Article

A Lightweight CNN Architecture for Automatic Modulation Classification

Zhongyong Wang , Dongzhe Sun , Kexian Gong , Wei Wang and Peng Sun *

School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China;
iezywang@zzu.edu.cn (Z.W.); dzsun@gs.zzu.edu.cn (D.S.); kxgong@zzu.edu.cn (K.G.);
iewwang@zzu.edu.cn (W.W.)

* Correspondence: iepengsun@zzu.edu.cn

Abstract: Automatic modulation classification (AMC) algorithms based on deep learning (DL) have been widely studied in the past decade, showing significant performance advantage compared to traditional ones. However, the existing DL methods generally behave worse in computational complexity. For this, this paper proposes a lightweight convolutional neural network (CNN) for AMC task, where we design a depthwise separable convolution (DSC) residual architecture for feature extraction to prevent the vanishing gradient problem and lighten the computational burden. Besides that, in order to further reduce model complexity, global depthwise convolution (GDWConv) is adopted for feature reconstruction after the last (non-global) convolutional layer. Compared to recent works, the experimental results show that the proposed network can save approximately 70~98% model parameters and 30~99% inference time on two well-known benchmarks.

Keywords: automatic modulation classification; convolutional neural network; depthwise separable convolution; feature reconstruction; global depthwise convolution

Citation: Wang, Z.; Sun, D.; Gong, K.; Wang, W.; Sun, P. A Lightweight CNN Architecture for Automatic Modulation Classification. *Electronics* **2021**, *10*, 2679. <https://doi.org/10.3390/electronics10212679>

Academic Editor: Amir Mosavi

Received: 4 October 2021

Accepted: 30 October 2021

Published: 2 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Automatic modulation classification (AMC) is a vital technology between signal detection and demodulation in non-cooperative communication scenarios. AMC means to non-cooperatively classify the modulation scheme of a received radio signal, which can be regarded as a multi-class decision problem. As the foundation of signal demodulation, the correctness of AMC directly determines whether valid information can be recovered from the received signal. Rapid and accurate AMC of wireless signals is widely applied in various civilian and military fields, such as spectrum monitoring, radio fault detection, automatic receiver configuration, and signal interception and jamming [1–3].

Traditional AMC methods can be divided into two categories: likelihood based [4] and feature based [5,6]. The likelihood-based methods calculate likelihood function of candidate modulations and select the modulation mode with maximal likelihood value. This approach treats AMC as a multi-hypothesis test problem, whose implementation is impractical due to its high computational complexity. The traditional feature-based AMC algorithms can be realized by two steps: feature extraction and classificatory decision. For feature extraction, the typical realization methods contain wavelet transform-based features, high-order statistical features, cyclic spectrum-based features and so on. For classificatory decision, available classifiers include decision tree, support vector machine (SVM), fully connected neural network and so on. The feature-based ones behave well in some certain circumstances, whose performances, however, are limited by the design of manual features when the systems contain challenging modulation formats, i.e., high-order modulation 256QAM.

Inspired by the successful application of deep learning (DL) in face recognition, object detection and natural language processing [7,8], the DL-based AMC has become a research hotspot in recent years. Compared to the traditional AMC methods, DL-based

ones can achieve higher accuracy because of their capability of efficient feature automatic learning. We now briefly review the following relevant works. Several high-accuracy deep neural network architectures were explored in [9], including ResNet, DenseNet and convolutional long short-term deep neural network (CLDNN), which can achieve higher accuracy than simpler architecture such as convolutional neural network (CNN). In [1], an improved residual network was proposed for AMC, which achieves advanced classification performance over the DeepSig dataset compared to the feature-based approaches. In [3], the received signal is preprocessed as amplitude phase information and then fed into a long short-term memory network to extract signal features, resulting in significantly improved recognition accuracy. Peng et al. [10] adopted signal constellation for AMC, and the experimental results show that DL-based schemes provide a superior classification accuracy. Huynh-The et al. [11] designed convolution blocks using asymmetric convolution kernels to enhance feature extraction capability. Lin et al. [12] deployed more skip connections in each residual stack to capture the deep and shallow features of signal simultaneously. In [13], a three-stream DL framework was realized to extract the features from individual and combined in-phase/quadrature (I/Q) symbols of modulated signal. In [14], a novel data preprocessing method was proposed for AMC, which can provide more change cases between adjacent symbols for each input signal sample. Unfortunately, most existing DL-based models generally require many trainable parameters (i.e., a large model size). This leads to slow computation speed and can hardly meet the basic requirements of a wireless communication system for low-latency [11]. Moreover, for edge devices with limited memory and computation power [15,16], such as internet-of-things (IoT) devices and unmanned aerial vehicles (UAVs), most existing methods are not competent.

