УДК 004.82

CLASSIFICATION OF METHODS OF OPTIMIZATION OF CNN MODELS

Нечипуренко С. І., Сорокін А. Р.
stanislav.nechypurenko@nure.ua, anton.sorokin@nure.ua
Харківський національний університет радіоелектроніки, каф. ЕОМ,
м. Харків, Україна

Reducing the computational cost of CNNs without sacrificing accuracy is still one of the central problems in deep learning. This work analyzes four optimization families - pruning, quantization, knowledge distillation, and NAS - comparing their trade-offs in accuracy, compression, and hardware requirements. The analysis shows that quantization combined with pruning offers the most practical solution for edge deployment, while NAS achieves the best compression at high computational cost.

The first method, structural pruning, removes redundant weights or entire channels/filters that contribute little to the output. In the foundational magnitude-based approach [1], the saliency of a weight is measured directly by its absolute value:

$$S(w_i) = |w_i| \# \quad (1)$$

Weights with saliency below a threshold τ are removed.

The second method, quantization, maps full-precision (FP32) weights and activations to lower bit-widths. For uniform k-bit quantization, a floating-point value x is mapped to:

$$x_q = \lfloor \frac{x}{\Delta} \rceil \cdot \Delta, \Delta = \frac{x_{max} - x_{min}}{2^k - 1} \# \quad (2)$$

Post-training quantization (PTQ) applies this transformation to a pretrained model with a small calibration set. Quantization-aware training (QAT) simulates quantization noise during training using straight-through estimators, yielding higher accuracy at INT8 or even INT4 precision [2].

The third method, knowledge distillation, trains a compact student *S* to mimic a larger teacher *T* using a combined loss [3]:

$$L_{total} = \alpha \cdot L_{CE}(y, P_S) + (1 - \alpha) \cdot \tau^2 \cdot KL(p_T^\tau \,||\, p_S^\tau) \quad (3)$$

where τ is the temperature and α ∈ [0,1] balances the objectives. Inference speedup comes from the student architecture, not the distillation process itself. Beyond output probabilities, feature-based distillation additionally aligns intermediate layer representations between teacher and student, which often yields better accuracy for very compact student architectures [3].

The fourth method, neural architecture search (NAS), finds the architecture maximising accuracy under a latency constraint [4]:

$$\alpha^* = argmax_{\alpha \in A} A(\alpha)\ st.\ \Lambda(\alpha) \leq T\# \quad (4)$$

Differentiable methods such as DARTS relax the discrete search space using learnable operation weights. A more efficient alternative is one-shot NAS, where all candidate architectures share a single set of weights during search, dramatically reducing the GPU hours required compared to standard differentiable methods [4].

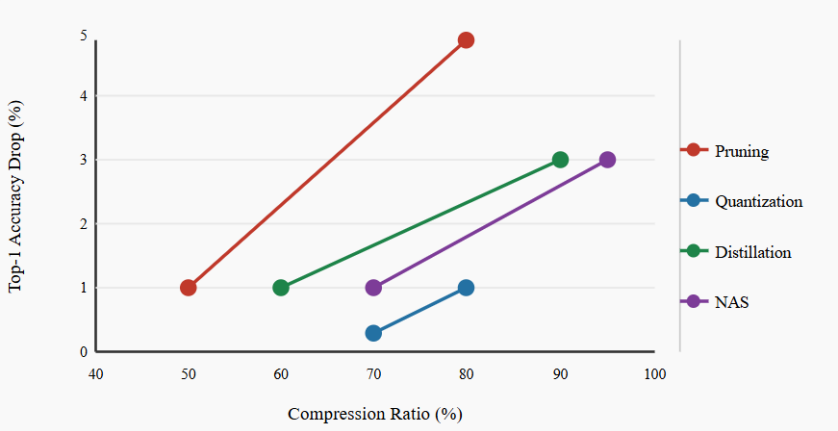

Figure 1. Top-1 accuracy drop vs. compression ratio (ResNet-50, ImageNet) [1, 2, 3, 4].

Table 1. Comparison of CNN optimization method families

Method	Accuracy Drop	Speedup*	Size Reduction	HW Required
Pruning	1-5%	2-4×	50-80%	Standard
Quantization	<1%	2-4×	~75%	INT8 support
Distillation**	1-3%	2-10×	60-90%	Standard (inference)
NAS***	1-3%	3-8×	70-95%	GPU cluster (search)

* Relative to FP32 baseline.

