A Data Preprocessing Method for Automatic Modulation Classification Based on CNN

Haozheng Zhang, Ming Huang, Jingjing Yang, *Member, IEEE*, and Wei Sun

***Abstract*— As a backbone of deep learning models, convolutional neural networks (CNNs) are widely used in the field of automatic modulation classification. Nevertheless, we speculate that the forms of signal samples make them inefficient for direct use as a CNN input. In this letter, a novel data preprocessing method is proposed to markedly improve CNN-based automatic modulation classification. The benchmark dataset used in this research is the well-known RadioML2016.10a dataset. The experimental results show that using the proposed method gains approximately 10% accuracy improvement in a simple CNN. Furthermore, according to the form of the preprocessed data, we designed a CNN with residual blocks to reach a maximum accuracy of 93.7% when the signal-to-noise ratio is 14 dB, which outperforms state-of-the-art automatic modulation classifiers.**

***Index Terms*— Automatic modulation classification, convolutional neural network (CNN), data preprocessing, residual block.**

I. Introduction

Automatic modulation classification (AMC) is a technology designed to classify the modulation scheme of an unknown received signal. As a necessary step to demodulate a detected signal, AMC is considered to be a crucial technology used in various civilian and military communication applications [1]. Algorithms to solve the AMC problem can be generally divided into two categories: likelihood-based (LB) and feature-based (FB) [1], [2]. The LB approach is based on the likelihood function of the received signal, and the LB algorithms are derived from three tests of the likelihood ratio: average likelihood, generalized likelihood, and hybrid likelihood. From Bayes decision theory, the LB algorithms offer an optimal solution to maximize the probability of correct classification [1]–[5]. However, the LB approach suffers from unacceptably high computational complexity. As a suboptimal solution, by comparison, the FB approach is much easier to implement. The FB approach is based on various kinds of features extracted from the received signal; e.g., cyclostationary features [6], high-order cumulants [7], and time-frequency distribution [8].

In recent years, with the development of artificial intelligence, more and more algorithms based on deep learning have been used as the FB approach to solving the AMC problem (e.g., [9]–[13]). Liu *et al.* [9] proposed a simple convolutional neural network (CNN) for AMC and explored the optimal depth of the proposed structure. Ramjee *et al.* [10] presented three high-accuracy models for AMC based on various neural networks, including a convolutional long short-term deep neural network (CLDNN), a long short-term memory (LSTM) neural network, and a deep residual network (ResNet). Rajendran *et al.* [11] proposed a data-driven model based on LSTM to solve the AMC problem. Their experimental results showed that by formatting the input data from rectangular coordinates (IQ data) to polar coordinates (amplitude and phase), simple LSTM models could achieve higher accuracy.

However, among those studies, some key characteristics of CNNs were ignored. It is well known that a CNN focuses on the local correlation of its input, whereas we propose that the features of the modulation scheme be distributed over the entire time series of the signal. Using the signal sample directly as the input of the CNN would lead to a waste of some useful information for modulation classification. On the other hand, note that convolutional layers and pooling layers compose a large part of a typical CNN, and those two types of layers play a key role in the CNN [14]. If the input of a CNN had a too narrow shape in one dimension like a signal sample, it would be easy to infer that the kernel size of the convolutional layers and the pool size of the pooling layers would be severely restricted; we postulate that this restriction would negatively affect the performance of the CNN. On the basis of this point of view, we assumed that preprocessing the signal data properly before inputting would improve the performance of a CNN in the AMC problem. Some algorithms transform the signals into color images (see, e.g., [12], [13]), which could avoid the negative effects caused by unsuitable input shapes for CNN-based models. Nevertheless, the high cost of image generation and storage greatly increases the difficulty of their implementation in practical applications.

In this letter, we propose a novel data preprocessing method (DPM) to address the above problems. Relative to direct use, using preprocessed signals as the inputs of a simple CNN gains a marked improvement in the classification accuracy rate. Subsequently, to further improve the classification accuracy, we specially designed a CNN with residual blocks based on the form of the preprocessed data. The rest of this letter is organized as follows: Section II describes the signal model adopted and the details of the proposed DPM including an analysis of its advantages. In Section III, the structure of the proposed network and a fine-tuned simple CNN are introduced. Section IV shows the experimental results, and Section V concludes the letter.

