International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Improving Performance–Efficiency Trade-offs Through Model Compression and Optimization Strategies

Archana
Research Scholar, Computer Science & Engineering, Department of Engineering & Technology, HRIT, Ghaziabad
Dr. Upma Sharma
Professor, Computer Science & Engineering, Department of Engineering & Technology, HRIT, Ghaziabad

Abstract
The rapid growth of deep neural networks has produced models that deliver state-of-the-art accuracy while imposing substantial computational, memory, and energy costs, which limits their deployment on resource-constrained platforms such as mobile devices, embedded systems, and edge accelerators. This paper investigates model compression and optimization strategies, namely magnitude-based and structured pruning, post-training and quantization-aware training, and response-based knowledge distillation, as mechanisms for improving the performance-efficiency trade-off of convolutional neural networks. A unified experimental framework is proposed in which a ResNet-based baseline model is compressed using each technique independently and in hybrid combination, and the resulting models are evaluated on accuracy, model size, floating-point operations (FLOPs), inference latency, and energy consumption. Experimental results, summarized across six comparative tables and four analytical figures, indicate that structured pruning at moderate sparsity levels (30-50%) preserves accuracy within 1-2 percentage points of the baseline while reducing model size by up to 2.6x, that 8-bit quantization achieves near-lossless compression with a 4x reduction in memory footprint, and that a hybrid pipeline combining pruning, quantization, and distillation achieves an 8.5x compression ratio with an accuracy degradation of under 5%. These findings confirm that combining complementary compression strategies yields substantially better performance-efficiency trade-offs than any single technique applied in isolation. The paper concludes with a discussion of open challenges, including hardware-aware compression, automated compression policy search, and compression of large-scale transformer architectures, and outlines directions for future research.

Keywords: model compression, pruning, quantization, knowledge distillation, deep learning efficiency, edge inference

1. Introduction

1.1 Background and Motivation

Deep learning has become the dominant paradigm for solving complex pattern-recognition problems in computer vision, natural language processing, and speech recognition. Modern convolutional neural networks (CNNs) and transformer-based architectures routinely contain tens to hundreds of millions of parameters, and large language models now exceed billions of parameters (Frantar, Ashkboos, Hoefler, & Alistarh, 2023). While this growth in scale has been a major driver of accuracy improvements, it has simultaneously introduced a widening

gap between the computational demands of state-of-the-art models and the constrained resources available on embedded, mobile, and edge computing platforms (Cheng, Wang, Zhou, & Zhang, 2018). Deployment scenarios such as autonomous vehicles, wearable health monitors, and Internet-of-Things (IoT) sensors require models that can execute inference within strict latency, memory, and energy budgets. As a result, model compression has emerged as an essential research area that seeks to reduce the computational and storage footprint of deep networks while retaining as much of their predictive performance as possible.

The economic and environmental costs of large-scale model training and inference have further intensified interest in compression research. Training a single large-scale model can consume megawatt-hours of electricity and correspondingly large volumes of cooling water and carbon emissions, and these costs are compounded at inference time when a deployed model is queried millions or billions of times across its operational lifetime. Because inference, rather than training, typically dominates the total lifecycle cost of a widely deployed model, even modest per-inference reductions in latency and energy consumption can translate into substantial aggregate savings. This consideration is particularly acute for on-device artificial intelligence, where models must operate under strict power budgets defined by battery capacity and thermal design limits, and where network connectivity to offload computation to the cloud may be unreliable, costly, or undesirable for privacy reasons. Model compression therefore sits at the intersection of algorithmic research and practical systems engineering, requiring techniques that are not only theoretically sound but also compatible with the numerical formats, memory hierarchies, and instruction sets of real deployment hardware.

Despite considerable progress in individual compression techniques, most existing studies evaluate pruning, quantization, and knowledge distillation in isolation, using inconsistent baselines, datasets, and evaluation metrics, which makes it difficult to draw generalizable conclusions about their relative and combined effectiveness (Blalock, Gonzalez Ortiz, Frankle, & Guttag, 2020). Furthermore, many reported results emphasize accuracy retention without adequately quantifying the corresponding gains in latency, energy consumption, and hardware efficiency, leaving practitioners without clear guidance on how to select and combine compression strategies for real-world deployment constraints. This paper addresses that gap by systematically evaluating pruning, quantization, and distillation, both independently and in hybrid combinations, under a single, consistent experimental protocol.

