Research Article

Deep Learning-Based Signal-To-Noise Ratio Estimation Using Constellation Diagrams

Xiaojuan Xie ⓘ,[1] Shengliang Peng ⓘ,[1] and Xi Yang ⓘ[2]

[1]*College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China*
[2]*College of Information Science and Engineering, Jishou University, Jishou 416000, China*

Correspondence should be addressed to Shengliang Peng; peng.shengliang@gmail.com

Signal-to-noise ratio (SNR) estimation is a fundamental task of spectrum management and data transmission. Existing methods for SNR estimation usually suffer from significant estimation errors when SNR is low. This paper proposes a deep learning (DL) based SNR estimation algorithm using constellation diagrams. Since the constellation diagrams exhibit different patterns at different SNRs, the proposed algorithm achieves SNR estimation via constellation diagram recognition, which can be easily handled based on DL. Three DL networks, AlexNet, InceptionV1, and VGG16, are utilized for DL based SNR estimation. Experimental results show that the proposed algorithm always performs well, especially in low SNR scenarios.

1. Introduction

Signal-to-noise ratio (SNR), which defines the difference in level between the signal and the noise, is one of the most important parameters for spectrum management [1], channel resource allocation [2, 3], transmission power control [4], and adaptive modulation and coding [5].

Based on whether auxiliary data is needed, existing SNR estimation methods can be divided into two groups, data-aided (DA) algorithms [6] and non-data-aided (NDA) algorithms [7]. The former are able to achieve high estimation accuracy but have to transmit a pilot sequence [8]. The latter require little prior information about the signal and increasingly become prevalent choices.

For NDA algorithms, an estimator based on the second- and fourth-order moments (M_2M_4) is used for SNR estimation in [9]. Higher-order moments are exploited to estimate SNR in multiple antenna systems according to [10]. In [11], sixth-order statistics based SNR estimation is proposed for the signals with two different amplitude levels. SNR estimation based on metric normalization frequency in Viterbi decoder is discussed in [12]. SNR of orthogonal frequency division multiplexing signal without the knowledge of any pilot sequences is estimated by the use of cyclic prefix in [13]. In [14], how to use a look-up table for SNR estimation based on rank discrimination test is depicted. In [15], a modified SNR estimation algorithm based on singular value decomposition is investigated. In order to alleviate the impact of signal leakage that impairs the performance of SNR estimation, a weighting operator is suggested to predict and retrieve the leaked signal in [16]. Although these algorithms do not require pilot sequences, they usually suffer from significant estimation errors, especially when SNR is very low.

In recent years, with the prosperity of deep learning (DL), DL based communications techniques have attracted great attention. In [17], an intelligent eye-diagram analyzer based on convolutional neural networks (CNNs) is constructed for performance monitoring in optical communications. In [18, 19], DL based mobile traffic classification is discussed. In [20], multiple input multiple output detection using deep neural networks is investigated. Some applications of CNNs in wireless communications are summarized in [21]. Due to the fact that the constellation diagrams of different modulation types are markedly different, [22] utilizes CNNs to recognize constellation diagrams and achieve DL based modulation classification.

Note that constellation diagrams also exhibit different patterns at different SNRs. When SNR is higher, the points on constellation diagrams are more concentrated. Otherwise, the points are more scattered. Therefore, DL based SNR estimation may be similarly achieved via constellation diagram recognition.

This paper focuses on DL based SNR estimation using constellation diagrams. The issue of SNR estimation is converted into a problem of constellation diagram recognition and solved by DL. The proposed algorithm comprises two stages. In the off-line training stage, a CNN model is trained by the use of a large number of constellation diagrams labeled by the SNR values. In the online estimating stage, the received signal is preprocessed into a constellation diagram and fed into the trained CNN model, with which the SNR of the received signal is estimated.

The rest of this paper is organized as follows. Section 2 formulates the signal model as well as the SNR estimation task. Section 3 introduces two traditional algorithms for SNR estimation. Section 4 illustrates the detailed steps of the DL based SNR estimation algorithm. Experiments results are provided in Section 5. Finally, Section 6 concludes this paper.

2. Problem Formulation

In communications systems, SNR estimation is usually performed by the receiver, where the received signal at the kth observation can be generally expressed as

$$y_k(n) = h_k(n) \cdot s_k(n) + \omega_k(n), \quad n = 1, 2, \ldots, L, \qquad (1)$$

where $h_k(n)$ represents the channel impact, $s_k(n)$ is the transmitted signal, and $\omega_k(n)$ stands for the additive noise with zero mean and follows the Gaussian distribution. $h_k(n)$, $s_k(n)$ and $\omega_k(n)$ are assumed to be uncorrelated to each other and stationary. For the nonstationary signals, one possible solution is to split the receiving period into multiple slots in time domain. Each slot is very short, and the signal in each slot is approximately stationary. In a detailed way, given a signal observation composed of L samples $\mathbf{y}_k = [y_k(1), y_k(2), \ldots, y_k(L)]^T$, if $\mathbf{y}_k$ is nonstationary, we can split it into P segments,

$$\mathbf{y}_{k,p} = \left[y_k\left(1 + \frac{(p-1)L}{P}\right), y_k\left(2 + \frac{(p-1)L}{P}\right), \ldots, y_k\left(\frac{p \cdot L}{P}\right) \right]^T, \quad 1 \le p \le P. \qquad (2)$$

