Software Engineering and Artificial Intelligence Systems

Volume: 02 Issue: 01 (2026)

www.seais.org/

Deep Model Compression for Efficient Edge Inference Under Resource Constraints

Lukas Meier, Anna Keller, Sophie Baumann*
Faculty of Arts, University of Melbourne, Melbourne, Australia

Key word: model compression, structured pruning, 8-bit quantization, knowledge distillation, edge computing, efficient inference, Jetson Nano

A B S T R A C T

Deep neural networks often need large memory, high computation, and high energy use, which limits their use on embedded and edge devices. This study proposes a multi-stage compression method for resource-limited inference. The method uses structured pruning to remove redundant channels, 8-bit quantization to reduce memory and computation cost, and teacher-student distillation to recover accuracy after compression. Experiments were conducted on MobileNet and ResNet and deployed on Jetson Nano. The results showed that the method reduced model parameters by about 72%, lowered inference latency to 38% of the original model, and reduced energy use by about 45%. Top-1 accuracy decreased by only 1.6%, showing that the compressed model maintained stable recognition performance. These results indicate that the method can improve deployment efficiency with limited accuracy loss. The method provides a practical approach for low-power vision tasks, mobile sensing, and edge monitoring systems.

1. INTRODUCTION

Deep neural networks have been widely used in image recognition, object detection, medical monitoring, and industrial inspection. Their strong feature representation ability has improved prediction accuracy in many visual and sensing tasks. However, modern neural networks often require large memory capacity, high computational cost, and considerable energy consumption, which limits their deployment on embedded and edge devices such as Jetson Nano [1,2]. This problem is more serious in real-time inference scenarios, where the model must maintain acceptable accuracy while meeting strict latency and power constraints. Evidence from fast MRI reconstruction also shows that different algorithmic strategies may perform differently under specific hardware and task conditions, suggesting that model evaluation should not rely only on theoretical performance indicators but should also consider practical deployment efficiency and application-specific constraints [3]. Model compression has become an important approach for resource-limited deep learning deployment. Pruning reduces model size by removing less important weights, channels, filters, or layers. Compared with unstructured pruning, structured pruning is more suitable for edge devices because it can directly reduce tensor dimensions and improve hardware execution efficiency [4,5]. Quantization further reduces memory usage and computational cost by replacing full-precision weights and activations with low-bit representations, such as 8-bit integers [6,7]. Knowledge distillation provides another effective way to improve the performance of compressed models. In this strategy, a compact student model learns from the output distribution or intermediate

features of a larger teacher model, which helps reduce the accuracy loss caused by compression [8,9]. Although these methods have been widely studied, existing research still has several limitations. Many studies evaluate compression mainly through parameter size, FLOPs, or theoretical compression ratio. These indicators can describe model complexity, but they do not always reflect real inference performance on edge hardware. A model with fewer parameters may not necessarily achieve lower latency if its operators are not hardware-friendly [10,11]. Similarly, a lower FLOP count may not directly lead to lower power consumption because memory access, kernel scheduling, and hardware utilization also affect runtime efficiency [12,13]. Therefore, real-device testing is necessary for evaluating the practical value of compressed models. Another limitation is that pruning, quantization, and distillation are often tested separately. This makes it difficult to understand the combined effect of a full compression pipeline. In practical deployment, a single compression technique may not be sufficient to satisfy both accuracy and efficiency requirements. Strong pruning can reduce computation but may damage feature representation. Quantization can reduce memory and accelerate inference, but it may introduce numerical errors. Distillation can recover part of the lost accuracy, but its effectiveness depends on the teacher model, training strategy, and compression strength [14]. Therefore, a multi-stage compression framework is needed to balance model compactness, accuracy retention, inference speed, and energy efficiency. This issue is especially important for lightweight and edge-oriented networks. Models such as MobileNet are already designed for efficient inference, so aggressive compression may lead to clear accuracy degradation [15,16]. Larger networks such as ResNet contain more redundancy, but their deployment cost remains high on low-power devices [17,18]. A practical compression method should therefore be evaluated across different model architectures and should include both algorithm-level indicators and hardware-level indicators. This can provide a more reliable understanding of whether the compressed model is truly suitable for resource-limited environments [19,20]. To address this problem, this study proposes a multi-stage compression method for resource-limited inference. The method combines structured pruning, 8-bit quantization, and teacher-student knowledge distillation. Structured pruning is used to remove redundant channels and reduce computational cost. Quantization is used to lower memory usage and improve inference efficiency. Knowledge distillation is used to recover accuracy after compression and improve the stability of the compact model. Experiments are conducted on MobileNet and ResNet, and the compressed models are deployed on Jetson Nano for real-device evaluation. The purpose of this study is to examine whether a combined compression pipeline can achieve a better balance between accuracy and deployment efficiency than isolated compression strategies. Parameter size, Top-1 accuracy, inference latency, and power consumption are used as evaluation metrics. The significance of this study lies in shifting model compression evaluation from theoretical complexity reduction to practical edge deployment performance. The proposed method aims to provide a compression strategy that can reduce memory and computation requirements while maintaining acceptable accuracy, making deep neural networks more practical for embedded vision, medical monitoring, and industrial inspection tasks under limited hardware resources.