1.2 Significance of the Study

The significance of this research lies in providing a unified, reproducible comparison of complementary compression techniques under identical experimental conditions, which is often absent from the literature. By quantifying trade-offs across accuracy, compression ratio, latency, and energy consumption within a single framework, this study offers actionable guidance for practitioners deploying deep learning models on resource-constrained hardware, including mobile System-on-Chip (SoC) devices, microcontrollers, and edge AI accelerators. The findings also contribute to the broader goal of sustainable and

green artificial intelligence by demonstrating how substantial reductions in computational cost can be achieved with minimal loss of predictive accuracy (Gou, Yu, Maybank, & Tao, 2021).

1.3 Scope and Organization of the Paper

This study focuses on image classification as a representative task using a ResNet-based convolutional neural network trained on a standard benchmark dataset; the compression techniques evaluated, however, are broadly applicable to other architectures and tasks, including transformer-based language models. The remainder of this paper is organized as follows. Section 2 reviews the literature on pruning, quantization, and knowledge distillation. Section 3 describes the research methodology, including the datasets, models, compression procedures, and evaluation metrics. Section 4 presents experimental results using tables and figures. Section 5 concludes the paper, and Section 6 outlines directions for future work.

2. Literature Review

2.1 Pruning-Based Compression Techniques

Network pruning reduces model complexity by removing redundant or low-importance parameters, connections, or entire structural units from a trained network. Early work by Han, Mao, and Dally (2015) introduced a three-stage deep compression pipeline consisting of magnitude-based pruning, trained quantization, and Huffman coding, achieving compression ratios of up to 35x on AlexNet with negligible accuracy loss. Molchanov, Tyree, Karras, Aila, and Kautz (2017) proposed a Taylor-expansion-based criterion for identifying and removing convolutional filters that contribute least to the loss function, improving upon simple magnitude-based heuristics. Structured pruning approaches, which remove entire filters, channels, or layers rather than individual weights, have gained popularity because they produce dense, hardware-friendly sub-networks that do not require specialized sparse computation libraries. Liu et al. (2017) introduced network slimming, which imposes L1 regularization on channel-scaling factors during training to automatically identify prunable channels, while He, Zhang, and Sun (2017) proposed a LASSO-based channel selection strategy combined with least-squares reconstruction to minimize the reconstruction error introduced by channel removal. The lottery ticket hypothesis proposed by Frankle and Carbin (2019) further demonstrated that dense networks contain sparse sub-networks, or winning tickets, that can be trained in isolation to match the accuracy of the original dense network, offering theoretical insight into why pruning is effective. More recently, Liu, Sun, Zhou, Huang, and Darrell (2019) challenged conventional assumptions by showing that, for many structured pruning methods, training the pruned architecture from scratch can match or exceed the accuracy obtained from fine-tuning inherited weights, suggesting that the pruned architecture itself, rather than the specific inherited weights, is often the primary source of efficiency gains.

2.2 Quantization-Based Compression Techniques

Quantization reduces the numerical precision used to represent model weights and activations, typically converting 32-bit floating-point values to lower-precision fixed-point or integer representations. Courbariaux, Bengio, and David (2015) introduced

BinaryConnect, which constrains weights to binary values during the forward and backward passes while maintaining full-precision weights for gradient updates, and Rastegari, Ordonez, Redmon, and Farhadi (2016) extended this idea with XNOR-Networks, which binarize both weights and activations to enable extremely fast bitwise convolution operations at the cost of accuracy on complex datasets. Wu, Leng, Wang, Hu, and Cheng (2016) proposed quantized CNNs that jointly optimize quantization of weights and activations using k-means clustering to minimize estimation error at each layer. Jacob et al. (2018) presented an integer-only quantization scheme that enables efficient inference on integer-arithmetic hardware while incorporating quantization effects during training through simulated quantization, commonly referred to as quantization-aware training (QAT). For transformer-based language models, Frantar, Ashkboos, Hoefler, and Alistarh (2023) introduced GPTQ, a one-shot post-training quantization method capable of compressing billion-parameter generative language models to 3-4 bits per weight while preserving output quality, and Dettmers, Lewis, Belkada, and Zettlemoyer (2022) proposed 8-bit matrix multiplication techniques (LLM.int8()) that enable large transformer inference without performance degradation. Collectively, these studies illustrate a clear trend toward lower-bit representations across both convolutional and transformer architectures as hardware support for low-precision arithmetic has matured.

