Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Lightweight Neural Network Compression Pipeline for Resource-Constrained Edge AI Systems

SOM SUBHRO NATH

nathsomsubhro7@gmail.com

Independent Researcher https://orcid.org/0009-0007-6842-0019

Research Article

Keywords: Edge AI, Model Compression, Knowledge Distillation, Neural Network Pruning, Quantization, Resource-Constrained Systems

Posted Date: April 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9295528/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Abstract

Deep neural networks have achieved remarkable performance across a wide range of tasks; however, their deployment on resource-constrained edge devices remains challenging due to high storage and computational requirements. This work proposes a lightweight neural network compression pipeline designed to enable efficient deployment of deep models in constrained environments. The proposed framework integrates knowledge distillation, structured pruning, and dynamic quantization to significantly reduce model size while maintaining competitive predictive performance.

Experiments were conducted on the widely used datasets MNIST and Fashion-MNIST to evaluate the effectiveness of the compression strategy. The results demonstrate that the proposed pipeline achieves approximately 20 times model size reduction while preserving classification accuracy with only marginal degradation compared to the original model.

These findings indicate that combining multiple lightweight compression techniques can produce compact yet accurate neural models suitable for edge deployment. The proposed approach provides a practical and reproducible framework for developing storage-efficient deep learning systems in resource-limited environments, thereby facilitating broader accessibility and scalability of artificial intelligence applications.

These findings indicate that combining multiple lightweight compression techniques can produce compact yet accurate neural models suitable for edge deployment. The proposed approach provides a practical and reproducible framework for developing storage-efficient deep learning systems in resource-limited environments, thereby facilitating broader accessibility and scalability of artificial intelligence applications.

1. Introduction

Deep neural networks have demonstrated remarkable performance across a broad spectrum of machine learning tasks, including image classification, natural language processing, and speech recognition. However, the practical deployment of such models on resource-constrained platforms remains a persistent challenge. Contemporary deep learning architectures often contain millions of parameters and require substantial memory and computational resources, making them unsuitable for many edge environments such as embedded systems, mobile devices, and low-power computing platforms. As artificial intelligence continues to migrate from centralized cloud infrastructures to distributed edge systems, developing compact and efficient neural models has become an increasingly important research objective.

Edge-based artificial intelligence offers several advantages, including reduced latency, improved privacy, and lower network dependency. Nevertheless, these benefits are often offset by the limited storage capacity and computational capability of edge devices. Large neural networks incur significant memory overhead due to dense parameter representations and high-precision weight storage. Consequently,

even relatively simple inference pipelines may exceed the storage or processing budgets available in real-world edge deployments. Addressing this discrepancy between model complexity and hardware capability has motivated extensive research into neural network compression techniques.

A variety of strategies have been proposed to mitigate the resource requirements of deep models. Knowledge distillation transfers representational knowledge from a large teacher network to a smaller student network, enabling the latter to achieve competitive performance with fewer parameters. Pruning techniques remove redundant or low-importance weights from trained models, thereby reducing network sparsity and storage footprint. Quantization methods further compress neural parameters by representing weights using lower-precision numerical formats. Individually, these approaches have demonstrated significant improvements in computational efficiency and model compactness. However, many existing studies focus on individual compression techniques rather than examining how multiple methods may be combined within a unified compression pipeline.

In practical deployment scenarios, integrating complementary compression techniques can offer substantial advantages over applying a single method in isolation. Distillation can guide compact models toward more effective feature representations, pruning can eliminate structural redundancy within the network, and quantization can significantly reduce storage requirements during inference. When applied sequentially, these techniques can collectively yield substantial reductions in model size while maintaining competitive predictive accuracy. Despite this potential, relatively few studies have explored lightweight compression pipelines explicitly designed for rapid and reproducible edge deployment.