2. Materials and Methods

2.1 Samples and Study Description

This study used ImageNet-1K validation images and CIFAR-100 test images to test model compression on edge devices. A total of 50,000 ImageNet-1K images and 10,000 CIFAR-100 images were used. The images included common objects, natural scenes, animals, vehicles, tools, and indoor items. MobileNetV2 and ResNet-50 were selected as test models. MobileNetV2 represents a lightweight network, while ResNet-50 represents a deeper residual network. All images were resized according to the input size of each model. ImageNet-1K images were resized to 224 × 224 pixels, and CIFAR-100 images were resized to 32 × 32 pixels. The deployment test was conducted on NVIDIA Jetson Nano to reflect a low-power edge inference setting.

2.2 Experimental Design and Control Groups

The experiment compared the proposed multi-stage compression method with three control settings. The proposed group used structured pruning, 8-bit quantization, and teacher-student distillation in sequence. The first control group used the original full-precision model without compression. The second group used structured pruning only. The third group used pruning and 8-bit quantization without distillation. These groups were used to test the separate and combined effects of pruning, quantization, and distillation. The same training and testing samples were used for all groups. For each model, the pruning ratio was set from 30% to 70%. The final setting was selected according to validation accuracy and inference latency.

2.3 Measurement Methods and Quality Control

Top-1 accuracy, parameter size, inference latency, and power use were measured for each model. Top-1 accuracy was calculated on the full test set. Parameter size was obtained from the number of trainable weights after compression. Inference latency was measured on Jetson Nano with a batch size of 1 to match real edge inference. Each model was warmed up for 100 runs before testing. Latency was then averaged over 1,000 inference runs. Power use was recorded during inference by platform-level power monitoring. All deployment tests were repeated five times, and the mean value was reported. The same software environment, input size, batch size, and device power mode were used for all models.

2.4 Data Processing and Model Formulation

Before training, image pixels were normalized using the mean and standard deviation of the training dataset. Structured pruning removed channels with low importance scores. The channel importance score was calculated by the L1 norm of convolutional filters:

$$S_j=\sum_{i=1}^{C_{in}}\sum_{k=1}^{K}\sum_{l=1}^{K}|W_{j,i,k,l}|$$

where S_j is the importance score of the j-th output channel, and $W_{j,i,k,l}$ is the convolutional weight. Channels with lower scores were removed first. After pruning and 8-bit quantization, teacher-student distillation was used to recover accuracy. The student model was trained with hard labels and teacher soft outputs:

$$L=(1-\alpha)L_{CE}(y,\hat{y}_s)+\alpha T^2 KL(p_t^T||p_s^T)$$

where L_{CE} is the cross-entropy loss, KL is the Kullback-Leibler divergence, p_t^T and p_s^T are the teacher and student outputs under temperature T, and α\alphaα controls the distillation weight.

2.5 Statistical Analysis and Evaluation Criteria

The compressed models were evaluated by accuracy loss, parameter reduction, latency reduction, and energy reduction. Accuracy loss was calculated as the difference between the original model and the compressed model. Parameter reduction was calculated as the percentage decrease in trainable parameters. Latency reduction was calculated from the average inference time before and after compression. Energy reduction was estimated from measured power use and inference time. A compression method was considered effective when it reduced model size and latency while keeping Top-1 accuracy loss within 2%. Paired comparisons were used because all models were tested on the same image sets and the same hardware platform. The final analysis examined whether the proposed multi-stage method achieved a better balance among accuracy, speed, and energy use than single-stage compression methods.

