the NARX model used for demodulation is shown in Figure 4. Figure 5 shows the proposed CNN structure in our study.

## 4. Experimental Results

*4.1. Classification Results.* The advantage of CNN is that they can process data with different spatial dimensions, resulting in less computational costs than traditional matrix multiplication neural networks [33]. Additionally, convolution is simple to implement. Convolution is applied to images of different widths and heights, a different number of times depending on the input size. The output of the convolution process is scaled accordingly [34].

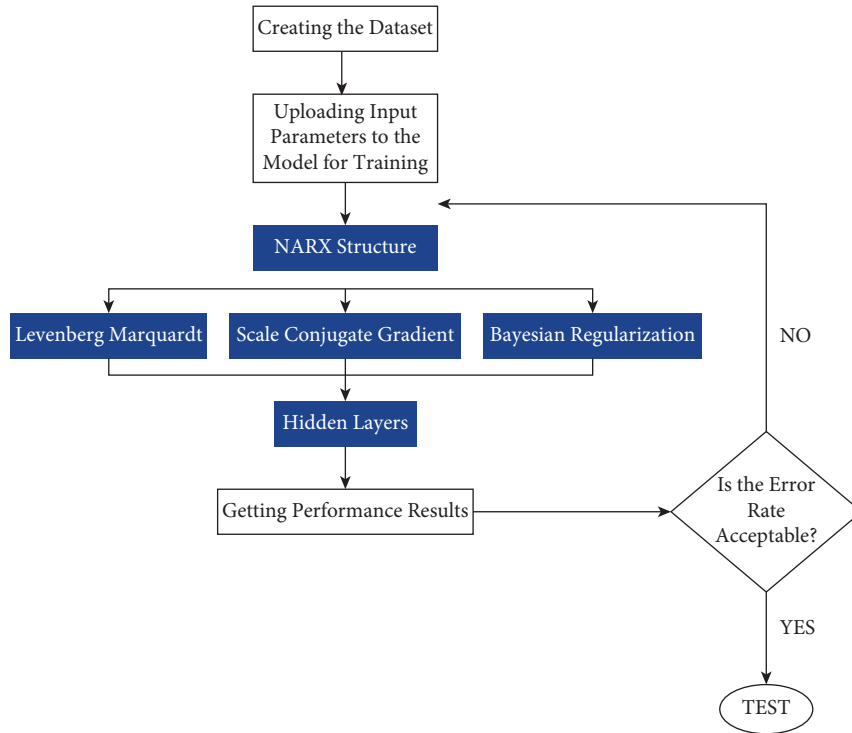


FIGURE 4: NARX algorithm flowchart.

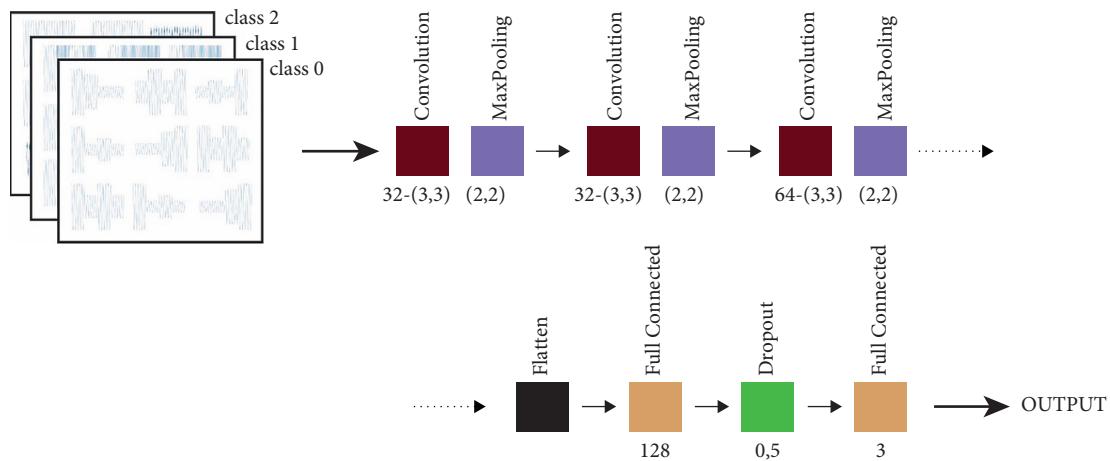


FIGURE 5: Proposed CNN structure in our study.

The dimensions are made the same thanks to the pooling layer that comes after the convolution layer, independent of the dimensions. In the classification made with CNN, images of the QASK, QFSK, and QPSK signals of 0 dB, 5 dB, 10 dB, and 15 dB noise ratios were classified into three different folders. Python libraries read these folders from their location and make them ready for the CNN algorithm. The operation system library has been used to read the images from the folders for processing. The images included in the algorithm were resized and brought to  $200 \times 200$  dimensions. This ensures that if the images are of different sizes, they are all the same format for CNN. Image matrices were saved, and classes were created by determining 0 for QASK, 1 for QFSK, and 2 for QPSK.