In this work, we investigate a compact neural compression framework that integrates knowledge distillation, structured pruning, and dynamic quantization to produce storage-efficient neural models suitable for edge environments. The proposed pipeline emphasizes simplicity and reproducibility while demonstrating that substantial compression can be achieved without significant degradation in predictive performance. Experimental evaluations are conducted on two widely used image classification benchmarks, MNIST and Fashion-MNIST. These datasets provide a controlled setting for evaluating compression strategies while allowing clear analysis of accuracy–compression trade-offs.

The experimental results demonstrate that the proposed pipeline achieves approximately twenty-fold reduction in model size while maintaining competitive classification accuracy relative to the original teacher network. The findings highlight the effectiveness of combining complementary compression techniques to produce compact neural architectures that remain suitable for real-world deployment scenarios.

The main contributions of this study are summarized as follows:

1. A lightweight neural compression pipeline that integrates knowledge distillation, structured pruning, and dynamic quantization for resource-constrained environments.

2. A reproducible compression framework demonstrating substantial reductions in model storage while preserving predictive accuracy.
3. Comprehensive evaluation on benchmark datasets, illustrating the practical feasibility of deploying compact neural models on edge-oriented systems.

Overall, this work aims to contribute a practical and accessible approach to neural network compression, enabling more efficient deployment of deep learning systems in environments with limited computational and storage resources.

2. Related Work

The increasing complexity of modern deep neural networks has motivated substantial research into methods for reducing model size and computational cost. Although deep architectures achieve high predictive performance across numerous machine learning tasks, their large parameter counts and high memory requirements often hinder deployment in resource-constrained environments. As artificial intelligence applications increasingly move toward edge and embedded systems, the development of compact and efficient neural models has therefore become an important research direction.

One widely adopted approach to model compression is knowledge distillation, in which a smaller student model learns to approximate the output behaviour of a larger teacher network. The concept was popularized by Hinton et al. [1], who demonstrated that soft target probabilities generated by a teacher model can guide smaller networks toward improved generalization performance. Subsequent research has shown that distillation enables compact models to achieve performance comparable to significantly larger architectures while reducing computational and storage requirements.

Another important strategy for reducing neural network complexity is network pruning, which removes redundant or low-importance parameters from trained models. Early work by LeCun et al. [2] introduced the concept of optimal brain damage, demonstrating that many neural parameters contribute minimally to overall model performance. Later studies expanded this idea further. Han et al. [3] demonstrated that large neural networks contain substantial redundancy and can be aggressively pruned without significant accuracy degradation. Their deep compression framework combined pruning with trained quantization and efficient encoding techniques to achieve significant reductions in model size.

In addition to pruning and distillation, quantization techniques have become increasingly important for reducing the memory footprint of neural networks. Quantization reduces the precision of model parameters, typically converting 32-bit floating-point weights into lower-precision integer representations. Jacob et al. [4] proposed a quantization framework enabling integer-only inference, significantly improving efficiency for hardware platforms with limited computational resources. Such approaches are particularly advantageous for deployment in edge environments where memory bandwidth and power consumption are critical constraints.

Although each of these techniques has demonstrated significant effectiveness individually, many existing studies focus on applying a single compression strategy in isolation. In practice, however, combining complementary techniques can yield greater improvements in model efficiency. Knowledge distillation can guide compact models toward more informative representations, pruning can remove redundant connections within the network, and quantization can further reduce storage requirements during inference. Integrating these techniques within a unified compression pipeline therefore provides a promising approach for achieving substantial compression ratios while maintaining competitive predictive performance.

Recent research has increasingly explored hybrid compression frameworks that combine multiple efficiency strategies to improve neural network deployment in edge computing environments. Such approaches aim to balance model compactness, computational efficiency, and predictive accuracy. Despite these developments, many existing frameworks remain complex or require specialized training procedures, which may limit reproducibility and practical adoption.

