Proceedings of CONF-SEML 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA31902

Research on Automatic Modulation Recognition of Medium-SNR Signals Based on Deep Learning

Yubin Wang

City University of Hong Kong, Hong Kong, China
YubinWang0925@gmail.com

Abstract. With the rapid development of electronic technology and increasing requirements on signal utilization efficiency and low bit error rates, diverse signal modulation schemes have emerged to accommodate diverse scenarios, including electronic countermeasures and military competition. The growing complexity of modulation schemes has made modulation recognition increasingly challenging, positioning deep learning-based automatic modulation recognition as a major research focus. This study uses the RadioML 2018.01A dataset from Kaggle to evaluate modulation recognition performance under signal-to-noise ratios ranging from 0 dB to 12 dB. A conventional convolutional neural network (CNN) and a residual neural network (ResNet), both commonly employed in image classification, are adopted and compared. The objective is to identify the modulation scheme used by the signals in the dataset and classify them into the corresponding categories. For certain modulation schemes, such as 64QAM and 128QAM, the residual neural network achieves classification accuracies that are up to approximately 20% higher than those of the convolutional neural network. Although the recognition accuracy of ResNet for some modulation schemes is slightly lower than that of the CNN, it remains within an acceptable range. Experimental results demonstrate that under medium signal-to-noise ratio conditions, the residual neural network outperforms the convolutional neural network.

Keywords: CNN, ResNet, Automatic Modulation Recognition

1. Introduction

Automatic Modulation Recognition (AMR) methods can generally be divided into traditional approaches and deep learning-based approaches. Traditional AMR mainly includes likelihood-based (LB) methods and feature-based (FB) methods [1-3]. However, these approaches face three major bottlenecks in practical applications. Low computational efficiency in large-scale signal processing scenarios makes it difficult to meet real-time monitoring requirements. As the complexity of the electromagnetic environment continues to increase, traditional methods struggle to maintain stable recognition performance in interference-intensive or highly spectrum-reused scenarios. And with the growing diversity of modulation schemes in modern communication systems, the recognition accuracy of traditional methods still requires improvement when handling such variations [4].

In recent years, rapid advances in deep learning, particularly in image recognition and natural language processing, have demonstrated strong potential for AMR due to their powerful feature

DOI: 10.54254/2755-2721/2026.BA31902

extraction and representation capabilities [5]. Existing studies have shown that various network architectures—including convolutional neural networks (CNNs), recurrent neural networks (RNNs), convolutional long short-term deep neural networks (CLDNNs), and multi-channel convolutional long short-term deep neural networks (MCLDNNs)—can significantly improve modulation classification performance and robustness in complex environments [6-9]. In addition, mechanisms such as time–frequency feature fusion, cross-modal contrastive learning, and SCAM have been introduced to strengthen feature representation under low signal-to-noise ratio conditions [10,11].

Despite these improvements, the aforementioned methods still exhibit limitations in terms of high-order modulation recognition, performance under low signal-to-noise ratio environments, and model complexity. This indicates that exploring more efficient, robust, and deployable network architectures under medium signal-to-noise ratio conditions remains an urgent and valuable research direction. Based on the RadioML 2018.01A dataset from Kaggle, this paper primarily employs CNN and ResNet models to perform modulation recognition on medium-SNR signals using 24 modulation schemes. By comparing the strengths and weaknesses of these two networks in recognizing medium-SNR signals, this study aims to provide data-driven insights and references for subsequent research.

2. Methods

2.1. Dataset and preprocessing

The RadioML 2018.01A dataset is organized into 24 major categories according to modulation schemes. Within each major category, the data are further divided into 26 subcategories based on signal-to-noise ratio (SNR), arranged in ascending order. The SNR range spans from −20 dB to 30 dB.

The RadioML 2018.01A dataset consists of the following three components:

X: A total of 24 × 26 × 4096 sampled matrices, each with a shape of (1021, 2).

Y: The modulation scheme labels corresponding to each matrix in X. The 24 modulation formats are, in order: 'OOK', '4ASK', '8ASK', 'BPSK', 'QPSK', '8PSK', '16PSK', '32PSK', '16APSK', '32APSK', '64APSK', '128APSK', '16QAM', '32QAM', '64QAM', '128QAM', '256QAM', 'AM-SSB-WC', 'AM-SSB-SC', 'AM-DSB-WC', 'AM-DSB-SC', 'FM', 'GMSK', and 'OQPSK'.