Then, the image matrices were assigned to a variable, and their labels were transferred to another variable and stored to be used for CNN.

CNN model was created using Keras. ReLU was chosen as the activation function in the CNN model. In the compiler of the model, the algorithm is run by selecting the loss function sparse categorical cross-entropy and Adam as the optimization function. Adam optimization is known as an algorithm that follows a probabilistic approach. Thus, it works fast for high-dimensional datasets [35]. Further, CNN has chosen as a good option for optimization problems for machine learning. As seen in the functional diagram of the layers belonging to the CNN model used, it is given. There are

many regularization methods in the CNN model. Besides dropout, data augmentation, normalization, and weight decay are some of them [36]. The sequential structure of the CNN model because of compiling the code in Python is given in Figure 6. While fitting the dataset to the model, 0.3 was determined as the validation split ratio, and 30% of the dataset was used for validation.

As can be seen in the CNN model used, there is a dropout layer, one of the regularization methods. In addition, data augmentation is also used. The rotation angle is 30 degrees, the zoom ratio is 0.2, the horizontal and vertical shift value is 0.1, and the horizontal and vertical random flip is chosen as true. The data augmentation stage is used to avoid overfitting and increase the model's predictive power.

Thanks to this stage, the quality of any input image or the angle of the objects in the image becomes insignificant. Since we have three classes and the targets are expressed in terms of the index (0, 1, 2), we used sparse categorical cross-entropy. The formula for it is

$$\text{CCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)], \quad (3)$$

where  $y_i$  represents true labels and  $\hat{y}_i$  represents predicted labels [37].

It was placed in two different folders to share the data as train and test. These two folders have different folders for every three classes. Each class comprises of 0 dB, 5 dB, 10 dB, and 15 dB signal images. These decibel values have 255 images. So, a class has 1020 images. Consequently, all classes have a total of 3060 images (Figure 7). Visualizing numbers of train and test data has a visualization of class numbers using the seaborn library. Learning rate, which is an important parameter in the training of the CNN model, is used as the update rate of the estimated weight errors. The weights in the hidden layers have updated the model. The purpose of this is to minimize the loss of function.

The magnitude of the learning rate also affects the speed of the training [38]. For example, a big ratio results in an unacceptable level of the loss function, while small level training is slow, and a very small amount of weight is updated [19]. In other words, the optimal learning rate to be selected for the model leads to approaching the best result.

**4.2. Demodulation Results Using Deep Learning and NARX Models.** The demodulation process using the NARX structure is carried out to obtain the information signal superimposed on the carrier. NARX consists of three parts: training, testing, and verification. Performance evaluation is made at the validation stage. At this stage, the success of the model is predicted according to the training and test results. 70% of the dataset was reserved for training, and 15% of the rest was divided for validation and 15% for testing. The dataset included in the algorithm from the MATLAB workspace and consisting of  $3060 \times 400$  dimensions was determined as target values, that is, information mark,

$3060 \times 1$  sized sequences from 1 to 255 for each signal type. Once the network is trained and used for prediction, the output is reported back to the network to obtain the forecast to generate the next prediction step. The serial-parallel model does the training by reducing the iteration time. The model is created for initial training by randomly assigning the weights to the initial training. Then, in each iteration, the model adjusts itself thanks to the feedback. The structure of the NARX model used is shown in Figure 8.

**4.3. The Obtained Results.** The model's results in training performed according to these three learning rates of 0.01 and 0.001 are shown in Figure 9. The epoch number is chosen as 20, and the process is performed according to the same epoch number. In consequence of compiling the model in Python, approximately 1.6 million trainable parameters were obtained. As can be seen in Figure 9, after the two learning rate trainings, the graphics are similar to each other. The training time was longer for the 0.0001 rates. The loss curve, which is one of the most used graphs for error detection in CNN models, gives us visual results to reveal the learning ability of the training process and the network. The lower the learning rate, the more delayed the convergence [19]. As can be seen in Figure 10, the convergence of the graph with a 0.001 learning rate was earlier. As it can be understood from the graphics in both Figures 9 and 10, there is no overfitting compatibility of the model in training. Although our dataset is not large enough, the result shows that the model is successful. According to the learning rates, the success of the model was 98.37% for 0.001, while it was 99.45% for 0.0001.

Figure 11 represents the confusion matrices of CNN models according to both learning rates. Using the CPU, the model's training took about 20 minutes for a learning rate of 0.001, compared with about 30 minutes at a rate of 0.0001. However, the training time for both operations performed when the GPU is used is less than one minute.

As it can be seen in the classification report in Figures 11 and 12, there has been a slight increase in the amount of data in the dataset as data increase is applied. Due to the classification report, individual precision, accuracy, and f-score values can be obtained for each class. These values have been revealed by using the sklearn library.