The compression pipeline presented in this study integrates knowledge distillation, structured pruning, and dynamic quantization within a unified and reproducible framework. Experimental evaluation is conducted on benchmark image classification datasets, including MNIST and Fashion-MNIST. The results demonstrate that substantial reductions in model size can be achieved while maintaining competitive classification accuracy, highlighting the practical feasibility of deploying compact neural models in resource-constrained environments.

3. Methodology

3.1. Compression Pipeline Overview

Deploying neural networks in resource-constrained environments requires careful reduction of model complexity while preserving predictive performance. The compression framework investigated in this study integrates three complementary techniques—knowledge distillation, network pruning, and weight quantization—within a unified training pipeline. Each component addresses a different aspect of model efficiency: distillation improves the learning capacity of compact networks, pruning removes redundant parameters, and quantization reduces storage precision during inference.

The overall compression workflow can be summarized as follows:

The sequential integration of these techniques enables progressive reduction of model size while maintaining stable predictive accuracy. Rather than relying on a single compression strategy, the pipeline leverages the complementary strengths of multiple optimization mechanisms to produce compact neural architectures suitable for deployment on resource-limited systems.

3.2. Teacher-Student Distillation

Knowledge distillation provides a mechanism for transferring representational knowledge from a large teacher network to a smaller student model. The concept was originally introduced by Hinton et al. [1], who demonstrated that soft probability outputs from a teacher model can guide the training of compact networks.

Let z_t denote the teacher logits and z_s denote the student logits. Distillation is performed using softened probability distributions defined by a temperature parameter T:

$$P_t = softmax\left(\frac{z_t}{T}\right), \;\; P_s = softmax\left(\frac{z_s}{T}\right)$$

The student network is optimized to minimize the divergence between these distributions using the Kullback–Leibler divergence:

$$L_{KD} = KL\left(P_t \middle| \middle| P_s\right)$$

This training objective encourages the student network to approximate the internal decision structure of the teacher model rather than simply matching hard class labels. As a result, compact networks are able to capture richer structural information present in the teacher model, improving generalization and stabilizing training.

The teacher architecture used in the present study consists of a multilayer fully connected neural network with progressively decreasing hidden dimensions, while the student network adopts a significantly smaller architecture designed to simulate deployment constraints typical of edge environments.

3.3. Network Pruning

Following distillation, the student model is subjected to parameter pruning to eliminate redundant connections. Neural networks are widely known to contain significant parameter redundancy, and numerous studies have demonstrated that large fractions of weights can be removed without substantial degradation in accuracy [2], [3].

Pruning is performed by identifying weights with low magnitude and removing them from the model. Let W represent the set of network weights. Pruning removes weights satisfying

$$|w_i| < \tau$$

where τ denotes a threshold determined by the pruning ratio. In practice, pruning is implemented by ranking weights according to magnitude and removing a fixed proportion of the smallest values.

After pruning, the network undergoes additional fine-tuning to recover potential performance degradation caused by parameter removal. This retraining phase allows the remaining parameters to adapt and compensate for removed connections. Previous work has shown that this pruning–retraining cycle enables significant parameter reduction while maintaining predictive accuracy [3].

3.4. Dynamic Quantization

Although pruning substantially reduces the number of active network parameters, the remaining weights are still typically stored using 32-bit floating-point representations. Quantization addresses this limitation by reducing the numerical precision of network parameters.

Dynamic quantization converts floating-point weights into lower-precision integer representations, typically 8-bit integers. The quantized weight representation can be expressed as

$$w_q = round\left(\frac{w}{s}\right) + z$$

Where s represents a scaling factor and z denotes a zero-point offset used to map floating-point values to integer ranges.

This transformation significantly reduces memory consumption while maintaining computational efficiency during inference. Quantization techniques enabling integer-only neural inference were explored by Jacob et al. [4], demonstrating that substantial efficiency improvements can be achieved without significant loss of model accuracy.

Dynamic quantization is particularly well suited for inference-time optimization because weight conversion occurs after training, allowing compression to be applied without modifying the underlying training procedure.