Z: The SNR labels corresponding to each matrix in X. A total of 26 SNR levels are included, with a step size of 2 dB, covering the interval from −20 dB to 30 dB.

Since this study focuses on modulation recognition under medium signal-to-noise ratio conditions, the dataset is sliced accordingly. Specifically, seven subsets corresponding to SNR levels of 0 dB, 2 dB, 4 dB, 6 dB, 8 dB, 10 dB, and 12 dB are extracted for subsequent experiments.

2.2. Algorithm overview

2.2.1. CNN architecture

The CNN employed in this study consists of four convolutional layers, and the ReLU function is used as the activation function for all layers:

$$f(x) = \max(0, x) \tag{1}$$

Here, x denotes the input value. The max-pooling layer is configured with a pooling window size of (1, 2). A Dropout layer with a dropout rate of 0.5 is employed to reduce the risk of overfitting.

DOI: 10.54254/2755-2721/2026.BA31902

Two fully connected layers are used, with the numbers of neurons and activation functions set to 128 with ReLU, and 24 with Softmax, respectively.

$$\text{Softmax} = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{2}$$

Here, z^i denotes the raw output corresponding to the i-th class, and N represents the total number of classes. After the input layer receives the preprocessed dataset (where the dataset is sliced according to the signal-to-noise ratio, and all 24 modulation schemes under SNR levels of 0, 2, 4, 6, 8, 10, and 12 dB are extracted and stored separately), the data are fed into the convolutional layers for processing. The specific workflow is as follows:

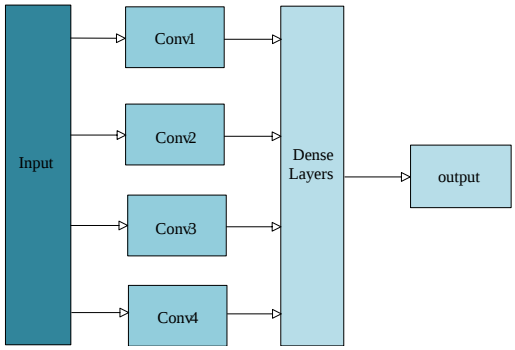

Figure 1. CNN workflow

The cross-entropy function is selected as the loss function of the algorithm:

$$H\left(y, \hat{y}\right) = \sum_i y_i \log\left(\hat{y_i}\right) \tag{3}$$

Here, y_i denotes the true label probability distribution (with one class equal to 1 and the others equal to 0), and $\hat{y_i}$ represents the label probability distribution predicted by the model.

2.2.2. ResNet architecture

The loss function and activation functions used in ResNet are the same as those of the CNN. The key difference is that Conv1 and Conv2 layers are combined to form a residual block, which helps the original CNN network mitigate the effects of gradient explosion and gradient vanishing. The specific workflow is as follows:

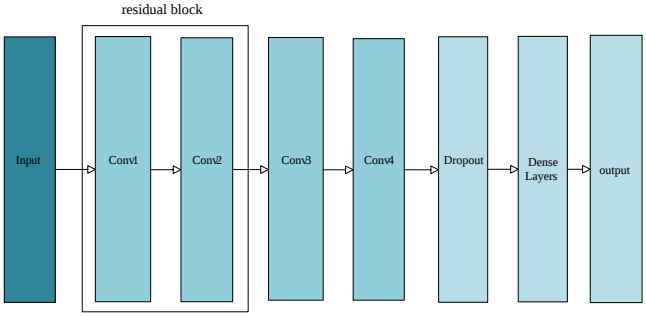

Figure 2. ResNet

DOI: 10.54254/2755-2721/2026.BA31902

2.3. Evaluation metrics

Accuracy: The proportion of correctly predicted samples among all samples.

Calculation: Accuracy = (TP + TN) / (TP + TN + FP + FN).

Precision: The proportion of samples that truly belong to a given class among all samples predicted as that class.

Calculation: Precision = TP / (TP + FP), which measures the accuracy of the predictions.

Recall: The proportion of samples that are correctly predicted among all samples that actually belong to a given class.

Calculation: Recall = TP / (TP + FN), which measures the ability to identify all positive samples.

F1-score: The harmonic mean of Precision and Recall.

Calculation: F1 = 2 × (Precision × Recall) / (Precision + Recall).