Figure 13 shows the closeness of demodulation estimation data obtained from modulated signals carrying 0-255 8-bit binary data in 0-5-10-15 dB noise of QASK, QFSK, and QPSK signals collectively. The first four signals belong to QASK 0-5-10-15 dB, the next four to QFSK 0-5-10-15 dB, and the last four to QPSK 0-5-10-15 dB. When looking at all modulated signals, it is understood that the estimation of demodulation estimation data of 5 dB noisy signals contains more errors than others.

Mean square error rate (MSE), one of the methods used for performance evaluation, provides information about the success of the neural network. Figure 14 shows the harmony between target values and estimated values. Therefore, the predictions here should be along the

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 120, 120, 32)	320
activation (Activation)	(None, 120, 120, 32)	0
max_pooling2d (MaxPooling2D)	(None, 60, 60, 32)	0
conv2d_1 (Conv2D)	(None, 60, 60, 32)	9248
activation_1 (Activation)	(None, 60, 60, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
activation_2 (Activation)	(None, 28, 28, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 128)	1605760
activation_3 (Activation)	(None, 128)	0
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
activation_4 (Activation)	(None, 3)	0

FIGURE 6: Sequential structure of our CNN Model.

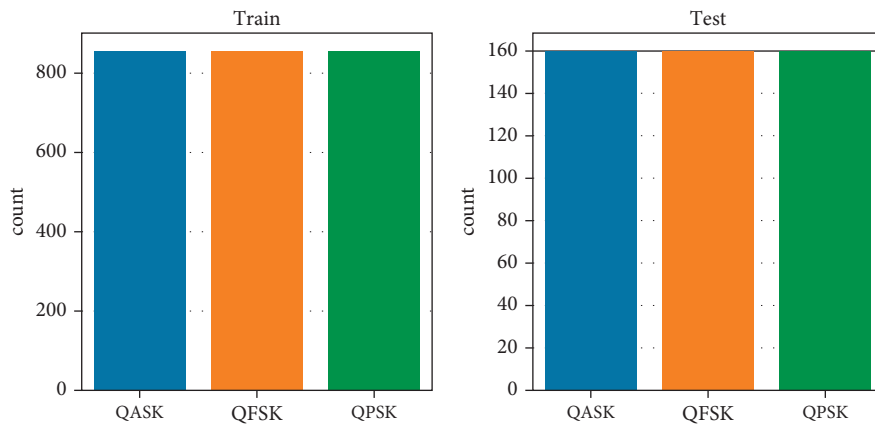


FIGURE 7: Visualizing numbers of train and test data.

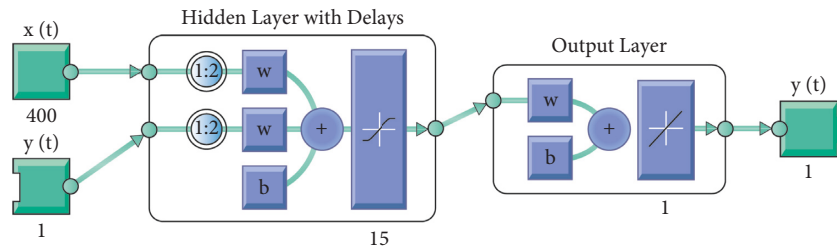


FIGURE 8: NARX neural network structure.

points expressed with dashed lines starting from the center point (0, 0).

As can be seen from Figure 14, the rate for validation is around 94%. Demodulation of noiseless signal images has been achieved with a high success rate. In addition, the epoch value is 107 for the lowest mean square error rate, as shown in Figure 15.

### 5. Discussion

As in the image given in Figure 16, individual test images for each of the three types of signals were loaded into the model and asked to classify. The model was assigned one for the images that made the class estimation correctly and zero for the others. Finally, the saved model was loaded and