II. Signal Model and the Proposed DPM

A. Signal Model

As a typical signal model described in [15], the received signal can be expressed as

$$y(t) = h(t) * x(t) + n(t) \tag{1}$$

Manuscript received November 14, 2020; accepted December 7, 2020. Date of publication December 14, 2020; date of current version April 9, 2021. This work was funded by the National Natural Science Foundation of China (Grant Nos. 61863035, 61963037, and 11564044). The associate editor coordinating the review of this letter and approving it for publication was M. Chafii.
(Corresponding author: Ming Huang.)

The authors are with the School of Information Science and Engineering, Yunnan University, Kunming 650091, China (e-mail: zhanghaozheng@mail.ynu.edu.cn; huangming@ynu.edu.cn; yangjingjing@ynu.edu.cn; sunwei2018ynu@163.com).

Digital Object Identifier 10.1109/LCOMM.2020.3044755

where $n(t)$ is the channel interference commonly regarded as additive white Gaussian noise with a zero mean, $h(t)$ and $*$ are the channel impulse response and the convolution operation respectively, $x(t)$ is the modulated signal, and $y(t)$ is the received signal. Modulation classification can be interpreted as identifying the modulation scheme of $x(t)$ only by the received $y(t)$. The signal sample matrix Y_{dts} is generated from a discrete time sampling of $y(t)$ and can be expressed as

$$Y_{dts} = [s_1, s_2, \ldots, s_l, \ldots, s_L],\ 1 \le l \le L \qquad (2)$$

where L is the number of sample points and s_l is the l-th sample point of Y_{dts}. Because the signal samples are IQ samples, s_l can be expressed as a vector:

$$s_l = [I_l, Q_l]^T \qquad (3)$$

where A^T represents the transposed A, I_l and Q_l are the real value and imaginary value of the sample point s_l respectively. Therefore, the size of the matrix Y_{dts} is $2 \times L$. In [9] and [10], Y_{dts} is the input of CNN-based modulation classifiers.

B. The Proposed DPM

Hypothesize that $x(t)$ is a digital modulation signal, Y_{dts} comprises M symbols $\{y_m\}_{m=1}^{M}$, and each symbol y_m consists of B sample points $\{s_{mb}\}_{b=1}^{B}$ $(M \times B = L)$. We assume that the features of the modulation scheme exist in the differences between signal symbols, and CNN classifiers achieve the modulation classification by comparing the values of each B sample point in the M signal symbols. Although the CNN classifiers do not know the values of M and B, and do not even commit to calculating those values, the comparison of symbols will be achieved by local connecting, weight sharing, and pooling operations. However, a CNN focuses on the local correlation of its input. Luo *et al.* [16] proved that although the receptive field size of a CNN can be increased by stacking more convolutional layers and pooling layers, only the central region of the receptive field is effective. Consequently, a CNN can effectively compare only symbol y_m with a small number of symbols that are located close to y_m on the signal sample matrix Y_{dts}, and the features that can be extracted by comparing y_m with other symbols are wasted. We assumed that if the CNN compared more symbols effectively, more features of the modulation scheme would be extracted, and the performance of the CNN would be improved. By the same token, we further speculated that whether $x(t)$ is a digital modulation signal or an analog modulation signal, CNN classifiers achieve the modulation classification by comparing periods of sample points, and comparing more periods effectively would improve the performance.

Table I shows the algorithm of the proposed DPM. To facilitate understanding, Fig. 1 describes the process of DPM. R is a hyperparameter of DPM. To enhance the features, steps 1 to 3 of the Table I algorithm add the conversion results in polar coordinates to the signal. The concatenation of Y_0 and Y_1 makes $\{s_l\}_{l=1}^{R}$ spatially close to $\{s_l\}_{l=L-R+1}^{L}$ and makes $\{s_l\}_{l=R+1}^{L}$ spatially close to $\{s_l\}_{l=1}^{L-R}$; the concatenation of Y_1 and Y_2 makes $\{s_l\}_{l=1}^{2*R}$ spatially close to $\{s_l\}_{l=L-2*R+1}^{L}$ and makes $\{s_l\}_{l=2*R+1}^{L}$ spatially close to $\{s_l\}_{l=1}^{L-2*R}$; and so on, until the concatenation of Y_{N-1} and Y_N makes $\{s_l\}_{l=1}^{N*R}$ spatially close to $\{s_l\}_{l=L-N*R+1}^{L}$, and $\{s_l\}_{l=N*R+1}^{L}$ spatially close to $\{s_l\}_{l=1}^{L-N*R}$. We mark the DPM-processed Y_{dts} as Y_{DPM}. It is clear that by changing the input of a CNN from Y_{dts} to Y_{DPM}, the CNN could effectively compare more periods of sample points, and the input shape would be more suitable for the CNN. We found that $R = 4$ was best for processing the RadioML2016.10a dataset to be used as the