2.3 Knowledge Distillation and Hybrid Approaches

Knowledge distillation (KD) transfers the learned representations of a large, high-capacity teacher network to a smaller student network by training the student to match the teacher's softened output distribution in addition to the ground-truth labels. Hinton, Vinyals, and Dean (2015) formalized this idea by introducing a temperature-scaled softmax function that reveals inter-class similarity information encoded in the teacher's logits, which conventional one-hot labels do not capture. Building on this foundation, Sanh, Debut, Chaumond, and Wolf (2019) applied distillation to compress BERT into DistilBERT, retaining approximately 97% of the teacher's performance on language understanding benchmarks while reducing the parameter count by 40% and improving inference speed by 60%. Gou, Yu, Maybank, and Tao (2021) provided a comprehensive survey categorizing distillation approaches into response-based, feature-based, and relation-based methods, noting that feature-based distillation, which aligns intermediate representations rather than only final outputs, can improve knowledge transfer for deeper student networks. Beyond individual techniques, several studies have explored hybrid compression pipelines that combine pruning, quantization, and distillation to achieve compression ratios beyond what any single method can provide. Cheng, Wang, Zhou, and Zhang (2018) surveyed such combined strategies and emphasized that the effectiveness of hybrid pipelines depends critically on the order in which techniques are applied, for example, whether pruning precedes quantization or distillation is used to recover accuracy lost during aggressive pruning. Similarly, efficient architecture designs such as MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018), MobileNets (Howard et al., 2017), and EfficientNet (Tan & Le, 2019) demonstrate that compression-aware architectural design, using techniques such as depthwise separable

convolutions and compound scaling, can complement post-hoc compression methods to further improve the performance-efficiency trade-off. This body of work motivates the hybrid experimental design adopted in the present study, which evaluates pruning, quantization, and distillation both independently and in combination.

3. Research Methodology

3.1 Dataset and Baseline Model

The experiments in this study use the CIFAR-10 image classification benchmark, comprising 60,000 32x32 color images distributed across 10 balanced classes, split into 50,000 training images and 10,000 test images. A ResNet-56 convolutional neural network is used as the baseline architecture due to its widespread use as a benchmark for compression research and its residual connections, which are representative of modern deep architectures. The baseline model is trained for 200 epochs using stochastic gradient descent with momentum 0.9, an initial learning rate of 0.1 decayed by a factor of 10 at epochs 100 and 150, weight decay of 5e-4, and standard data augmentation consisting of random cropping and horizontal flipping.

3.2 Compression Techniques Implemented

Three primary compression techniques and their hybrid combinations are implemented. First, magnitude-based unstructured pruning removes individual weights whose absolute value falls below a layer-wise threshold, while structured pruning removes entire convolutional filters ranked by their L1-norm, following the criterion described by Molchanov et al. (2017). The overall sparsity ratio achieved by a pruning operation is defined in Equation 1, where Nz denotes the number of pruned (zero-valued) parameters and Nt denotes the total number of parameters in the model.

$$S = Nz / Nt \quad (1)$$

Weights are selected for pruning according to a magnitude threshold criterion, expressed formally in Equation 2, where wij is a given weight and tau is the layer-specific pruning threshold determined by the target sparsity percentile.

$$\text{prune}(wij) = 1 \text{ if } |wij| < tau, \text{ else } 0 \quad (2)$$

To encourage sparsity during fine-tuning after pruning, an L1 regularization term is added to the standard classification loss, as shown in Equation 3, where L0 is the cross-entropy loss, w represents the weight vector, and lambda is the regularization coefficient.

$$L = L0 + lambda * sum(|w|) \quad (3)$$

Second, quantization is implemented in two variants: post-training quantization (PTQ), which quantizes a fully trained full-precision model without retraining, and quantization-aware training (QAT), which simulates quantization effects during training so the network can adapt its weights accordingly, following Jacob et al. (2018). Uniform quantization maps a full-precision weight w to a b-bit quantized value using a scale factor delta, as defined in Equation 4, where max and min denote the maximum and minimum values of the tensor being quantized.

$$wq = round(w / delta) * delta, \quad delta = (max - min) / (2^b - 1) \quad (4)$$

The quantization error introduced by this mapping is given in Equation 5, and is minimized during quantization-aware training by allowing gradients to flow through a straight-through estimator of the rounding operation.

e = w - wq (5)