3.Results and Discussion

3.1 Compression Effect on Model Size

The proposed method reduced the number of model parameters by about 72%. This result shows that structured pruning removed redundant channels while keeping the main feature path. Compared with pruning alone, the full method gave a better balance between model size and accuracy. This was mainly because distillation helped the compressed model retain useful class information. This result is consistent with previous compression studies showing that pruning and distillation can reduce model size with limited accuracy loss [21,22].

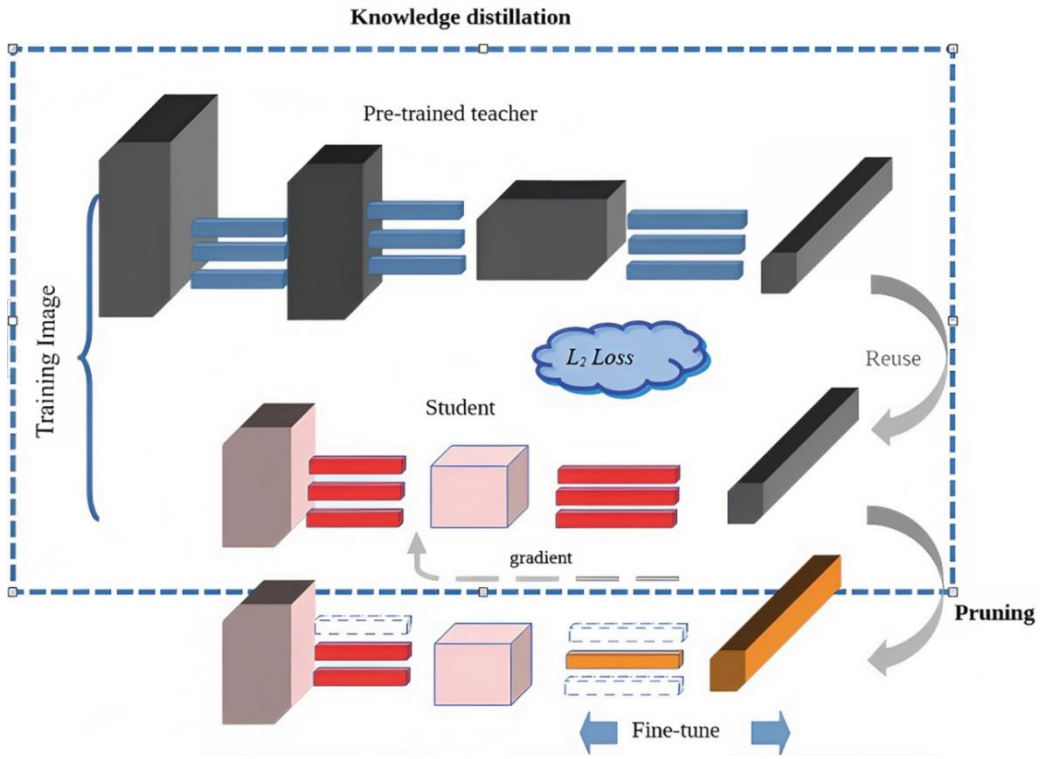


Fig. 1. Parameter size after different compression settings.

3.2 Inference Latency on Jetson Nano

The compressed models showed clear speed gains on Jetson Nano. The average inference latency decreased to 38% of the original model. This reduction came from two parts. First, structured pruning reduced active channels and convolution operations. Second, 8-bit quantization reduced memory access and improved inference speed. Pruning alone reduced latency, but its effect was weaker than the full compression process. This result supports previous edge deployment studies showing that real hardware tests are needed, because parameter reduction does not always lead to the same speed gain [23].