TABLE I

ALGORITHM OF THE PROPOSED DPM

DPM Algorithm	
1.	Reshape Y_{dts} to size $1 \times L \times 2$, and each $1 \times 1 \times 2$ unit represents a sample point.
2.	Calculate the amplitude value $A_l = \sqrt{{I_l}^2 + {Q_l}^2}$ and the phase value $P_l = \tan^{-1}(\frac{I_l}{Q_l})$ of each sample point s_l.
3.	Concatenate each $[A_l, P_l]$ and s_l to generate Y_0 with size $2 \times L \times 2$, and each $2 \times 1 \times 2$ unit represents a sample matrix $\begin{bmatrix} I_l & Q_l \\ A_l & P_l \end{bmatrix}$.
4.	Set a positive integer R; set $n = 0$.
5.	$ns = n*(n+1)*R/2$; implement a length-ns circular right shift on Y_0 in the second dimension to get Y_n.
6.	$n = (n+1)$.
7.	If $n*(n+1)*R/2 < L$: back to 5. Else: $NS = ns, N = n$.
8.	Concatenate $\{Y_n\}_{n=0}^{N}$ to get Y_{DPM} with size $(2*N+2) \times L \times 2$ as the input of CNN.

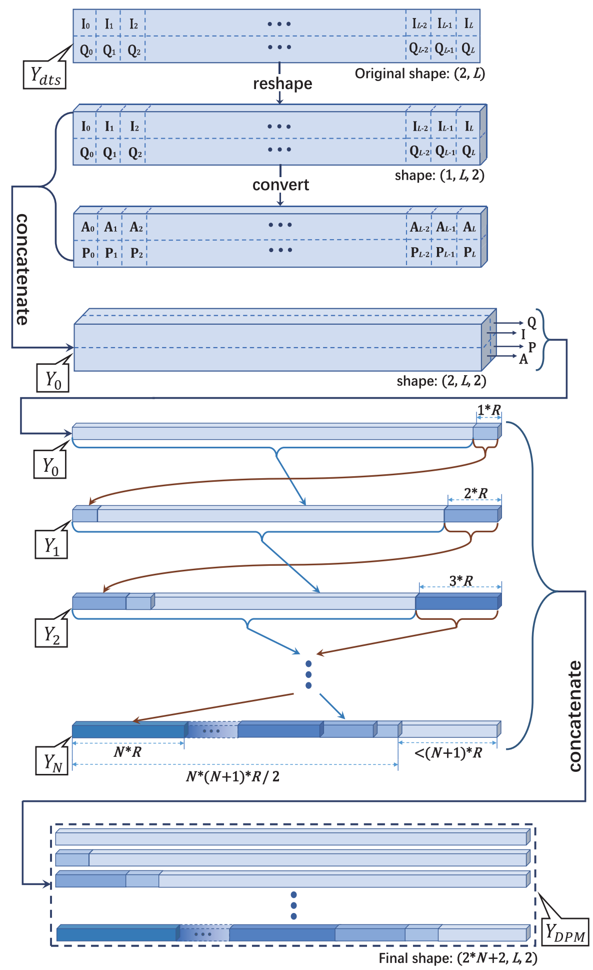
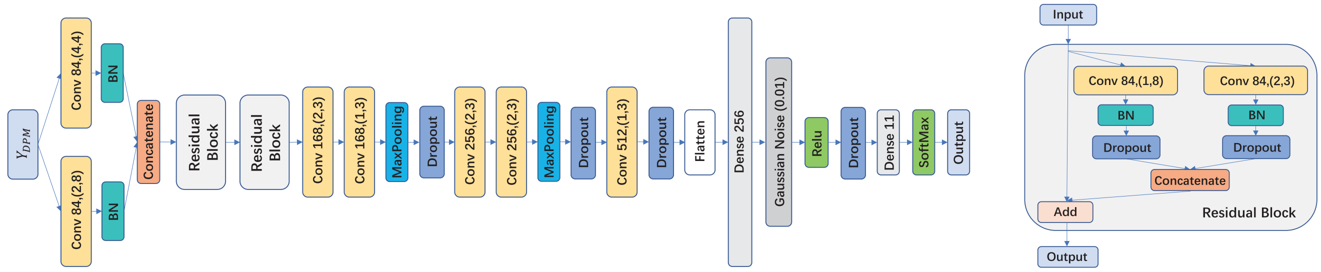