3.5. Combined Compression Strategy

The effectiveness of the proposed pipeline arises from the complementary interaction between its components. Distillation enables compact models to learn meaningful feature representations, pruning removes structurally redundant parameters, and quantization reduces the storage cost of remaining weights. When applied sequentially, these techniques progressively reduce network complexity while preserving predictive capability.

In contrast to many existing compression frameworks that focus on a single optimization strategy, the pipeline presented here demonstrates that a simple integration of widely established techniques can produce highly compact neural models suitable for practical deployment. Experimental evaluation in the following section demonstrates that this approach achieves substantial reductions in model size while maintaining competitive classification accuracy.

4. Experimental Setup

4.1. Datasets

The experimental evaluation is conducted using two widely adopted benchmark datasets for image classification: MNIST and Fashion-MNIST. These datasets are frequently used for assessing neural

network compression strategies due to their standardized structure and reproducible evaluation protocols.

The MNIST dataset consists of grayscale handwritten digit images belonging to ten classes (digits 0–9). Each image has a spatial resolution of 28 x 28 pixels. The dataset contains 60,000 training samples and 10,000 testing samples. Despite its relative simplicity, MNIST provides a controlled experimental environment for evaluating compression frameworks and analysing accuracy–compression trade-offs.

Fashion-MNIST follows the same structural format as MNIST but contains images representing clothing categories such as shirts, trousers, and footwear. While the image dimensions and dataset sizes remain identical to MNIST, Fashion-MNIST presents a more challenging classification task due to greater intra-class similarity and higher visual complexity. Consequently, it provides a useful benchmark for evaluating the robustness of compression pipelines under moderately more complex visual distributions.

Prior to training, all images are normalized using standard preprocessing transformations. Each input image is converted into a tensor representation and normalized to stabilize training dynamics. The images are then flattened into vectors of length 784, corresponding to the 28 x 28 pixel input resolution.

4.2. Model Architectures

Two neural network architectures are used in the experiments: a larger teacher model and a compact student model.

The teacher network consists of a multilayer fully connected architecture with progressively decreasing hidden dimensions. This architecture provides sufficient representational capacity to achieve strong classification performance while maintaining architectural simplicity. The teacher model is composed of the following layers:

- Input layer: 784 units
- Hidden layer 1: 512 units
- Hidden layer 2: 256 units
- Hidden layer 3: 128 units
- Hidden layer 4: 64 units
- Output layer: 10 units

Rectified Linear Unit (ReLU) activations are used between hidden layers.

The student network is designed to represent a compact architecture suitable for deployment in resource-constrained environments. The student model consists of three fully connected layers:

- Input layer: 784 units
- Hidden layer 1: 128 units

- Hidden layer 2: 64 units
- Output layer: 10 units

This reduced architecture significantly lowers parameter count relative to the teacher network while maintaining sufficient representational capacity for classification tasks.

4.3. Training Procedure

Training is performed in three sequential stages corresponding to the proposed compression pipeline.

First, the teacher network is trained using standard supervised learning. The model is optimized using the Adam optimizer with a learning rate of 10^{-3}. Cross-entropy loss is used as the primary training objective. Training is conducted for multiple epochs until stable classification performance is achieved.

Second, knowledge distillation is applied to transfer information from the teacher network to the student model. During this stage, the student network learns from the soft probability distributions produced by the teacher model rather than relying solely on hard class labels. The distillation process uses a temperature parameter to soften probability distributions, enabling the student model to capture richer structural information from the teacher network.

Third, the trained student model undergoes network pruning followed by fine-tuning. Pruning removes low-magnitude weights from the network based on a predefined sparsity ratio. After pruning, the network is retrained for several additional epochs to recover potential accuracy degradation resulting from parameter removal.

Finally, dynamic quantization is applied to the pruned network to reduce parameter precision from 32-bit floating-point values to 8-bit integer representations. This step significantly reduces model storage requirements while preserving inference performance.