Considering the aforementioned problems of DL-based methods in AMC, in this work, we propose an efficient and lightweight CNN architecture, namely LWAMCNet. The major contributions of this paper are summarized as follows:

- The residual architecture is designed with depthwise separable convolution (DSC) to prevent the vanishing gradient problem and reduce the computational burden.
- After the last (non-global) convolutional layer (last feature map), we use a nonlinear global depthwise convolution (GDWConv) layer to reconstruct the discriminative feature vector.

The rest of this article is organized as follows. Section 2 describes the existing CNN-based AMC. Details of the proposed LWAMCNet are introduced in Section 3. The simulation results are discussed in Section 4. Section 5 concludes the paper.

2. Existing CNN-Based Method

In this section, we will briefly introduce the existing CNN-based AMC methods. The structure of the network is shown in Figure 1a, where we divide the whole procedure of CNN-based AMC into three parts: feature extraction, feature reconstruction and classification.

Generally speaking, the existing CNN-based approaches perform standard convolution (SC) [1,2,9–17] in the lower layers of the network for feature extraction. The SC operation considers all channels simultaneously. For a given convolution layer, the input feature map with size $F_w \times F_h \times M$ and the convolution kernel with size $K_w \times K_h \times M$ are denoted as F and K , respectively, where $F_w \times F_h$ is the spatial width and height of F , $K_w \times K_h$ is the spatial dimension of K and M is the number of input channel (depth). Then the output of each convolution kernel can be expressed as:

$$G = \sigma_{\text{ReLU}}\left(\sum_{m=1}^M K_m * F_m\right), \quad (1)$$

where m and $*$ denote channel index and convolution operation, respectively. The nonlinear activation function, rectified linear unit (ReLU), among each layer is defined as:

$$\sigma_{\text{ReLU}}(x) = \max\{0, x\}. \quad (2)$$

Figure 1. The overall structure of CNN-based method. We divide the structure into three parts: feature extraction, feature reconstruction and classification. (a) The existing methods perform SC operation in the feature extraction part, and FC layers are adopted in feature reconstruction part; (b) the proposed method performs DSC operation in the feature extraction part, and GDWConv is adopted in feature reconstruction part.

In the sequel, one or more fully connected (FC) layers [1,2,9,10,12–17] are deployed for feature reconstruction. The FC layer is fed with the N_i -dimensional v_i , the vectorized version of last feature map, and outputs the N_o -dimensional vector:

$$v_o = \sigma_{\text{ReLU}}(Xv_i + b), \quad (3)$$

where $X \in \mathbb{R}^{N_o \times N_i}$ and $b \in \mathbb{R}^{N_o}$ are weight matrix and bias, respectively. In addition, global average pooling (GAP) layer is heuristically adopted after the last feature map in [11].

Finally, after the last FC layer for classification, the softmax activation function outputs the predicted probabilities of C categories, that is:

$$\sigma_{\text{softmax}}(z_i) = \frac{e^{z_i}}{\sum_{j=0}^{C-1} e^{z_j}}. \quad (4)$$

The above SC operation consumes numerous parameters and calculations, leading to tremendous computational cost of the network, especially in the case of a large number of convolution layers. Moreover, the FC layers used for feature reconstruction are prone to overfitting [18], hence hampering the generalization ability of the network.