Fig. 1. Description of the process of DPM.



(a) (b)

Fig. 2. Specially designed CNN structures. (a) Structure of the SCNN. (b) Structure of the residual block in SCNN. BN represents batch normalization.

input of a fine-tuned simple CNN [9]; detailed results are shown in Section IV.

III. Structure of Networks

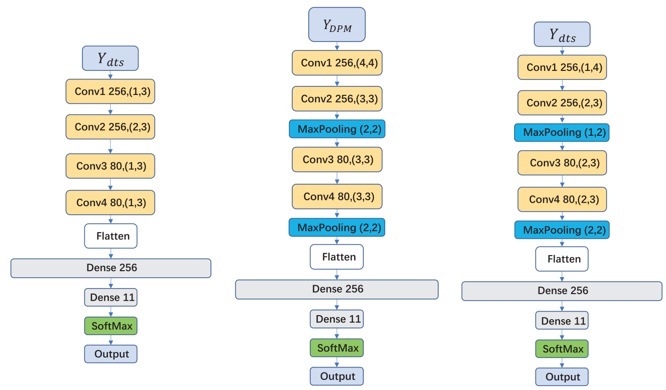
A. Proposed Network

In this letter, we propose a specially designed CNN (SCNN) with residual blocks to identify the modulation scheme of a DPM-processed signal dataset. Fig. 2a shows the structure of the proposed network. Fig. 2b shows the architecture of the residual block used in this model. The dropout rate of the dropout layers is 0.5, and the pool size of the pooling layers is 2×2. All the convolutional layers use a rectified linear unit as the activation function. A Gaussian noise layer is adopted to reduce overfitting. There are two special designs in this network:

1) Note that each $2 \times 1 \times 2$ unit in $\{Y_n\}_{n=0}^{N}$ is a sample point, thereby the strides of the first two convolutional layers, which are both connected with input, are set to $(2, 1)$. Strides of other convolutional layers are the default $(1, 1)$.
2) The first two convolutional layers are parallel to each other, as are the two convolutional layers in the same residual block. To focus on extracting the horizontal and vertical features, the filters of these pairs of convolutional layers are set with a narrow shape and a comparably wide and short shape respectively. As mentioned in Section II, we assumed that the CNN classifiers achieve the modulation classification by comparing periods of sample points. Because the arrangement of the sample points is no longer in sequence but in a matrix after DPM processing, the comparison of the periods of the sample points is not only horizontal but also vertical, and therefore the parallel convolutional layers are designed.

B. Fine-Tuned Simple CNN

Fig. 3a shows the structure of the simple CNN proposed in [9], namely CNN1; Fig. 3b shows the structure of the fine-tuned CNN1, namely CNN2, which is prepared to use a DPM-processed dataset as input; Fig. 3c shows the structure of the fine-tuned CNN1, namely CNN3, which is prepared to use a raw dataset as input. The fine-tuning is shown in Fig. 3 and described as follows: Compared with the original CNN1, we add two maxpooling layers with a pool size of $(2, 2)$ to form CNN2; they are located after the second and fourth



(a) (b) (c)

Fig. 3. Structure of CNN1 (a), CNN2 (b), and CNN3 (c).

convolutional layers. Meanwhile, we set the kernel size of the i-th convolutional layer of CNN2 from $(1, 3)$, $(2, 3)$, $(1, 3)$ and $(1, 3)$ to $(4, 4)$, $(3, 3)$, $(3, 3)$ and $(3, 3)$, respectively. The structure of CNN3 is more similar to that of CNN2, to verify that the change of performance is due to the use of DPM or the fine-tuning of CNN1. However, due to the limitation of the input shape, the kernel size and pool size of CNN3 cannot be exactly the same as those of CNN2. Also, the strides of the first convolutional layer of CNN2 are set to $(2, 1)$, whereas those of CNN1 and CNN3 are the default $(1, 1)$.