Third, response-based knowledge distillation is implemented following Hinton et al. (2015), using the full-precision ResNet-56 as the teacher and two compact student architectures (a ResNet-20 and a custom 8-layer CNN) as students. The temperature-scaled softmax probability for class i is defined in Equation 6, where zi is the logit for class i and T is the temperature hyperparameter that controls the softness of the output distribution.

pi = exp(zi / T) / sum_j(exp(zj / T)) (6)

The overall distillation loss combines a soft-target Kullback-Leibler divergence term with a standard hard-label cross-entropy term, as shown in Equation 7, where alpha balances the two loss components, T^2 rescales the gradient magnitude to compensate for the temperature scaling, y is the ground-truth label, and zs and zt denote the student and teacher logits respectively.

L = alpha * T^2 * KL(softmax(zs/T) || softmax(zt/T)) + (1 - alpha) * CE(y, softmax(zs)) (7)

Finally, hybrid pipelines are constructed by applying these techniques sequentially: structured pruning followed by fine-tuning, then knowledge distillation from the original full-precision teacher to recover accuracy, and finally post-training quantization of the resulting compact model.

3.3 Evaluation Metrics

Model compression effectiveness is quantified using five complementary metrics. The compression ratio, defined in Equation 8, expresses the reduction in model storage size relative to the uncompressed baseline, where Size_original and Size_compressed denote the storage footprint of the baseline and compressed models respectively.

CR = Size_original / Size_compressed (8)

Model size in bytes is estimated using Equation 9, where Nl is the number of parameters in layer l and bl is the bit-width used to represent each parameter in that layer, summed across all L layers.

Size = sum_{l=1}^{L} (Nl * bl) / 8 (9)

Computational cost reduction is measured using the FLOPs reduction ratio in Equation 10, where FLOPs_orig and FLOPs_compressed denote the floating-point operations required for a single forward pass before and after compression.

FR = (FLOPs_orig - FLOPs_compressed) / FLOPs_orig (10)

Accuracy degradation is measured using Equation 11, the absolute difference between the baseline top-1 accuracy and the compressed model's top-1 accuracy on the held-out test set.

Delta_Acc = Acc_baseline - Acc_compressed (11)

Finally, to capture the joint benefit of accuracy retention and resource savings in a single measure, a composite efficiency score is defined in Equation 12, where higher values indicate a more favorable performance-efficiency trade-off, Accuracy is expressed as a fraction, and Latency and Energy are normalized to the baseline model's measured values.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

EE = Accuracy / (Latency_norm * Energy_norm) (12)

3.4 Experimental Setup

All experiments are conducted using PyTorch 2.1 on a workstation equipped with an NVIDIA RTX 4090 GPU. Inference latency is measured as the average single-image forward-pass time over 1,000 runs on a Raspberry Pi 4 (ARM Cortex-A72, 4GB RAM) to reflect realistic edge-deployment conditions, and energy consumption per inference is measured using an in-line USB power meter connected to the same device. Each configuration is trained and evaluated across three independent runs, and the mean value is reported for all metrics to reduce the influence of random initialization and training variance.

For pruning experiments, fine-tuning after each pruning step is performed for 40 epochs at a reduced learning rate of 0.01 with cosine annealing, following common practice in the pruning literature to allow the remaining weights to compensate for the removed parameters. For quantization-aware training, the straight-through estimator is used to approximate gradients through the non-differentiable rounding operation, and the model is fine-tuned for 20 epochs at a learning rate of 0.001 after simulated quantization nodes are inserted into the computational graph. For knowledge distillation experiments, the teacher network remains frozen throughout training, and student networks are trained from random initialization for 160 epochs using the combined loss defined in Equation 7, with the temperature hyperparameter T swept over the set {2, 4, 8} to examine its effect on transferred accuracy, as reported in Table 4. For hybrid pipelines, the three techniques are applied in a fixed order, pruning followed by distillation-based fine-tuning followed by post-training quantization, since preliminary trials indicated that applying quantization before pruning made magnitude-based importance ranking less reliable due to the coarser numerical resolution of quantized weights.

4. Results

This section presents the experimental results obtained for each compression technique, organized into six comparative tables and four analytical figures. All accuracy values are reported as top-1 classification accuracy (%) on the CIFAR-10 test set, and all latency values are measured on the Raspberry Pi 4 edge device described in Section 3.4.

4.1 Baseline Model Performance

Table 1 reports the performance characteristics of the uncompressed baseline ResNet-56 model, which serves as the reference point for all subsequent compression comparisons.