3. The Proposed Method

In this section, we design a lightweight AMC network, as shown in Figure 1b, which mainly contains three parts: DSC operation for feature extraction, GDWConv method for feature reconstruction and a FC layer for classification.

3.1. DSC Residual Architecture

Inspired by the skip connection method of deep residual networks [1,19] that can deal with the vanishing gradient problem and improve performance effectively, in this work, we apply depthwise separable convolution (DSC) [8] rather than SC to realize residual architecture for feature extraction, which also brings the benefit of lower computational complexity. The designed DSC residual unit and DSC residual stack is shown in Figure 2.

For feature extraction, multiple DSC residual stacks are used in serial and max pooling is adopted generally at the end of each residual stack for feature dimension reduction. Moreover, a linear 1×1 convolution (a SC layer with convolution kernel size 1×1) is deployed at the beginning of each DSC residual stack to perform channel (feature) fusion after max pooling from previous DSC residual stack.

A SC kernel both filters and combines input feature map into a new channel of output in one step. The DSC splits this into two layers, a separate layer for filtering and a separate layer for combining [8]. We now specifically compare the complexities between SC and DSC. A convolutional layer receives a $W_i \times H_i \times M$ feature map and produces a $W_o \times H_o \times N$ feature map. The SC layer is calculated by N convolution kernels with each size $K_w \times K_h \times M$, while DSC divides the SC into two layers: depthwise convolution (kernel size is $K_w \times K_h \times M$) and pointwise convolution (N kernels with each size $1 \times 1 \times M$). In detail, the computational cost of SC and DSC is compared in Table 1 (SC considering the bias and DSC considering bias only for pointwise convolution layer). Obviously, DSC has fewer multiplications, which will speed up the computation of the model.

Figure 2. DSC residual unit and DSC residual stack.

Table 1. Complexity comparison between SC and DSC.

Method	Multiplications	Additions
SC	$W_o H_o N K_w K_h M$	$W_o H_o N K_w K_h M$
DSC	$W_o H_o (K_w K_h M + NM)$	$W_o H_o N K_w K_h M$
DSC/SC	$1/N + 1/K_w K_h$	1

3.2. GDWConv Feature Reconstruction Method

In the feature reconstruction part, we no longer consider adopting the traditional FC layers [1,2,9,10,12–17], because it usually brings a lot of model parameters and easily leads to network overfitting [18]. For CNN, each channel in last feature map contains the same type features of input signal. According to the nature of receptive field [20], for a single channel, each feature sequentially corresponds to the particular range of the input signal, as shown in Figure 1b. Furthermore, due to the randomness and variability of transmitted signal, the feature importance of each channel varies at different positions. Nevertheless, GAP layer from [18] suggests taking the average of each channel and cannot reflect the importance differences.

Considering the aforementioned problems, we use GDWConv to learn the feature importance of different positions, and then employ ReLU activation function and bias to enhance the classification ability. A GDWConv layer is a depthwise convolution layer with kernel size equaling the input size. In particular, the output of GDWConv layer is represented as:

$$\tilde{G}_m = \sigma_{\text{ReLU}} \left(\sum_{i,j} \tilde{K}_{i,j,m} \cdot \tilde{F}_{i,j,m} + \tilde{b}_m \right), \quad (5)$$

where $\tilde{F}$ is the last feature map of size $W \times H \times M$, $\tilde{K}$ is the global depthwise convolution kernel of size $W \times H \times M$, $\tilde{b}$ is bias and $\tilde{G}$ is the output of size $1 \times 1 \times M$. Furthermore, index (i, j) denotes spatial position and m is channel index. To be specific, the size of $\tilde{K}$ is equal to the size of $\tilde{F}$, and the reconstruction operation happens in the corresponding channel between $\tilde{K}$ and $\tilde{F}$. For a particular channel m , the multiplication in Equation (5) occurs at the corresponding position (i, j) between $\tilde{K}_m$ and $\tilde{F}_m$, and finally adds to output a feature vector.