2.4. Experiment setup

The experiments are conducted on a system equipped with an AMD Ryzen 7 7800X3D octa-core processor, featuring a base frequency of 4.20 GHz and a maximum boost frequency of 4.30 GHz. The processor supports 16 threads and is equipped with 512 KB of L1 cache, 8 MB of L2 cache, and 96 MB of L3 cache, as well as hardware virtualization technology. The system is configured with 32 GB of DDR5 memory operating at 4800 MT/s in a dual-channel architecture, providing sufficient bandwidth for large-scale data processing. An NVIDIA GeForce RTX 5070 graphics card with 12 GB of GDDR6 memory is used, supporting CUDA-based parallel computing and providing hardware acceleration for deep learning training. The primary storage device is a KIOXIA EXCERIA G2 solid-state drive with a capacity of 1.8 TB, which offers high-speed data read and write capabilities to ensure efficient loading of training data.

2.5. Software environment

The experiments are performed on the Ubuntu 20.04 LTS operating system using Python 3.8.10. The deep learning framework is based on TensorFlow and Keras 2.12.0, with GPU acceleration enabled via CUDA 11.8 and cuDNN 8.6.0. The core dependent libraries include NumPy for numerical computation, Matplotlib 3.7.2 for result visualization, H5py for dataset loading, Pickle for model serialization, and scikit-learn for performance evaluation.

3. Results

3.1. CNNs

Taking the analysis results at SNR levels of 8 dB and 10 dB as examples, the following figure illustrates the classification results of the CNN under these signal-to-noise ratio conditions:

DOI: 10.54254/2755-2721/2026.BA31902

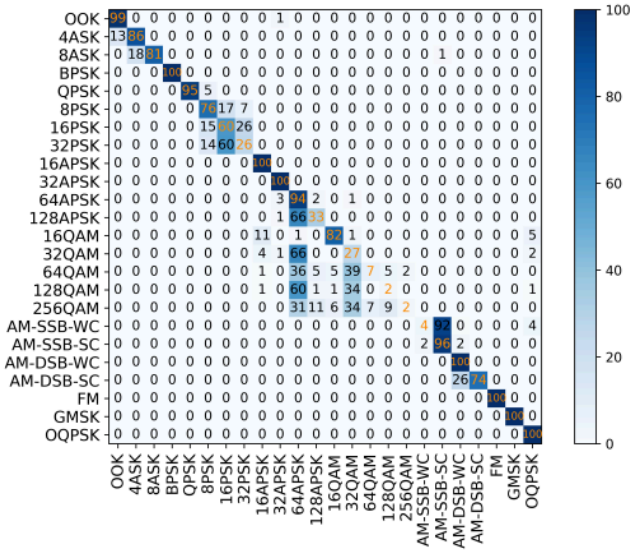

Figure 3. CNN confusion matrix (SNR=8)

As shown in Figures 3 and 4, when classifying modulation schemes such as 256QAM and AM-SSB-WC at an SNR of 8 dB, the CNN exhibits evident shortcomings, with classification accuracies failing to reach even 10%. In contrast, for modulation schemes such as OQPSK and BPSK, the CNN performs exceptionally well, achieving classification accuracies of up to 100%.

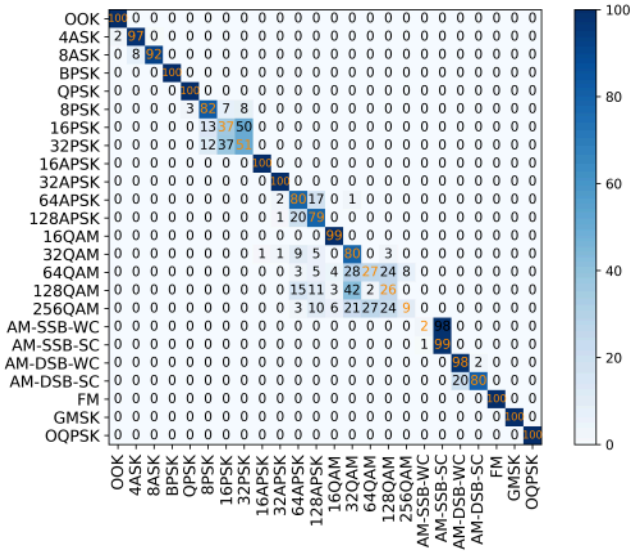

Figure 4. CNN confusion matrix (SNR=10)

3.2. ResNet

Next, the performance of ResNet at an SNR of 8 dB is analyzed. Figures 5 and 6 present the classification results of ResNet at SNR levels of 8 dB and 10 dB, respectively.