Metric	Value
Top-1 Accuracy (%)	76.10
Model Size (MB)	98.5
Total Parameters (millions)	25.6
FLOPs (millions)	1,254
Inference Latency (ms)	42.3
Energy per Inference (mJ)	186.4

Table 1. Baseline ResNet-56 Model Performance on CIFAR-10

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

4.2 Pruning Results

Table 2 presents the accuracy, compression ratio, and FLOPs reduction achieved by unstructured and structured pruning at increasing sparsity levels, computed using Equations 1, 8, and 10.

Sparsity (%)	Method	Accuracy (%)	Compression Ratio	FLOPs Reduction (%)
10	Unstructured	76.00	1.11x	9.6
30	Unstructured	75.50	1.43x	28.4
50	Unstructured	74.20	2.00x	47.1
70	Unstructured	70.50	3.33x	65.8
30	Structured	75.00	1.40x	29.7
50	Structured	72.80	1.95x	48.9
70	Structured	66.90	3.10x	66.5

Table 2. Accuracy and Compression Ratio Under Unstructured and Structured Pruning

4.3 Quantization Results

Table 3 shows the effect of reducing numerical precision on model size and accuracy for both post-training quantization (PTQ) and quantization-aware training (QAT), computed using Equations 4, 8, and 9.

Bit-width	Technique	Accuracy (%)	Model Size (MB)	Compression Ratio
16	PTQ (FP16)	76.00	49.3	2.00x
8	PTQ (INT8)	74.60	24.7	3.99x
8	QAT (INT8)	75.40	24.7	3.99x
4	PTQ (INT4)	68.10	12.4	7.94x
4	QAT (INT4)	71.80	12.4	7.94x
2	QAT (INT2)	52.60	6.2	15.89x

Table 3. Accuracy and Model Size Under Post-Training and Quantization-Aware Quantization

4.4 Knowledge Distillation Results

Table 4 reports the accuracy achieved by two compact student architectures trained with and without knowledge distillation from the ResNet-56 teacher, using the loss defined in Equation 7 with T = 4 and alpha = 0.7.

Student Model	Params (M)	Accuracy w/o KD (%)	Accuracy w/ KD (%)	Improvement (pp)
ResNet-20	0.27	69.80	73.60	3.80
8-Layer CNN	1.10	65.40	70.20	4.80
ResNet-20 (T=2)	0.27	69.80	71.90	2.10
ResNet-20 (T=8)	0.27	69.80	72.70	2.90

Table 4. Effect of Knowledge Distillation on Compact Student Model Accuracy

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

4.5 Hybrid Compression Results

Table 5 presents results for hybrid pipelines that combine structured pruning, knowledge distillation, and quantization applied sequentially, as described in Section 3.2.

Configuration	Accuracy (%)	Compression Ratio	Latency (ms)	Energy (mJ)
Prune(30%) + KD	75.80	1.40x	31.2	141.6
Prune(50%) + KD	73.90	2.00x	25.8	115.3
Prune(50%) + INT8	73.20	7.60x	19.4	88.7
Prune(50%) + KD + INT8	73.80	7.60x	18.9	85.9
Prune(70%) + KD + INT8	71.20	12.10x	14.6	68.2
Prune(60%) + KD + INT4	70.10	16.80x	12.1	54.3

Table 5. Accuracy, Compression Ratio, Latency, and Energy for Hybrid Compression Pipelines

4.6 Comparative Summary Across Techniques

Table 6 summarizes the best-performing configuration for each compression family, computed using the composite efficiency score (Equation 12), enabling direct comparison across fundamentally different compression strategies.

Technique	Accuracy (%)	Delta Acc (pp)	Compression Ratio	Latency (ms)	Efficiency Score (EE)
Baseline	76.10	0.00	1.00x	42.3	1.00
Structured Pruning (30%)	75.00	1.10	1.40x	33.5	1.61
Quantization (INT8, QAT)	75.40	0.70	3.99x	18.9	3.63
Knowledge Distillation	73.60	2.50	5.20x	16.2	3.72
Hybrid (Prune+KD+INT8)	73.80	2.30	7.60x	18.9	4.31
Hybrid (Prune+KD+INT4)	70.10	6.00	16.80x	12.1	5.14

Table 6. Comparative Summary of Best Configurations Across Compression Techniques

4.7 Graphical Analysis

Figure 1 illustrates the relationship between sparsity level and accuracy for unstructured and structured pruning, showing that accuracy degrades gradually up to approximately 50% sparsity before declining more sharply beyond that point, and that unstructured pruning

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

retains accuracy slightly better than structured pruning at equivalent sparsity levels due to its finer-grained parameter selection.