4.4. Evaluation Metrics

The effectiveness of the compression pipeline is evaluated using several quantitative metrics:

Classification Accuracy:

Accuracy measures the proportion of correctly classified samples within the test dataset.

$$Accuracy = \frac{No.\ of\ Correct\ Predictions}{Total\ No.\ of\ Samples}$$

Model Size:

Model size is estimated based on the number of parameters and the numerical precision used to store weights.

Compression Ratio:

Compression ratio quantifies the reduction in storage achieved by the compressed model relative to the original teacher model.

$$Compression\ Ratio = \frac{Size_{original}}{Size_{compressed}}$$

These metrics provide a comprehensive assessment of both predictive performance and compression efficiency.

4.5. Implementation Details

All experiments are implemented using the PyTorch deep learning framework. Training is performed using GPU acceleration when available, while quantized models are evaluated on CPU environments consistent with typical edge deployment scenarios.

Batch sizes of 128 samples are used during training, while evaluation is conducted using batches of 256 samples. Dynamic quantization is applied to fully connected layers during the final compression stage. The complete experimental pipeline is designed to ensure reproducibility and facilitate straightforward replication of results.

5. Results

5.1. Overall Compression Performance

The effectiveness of the proposed compression pipeline was evaluated using two benchmark datasets: MNIST and Fashion-MNIST. The experiments compare four configurations: the original teacher network, the distilled student model, the pruned network, and the final quantized model. The objective of this evaluation is to assess the trade-off between predictive accuracy and model compactness as successive compression stages are applied.

The experimental results indicate that the proposed pipeline achieves substantial reductions in model size while maintaining competitive classification accuracy. On the MNIST dataset, the final quantized model achieves an approximate 21 times reduction in model storage relative to the original teacher network. Importantly, this compression is obtained with negligible loss in predictive accuracy. The classification accuracy of the quantized model remains comparable to that of the teacher network, indicating that the sequential integration of distillation, pruning, and quantization does not significantly degrade the learned decision boundaries.

A similar pattern is observed for the Fashion-MNIST dataset. Despite the greater visual complexity of the dataset, the compressed model maintains accuracy close to the teacher network. This result suggests that the proposed pipeline preserves useful feature representations even after aggressive reductions in parameter count and numerical precision.

Table 1
summarizes the overall performance trends observed during compression.

Model Stage	Accuracy (MNIST)	Accuracy (Fashion-MNIST)	Relative Model Size
Teacher Network	Baseline accuracy	Baseline accuracy	1x
Distilled Student	Slightly reduced	Slightly Reduced	~ 0.5x
Pruned Network	Comparable to student	Comparable	~ 0.3x
Quantized Model	Near-baseline accuracy	Near-baseline accuracy	~ 0.05x

Table 1: **Classification accuracy and compression ratio across different pipeline stages.**

These results demonstrate that the compression pipeline effectively balances model compactness and predictive accuracy.

As illustrated in Chart 1 and Chart 2, the compressed model maintains accuracy comparable to the teacher network.

5.2 Role of Knowledge Distillation

Knowledge distillation serves as the first stage of the compression process by transferring representational knowledge from the teacher model to the compact student network. The original formulation proposed by Hinton et al. [1] showed that soft probability distributions produced by a teacher network contain richer information than hard class labels alone. By training the student network to match these softened outputs, the student can approximate the teacher's internal decision structure.

The results obtained in this study align with prior findings indicating that distillation enables compact networks to achieve competitive performance relative to larger models. Similar observations have been reported in several subsequent studies investigating efficient neural architectures and model compression strategies [5], [6]. These works demonstrate that distillation improves generalization in compact networks by providing additional supervisory signals during training.

In the present experiments, the distilled student model retains classification accuracy close to that of the teacher network despite having substantially fewer parameters. This observation highlights the effectiveness of distillation in mitigating the representational limitations of compact neural architectures.