3.3. LWAMCNet Architecture

In this subsection, the proposed lightweight AMC network, LWAMCNet, is realized based on DSC residual architecture and GDWConv feature reconstruction algorithm. In this letter, we mainly consider two datasets, RadioML2018.01A and RadioML2016.10A [1]. For dataset RadioML2018.01A, the detailed configuration of the proposed network is shown in Table 2. In the feature extraction part, we deployed six DSC residual stacks; the details of DSC residual stack are shown in Section 3.1. It is worth noting here that a convolution operation is performed before and after the six DSC residual stacks. We set the number of convolution kernels used in the DSC residual stack to be 32, and the purpose of making the number of first convolution (First Conv) kernels 64 is to extract sufficient information from input signal. Similarly, the linear 1×1 convolution with 64 kernels is adopted for rising feature dimension. For feature reconstruction part, we leverage GDWConv described in Section 3.2. Finally, a FC layer activated by softmax function is deployed for classification.

With regard to another benchmark, RadioML2016.10A, the network structure is fine-tuned in the feature extraction part. Since the number of samples in this dataset is relatively small (compared to RadioML2018.01A) and the deep network is easy to become overfitted, we only configure three DSC residual stacks. After that, the feature dimension will not be raised by linear 1×1 convolution after the three DSC residual stacks because we believe that 32 features are sufficient to discriminate 11 classes of signals.

Table 2. Proposed LWAMCNet architecture.

Layer	Output Dimension	Procedure
Input	$2 \times 1024 \times 1$	—
First Conv 2×5 ReLU	$1 \times 1024 \times 64$	feature extraction part
DSC Residual Stack	$1 \times 512 \times 32$	
DSC Residual Stack	$1 \times 256 \times 32$	
DSC Residual Stack	$1 \times 128 \times 32$	
DSC Residual Stack	$1 \times 64 \times 32$	
DSC Residual Stack	$1 \times 32 \times 32$	
DSC Residual Stack	$1 \times 16 \times 32$	
Conv 1×1 Linear	$1 \times 16 \times 64$	
GDWConv 1×16 ReLU	1×64	feature reconstruction part
FC Softmax	24	classification part

4. Simulation Results and Discussion

4.1. Dataset and Experimental Background

In this section, the proposed LWAMCNet is evaluated on the datasets RadioML2018.01A and RadioML2016.10A [1]. The datasets are generated by synthetic simulated channel effects (including carrier frequency offset, symbol rate offset, delay spread). The parameters of datasets utilized in the experiment are summarized in Table 3. The configuration of the experiment for training is Ubuntu, a 2.40-GHz Intel(R) Xeon(R) E-5 2640 v4 CPU and a 12 GB NVIDIA TITAN V GPU (the core frequency is 1200/1455 MHz). The DL environment is a Keras framework based on TensorFlow.

Table 3. RadioML dataset parameters.

Dataset Version	RadioML2018.01A	RadioML2016.10A
Number of Modulation Types	24	11
Sample Size	2×1024	2×128
SNR Range (dB)	−20:2:20	−20:2:18
Number of Training Samples	722,534	154,000
Number of Testing Samples	309,658	66,000

The correct classification probability ($P_{cc} = \{P^k | k = -20:2:18(20)\}$) is used to measure performance in this correspondence, that is:

$$P_{cc}^k = \frac{S_{correct}^k}{S_{total}^k} \times 100\%, \quad (6)$$

where $S_{correct}^k$ and S_{total}^k are the number of correctly classified samples and totally testing samples under signal-to-noise ratio (SNR) = k dB, respectively. Furthermore, we adopt maximum accuracy (MaxAcc) and average accuracy (AvgAcc) to evaluate the overall classification performance of all methods. Specifically, MaxAcc and AvgAcc are the maximum value and average value of P_{cc} , respectively.