Since the length of $\mathbf{y}_{k,p}$ is much shorter than $\mathbf{y}_k$, each segment $\mathbf{y}_{k,p}$ is approximately stationary and can be used for SNR estimation. Our task is to estimate SNR of the received signal, which is defined by

$$\gamma_k = \frac{E\left\{\left|h_k(n) \cdot s_k(n)\right|^2\right\}}{E\left\{\left|\omega_k(n)\right|^2\right\}}. \qquad (3)$$

The signal may be modulated according to various modulation types, such as binary-phase shift keying (BPSK), quadrature-phase shift keying (QPSK), eight-phase shift keying (8PSK), and sixteen-phase shift keying (16PSK). The M-ary PSK modulated signal can be expressed as

$$s_k(n) = e^{j\theta_n}, \qquad (4)$$

where θ_n is one of the M phases spaced evenly around the unit circle. Taking the BPSK modulated signal for example, the amplitude of $s_k(n)$ is always 1 ($|s_k(n)| = 1$) and the phase is randomly either 0 or π. Figure 1 shows both the amplitude and the phase of the BPSK modulated signal. Moreover, the corresponding received signal corrupted by the additive white Gaussian noise (AWGN) channel at SNR = 6 dB is also depicted.

For each type of modulation, constellation diagrams of the received signal are obviously different at different SNRs, as shown in Figure 2. In a detailed way, the sample points on constellation diagrams are more scattered when SNR is lower, and it is more concentrated when SNR is higher, which motivates us to perform SNR estimation via constellation diagram recognition.

Furthermore, the signal may be corrupted by various channels. Three channels, including AWGN channel, Rician fading channel, and Rayleigh fading channel, are considered in this paper. Figure 3 shows the constellation diagrams of QPSK modulation at different SNRs under three channels. Likewise, for each channel, constellation diagrams exhibit different patterns at different SNRs, and thus SNR estimation using constellation diagrams can be performed regardless of channel conditions.

3. Traditional Algorithms for SNR Estimation

3.1. M_2M_4 Based SNR Estimation.
The derivation of M_2M_4 based algorithm provided in [9] for complex channel is described below; the second moment and the fourth moment of received signal $y_k(n)$ can be calculated as

$$M_{2,k} = E\left\{\left|y_k(n)\right|^2\right\} \approx \frac{1}{L} \sum_{n=1}^{L} y_k(n) y_k^*(n),$$

$$M_{4,k} = E\left\{\left|y_k(n)\right|^4\right\} \approx \frac{1}{L} \sum_{n=1}^{L} \left[y_k(n) y_k^*(n)\right]^2. \qquad (5)$$

For any M-ary PSK signal, the SNR can be estimated by

$$\widehat{\gamma}_k = \frac{\sqrt{2M_{2,k}^2 - M_{4,k}}}{M_{2,k} - \sqrt{2M_{2,k}^2 - M_{4,k}}}. \qquad (6)$$
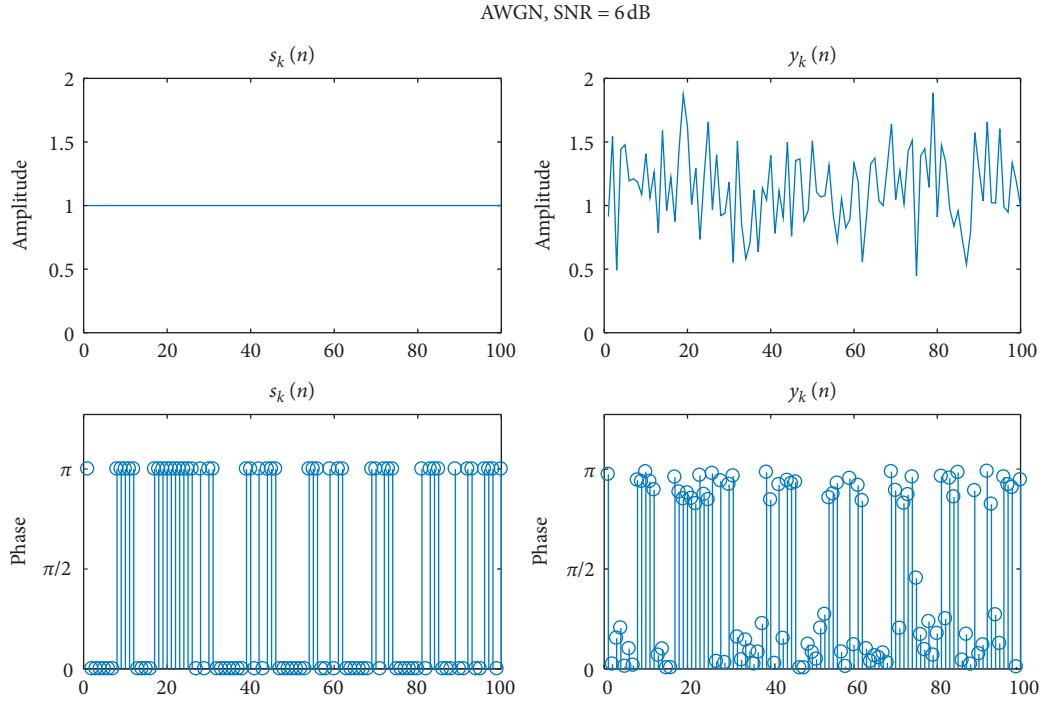
AWGN, SNR = 6 dB