5.3. Impact of Network Pruning

Following distillation, network pruning is applied to remove redundant parameters. Neural networks are known to contain significant structural redundancy, and many connections can be removed without substantially affecting predictive performance. Early investigations into parameter redundancy were conducted by LeCun et al. [2], who demonstrated that neural networks often contain unnecessary weights that can be eliminated without degrading model accuracy.

More recent work has further explored the effectiveness of pruning for neural network compression. Han et al. [3] demonstrated that large neural networks can be pruned aggressively while maintaining high predictive performance. Subsequent studies have explored both structured and unstructured pruning strategies for reducing model complexity [7], [8].

In the experiments presented here, pruning removes a large fraction of the student network's parameters based on weight magnitude. Interestingly, the pruned model occasionally exhibits performance comparable to or slightly better than the distilled student model. This behaviour is consistent with prior findings suggesting that pruning can introduce a regularization effect by eliminating noisy or redundant connections [3], [7]. Such regularization may encourage the network to rely on more informative feature representations.

5.4. Quantization and Storage Efficiency

Dynamic quantization constitutes the final stage of the compression pipeline and is responsible for the largest reduction in model storage requirements. Quantization reduces the precision of neural parameters by converting floating-point weights into lower-precision integer representations. This approach has been widely adopted in efficient inference frameworks for neural networks [4], [9].

Previous work has demonstrated that carefully designed quantization strategies can significantly reduce model size without substantial loss in predictive accuracy [4], [10]. The results obtained in this study confirm these observations. After quantization, the compressed model retains classification accuracy comparable to the pruned network while achieving a substantial reduction in storage requirements.

Because quantization is applied after pruning, fewer parameters remain in the model, allowing the reduction in numerical precision to produce an even greater improvement in storage efficiency. This synergy between pruning and quantization has been previously observed in neural compression pipelines and is widely regarded as an effective strategy for enabling efficient edge deployment [3], [9].

Chart 3 and Chart 4 demonstrates the estimated model size for MNIST and Fashion-MNIST respectively.

Chart 4: Comparison of Model Size in each stage for Fashion-MNIST Dataset

5.5. Comparison with Existing Compression Strategies

The experimental findings demonstrate that combining multiple compression techniques can significantly improve neural network efficiency while preserving predictive performance. Previous studies have explored various compression strategies, including low-precision networks [10], binary neural networks [11], and sparsity-based optimization techniques [7]. While these approaches often achieve substantial reductions in computational cost, they may require specialized architectures or complex training procedures.

In contrast, the compression pipeline investigated in this work emphasizes simplicity and reproducibility. The integration of distillation, pruning, and dynamic quantization leverages well-established techniques

that can be implemented using standard deep learning frameworks without requiring architectural modifications. This design choice improves the practical applicability of the proposed pipeline, particularly for developers seeking straightforward methods for deploying neural models in constrained environments.

The experimental results therefore suggest that hybrid compression pipelines represent a promising approach for enabling efficient deployment of neural networks in edge computing scenarios. By combining complementary optimization strategies, compact neural models can be constructed that significantly reduce storage requirements while preserving predictive accuracy.

6. Conclusion

The rapid expansion of artificial intelligence toward edge and embedded platforms has intensified the need for neural models that balance predictive capability with strict resource constraints. Although modern deep learning architectures achieve impressive performance across a wide range of tasks, their large parameter counts and high memory requirements often limit practical deployment on resource-constrained systems. Addressing this challenge requires effective strategies for reducing model complexity while preserving predictive performance.

This study investigated a lightweight neural compression pipeline that integrates knowledge distillation, network pruning, and dynamic quantization within a unified framework. Each stage of the pipeline contributes a complementary form of efficiency improvement: distillation enables compact networks to capture informative representations from larger models, pruning removes redundant parameters, and quantization reduces the storage cost of the remaining weights. When applied sequentially, these techniques produce substantial reductions in model size while maintaining stable classification performance.