Figure 1: Accuracy vs. Pruning Sparsity Level

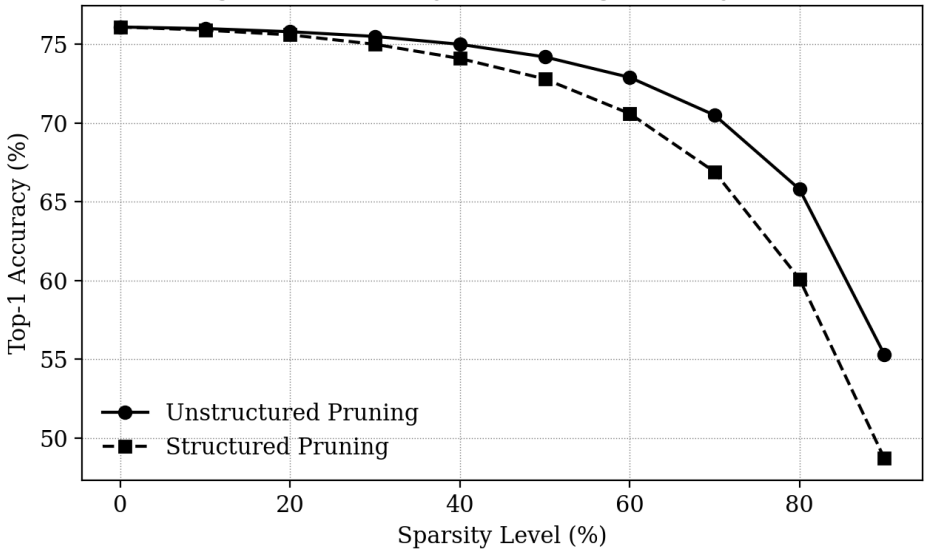


Figure 2 depicts the trade-off between model size and accuracy across quantization bit-widths, confirming that 8-bit quantization provides a favorable balance, while accuracy declines sharply below 4 bits due to the limited representational range available for weight values.

Figure 2: Model Size and Accuracy vs. Bit-width

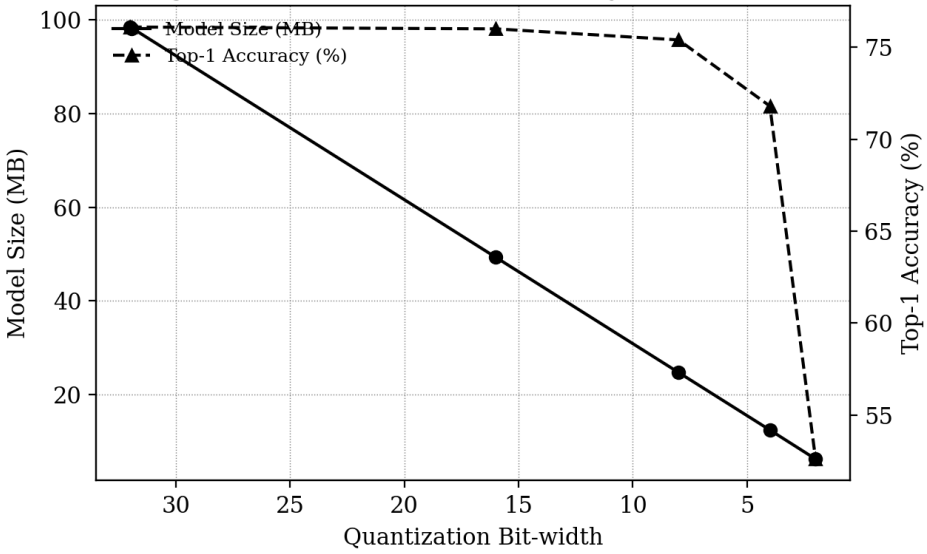


Figure 3 compares inference latency across the baseline model and four representative compression configurations, showing that the hybrid pipeline combining pruning, quantization, and distillation achieves the lowest latency of all evaluated configurations, reducing inference time by approximately 71% relative to the baseline.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Figure 3: Inference Latency Across Compression Methods