IV. Experimental Results

All the tested models were simulated in Keras with a GPU GTX 1080 Ti.

A. Results on RadioML2016.10a

This research used the RadioML2016.10a dataset as a benchmark to evaluate the classification models. The RadioML2016.10a dataset was generated with GNU Radio by O'Shea *et al.* [17] and is available at *https://www.deepsig.ai*. It contains some practical channel impairments, such as sampling rate offset and carrier frequency offset, and is widely used as a benchmark in the AMC field. The dataset comprises 11 classes of modulation signals, each class contains 20,000 signal examples, and each example consists of 128 IQ sample points. The 11 modulation classes are SSB-AM, DSB-AM, BPSK, QPSK, 8PSK, BFSK, CPFSK,

TABLE II
PERFORMANCE OF DIFFERENT MODELS ON RADIOML

	CNN1	CNN2					CNN3
		$R=2$	$R=3$	$R=4$	$R=5$	$R=6$	
MaxAcc(%)	81.38	91.01	**91.32**	91.05	90.77	89.48	82.11
AvgAcc(%)	68.82	77.55	77.38	**77.72**	76.93	75.70	68.30
Input shape	2,128	22,128	18,128	16,128	14,128	14,128	2,128
	$c=1$	$c=2$	$c=2$	$c=2$	$c=2$	$c=2$	$c=1$
CT(ms)	0.162	0.591	0.429	0.402	0.377	0.338	0.208

c represents the channel size.

16-QAM, 64-QAM, PAM4, and WBFM. The signal-to-noise ratio (SNR) of those signals ranges from –20 dB to 18 dB at intervals of 2 dB. In our experiments, the dataset was randomly divided into 70% for training and 30% for testing for all the classification models. For further research, the code for DPM and the proposed network was made public and is available at *https://github.com/haozheng61/DPM-for-AMC-based-on-CNN*.

In this research, we compared the results of CNN1, CNN3, and CNN2 with various values of the hyperparameter R for the DPM to preprocess the dataset. Table II shows the maximum accuracy (MaxAcc), the average accuracy (AvgAcc) when the SNR was not less than –10 dB, the input shape, and the computing time per sample (CT) of each model. We found that a DPM with $R = 3$ achieved the highest maximum accuracy, and a DPM with $R = 4$ achieved the highest average accuracy. Note that CNN2 with other values of R for DPM also gained a marked improvement in accuracy, whereas the accuracy of CNN3 and CNN1 did not differ by more than 1%. On the other hand, as the sample size became larger after the DPM, the computing time for each sample became longer. Based on the algorithm of the DPM, the smaller the value of R, the larger the sample size would be. Also, the raw signal sample needed to be added a dimension as its channel before being fed into CNN1 or CNN3, and the last dimension of DPM-processed samples was regarded as the channel of input of CNN2.

Subsequently, we used an SCNN to identify the modulation of a DPM-processed dataset with $R = 4$ and compared the results with CNN2 as $R = 4$ and five state-of-the-art models (CNN1 [9], CLDNN [10], ResNet [10], LSTM [11], and GrrNet [18]). Fig. 4 shows the classification accuracy of models at varying SNRs from -20 dB to 18 dB. According to our results, 93.72%—the highest maximum accuracy—was achieved by an SCNN with a DPM when the SNR was 14 dB, whereas the maximum accuracy of CNN1, CLDNN, ResNet, LSTM, GrrNet, and CNN2 with a DPM were 81.38%, 84.61%, 84.53%, 91.15%, 90.78%, and 91.05% respectively. Meanwhile, the highest average accuracy (80.32%) when the SNR was not less than -10 dB was also achieved by an SCNN with a DPM, whereas the average accuracy of CNN1, CLDNN, ResNet, LSTM, GrrNet, and CNN2 with a DPM were 68.82%, 71.39%, 68.97%, 77.13%, 76.81%, and 77.72% respectively. Fig. 5 shows the computing time per sample of the above models. We found that although LSTM and GrrNet achieved accuracy above 90% when the SNR was high,