Experimental evaluation on benchmark datasets, including MNIST and Fashion-MNIST, demonstrates that the proposed compression strategy achieves approximately twenty-fold reduction in model storage requirements with minimal degradation in predictive accuracy. These results confirm that carefully integrated compression techniques can produce compact neural models that remain suitable for real-world inference tasks. The compressed models retain performance comparable to the original teacher networks despite drastic reductions in parameter count and numerical precision.

Beyond the quantitative improvements in model size, the results highlight the broader feasibility of constructing efficient neural inference pipelines without requiring complex architectural modifications. Many existing compression strategies rely on specialized network designs or intricate training procedures. In contrast, the pipeline examined in this study leverages widely available techniques that can be implemented within standard deep learning frameworks, thereby improving reproducibility and accessibility.

The findings therefore reinforce the growing consensus that hybrid compression strategies represent a practical pathway for deploying deep learning systems in resource-constrained environments. By combining complementary efficiency mechanisms, it becomes possible to significantly reduce storage requirements while preserving useful predictive capabilities.

Future work may explore extending this compression framework to more complex neural architectures and larger datasets. Additional investigation into adaptive pruning strategies, hardware-aware quantization schemes, and automated compression policies could further improve the efficiency of neural models designed for edge computing environments. As artificial intelligence continues to expand beyond centralized computing infrastructures, the development of compact and efficient neural architectures will remain a critical research priority.

Overall, the compression pipeline investigated in this study demonstrates that substantial reductions in neural model size can be achieved through the careful integration of established optimization techniques. Such approaches offer a practical foundation for enabling efficient deployment of machine learning systems in environments where computational resources and memory capacity are inherently limited.

Limitations

Despite the promising results demonstrated by the proposed compression pipeline, several limitations should be acknowledged.

First, the experimental evaluation is conducted on relatively small-scale benchmark datasets, namely MNIST and Fashion-MNIST. Although these datasets are widely used for validating compression strategies and provide controlled environments for reproducible experimentation, they do not fully capture the complexity of real-world visual recognition tasks. Larger datasets with higher spatial resolution and greater semantic diversity may present additional challenges for compression pipelines. Consequently, further evaluation on more complex datasets would be necessary to comprehensively assess the generalizability of the proposed framework.

Second, the neural architectures considered in this study are relatively simple fully connected networks. While such architectures provide a clear experimental setting for analysing compression behaviour, many modern computer vision systems rely on convolutional or transformer-based architectures. Compression strategies that perform well on compact multilayer perceptron may exhibit different behaviour when applied to deeper and more complex architectures. Extending the proposed pipeline to convolutional networks or attention-based models therefore represents an important direction for future investigation.

Third, the compression pipeline relies on sequential integration of established techniques rather than introducing a fundamentally new compression algorithm. Although the results demonstrate that combining distillation, pruning, and quantization can produce substantial reductions in model size, the

pipeline primarily represents an engineering integration of existing methods. As such, the approach should be interpreted as a practical deployment framework rather than a novel theoretical contribution to neural network compression.

Another limitation concerns the evaluation of computational efficiency. While the study focuses primarily on storage reduction and classification accuracy, additional performance metrics such as inference latency, energy consumption, and hardware-specific optimization were not explicitly analysed. These factors are often critical for real-world edge deployments and may influence the practical impact of compression strategies.

Finally, the pruning strategy employed in this work relies on magnitude-based weight removal. Although this approach is widely used and computationally efficient, it does not explicitly account for layer-wise sensitivity or structural dependencies within the network. More sophisticated pruning methods, including structured pruning or adaptive sparsity learning, may yield improved compression–accuracy trade-offs.

Despite these limitations, the findings presented in this study demonstrate that substantial reductions in neural model size can be achieved through the careful integration of widely established compression techniques. The proposed framework therefore serves as a practical foundation for exploring lightweight neural inference pipelines in resource-constrained environments.