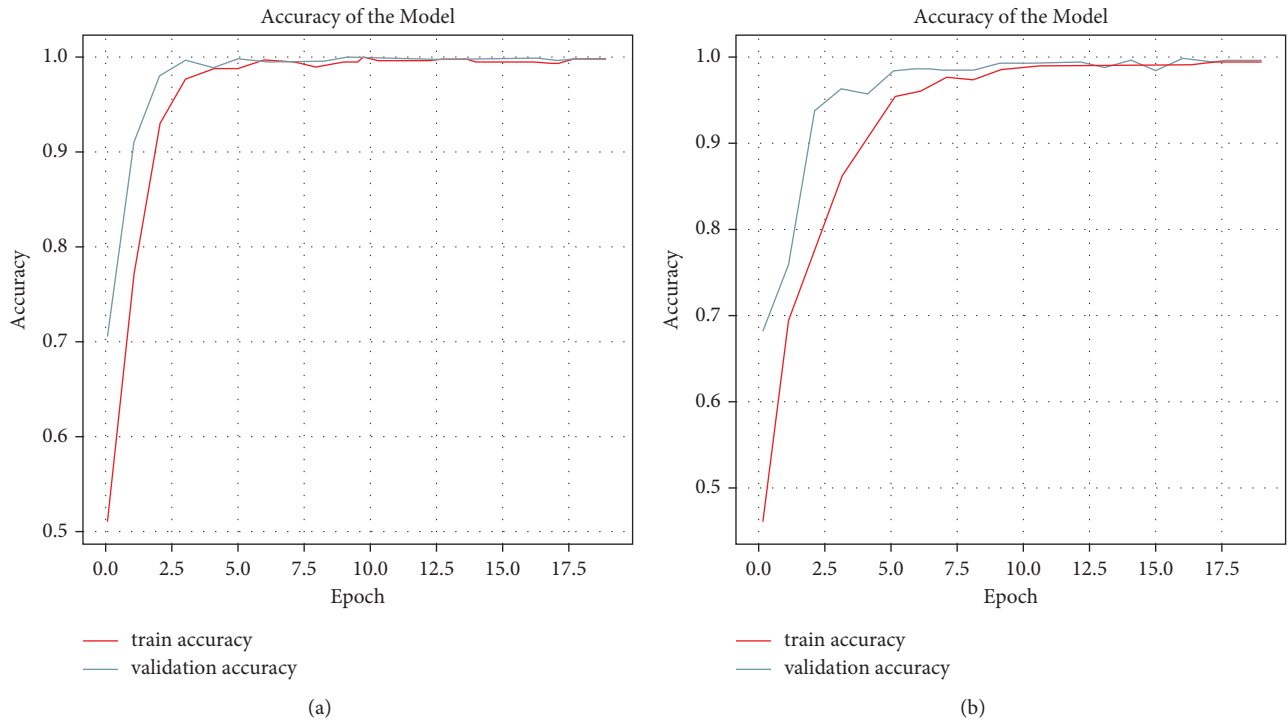


FIGURE 9: Training and test graphs according to the learning rates of the CNN model. (a) Learning rate graph selected as 0.001. (b) learning rate graph selected as 0.0001.

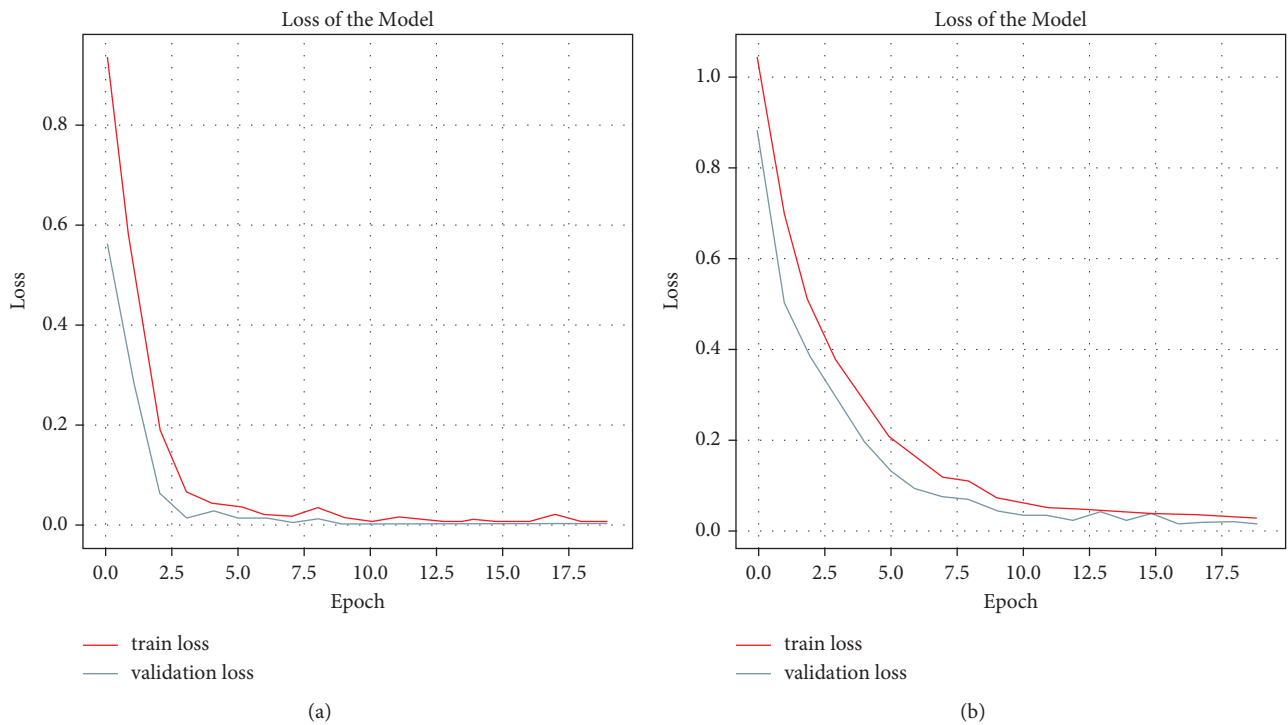


FIGURE 10: Loss plots according to train and test results of CNN model. (a) Learning rate graph selected as 0.001. (b) Learning rate graph selected as 0.0001.