4.2. Hyperparameters Optimization of LWAMCNet

In the sequel, our first task is to determine the hyperparameters of the proposed network architecture. Considering the number of DSC residual stacks L , from Table 4, we see that $L = 6$ and $L = 3$ yield the highest MaxAcc and AvgAcc for RadioML2018.01A and RadioML2016.10A, respectively. After that, we choose 1×5 kernel size for both datasets. Although 1×5 is not the best for RadioML2018.01A, it can achieve the best trade-off between accuracy and model parameters. Furthermore, we test the influence of batch size on these two datasets and find that 128 and 32 achieve the best accuracy for RadioML2018.01A and RadioML2016.10A.

Table 4. Comparison of different network hyperparameters.

RadioML2018.01A Dataset				RadioML2016.10A Dataset		
Hyperparameters		MaxAcc (%)	AvgAcc (%)	Hyperparameters	MaxAcc (%)	AvgAcc (%)
L	4	96.61	53.62	L	1	83.82
	5	96.80	53.69		2	84.52
	6	97.12	53.73		3	85.22
	7	96.71	53.38		4	83.70
Kernel Size	1×3	95.42	52.40	Kernel Size	1×3	84.87
	1×5	96.46	53.38		1×5	85.83
	1×7	96.36	53.18		1×7	85.47
	1×9	96.55	53.50		1×9	85.05
Batch Size	64	97.09	53.49	Batch Size	16	85.46
	128	97.35	53.85		32	86.41
	256	96.22	53.10		64	85.44
	512	96.46	53.38		128	85.17

4.3. Performance of Residual Architectures

For the sake of fair comparison, we fix that the FC method is adopted in both the feature reconstruction part and classification part, and then DSC residual architecture and SC

residual architecture are exploited in the feature extraction part, respectively. Particularly, the MaxAcc, AvgAcc, trainable parameters and average inference time are evaluated and summarized in Table 5 for RadioML2018.01A and Table 6 for RadioML2016.10A, respectively. Inference time refers to the time consumed in forward calculation of a single sample input network, and the results presented are the average value of 10,000 realizations. It should be noted that the parameters and inference time considered here are only for the feature extraction part, that is, from the input layer to last feature map. For both datasets, here we can see that, compared to SC, DSC residual architecture can significantly reduce the model parameters and inference time by approximately 70 and 30%, respectively, with the cost of a slight accuracy loss (roughly about 0.5%), which demonstrates the high efficiency of DSC.

Table 5. Performance comparison between DSC and SC on RadioML2018.01A dataset.

Residual Architecture	MaxAcc (%)	AvgAcc (%)	Parameters	CPU Inference Time (ms)
SC [1]	96.81	52.91	151,072	13.204
DSC	96.33	52.74	37,248	7.958

Table 6. Performance comparison between DSC and SC on RadioML2016.10A dataset.

Residual Architecture	MaxAcc (%)	AvgAcc (%)	Parameters	CPU Inference Time (ms)
SC [1]	85.22	57.47	66,720	2.406
DSC	84.76	56.68	19,488	1.720

4.4. Performance of Feature Reconstruction Methods

In this subsection, we fix that the feature extraction part and classification part are realized by SC residual architecture and FC layer, respectively, and four different schemes (FC, GAP, proposed GDWConv Linear and GDWConv ReLU) are tested in the feature reconstruction part. Tables 7 and 8 summarize the MaxAcc, AvgAcc, model parameters and average inference time (for feature reconstruction part only), where we find: (1) for model parameters and inference time, FC consumes the most but does not bring the best accuracy; (2) the proposed GDWConv ReLU provides slightly improved accuracy, and the number of parameters and inference time it consumes are comparable to those of GAP; and (3) compared with FC layers, the three global algorithms can achieve considerable accuracy while consuming less in parameters and inference time. As we have seen, the global operations contains very few trainable parameters, thus overfitting is prevented in the feature reconstruction part. In addition, the global algorithm sums out the whole information of the signal sample, which is more robust to AMC.

Table 7. Performance comparison with different feature reconstruction methods on RadioML2018.01A dataset.

Method	MaxAcc (%)	AvgAcc (%)	Parameters	CPU Inference Time (ms)
FC [1]	96.81	52.91	85,272	0.369
GAP [18]	96.30	52.76	0	0.032
GDWConv Linear	96.58	53.03	544	0.059
GDWConv ReLU	97.09	53.54	544	0.066

Table 8. Performance comparison with different feature reconstruction methods on RadioML2016.10A dataset.