FIGURE 1: Amplitude and phase of BPSK modulated signal as well as the received signal under AWGN channel at SNR = 6 dB.
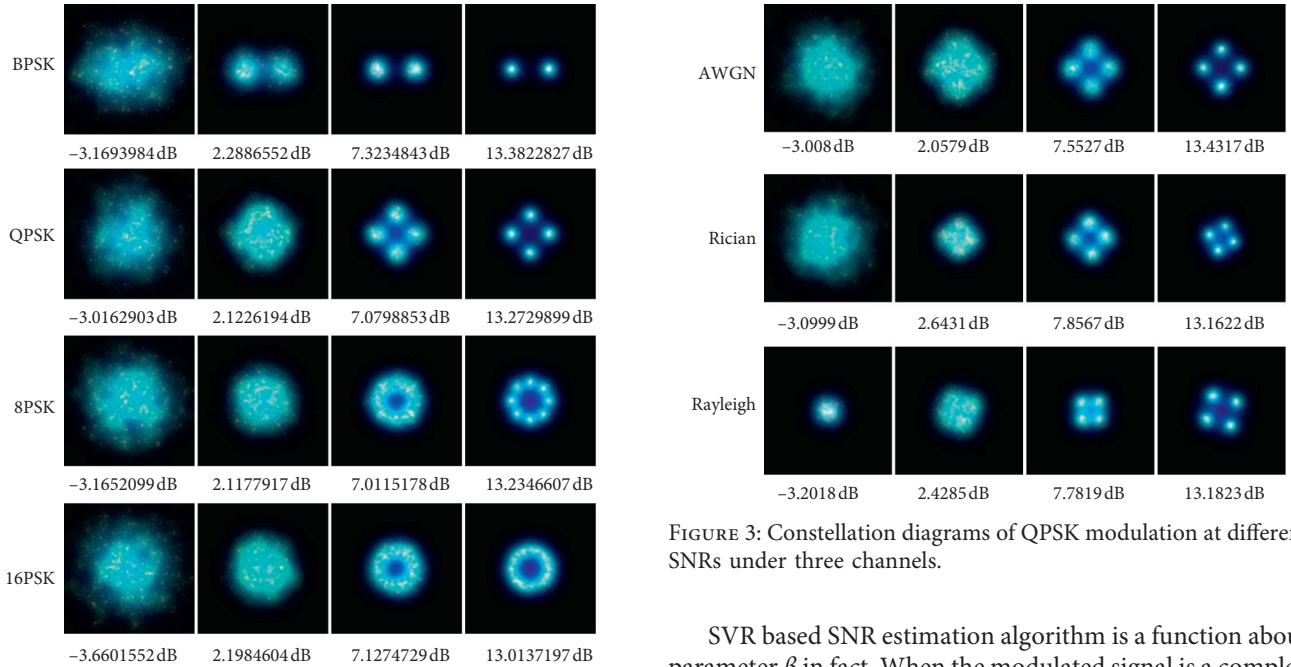

FIGURE 2: Constellation diagrams of four modulation types at different SNRs under AWGN channel.

FIGURE 3: Constellation diagrams of QPSK modulation at different SNRs under three channels.

3.2. SVR-Based SNR Estimation.

A parameter named signal-to-variation ratio (SVR) based on moments is proposed in [23], which is developed to monitor channel quality. In [9], SVR based SNR estimation algorithm is derived in detail under AWGN channel.
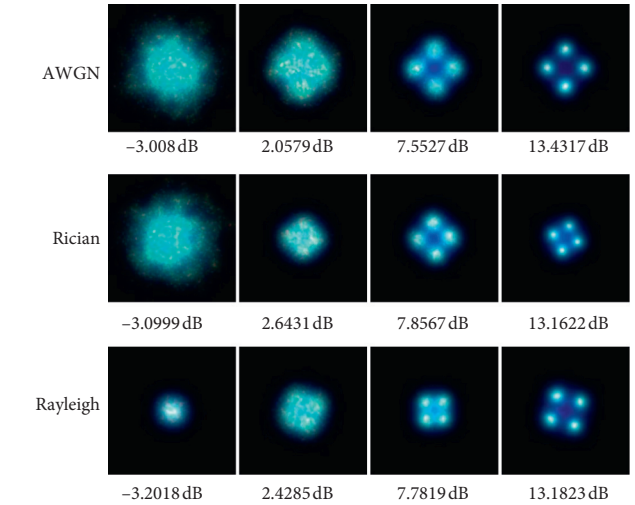
SVR based SNR estimation algorithm is a function about parameter β in fact. When the modulated signal is a complex signal, β is defined as

$$\beta = \frac{E\{y_k(n)y_k(n)^* y_k(n-1)y_k(n-1)^*\}}{E\{(y_k(n)y_k(n)^*)^2\} - E\{y_k(n)y_k(n)^* y_k(n-1)y_k(n-1)^*\}}. \tag{7}$$