run for test images. Test images tested anonymously are different from images produced using MATLAB. The purpose of this is to measure the model's reaction to other

images. Since the noiseless type of the QASK modulation signal is easy to detect, the noisy signal is used. The generated model is not affected by noise ratios, although the



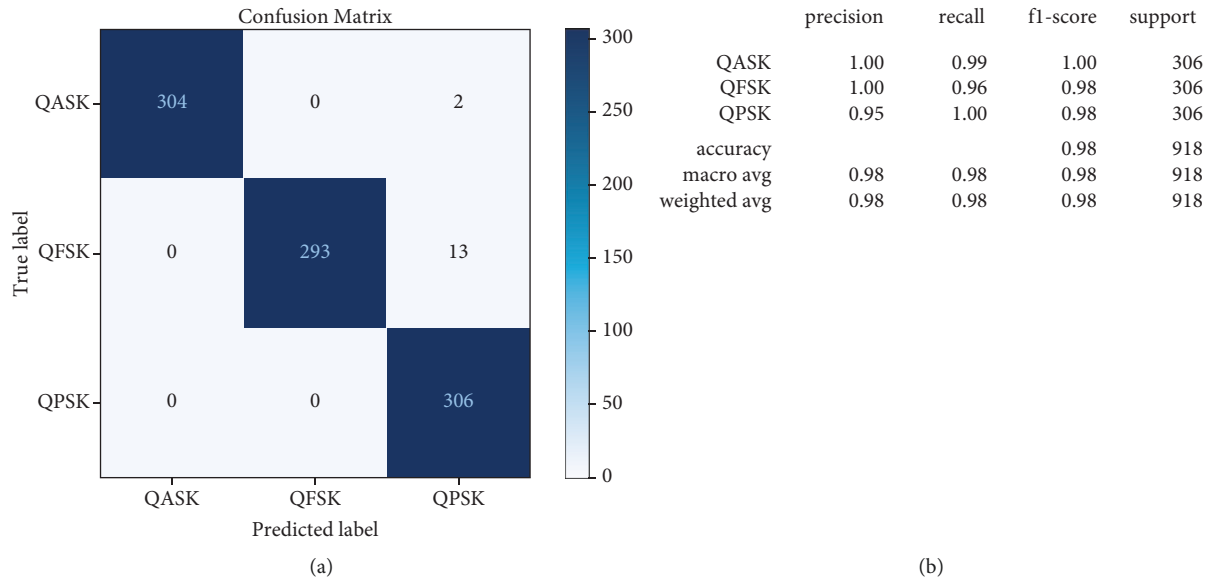


FIGURE 11: Model output for 0.001 learning rate. (a) Confusion matrices. (b) Classification report.

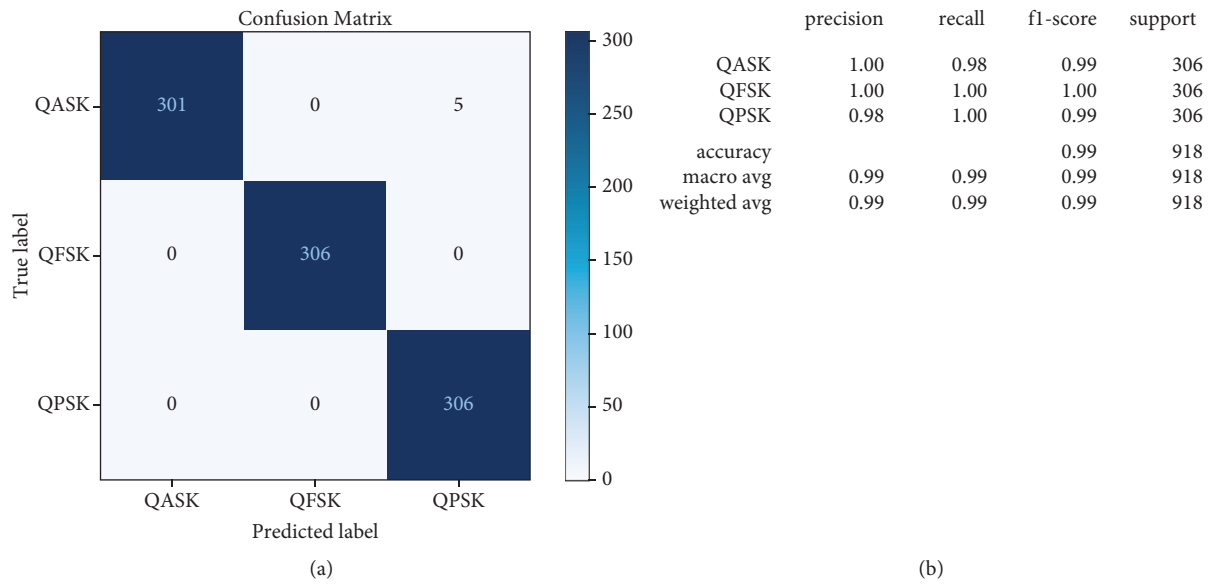


FIGURE 12: Model output for 0.0001 learning rate. (a) Confusion matrices. (b) Classification report.