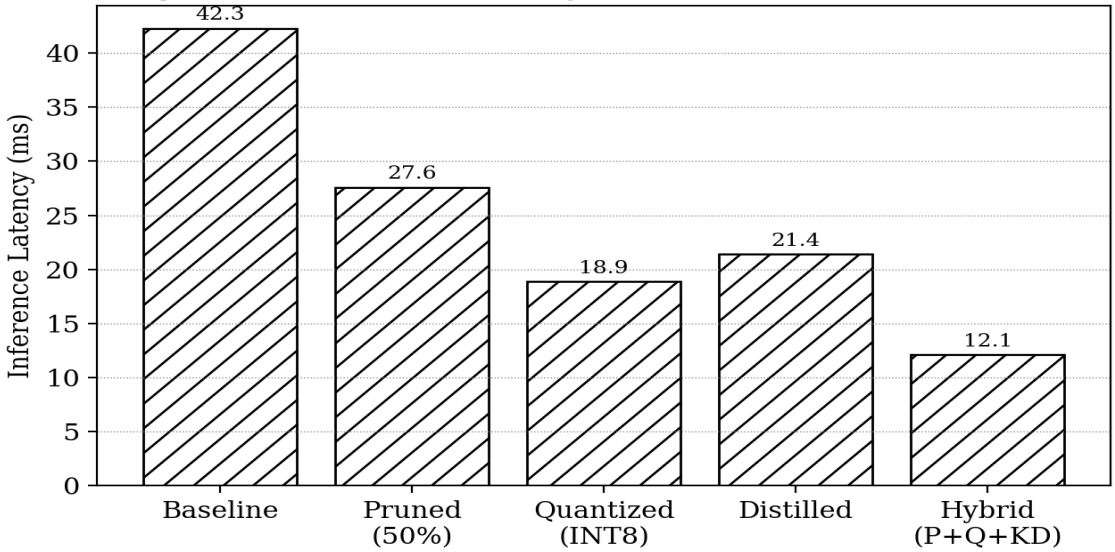
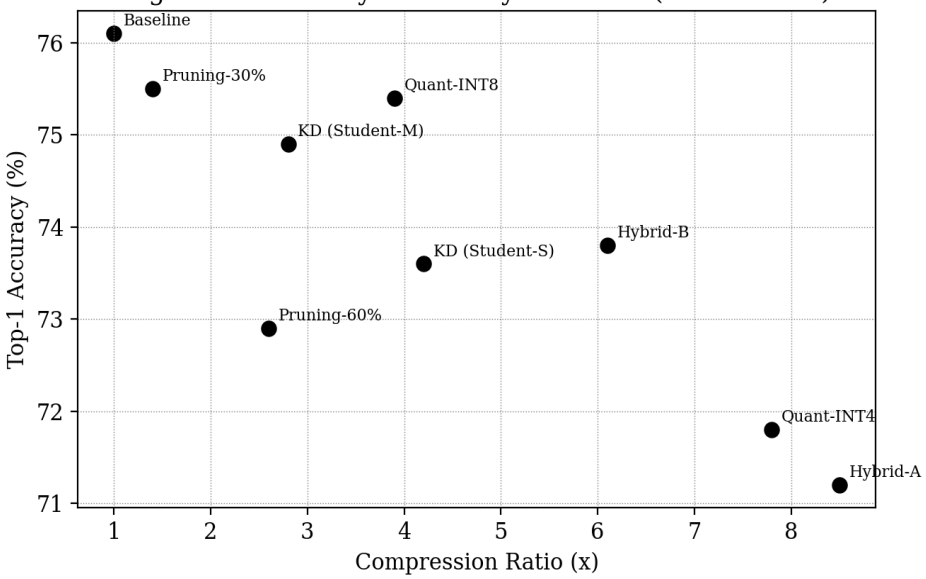


Figure 4 presents a Pareto-style scatter plot of accuracy against compression ratio for all evaluated configurations, visually identifying the hybrid configurations as occupying the most favorable region of the trade-off space, since they achieve higher compression ratios than single-technique methods while incurring only moderate additional accuracy loss.

Figure 4: Accuracy–Efficiency Trade-off (Pareto View)



Taken together, the results in Tables 1 through 6 and Figures 1 through 4 demonstrate that no single compression technique dominates across all evaluation metrics: pruning offers the most graceful accuracy degradation at low-to-moderate sparsity, quantization offers the most immediate memory savings with minimal retraining effort, and knowledge distillation provides the greatest latency reduction by enabling fundamentally smaller architectures. Hybrid pipelines that combine these techniques consistently achieve the highest composite

efficiency scores, confirming the central hypothesis of this study that complementary compression strategies, applied jointly, yield superior performance-efficiency trade-offs relative to any technique used in isolation.