For any M-ary PSK signal, the SNR can be estimated by

$$\hat{\gamma}_k = \beta - 1 + \sqrt{\beta(\beta-1)}. \tag{8}$$

4. DL Based SNR Estimation

Since DL is a powerful tool for image recognition, this section utilizes DL to recognize constellation diagrams and achieve SNR estimation. The algorithm comprises two stages, as shown in Figure 4. The off-line training stage has four steps, including signal generation, signal preprocessing, data labeling, and network training, in which massive signal observations with known SNRs are generated and utilized to train a network model. The online estimating stage has two steps, including signal preprocessing and SNR inferring, in which a signal observation with unknown SNR is preprocessed and fed into the trained model for SNR estimation.
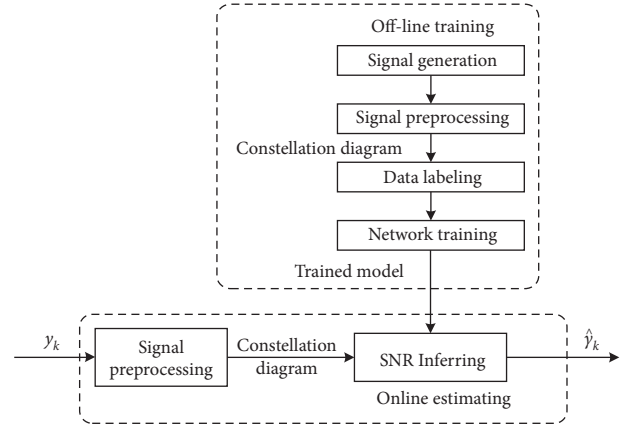
4.1. Signal Generation.

Considering a type of modulation, a large amount of bits are generated and modulated. Then, the modulated signals are transmitted over either AWGN, Rician fading, or Rayleigh fading channel. The power of modulated signals is assumed to be unit, and that of noise is randomly configured, producing massive signal observations with random SNRs following a uniform distribution.

4.2. Signal Preprocessing.

The purpose of signal preprocessing is to convert each signal observation into a constellation diagram. Various formats of constellation diagram can be exploited, and this paper uses three-channel constellation image stated in [22]. In a detailed way, L samples of the signal observation are mapped into a 7×7 complex plane. An exponential decay model is applied to alleviate the information loss that resulted from signal mapping. The three-channel constellation image consists of three constellation diagrams derived from the same signal observation but different exponential decay rates. Finally, all images are resized to a proper resolution according to the input layer of DL network.

4.3. Data Labeling.

Supervised learning is conducted to recognize constellation diagrams. In supervised learning, labeling is necessary to pair the input training data with the desired output labels. For the task of DL based SNR estimation, the constellation diagrams of generated signals are labeled by the corresponding SNR values, which can be obtained according to the powers of signal and noise in the step of signal generation.

4.4. Network Training.

CNN is one of the representative DL networks to process image data [24]. A complete CNN consists of one input layer, one output layer, and multiple hidden layers. Three CNNs, including AlexNet, InceptionV1, and VGG16, are selected for network training in this paper.

4.4.1. AlexNet.

AlexNet, as the winner of the ImageNet Large Scale Visual Recognition Challenge-2012 (ILSVRC-2012) competition, sparked a craze for DL. AlexNet consists

FIGURE 4: Algorithm flowchart for DL based SNR estimation.

of five convolutional layers and three fully connected layers, with a 1000-way softmax layer and a total of 62378344 parameters. There are several reasons for AlexNet's breakthrough.

AlexNet adopts the Rectified Linear Unit (ReLU) to greatly shorten the training time. It is a piecewise linear function, and the value of ReLU is 0 when the input is less than or equal to 0, and it is the input itself if the input is greater than 0. ReLU can make the output of partial neurons zero, which contributes to the sparsity of the network. At the same time, the codependence of parameters is reduced, and the overfitting problem can be efficiently alleviated.

The introduction of Dropout effectively alleviates overfitting. Dropout reduces the codependence between nodes by randomly zeroing some weights that the hidden layer outputs, so as to achieve regularization of the neural network. The computation cost can be slashed as well.

When the training data is limited, some new data can be generated from the existing training data set through some transformations to rapidly expand the training data. AlexNet randomly cropped an image from 256×256 to 227×227 and then flipped it horizontally.

The principle of Local Response Normalization (LRN) is to mimic the biologically active neurons' inhibition of neighboring neurons (lateral inhibition), which is simply the normalization of neurons in the same location among different feature map layers. The value is proportional to the size of the neuron, thus increasing the generalization ability of the network. In essence, LRN layer is designed to prevent the saturation of activation function.