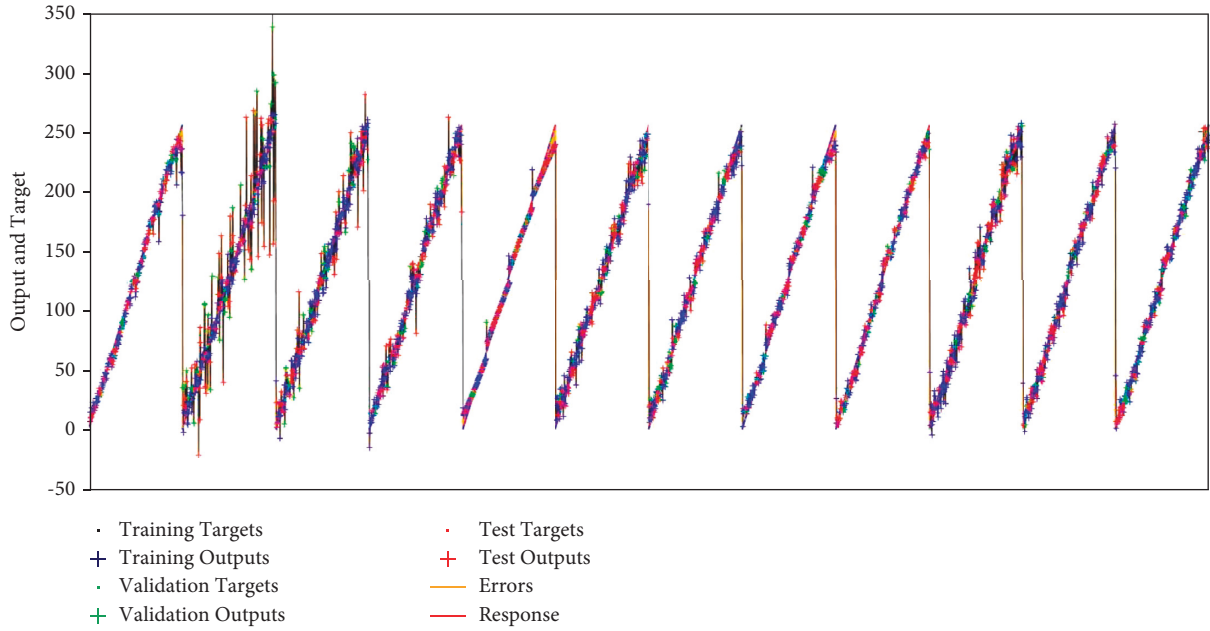


FIGURE 13: Response to output values.

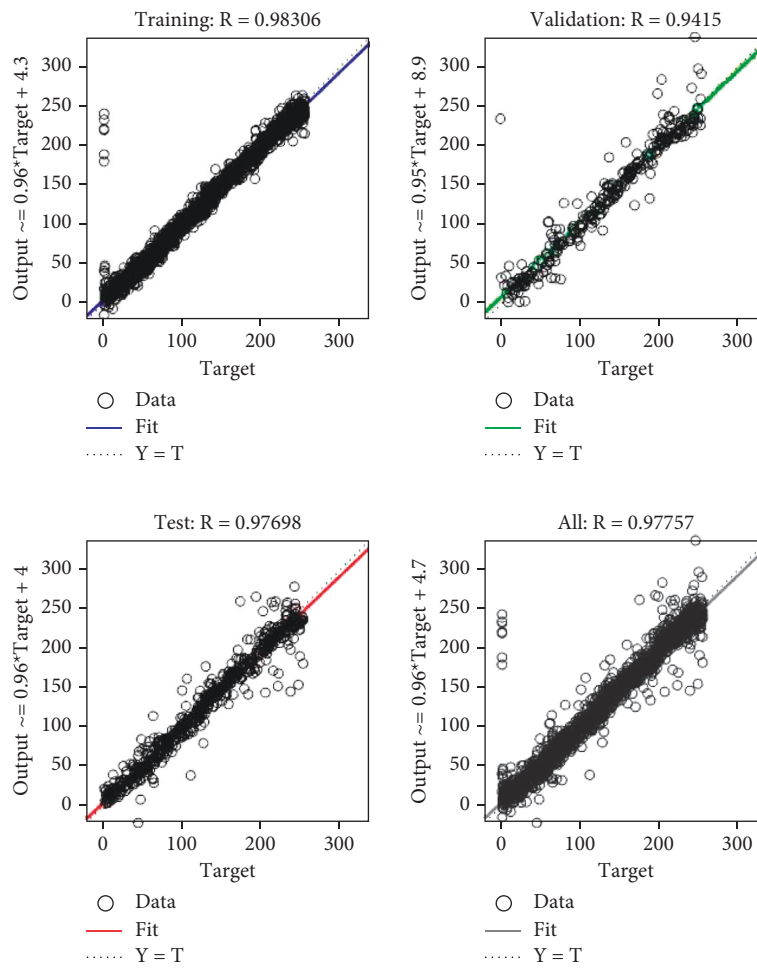


FIGURE 14: Error value graphics of the model's output.