Method	MaxAcc (%)	AvgAcc (%)	Parameters	CPU Inference Time (ms)
FC [1]	85.22	57.47	82,176	0.348
GAP [18]	86.01	57.95	0	0.029
GDWConv Linear	85.89	57.63	544	0.049
GDWConv ReLU	86.10	58.42	544	0.062

4.5. Performance of Different Networks

In this experiment, the accuracy performance of LWAMCNet is compared with those of the CNN/VGG neural network [1], residual neural network (ResNet) [1], modulation classification convolutional neural network (MCNet) [11] and multi-skip residual neural network (MRNN)[12] using RadioML2018.01A dataset, respectively, in Figure 3. Here we find that: (1) VGG network presents the worst accuracy due to its relatively simpler structure and the usage of less convolution layers; (2) MCNet behaves best when SNR is less than 0 dB; however, converges to relatively worse point at high SNRs; and (3) LWAMCNet achieves the best at higher SNRs, with an improvement of 0.42 to 7.55% at 20 dB compared to the others.

For the model complexity analysis, the network parameters and average inference time are evaluated in Table 9. We see that LWAMCNet ($L = 6$) consumes about 70–84% less model parameters than those of other schemes. In addition, LWAMCNet saves approximately 41% inference time compared to ResNet. Although CNN/VGG takes the shortest inference time, it has the worst accuracy with the most trainable parameters.

Figure 3. Correct classification probability of different networks on RadioML2018.01A dataset.

Table 9. Performance comparison using RadioML2018.01A dataset.

Network	MaxAcc (%)	AvgAcc (%)	Parameters (K)	CPU Inference Time (ms)
CNN/VGG [1]	89.80	49.76	257	4.967
ResNet [1]	96.81	52.91	236	13.701
MCNet [11]	93.59	50.80	142	11.731
MRNN [12]	96.00	51.20	155	11.765
LWAMCNet ($L = 4$)	96.61	53.62	33	7.756
LWAMCNet ($L = 5$)	96.80	53.69	37	7.928
LWAMCNet ($L = 6$)	97.35	53.85	42	8.147

To show the robustness of the proposed method, we re-evaluate LWAMCNet using the RadioML2016.10A dataset and compare it with previous works [3,9,17]. The classification accuracy performances versus SNR is shown in Figure 4, where we see that: (1) LSTM2 network from [3] presents the highest accuracy, and it should be noted that the network input is preprocessed; and (2) LWAMCNet is slightly better than those of a simple CNN network (CNN2) [9], CLDNN [9], and the specially designed IC-AMCNet [17]. Table 10 summarizes the model complexity of these networks. The results illustrate that our LWAMCNet is still significantly ahead of other algorithms in terms of model parameters and inference time.

Figure 4. Correct classification probability of different networks on RadioML2016.10A dataset.**Table 10.** Performance comparison using RadioML2016.10A dataset

Network	MaxAcc (%)	AvgAcc (%)	Parameters (K)	CPU Inference Time (ms)
CNN2 [9]	80.49	53.11	1,706	17.789
CLDNN [9]	84.42	56.80	509	50.602
LSTM2 [3]	91.76	59.86	217	308.78
IC-AMCNet [17]	83.40	55.14	527	5.175
LWAMCNet ($L = 1$)	84.60	56.78	10	1.230
LWAMCNet ($L = 2$)	85.54	57.90	15	1.597
LWAMCNet ($L = 3$)	86.41	57.96	20	1.915

5. Conclusions

In this paper, an efficient and lightweight CNN architecture, namely LWAMCNet, is proposed for AMC in wireless communication systems. Firstly, a residual architecture is designed by DSC for feature extraction, which can significantly reduce the computational complexity of the model. Additionally, after the last feature map, GDWConv method is adopted for feature reconstruction to output a feature vector, which also lightens the model. The simulation results show the superiority of the LWAMCNet in terms of both model parameters and inference time. In future work, we consider combining the proposed model with network pruning techniques to further reduce model complexity. Furthermore, the semi-supervised AMC algorithm based on few labeled samples and a large number of unlabeled samples will be investigated.