4.4.2. InceptionV1.

InceptionV1, also known as GoogLeNetV1, is an efficient CNN classification model proposed by Google in the ILSVRC-2014 competition. Inception module is proposed to increase the depth and width of the network while maintaining computational costs. In order to reduce the thickness of the feature map, the 1×1 convolution kernel is added separately before the 3×3 as well as 5×5 convolution kernels and after the max pooling layer. In addition, global average pooling layer is utilized to replace the fully connected layer inspired by Network In Network, which
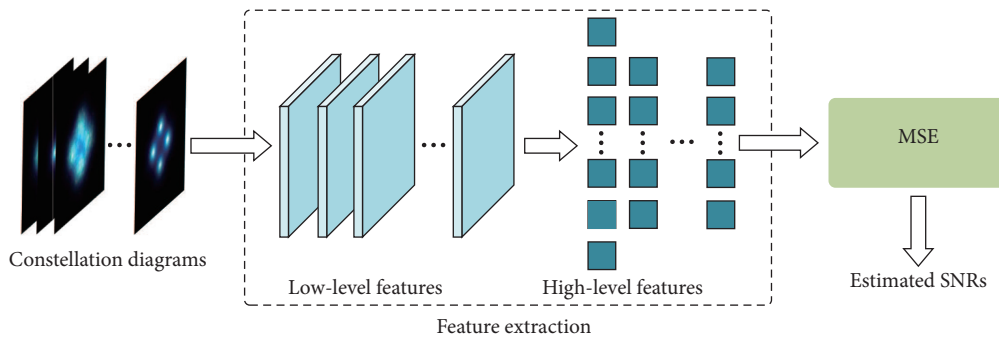
FIGURE 5: Network training in regression tasks.

saves computation overhead a lot, effectively alleviating overfitting. As a 22-layer CNN, only 5000000 parameters are owned, and its input size is 224×224.

4.4.3. VGG16. The core of VGG lies in the repeated use of 3×3 convolution kernels and the 2×2 pooling layers to increase the depth of the network. The VGG16 adopted in this paper consists of 13 convolution layers and 3 fully connected layers. One improvement of VGG16 over Alex-Net is to replace the larger convolution kernels in AlexNet (11×11, 5×5) with successive 3×3 convolution kernels. The input size of VGG16 is 224×224, and a total of 138M parameters lead to a long training time.

Note that, different from conventional image classification tasks, DL based SNR estimation is a regression task. In order to handle this task, some modifications on the CNNs are necessary. As depicted in Figure 5, the mean squared error (MSE) is utilized instead of the cross-entropy function as the loss function, and the loss value is employed instead of the accuracy value as the monitor index.

In order to train the networks above, labeled constellation diagrams are fed according to supervised learning. Partial settings of network training are listed in Table 1. The training is conducted on a computing server configured with Ubuntu 16.04, Keras, CUDA, Intel i7 8700, and Nvidia GTX 1080Ti. AlexNet, InceptionV1, and VGG16 take about 2, 9, and 20 hours, respectively, to obtain a trained network model.

4.5. SNR Inferring. In online estimating stage, the received signal observation $\mathbf{y}_k$ with unknown SNR is firstly preprocessed into a constellation diagram similar to that in offline training stage. After that, according to the step of SNR inferring, the constellation diagram is fed into a trained network model derived from the off-line training stage to infer the estimated result $\hat{\gamma}_k$.

5. Experiments Results

5.1. Experiment Settings. In experiments, the signal may be modulated according to different modulation types and corrupted by different channels. For each modulation type and channel, $K = 2000$ testing observations with random SNRs uniformly distributed in a closed interval from $-4\,\mathrm{dB}$ to $14\,\mathrm{dB}$ are utilized. Each testing observation is composed of

TABLE 1: Partial settings of network training.

Parameters	Values
Activation function	Linear
Loss function	MSE
Monitor index	Loss
Batch size	128
Epochs	50

$L = 1000$ samples and converted into a constellation diagram. The resolution of constellation diagrams is 227×227 when utilizing AlexNet and is 224×224 when utilizing InceptionV1 and VGG16. Both the Rayleigh and Rician fading channels are constructed by filters, where the sampling interval is $T_s = 10^{-6}\,\mathrm{s}$, the maximum Doppler shift is $F_d = 50\,\mathrm{Hz}$, the vector of path time delays is $\tau = [0, 0.004, 0.008, 0.012]\,\mathrm{s}$, the vector of average path gains is $P_{\mathrm{dB}} = [0, -3, -6, -10]\,\mathrm{dB}$, and the Rician factor is $\kappa = 3$. Moreover, $M_2 M_4$ based and SVR based SNR estimation algorithms are adopted for comparison.

In our task, the signal and the noise are possibly confused with each other. When mistaking the signal for the noise, the estimated SNR value becomes smaller, causing a negative estimation error. On the contrary, when mistaking the noise for the signal, the estimated value becomes larger, producing a positive estimation error. This paper quantifies the estimation errors using MSE. Based on the estimated SNR $\hat{\gamma}_k$, MSE can be calculated as follows:

$$\mathrm{MSE}(\hat{\gamma}_k, \gamma_k) = \frac{1}{K} \sum_{k=1}^{K} (\hat{\gamma}_k - \gamma_k)^2. \tag{9}$$