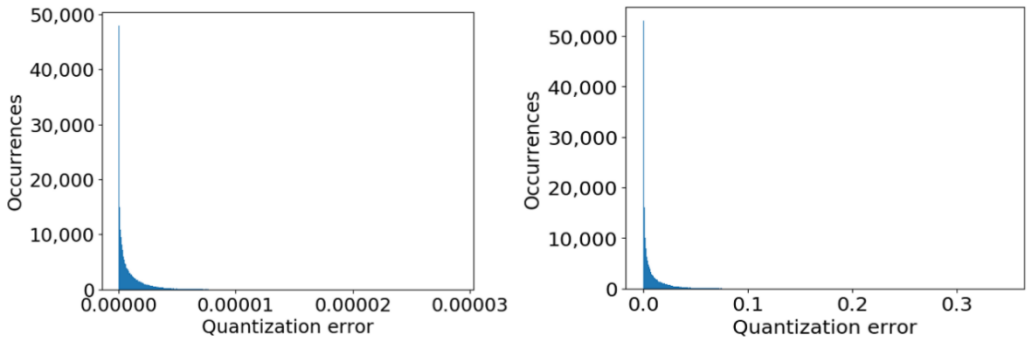


Fig. 2. Inference latency and energy use on Jetson Nano after compression.

3.3 Accuracy Change after Compression

The proposed method kept the Top-1 accuracy loss within 1.6% after compression. This shows that the teacher-student stage reduced the accuracy loss caused by pruning and quantization. MobileNet was more sensitive to compression because it was already a compact model. ResNet had a smaller accuracy drop because it had more redundant channels. Compared with pruning-only and pruning-plus-quantization settings, the full method gave more stable accuracy. This suggests that distillation is useful after structural reduction, as it guides the compressed model to learn the teacher model's output pattern rather than only fitting hard labels [24,25,26].

3.4 Energy Use and Deployment Value

Power tests showed that the proposed method reduced energy use by about 45%. This is important for edge devices, where battery life and heat control affect real use. The compressed model reduced both inference time and active computation, so energy use decreased on the same hardware. Overall, the results show that the proposed method improves deployment by reducing model size, latency, and energy use while keeping accuracy loss small. Some limits remain. The experiments used only MobileNet and ResNet on Jetson Nano. The results may not fully apply to transformer models or other edge chips. Future work should test more model types, more hardware platforms, and longer continuous-running conditions.

4.Conclusion

This study developed a multi-stage compression method for deep model inference on resource-limited devices. The method uses structured pruning, 8-bit quantization, and teacher-student distillation to reduce model size, inference time, and energy use while maintaining accuracy. Tests on MobileNet and ResNet deployed on Jetson Nano showed that the method reduced parameters by about 72%, lowered inference latency to 38% of the original model, and reduced energy use by about 45%. Top-1 accuracy decreased by only 1.6%, which shows that distillation helped recover part of the information lost after pruning and quantization. These results indicate that the method can support efficient deep model deployment on embedded and edge platforms. The main value of this study is that it evaluates compression through real latency and energy use on edge hardware, rather than parameter size alone. This provides a practical basis for low-power vision tasks, mobile sensing, and field monitoring systems. Some limits remain. The experiments used only MobileNet, ResNet, and Jetson Nano. Future work should test transformer models, other edge chips, and longer continuous-running conditions.

References

Zhu, W., Yao, Y., & Yang, J. (2025). Optimizing Financial Risk Control for Multinational Projects: A Joint Framework Based on CVaR-Robust Optimization and Panel Quantile Regression.

Swaminathan, T. P., Silver, C., Akilan, T., & Kumar, J. (2025). Benchmarking deep learning models on nvidia jetson nano for real-time systems: An empirical investigation. Procedia Computer Science, 260, 906-913.

Yan, B. (2026). A Comparative Study: Greedy versus GAN-Based Compressed Sensing for Fast MRI Reconstruction. International Journal of Advance in Applied Science Research, 5(3), 36-42.

Hernández, N., Toledo, P., Blanco, V., & Almeida, F. (2025, April). 2: 4 Pruning on Edge Devices: Performance, Energy Efficiency and Accuracy. In European Conference on Parallel Processing (pp. 278-291). Cham: Springer Nature Switzerland.

Feng, H. (2024, October). Design of Intelligent Charging System for Portable Electronic Devices Based on Internet of Things (IoT). In 2024 5th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE) (pp. 568-571). IEEE.

Tripathi, K., Malik, D., Akshat, A., & Lata, K. (2025). A survey on 1-bit quantized large language models. Neural Computing and Applications, 37(25), 20823-20878.

Chen, F., Li, S., Liang, H., Xu, P., & Yue, L. (2025). Optimization Study of Thermal Management of Domestic SiC Power Semiconductor Based on Improved Genetic Algorithm.