Author Contributions: Conceptualization, Z.W. and D.S.; methodology, Z.W., D.S. and K.G.; software, D.S.; validation, Z.W., D.S. and W.W.; writing—original draft preparation, D.S. and P.S.; writing—review and editing, Z.W., D.S. and P.S.; project administration, K.G., P.S. and W.W. All authors read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61901417, in part by Science and Technology Research Project of Henan Province under Grants 212102210173 and 212102210566 and in part by the Development Program “Frontier Scientific and Technological Innovation” Special under Grant 2019QY0302.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. O’shea, T.J.; Roy, T.; Clancy, T.C. Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [\[CrossRef\]](#)
2. Meng, F.; Chen, P.; Wu, L.; Wang, X. Automatic modulation classification: A deep learning enabled approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10760–10772. [\[CrossRef\]](#)
3. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Distributed deep learning models for wireless signal classification with low-cost spectrum sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. [\[CrossRef\]](#)
4. Hameed, F.; Dobre, O.; Popescu, D. On the likelihood-based approach to modulation classification. *IEEE Trans. Wireless Commun.* **2009**, *8*, 5884–5892. [\[CrossRef\]](#)
5. Huang, S.; Yao, Y.; Wei, Z.; Feng, Z.; Zhang, P. Automatic modulation classification of overlapped sources using multiple cumulants. *IEEE Trans. Veh. Technol.* **2017**, *66*, 6089–6101. [\[CrossRef\]](#)
6. Xie, L.; Wan, Q. Cyclic feature-based modulation recognition using compressive sensing. *IEEE Wireless Commun. Lett.* **2017**, *6*, 402–405. [\[CrossRef\]](#)
7. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
9. Liu, X.; Yang, D.; Gamal, A.E. Deep neural network architectures for modulation classification. In Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers (ACSSC), Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 915–919.
10. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y.-D. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 718–727. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Huynh-The, T.; Hua, C.-H.; Pham, Q.-V.; Kim, D.-S. MCNet: An efficient CNN architecture for robust automatic modulation classification. *IEEE Commun. Lett.* **2020**, *24*, 811–815. [\[CrossRef\]](#)
12. Lin, C.; Yan, W.; Zhang, L.; Wang, W. A real-time modulation recognition system based on software-defined radio and multi-skip residual neural network. *IEEE Access* **2020**, *8*, 221235–221245. [\[CrossRef\]](#)
13. Xu, J.; Luo, C.; Parr, G.; Luo, Y. A spatiotemporal multi-channel learning framework for automatic modulation recognition. *IEEE Wireless Commun. Lett.* **2020**, *9*, 1629–1632. [\[CrossRef\]](#)
14. Zhang, H.; Huang, M.; Yang, J.; Sun, W. A data preprocessing method for automatic modulation classification based on CNN. *IEEE Commun. Lett.* **2020**, *25*, 1206–1210. [\[CrossRef\]](#)
15. Tu, Y.; Lin, Y. Deep neural network compression technique towards efficient digital signal modulation recognition in edge device. *IEEE Access* **2019**, *7*, 58113–58119. [\[CrossRef\]](#)

16. Wang, Y.; Guo, L.; Zhao, Y.; Yang, J.; Adebisi, B.; Gacanin, H.; Gui, G. Distributed learning for automatic modulation classification in edge devices. *IEEE Wireless Commun. Lett.* **2020**, *9*, 2177–2181. [[CrossRef](#)]
17. Hermawan, A.P.; Ginanjar, R.R.; Kim, D.-S.; Lee, J.-M. CNN-based automatic modulation classification for beyond 5G communications. *IEEE Commun. Lett.* **2020**, *24*, 1038–1041. [[CrossRef](#)]
18. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2014**, arXiv:1312.4400.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 4905–4913.