DOI: 10.54254/2755-2721/2026.BA31902

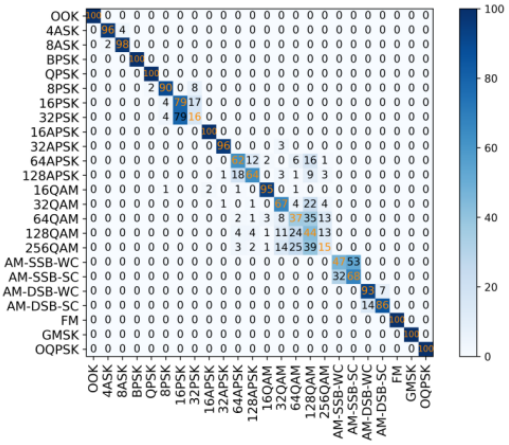

Figure 5. ResNet confusion matrix (SNR=8)

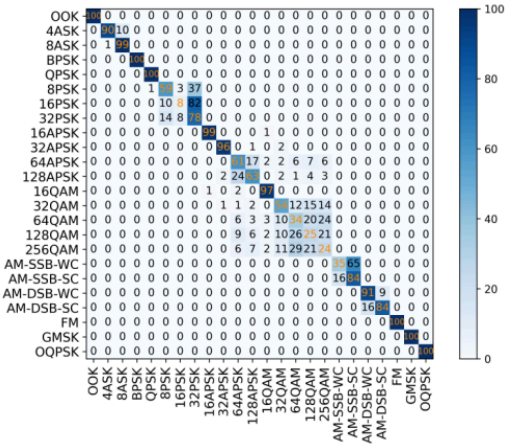

Figure 6. ResNet confusion matrix (SNR=10)

Similar to the CNN results, ResNet also exhibits modulation schemes with outstanding classification performance at a given SNR (such as 16QAM and 16PSK), as well as modulation schemes with relatively weaker performance (such as 256QAM and 32PSK).

3.3. Results comparison

Based on the comparison of the two neural networks at SNR levels of 8 dB and 10 dB, it can be readily observed that under an SNR of 8 dB, the classification performance of ResNet is significantly superior to that of the CNN. In contrast, under an SNR of 10 dB, the CNN exhibits noticeably better classification performance than ResNet.

At the same SNR level, the CNN demonstrates markedly stronger performance than ResNet for certain modulation schemes, while ResNet outperforms the CNN for others. For example, at an SNR of 8 dB, the CNN achieves only a 4% recognition accuracy for the AM-SSB-WC modulation scheme, with a large proportion of samples misclassified as AM-SSB-SC, whereas ResNet attains a recognition accuracy of 35% for this modulation scheme. Therefore, ResNet clearly outperforms the CNN for this particular modulation format. Conversely, at an SNR of 10 dB, the CNN achieves an 82% recognition accuracy for signals with 8PSK modulation, while ResNet attains only 59% accuracy for the same modulation scheme.

DOI: 10.54254/2755-2721/2026.BA31902

Based on these analysis results, it can be concluded that CNN and ResNet each have their own advantages and disadvantages in modulation classification tasks across different signal-to-noise ratios and modulation schemes.

4. Discussion

Based on the above analysis, CNN and ResNet each exhibit their own strengths and weaknesses in classification performance. Although neural network–based algorithms have continued to advance, their development has reached a certain bottleneck. Achieving further breakthroughs solely by modifying CNN, ResNet, or other neural network architectures to improve AMR classification accuracy has become increasingly difficult. In contrast, ensemble learning—where multiple weak classifiers are combined to construct a stronger classifier with higher accuracy—offers a more cost-effective and often more effective alternative.

Through the analysis of the dataset using CNN and ResNet, it is evident that the two models demonstrate complementary advantages and disadvantages across different data segments, with substantial performance differences in certain cases. Therefore, employing ensemble learning to combine two or more deep learning models and allowing them to jointly determine the optimal final output can significantly enhance classification accuracy. For example, a CNN can first be used to identify a signal as belonging to the AM-SSB-SC modulation scheme, after which a ResNet model performs a secondary recognition. If the two recognition results are consistent, the result is directly output; if they differ, the output of the model with the higher classification accuracy is selected as the final result.