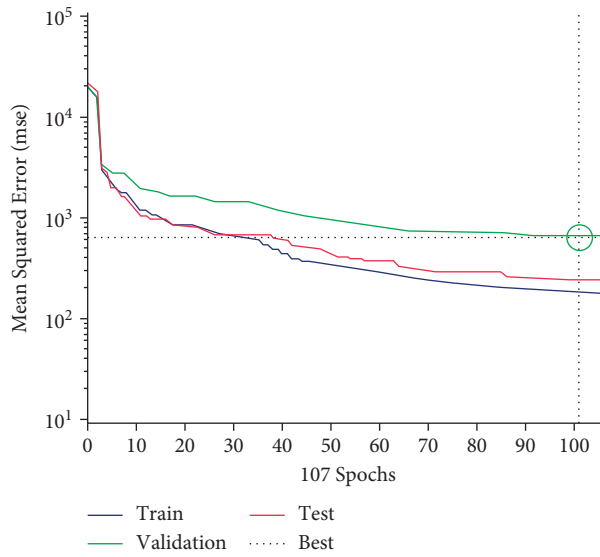
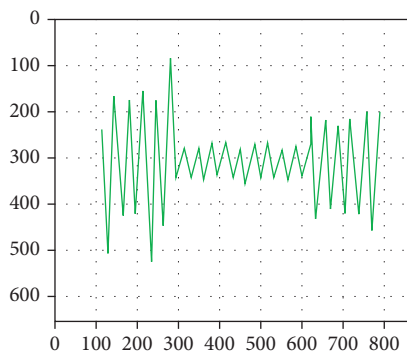


FIGURE 15: Best validation performance at epoch value. During these processes, Intel(R) Core i7-10750H, 16.0 GB RAM, 4 GB GDDR6 GTX1650, and 64-bit Windows 10 PC were used.

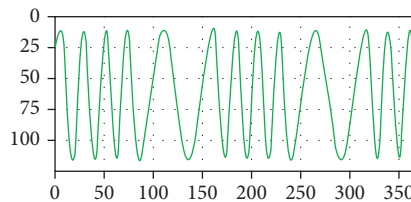
['QASK', 'QFSK', 'QPSK']

[[1. 0. 0.]]



['QASK', 'QFSK', 'QPSK']

[[0. 1. 0.]]



['QASK', 'QFSK', 'QPSK']

[[0. 0. 1.]]

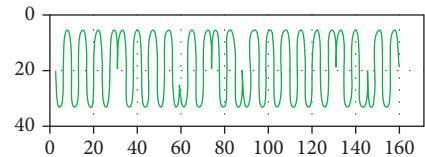


FIGURE 16: Test output of the model.

noise makes it difficult to distinguish the modulation signal type.

While demodulation of noiseless signals during the demodulation phase is easy, the algorithm has been challenged, especially for signals with 5 dB noise. One of the reasons for this is seen as the amount of data in the dataset. Therefore, increasing the amount of data or applying feature extraction can achieve higher success in obtaining the information signal from noisy signals.

### 6. Conclusions

The study used JPEG format images of the signals of digital modulation types QASK, QFSK, and QPSK with 0 dB, 5 dB, 10 dB, and 15 dB noise. The CNN model created using Python libraries has also been successful in 5 dB noisy signals. The model can be used for communication systems because it is challenging for the human eye to distinguish the images of signals with a 5 dB noise ratio, thanks to the

success rate of up to %99, the model can be used for communication systems. In other words, even if the amplitude, frequency, and phase values of the digital modulation signal are unknown, the signal's type can be easily understood.

Most of the studies for AMC are made from actual data of modulation signals. However, it may not always be possible to have all the data of a modulation signal. Modulation classification and demodulation, making it independent of a snapshot and its structure, size, shooting angle, or different contrasts, can provide great convenience. Therefore, modulation classification from images has yielded successful results thanks to this model without extra hardware or software. It has been demonstrated that the model has high success in terms of both training time and accuracy.

As a result of demodulation using a single algorithm, a beneficial effect has been obtained regarding time and cost. Furthermore, in areas where fast and secure communication

is essential, communication is also critical to security without different digital devices. Thus, the development of the algorithm will benefit users in areas where communication is vital.

## Data Availability

The datasets are available upon request from the authors.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

All authors contributed equally to this work. Additionally, all authors have read and approved the final manuscript and given their consent to publish the article.