Lee, S., Lee, D., Hyun, M., Kim, H., & Song, B. C. (2025). Compressing Neural Networks on Limited Computing Resources. IEEE Access.

Liang, R., Feifan, F. N. U., Liang, Y., & Ye, Z. (2025). Emotion-Aware Interface Adaptation in Mobile Applications Based on Color Psychology and Multimodal User State Recognition. Frontiers in Artificial Intelligence Research, 2(1), 51-57.

Liu, S., & Yim, J. (2025). Research on Generative AI Creation Systems Based on Visual Language Modeling: Human-Machine Collaboration and Cognitive Feedback Mechanisms. Available at SSRN 6139770.

Dritsas, E., & Trigka, M. (2026). Deployment-Aware Compression of Large Language Models for Edge Intelligence: A System-Level Survey. IEEE Transactions on Artificial Intelligence.

Suchý, I., & Turčaník, M. (2026). Review of large YOLOv8 and RT-DETR energy efficiency on edge devices for real-time detection. Scientific Reports.

Chen, X., Xiao, H., Zeng, Z., Zhang, S., & Xiao, R. (2025). Fine-Grained Multivariate Time Series Anomaly Detection via Causal Inference. Knowledge-Based Systems, 114765.

Mejia-Cajicá, F. A., Hernández, C. J. B., Riveill, M., & Henao, J. A. G. (2026). Optimizing Deep Neural Networks via Knowledge Distillation for Efficient and Sustainable Medical Image Classification. Revista UIS Ingenierías, 25(1), 49-70.

Gao, X., Chen, J., & Huang, M. (2026, January). Research on Risk Dependency Structures and Resource Allocation Optimization in New Energy Technology Collaboration within Enterprise Distributed Innovation. In 2026 International Conference on Electrical Engineering, Intelligent Control and Artificial Intelligence (EEICAI) (pp. 43-47). IEEE.

Raza, S. M., Abidi, S. M. H., Masuduzzaman, M., & Shin, S. Y. (2025). Survey on the application with lightweight deep learning models for edge devices. Authorea Preprints.

Xu, S., Ma, Q., Liu, H., & Yue, L. (2026). Continuous Reorganization and Performance Preservation of Agent Memory Structure Under Distributed Change Environments.

Rasheed, S. (2026). Lightweight Deep Learning Models for Face Mask Detection in Real-Time Edge Environments: A Review and Future Research Directions. Machine Learning and Knowledge Extraction, 8(4), 102.

Gao, G., Ma, X., Lu, C., & Gao, R. (2026). Reliability Analysis and Application Research of SMS Communication Systems in Medical Notification Scenarios.

Kim, G. I., Hwang, S., & Jang, B. (2025). Efficient compressing and tuning methods for large language models: A systematic literature review. ACM Computing Surveys, 57(10), 1-39.

Wang, Y., Yin, X., Chen, J., & Wang, Y. (2026). Evidence-Based Study on Low-Burden Digital Phenotyping for Precision Screening of Oral Anti-Obesity Drug Efficacy.

Kim, G. I., Hwang, S., & Jang, B. (2025). Efficient compressing and tuning methods for large language models: A systematic literature review. ACM Computing Surveys, 57(10), 1-39.

Jiao, Y., Wang, A., Zhao, B., & Shi, T. (2026). Quantitative Study on the Construction and Application Effectiveness of Graffiti Wall Painting Teaching Models in Public Space Contexts.

Aruna Gladys, A., Aishwarya, R., Vetriselvi, V., Daphne, S., Baskar, P. N., & Amaithi Rajan, A. (2026). Building Expert Small Models: A Comprehensive Survey of Model Compression, Knowledge Distillation, and Augmented Inference. Archives of Computational Methods in Engineering, 1-42.

Liu, S., Liu, X., & Feng, H. (2025, November). Research on AI-Driven Visual Design and Immersive Interactive Experiences Based on Multimodal Cognition and User. In Proceedings of the 2025 International Conference on Digital Society and Intelligent Computing (pp. 734-740).

Gu, X. (2026, January). Analysis of the Synergistic Role of User Natural Language Feedback and Behavioral Data in Adjusting Recommendation Strategies for Mental Health Platforms. In Proceedings of the 2026 5th International Conference on Big Data, Information and Computer Network (pp. 198-203).