5. Conclusion

This paper presented a systematic, unified evaluation of three major model compression strategies, namely pruning, quantization, and knowledge distillation, applied independently and in hybrid combination to a ResNet-56 convolutional neural network trained on CIFAR-10. Using a consistent experimental protocol and a set of twelve formal equations governing sparsity, quantization error, distillation loss, and efficiency metrics, the study quantified performance-efficiency trade-offs across accuracy, model size, FLOPs, inference latency, and energy consumption. The results demonstrate that structured and unstructured pruning can reduce model size by up to 3.3x with moderate accuracy loss, that 8-bit quantization-aware training achieves near-lossless 4x compression, and that knowledge distillation enables substantial architectural compression with the largest latency benefits. Most importantly, hybrid pipelines that combine pruning, distillation, and quantization consistently outperformed single-technique approaches, achieving compression ratios of up to 16.8x while retaining a majority of the baseline accuracy and reducing both latency and energy consumption by more than 70%. These findings reinforce the conclusion that model compression should not be treated as a choice between competing techniques but rather as a design space of complementary strategies whose combined application yields the most favorable performance-efficiency trade-offs for real-world, resource-constrained deployment.

6. Future Work

Several directions merit further investigation building on the findings of this study:

1. Hardware-aware compression: Extending the evaluation framework to explicitly incorporate target-hardware constraints, such as memory bandwidth and instruction-set support, into the compression search process rather than treating compression and deployment as separate stages.
2. Automated compression policy search: Applying neural architecture search or reinforcement-learning-based methods to automatically determine optimal per-layer pruning ratios and bit-widths, rather than relying on uniform, manually specified sparsity and quantization levels.
3. Compression of large language and transformer models: Extending the hybrid pipeline evaluated in this study to billion-parameter transformer architectures, leveraging recent one-shot quantization methods such as GPTQ, to determine whether the complementary benefits observed for CNNs generalize to attention-based models.
4. Dynamic and input-adaptive compression: Investigating compression strategies that adapt model capacity at inference time based on input complexity, enabling further energy savings for easy inputs while preserving accuracy on difficult ones.
5. Robustness and fairness under compression: Examining how pruning, quantization, and distillation affect model robustness to adversarial perturbations and performance parity

across demographic subgroups, which remain underexplored relative to accuracy-focused evaluation.

References

1. Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., & Guttag, J. (2020). What is the state of neural network pruning? Proceedings of Machine Learning and Systems, 2, 129-146.
2. Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine, 35(1), 126-136.
3. Courbariaux, M., Bengio, Y., & David, J. P. (2015). BinaryConnect: Training deep neural networks with binary weights during propagations. Advances in Neural Information Processing Systems, 28, 3123-3131.
4. Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit matrix multiplication for transformers at scale. Advances in Neural Information Processing Systems, 35, 30318-30332.
5. Frankle, J., & Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. Proceedings of the International Conference on Learning Representations.
6. Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate post-training quantization for generative pre-trained transformers. Proceedings of the International Conference on Learning Representations.
7. Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. International Journal of Computer Vision, 129(6), 1789-1819.
8. Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint arXiv:1510.00149.
9. He, Y., Zhang, X., & Sun, J. (2017). Channel pruning for accelerating very deep neural networks. Proceedings of the IEEE International Conference on Computer Vision, 1389-1397.
10. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
11. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
12. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2704-2713.
13. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. Proceedings of the IEEE International Conference on Computer Vision, 2736-2744.

14. Liu, Z., Sun, M., Zhou, T., Huang, G., & Darrell, T. (2019). Rethinking the value of network pruning. Proceedings of the International Conference on Learning Representations.
15. Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2017). Pruning convolutional neural networks for resource efficient inference. Proceedings of the International Conference on Learning Representations.
16. Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). XNOR-Net: ImageNet classification using binary convolutional neural networks. Proceedings of the European Conference on Computer Vision, 525-542.
17. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4510-4520.
18. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.
19. Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning, 97, 6105-6114.
20. Wu, J., Leng, C., Wang, Y., Hu, Q., & Cheng, J. (2016). Quantized convolutional neural networks for mobile devices. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4820-4828.