This approach offers several advantages. First, it requires a relatively lower level of technical complexity. Compared with attempting to achieve breakthroughs using a single neural network—an approach that demands extensive experimentation with uncertain cost–benefit outcomes—ensemble learning allows, through logical reasoning, the expectation that classification performance will surpass that of a single network. Second, ensemble learning improves model robustness, enabling more stable performance in complex data environments.

However, this method also has notable limitations. The primary drawback of ensemble learning lies in its increased computational cost, which leads to higher algorithmic complexity, reduced interpretability of results, and longer execution times. Nevertheless, in the context of practical AMR applications, strict requirements on runtime efficiency and interpretability are often not critical. Therefore, the increased computational overhead and reduced interpretability have a limited impact on real-world tasks. Overall, ensemble learning remains a feasible and effective option for improving classification performance.

5. Conclusion

This study performs modulation recognition on signals at seven signal-to-noise ratio levels ranging from 0 dB to 12 dB, and analyzes the recognition results at 8 dB and 10 dB as representative examples. The results indicate that, in terms of overall accuracy and loss values, residual neural networks exhibit superior performance to convolutional neural networks in modulation recognition tasks. However, convolutional neural networks still demonstrate better performance than residual neural networks for certain specific modulation schemes.

A limitation of this study is that only convolutional neural networks and residual neural networks are compared, without including other neural network architectures in the evaluation. Based on the recognition results obtained in this work, it can be anticipated that different neural networks each

DOI: 10.54254/2755-2721/2026.BA31902

have modulation schemes for which they achieve relatively high recognition accuracy. Therefore, in scenarios where real-time constraints are not particularly stringent, ensemble learning can be considered for AMR. The key advantage of ensemble learning lies in its ability to combine multiple models to overcome the limitations of individual models in signal recognition, thereby improving overall prediction accuracy. Ensemble models have already demonstrated superior robustness and generalization capability compared to single models in fields such as radar signal modulation recognition and natural language processing. With the continuous improvement of hardware performance, the high computational cost traditionally associated with ensemble learning is gradually becoming less prohibitive. Consequently, it is reasonable to predict that ensemble learning will become a mainstream technique for AMR in the future.

References

[1] Zhang, D.H., & Feng, J.D. (2025) A novel deep learning–based automatic signal modulation recognition algorithm. Modern Electronics Technique, 48(12): 179-186. DOI: 10.16652/j.issn.1004-373x.2025.12.026.
[2] Huang, S., Lin, C., Xu, W., et al. (2021) Identification of active attacks in Internet of Things: joint model and data-driven automatic modulation classification approach. IEEE Internet of Things journal, 8(3): 20512065.
[3] Wang, Y., Gui, G., Ohtsuki, T., et al. (2021) Multitask learning for generalized automatic modulation classification under non-gaussian noise with varying SNR conditions. IEEE transactions on wireless communications, 20(6): 587-596.
[4] Zhang, G.C., Lu, W.D., & Wu, W.B. (2025) An overview of the application of deep learning in signal modulation recognition. China Radio, (06): 65-67.
[5] Yin, Z., Wang, C. J., Cheng, Z. H., et al. (2023) An automatic modulation recognition algorithm based on attention-based convolutional neural networks. Journal of Radio Science, 38(05): 773-779.
[6] O'Shea, T.J., Corgan, J., Clancy, T.C. (2016) Convolutional radio modulation recognition networks. International Conference on Engineering Applications of Neural Networks, 213-226.
[7] Rajendran, S., Meert, W., Giustiniano, D., et al. (2018) Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. IEEE Transactions on Cognitive Communications and Networking, 4(3): 433-445.
[8] West, N. E., O'Shea, T. (2017) Deep architectures for modulation recognition. 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), 1-6.
[9] Xu, J. L., Luo, C.B., Parr, G., et al. (2020) A spatiotemporal multichannel learning framework for automatic modulation recognition. IEEE Wireless Communications Letters, 9(10): 1629-1632.
[10] Bo, D., Wang, K., Liu, Y. S., et al. (2025) Automatic modulation recognition based on time-frequency feature fusion. Radio Engineering, 55(11): 2163-2173.
[11] Chen, H., Guo, W. P., Ju, X. N., et al. (2025) A review of deep learning-based automatic modulation recognition methods for communication signals. Radio Engineering, 55(03): 526-539.