5.2. Performance Comparison. Figure 6 shows the MSEs of different algorithms versus SNR for QPSK modulation under AWGN channel. According to this figure, the MSEs of DL based algorithms, including AlexNet, InceptionV1, and VGG16, are all smaller than 0.055 throughout the SNR region. However, the $M_2 M_4$ based algorithm exhibits extremely high MSE when SNR is low, and so does the SVR based algorithm. Therefore, no matter which CNN is utilized, DL based SNR estimation algorithm is far better than the traditional algorithms, especially under low SNR scenarios.
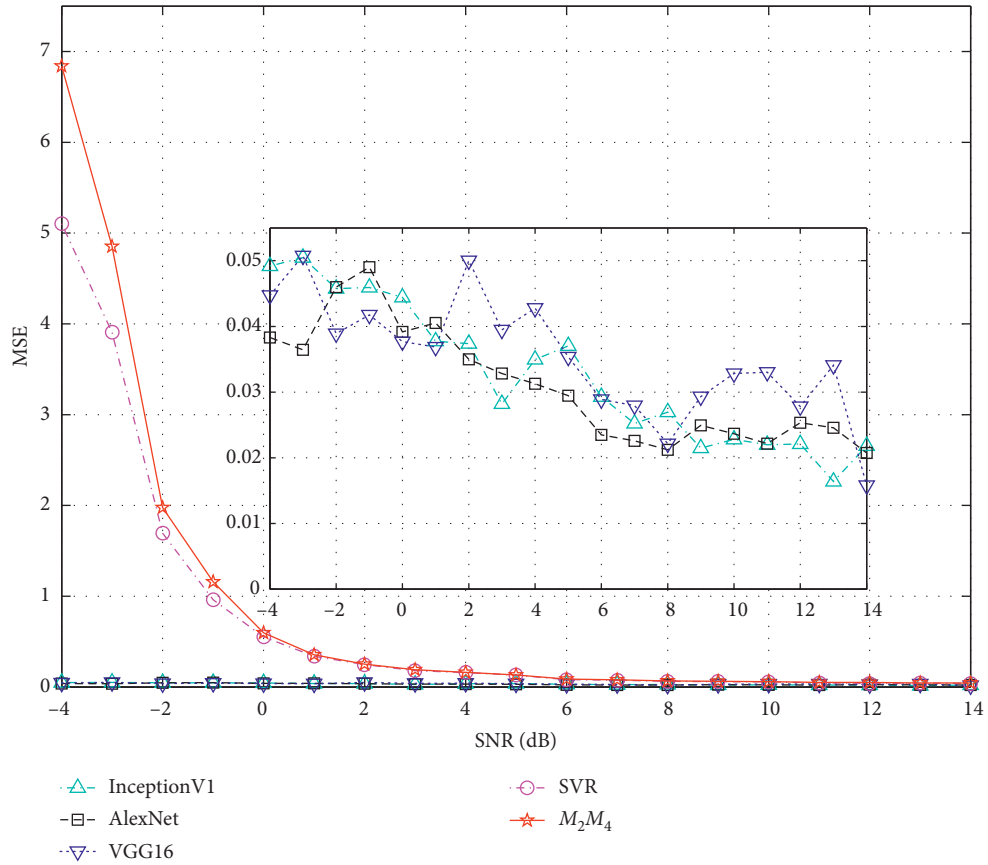
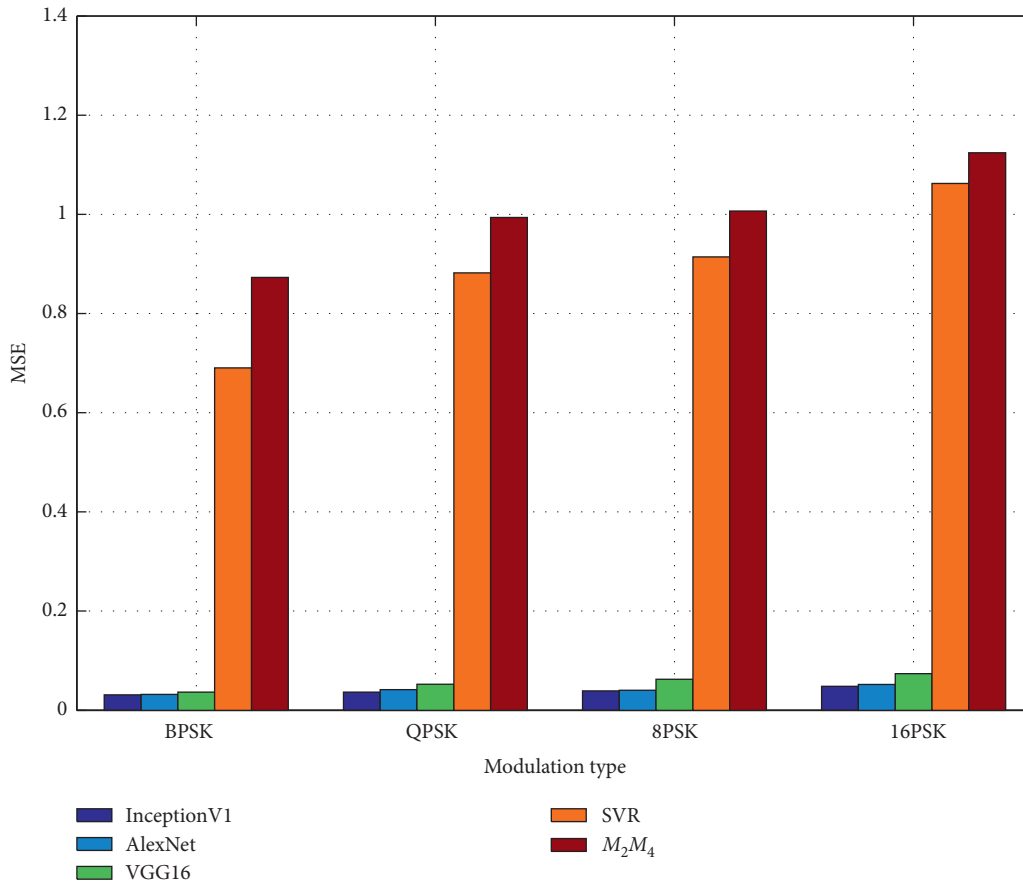FIGURE 6: MSEs of different algorithms for QPSK modulation under AWGN channel.