Declarations

Code Availability and Reproducibility

To facilitate transparency and reproducibility, the implementation of the compression pipeline used in this study is publicly available through the open-access research repository Zenodo. The archived repository contains the complete source code required to reproduce the experiments presented in this work, including scripts for model training, knowledge distillation, pruning procedures, quantization, and evaluation.

The repository also provides the experimental configurations used in the reported results, enabling independent verification of the compression pipeline. All experiments were implemented using the PyTorch deep learning framework, and the repository includes detailed instructions describing the steps required to replicate the experimental setup.

The archived dataset preprocessing procedures, model architectures, and training scripts are documented within the repository to ensure reproducibility across different computational environments. In addition, the repository contains example outputs and figure-generation scripts used to produce the plots presented in the Results section.

The code archive associated with this study can be accessed at the following DOI:

Zenodo DOI: https://doi.org/10.5281/zenodo.19373452

Providing an openly accessible implementation aims to encourage further experimentation and facilitate future research on neural network compression techniques for resource-constrained environments.

Competing Interests:

No competing interests were identified in relation to the content, methodology, or findings of this study. The research was carried out independently, without financial, commercial, or institutional influence. The objective was solely to advance academic understanding of emergent behaviours in language models.

Funding Information:

No funding was received from any government institution, commercial organization, or non-profit entity. The study was conducted out of independent academic interest and was entirely self-funded.

Author Contribution:

The conception of the idea, design of the methodology, execution of the experiments, data collection, analysis, and manuscript preparation were all carried out independently by the author. No external contributors were involved in any phase of the research process.

Data Availability Statement:

All research data generated during the study. The complete archive has been made publicly available to ensure transparency and reproducibility. The dataset can be accessed at: https://doi.org/10.5281/zenodo.19373452.

Research Involving Humans and/or Animals:

No experiments involving human participants or animals were conducted in the course of this study.

Informed Consent:

Informed consent was not applicable, as no human participants were involved in the study.

References

1. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint. 10.48550/arXiv.1503.02531
2. LeCun Y, Denker J, Solla S (1990) Optimal brain damage. Adv Neural Inf Process Syst. 10.5555/109230.109298
3. Han S, Mao H, Dally W (2016) Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. International Conference on Learning Representations (ICLR) 10.48550/arXiv.1510.00149

4. Jacob B, Kligys S, Chen B et al (2018) Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 10.1109/CVPR.2018.00286
5. Buciluǎ C, Caruana R, Niculescu-Mizil A, Model compression. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006). 10.1145/1150402.1150464
6. Romero A, Ballas N, Kahou S et al (2015) FitNets: Hints for thin deep nets. International Conference on Learning Representations (ICLR) 10.48550/arXiv.1412.6550
7. Gale T, Elsen E, Hooker S (2019) The state of sparsity in deep neural networks. arXiv preprint. 10.48550/arXiv.1902.09574
8. Molchanov P, Tyree S, Karras T et al (2017) Pruning convolutional neural networks for resource efficient inference. International Conference on Learning Representations 10.48550/arXiv.1611.06440
9. Banner R, Nahshan Y, Hoffer E, Soudry D (2019) Post training 4-bit quantization of convolutional networks for rapid deployment. Adv Neural Inf Process Syst. 10.48550/arXiv.1810.05723
10. Courbariaux M, Bengio Y, David J, BinaryConnect (2015) Training deep neural networks with binary weights during propagations. Adv Neural Inf Process Syst. 10.48550/arXiv.1511.00363
11. Rastegari M, Ordonez V, Redmon J, Farhadi A, XNOR-Net, European Conference on Computer Vision (2016) ImageNet classification using binary convolutional neural networks. DOI: 10.1007/978-3-319-46493-0_49

Chart 1 To 4

Chart 1 To 4 are available in the Supplementary Files section.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Charts.docx