** Speedup reflects student architecture properties. Teacher required only at training.

*** Accuracy drop may reach 2-4% under tight latency budgets.

Combining methods - e.g., structured pruning followed by INT8 quantization - produces compounding gains exceeding each method applied alone [5].

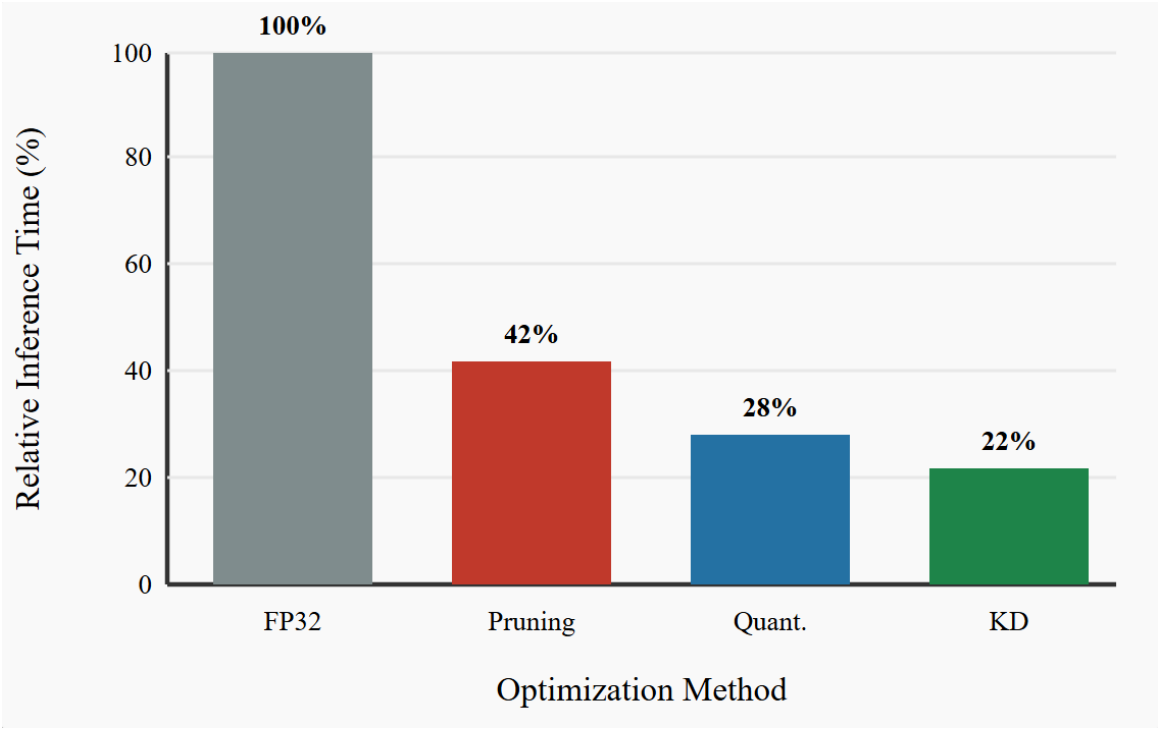

Figure 2. Relative inference time on ARM Cortex-A55 (batch size 1, PyTorch Mobile; FP32 = 100%). Authors' measurements.

No single method universally dominates. The optimal choice depends on the target hardware, acceptable accuracy budget, and available retraining resources. Combining complementary techniques yields compounding benefits exceeding any method applied in isolation [5]. Based on the analysis, quantization combined with structured pruning represents the most practical pipeline for edge deployment, as it requires no specialized hardware beyond INT8 support while achieving the best balance of compression and accuracy. NAS, despite its superior results, remains impractical for resource-limited research teams due to its search-phase cost.

References:

1. Han S., Pool J., Tran J., Dally W.J. Learning both Weights and Connections for Efficient Neural Networks. NIPS. 2015. P. 1135–1143.

2. Jacob B. et al. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. CVPR. 2018. P. 2704–2713. DOI: 10.1109/CVPR.2018.00286

3. Hinton G., Vinyals O., Dean J. Distilling the Knowledge in a Neural Network. 2015. arXiv:1503.02531.

4. Liu H., Simonyan K., Yang Y. DARTS: Differentiable Architecture Search. ICLR. 2019. arXiv:1806.09055.

5. Polino A., Pascanu R., Alistarh D. Model Compression via Distillation and Quantization. ICLR. 2018. arXiv:1802.05668.