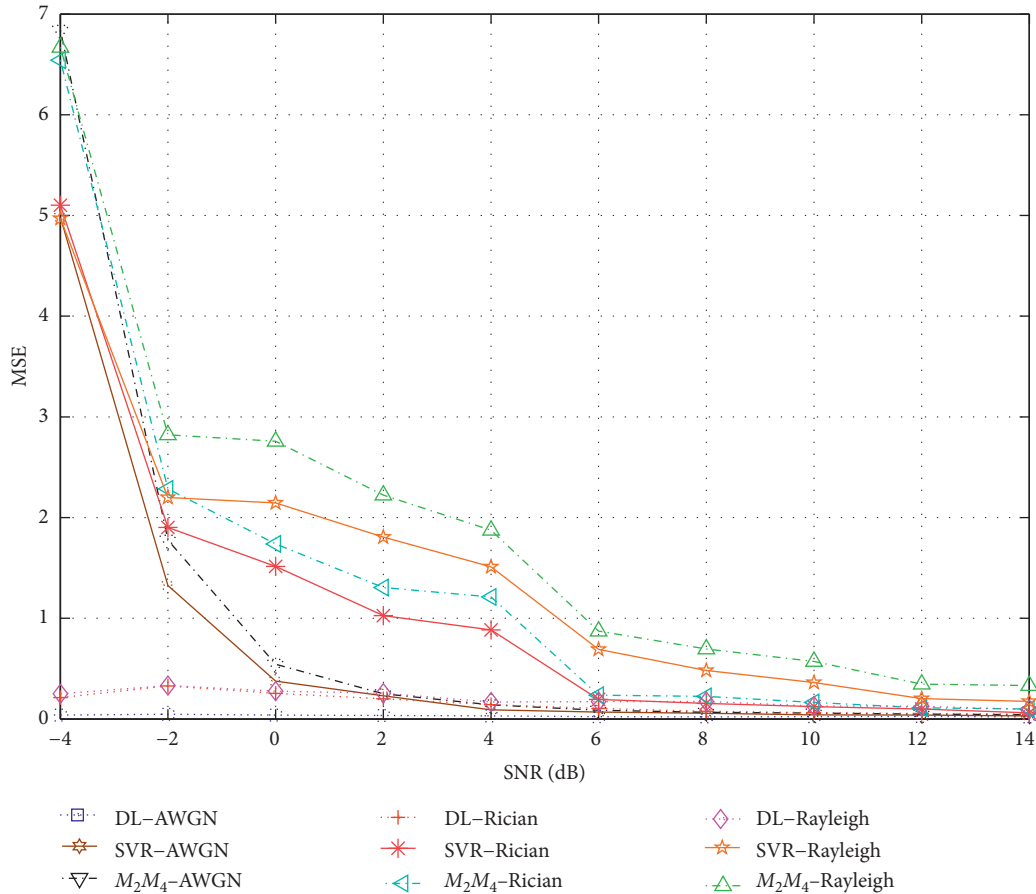FIGURE 7: Averaged MSEs of different modulation types under AWGN channel.

FIGURE 8: MSEs of two algorithms versus SNR under different channels.

5.3. Impacts of Modulation Types.

In order to demonstrate the impacts of modulation types on estimation accuracy, Figure 7 shows the MSEs of BPSK, QPSK, 8PSK, and 16PSK averaged in SNR under AWGN channel. For each estimation algorithm, MSE increases with the increase of modulation order, but the increment is not significant. The gap of performance between the M_2M_4 based algorithm and the SVR based algorithm narrows with the increase of modulation order. Furthermore, no matter which modulation type is adopted, the MSEs of three DL based algorithms are always lower than those of traditional algorithms. Among DL based algorithms, InceptionV1 performs slightly better than AlexNet and VGG16 and, thus, has the best estimation accuracy.

5.4. Impacts of Channels.

Besides AWGN channel, similar experiments are conducted under Rayleigh and Rician fading channels. InceptionV1 is taken as an example of DL based SNR estimation algorithm. Figure 8 shows the MSEs of two algorithms versus SNR under different channels. It can be seen from this figure that both algorithms perform the best under AWGN channel and the worst under Rayleigh fading channel. Compared with the traditional algorithms, DL based algorithm is less affected by the channel and always superior regardless of channel conditions. In addition, the difference of MSEs between the M_2M_4 based and the SVR

TABLE 2: Comparison on running time.

Algorithms		Running time
M_2M_4 based		21.11 ms
SVR based		16.9 ms
DL based	InceptionV1	12.46 ms
	VGG16	5.40 ms
	AlexNet	1.58 ms

based algorithm is the largest under Rayleigh fading channel, which decreases as the channel environment gets better, and the smallest under AWGN channel.