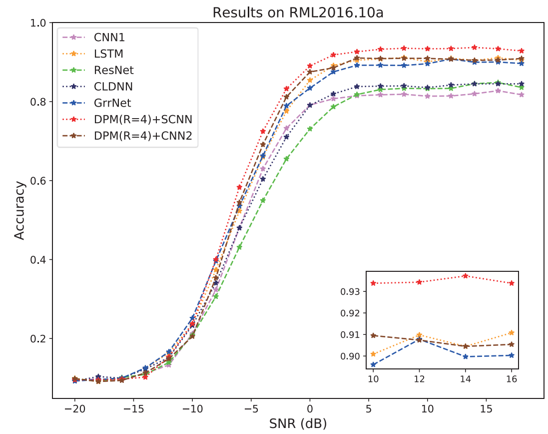


Fig. 4. Accuracy of different models on RadioML.

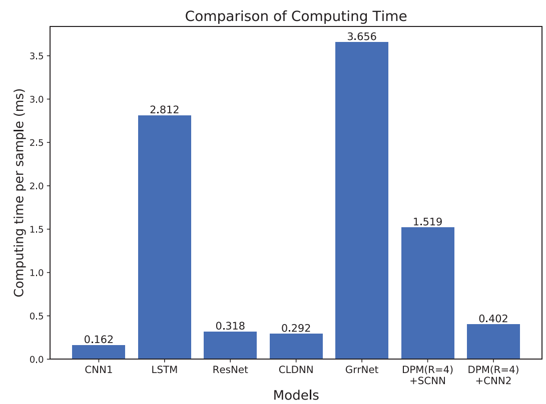


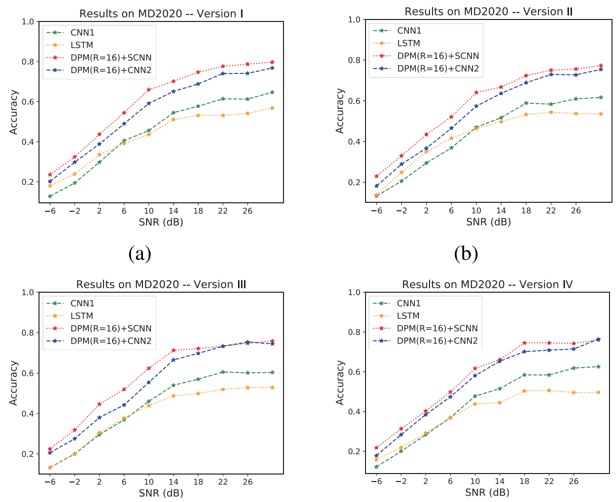
Fig. 5. Computing time per sample of the tested models.

the proposed SCNN ran faster and achieved higher accuracy on RadioML2016.10a.

B. Results on MD2020

In order to evaluate the robustness, we tested the CNN1, CNN2, SCNN, and LSTM as the optimal model among the others with another dataset generated with MATLAB R2020a, and we marked the dataset as MD2020. The code for MD2020 is provided by MathWorks, and is available at *https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html?s_tid=srchtitle*.

The MD2020 comprises 11 classes of modulation signals, each class contains 4,000 signal examples, and each example consists of 256 IQ sample points (for the original code, it was 1024 sample points per example, and we selected the last 256 sample points to reduce the computational complexity). The 11 modulation classes are SSB-AM, DSB-AM, BPSK, QPSK, 8PSK, GFSK, CPFSK, 16-QAM, 64-QAM, PAM4, and B-FM. The SNR of those signals ranges from –6 dB to 30 dB at intervals of 4 dB, and the signal examples were randomly divided into 50% for training and 50% for testing for the four models. To analyze the robustness of the proposed scheme against channel impairments, we tuned the code to generate four versions of MD2020, and the code



(a) (b) (c) (d)

Fig. 6. Accuracy of the four models on MD2020 Version I (a); Version II (b); Version III (c); and Version IV (d).