## References

- [1] M. Zhang, Y. Zeng, Z. Han, and Y. Gong, "Automatic modulation recognition using deep learning architectures," *IEEE Work. Signal Process. Adv. Wirel. Commun. SPAWC*, IEEE, vol. 2018, pp. 1–5, 2018- June.
- [2] S. Ertürk, *Sayısal Haberleşme*, Birsen Yayınevi, İstanbul, 1st edition, 2005.
- [3] F. Xiong, *Digital Modulation Techniques*, First, Norwood, MA USA, 2000.
- [4] S. Ma, J. Dai, S. Lu et al., "Signal demodulation with machine learning methods for physical layer visible light communications: prototype platform, open dataset and algorithms," *IEEE Access*, vol. 7, pp. 30588–30598, 2019.
- [5] D. Wang, M. Zhang, Z. Li et al., "Modulation format recognition and OSNR estimation using CNN-based deep learning," *IEEE Photonics Technology Letters*, vol. 29, no. 19, pp. 1667–1670, 2017.
- [6] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 431–436, 1998.
- [7] Y. Xu, D. Li, Z. Wang, Q. Guo, and W. Xiang, "A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals," *Wireless Networks*, vol. 25, no. 7, pp. 3735–3746, 2019.
- [8] A. Ali and F. Yangyu, "Unsupervised feature learning and automatic modulation classification using deep learning model," *Physical Communication*, vol. 25, 2017.
- [9] J. H. Lee, K. Y. Kim, and Y. Shin, "Feature image-based automatic modulation classification method using CNN algorithm," *1st Int Conf Artif Intell Inf Commun ICAIIC*, vol. 2019, pp. 560–3, 2019.
- [10] N. Daldal, Z. Cömert, and K. Polat, "Automatic determination of digital modulation types with different noises using Convolutional Neural Network based on time-frequency information," *Applied Soft Computing*, vol. 86, Article ID 105834, 2020.
- [11] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: a survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020.
- [12] R. G. Gallager, *Principles of Digital Communication*, Cambridge University Press, Cambridge uk, 2007.
- [13] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [14] Y. Lecun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the IEEE Int Symp circuits Syst*, pp. 253–256, IEEE, Paris, France, 30 June 2010.
- [15] Y. Bengio and Y. Lecun, *Convolutional Networks for Images Speech and Time-Series*, MIT Press, Cambridge, MA, USA, 1997.
- [16] H. C. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [17] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [18] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.
- [20] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Proc 25th Int Conf Neural Inf Process Syst*, vol. 25, no. 2, 2012.
- [21] A. V. Joshi, *Machine Learning and Artificial Intelligence*, Springer, Salmon Tower NY USA, 2020.
- [22] Y. T. Zhou and R. Chellappa, "Computation of Optical Flow Using a Neural Network," in *Proceedings of the IEEE 1988 International Conference on Neural Networks*, 24 July 1988.
- [23] W. Ballard, *Hands-On Deep Learning for Images with TensorFlow Build Intelligent Computer Vision Applications Using TensorFlow and Keras*, Packt Publishing Ltd, Birmingham, UK, 2018.
- [24] A. Zhang, Z. C. Lipton, M. Li, and J. Smola, *Dive into Deep Learning*, 2021, <https://arxiv.org/abs/2106.11342>.
- [25] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf*, vol. 1, pp. 655–665, 2014.
- [26] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2019.
- [27] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Lepton spectra as a measure of b quark polarization at LEP," *Journal of Machine Learning Research*, vol. 15, p. 1929, 2014.
- [29] T. Tsung-Nan Lin, C. L. Giles, B. G. Horne, and S. Y. Kung, "A delay damage model selection algorithm for NARX neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2719–2730, 1997.
- [30] L. Gonzaga, B. Ruiz, M. P. Cuéllar, M. D. Calvo-flores, M. Del, and C. Pegalajar, "An application of non-linear autoregressive neural public buildings," *Energies*, vol. 9, pp. 1–21, 2016.
- [31] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks," *Neurocomputing*, vol. 73, no. 13-15, pp. 2540–2553, 2010.

- [32] Z. Boussaada, O. Curea, A. Remaci, H. Camblong, and N. M. Bellaaj, "A non-linear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation," *Energies*, vol. 11, 2018.
- [33] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA*, pp. 806–813, March 2014.
- [34] H. Wang and B. Raj, *On the Origin of Deep Learning*, pp. 1–72, 2017, <https://arxiv.org/abs/1702.07800>.
- [35] D. P. Kingma and J. L. Ba, "ADAM: a method for stochastic optimization," in *Proceedings of the 3rd Int Conf Learn Represent*, DBLP, San Diego CA USA, 7 May 2015.
- [36] C. Farabet, *Towards Real-Time Image Understanding with Convolutional Networks* Clement, Université Paris-Est, Marne, France, 2013.
- [37] P. Meletis and G. Dubbelman, "On boosting semantic street scene segmentation with weak supervision," *IEEE Intell Veh Symp Proc*, vol. 2019, pp. 1334–1339, 2019.
- [38] B. Schölkopf and A. J. Smola, *Learning with Kernels Support Vector Machines, Regularization, Optimization, and beyond*, The MIT Pres, Cambridge, MA UK, 2001.