5.5. Complexity Analysis.

To analyze the computational complexity of different algorithms, Table 2 records their average running time of handling one testing observation. Note that the traditional algorithms are computed with a central processing unit (CPU) of Intel i7 8700, while DL based algorithms are computed with a graphic processing unit (GPU) of Nvidia GTX 1080Ti. As shown in Table 2, traditional algorithms have the longer running time, especially the M_2M_4 based algorithm. Benefiting from the parallel computing capability of GPU, the running time of DL based algorithms is reduced. Among the three implementations of DL based algorithms, VGG16 consumes more running time than AlexNet and less running time

than InceptionV1. It is because the network depth of AlexNet is the smallest, and that of InceptionV1 is the largest.

6. Conclusion

This paper proposes a DL based SNR estimation algorithm using constellation diagrams. The received signal is converted into a constellation diagram and recognized by AlexNet, InceptionV1, and VGG16 for SNR estimation. No matter which modulation is adopted and no matter which channel is considered, the proposed algorithm is always superior to the traditional algorithms, especially in low SNR scenarios. Moreover, the proposed algorithm consumes less running time and can be effectively calculated with GPU.

Our algorithm is characterized by the combination of constellation diagram representation and CNN based regression. The former can be explored to deal with similar communication problems, e.g., channel recognition and interference analysis. The latter is also beneficial for other parameters estimation tasks in communications systems, e.g., frequency offset estimation and time delay estimation.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest in the work.

Acknowledgments

References

[1] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2209–2221, 2013.

[2] S. Xie, B. Zhang, D. Guo, and B. Zhao, "Performance analysis and power allocation for NOMA-based hybrid satellite-terrestrial relay networks with imperfect channel state information," *IEEE Access*, vol. 7, pp. 136279–136289, 2019.

[3] J. C. S. Santos Filho, D. C. González, A. Wolf, L. L. Mendes, M. D. Yacoub, and G. Fettweis, "SNR-aware power allocation scheme for lossy-forward relaying systems," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 1018–1021, 2018.

[4] W. Xiang, M. Johnston, and S. Le Goff, "Low-complexity power control and energy harvesting algorithms for wiretap channels employing finite-alphabet input schemes," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 318–326, 2018.

[5] M. Torabi, S. Aliasghari, and C. Nerguizian, "Cross-layer design of T-ARQ and adaptive modulation and coding in a spectrum sharing with cooperative relaying system," *IET Communications*, vol. 13, no. 11, pp. 1662–1669, 2019.

[6] S. A. Khan, S. Saleem, S. A. Hassan, and M. U. Ilyas, "An improved data-aided linear estimator of modulation index for binary CPM signals," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 780–784, 2019.

[7] M. B. Ben Salah and A. Samet, "NDA SNR estimation using fourth-order cross-moments in time-varying single-input multiple-output channels," *IET Communications*, vol. 10, no. 11, pp. 1348–1354, 2016.

[8] N. Wu, H. Wang, and J.-M. Kuang, "Maximum likelihood signal-to-noise ratio estimation for coded linearly modulated signals," *IET Communications*, vol. 4, no. 3, pp. 265–271, 2010.

[9] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," *IEEE Transactions on Communications*, vol. 48, no. 10, pp. 1681–1691, 2000.

[10] M. Mohammadkarimi, O. A. Dobre, and M. Z. Win, "Non-data-aided SNR estimation for multiple antenna systems," in *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–5, Washington, DC, USA, December 2016.

[11] R. Lopez-Valcarce and C. Mosquera, "Sixth-order statistics-based non-data-aided SNR estimation," *IEEE Communications Letters*, vol. 11, no. 4, pp. 351–353, 2007.

[12] K.-T. Shr and Y.-H. Huang, "SNR estimation based on metric normalization frequency in Viterbi decoder," *IEEE Communications Letters*, vol. 15, no. 6, pp. 668–670, 2011.

[13] F.-X. Socheleau, A. Aissa-El-Bey, and S. Houcke, "Non data-aided SNR estimation of OFDM signals," *IEEE Communications Letters*, vol. 12, no. 11, pp. 813–815, 2008.

[14] A. W. Azim, S. S. Khalid, and S. Abrar, "Non-data-aided SNR estimation method for APSK exploiting rank discrimination test," *Electronics Letters*, vol. 48, no. 14, pp. 837–839, 2012.

[15] T. Wu, H. Zheng, Di Yan, and S. Liu, "A modified SNR estimation algorithm based on singular value decomposition," in *Proceedings of the 2014 International Conference on Information and Communications Technologies (ICT 2014)*, pp. 1–5, Nanjing, China, May 2014.

[16] Y. Chen and S. Fomel, "Random noise attenuation using local signal-and-noise orthogonalization," *Geophysics*, vol. 80, no. 6, pp. WD1–WD9, 2015.

[17] 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.

[18] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[19] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "MIMETIC: mobile encrypted traffic classification using multimodal deep learning," *Computer Networks*, vol. 165, p. 106944, 2019.

[20] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[21] N. Wu, X. Wang, B. Lin, and K. Zhang, "A CNN-based end-to-end learning framework toward intelligent communication systems," *IEEE Access*, vol. 7, pp. 110197–110204, 2019.

[22] S. Peng, H. Jiang, H. Wang et al., "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 718–727, 2019.

[23] A. L. Brandao, L. B. Lopes, and D. C. McLemon, "In-service monitoring of multipath delay and cochannel interference for indoor mobile communication systems," in *Proceedings of ICC/SUPERCOMM'94—1994 International Conference on Communications*, pp. 1458–1462, New Orleans, LA, USA, May 1994.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.