TABLE III

Overall AveAcc of the Four Models on MD2020

	CNN1	LSTM	SCNN	CNN2
Version I	44.75%	42.63%	**60.06%**	55.55%
Version II	43.85%	42.61%	**58.24%**	54.11%
Version III	43.74%	40.17%	**58.04%**	54.49%
Version IV	43.76%	39.18%	**57.00%**	54.39%

was also uploaded into the above-mentioned Github link. The description of the four versions of MD2020 are as follows:

- Version I: Rician fading channel, and the maximum Doppler shift is 5 Hz.
- Version II: Rician fading channel, and the maximum Doppler shift is 50 Hz.
- Version III: Rician fading channel, and the maximum Doppler shift is 100 Hz.
- Version IV: Rayleigh fading channel, and the maximum Doppler shift is 5 Hz.

As each signal example of MD2020 has 256 sample points, we set the value of R to 16 to reduce the calculation complexity for the CNN2 and SCNN, and so the input shape of CNN2 and SCNN were $(12, 256, 2)$.

Fig. 6 shows the results of the four models on different versions of MD2020. The classification accuracy kept increasing with the increase of SNR, so we recorded the overall average accuracy of each model to evaluate the performance on MD2020, and the results are shown in Table III. We found that the SCNN achieved the highest average accuracy on all the four versions of MD2020, which showed great robustness.

V. Conclusion

In this letter, we propose a DPM for a CNN-based automatic modulation classifier and combine the proposed DPM with an SCNN to form a novel AMC system. We also analyze the advantages of DPM and the idea of two special designs of SCNN. According to experimental results, using the DPM greatly improved the performance of a simple CNN modulation classifier, and combining the DPM with SCNN formed a high-accuracy AMC system that outperformed state-of-the-art AMC models. Also, note that DPM has the potential to be combined with the developments of artificial intelligence in the future to obtain better AMC performance. On the other hand, the use of DPM makes the sample size larger therefore reduces the real-time performance of an AMC system, and the smaller the value of R, the greater the effect on real-time performance.

References

[1] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, Apr. 2007.
[2] O. Dobre, "Signal identification for emerging intelligent radios: Classical problems and new challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 2, pp. 11–18, Apr. 2015.
[3] F. Hameed, O. Dobre, and D. Popescu, "On the likelihood-based approach to modulation classification," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 5884–5892, Dec. 2009.
[4] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 189–193, Feb. 2000.
[5] Y. Yuan, P. Zhao, B. Wang, and B. Wu, "Hybrid maximum likelihood modulation classification for continuous phase modulations," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 450–453, Mar. 2016.
[6] O. A. Dobre, M. Oner, S. Rajan, and R. Inkol, "Cyclostationarity-based robust algorithms for QAM signal identification," *IEEE Commun. Lett.*, vol. 16, no. 1, pp. 12–15, Jan. 2012.
[7] V. Orlic and M. Dukic, "Automatic modulation classification algorithm using higher-order cumulants under real-world channel conditions," *IEEE Commun. Lett.*, vol. 13, no. 12, pp. 917–919, Dec. 2009.
[8] H. Zhang, L. Yu, and G.-S. Xia, "Iterative time-frequency filtering of sinusoidal signals with updated frequency estimation," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 139–143, Jan. 2016.
[9] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 915–919.
[10] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El Gamal, and Y. C. Eldar, "Fast deep learning for automatic modulation classification," 2019, *arXiv:1901.05850*. [Online]. Available: http://arxiv.org/abs/1901.05850
[11] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.
[12] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 3, pp. 469–478, Sep. 2019.
[13] Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav, "Automatic modulation classification based on constellation density using deep learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1275–1278, Jun. 2020.
[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
[15] S. Haykin, "Band-pass data transmission," *Communication Systems*, 2nd ed. Hoboken, NJ, USA: Wiley, 1983, pp. 506–510.
[16] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," 2017, *arXiv:1701.04128*. [Online]. Available: http://arxiv.org/abs/1701.04128
[17] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2016, pp. 213–226.
[18] S. Huang *et al.*, "Automatic modulation classification using gated recurrent residual network," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7795–7807, Aug. 2020.