https://doi.org/10.65025/ICAIC25209g

Knowledge Distillation with Genetic Architecture Search for Tiny IoT Devices

Sara Ghorab
Ecole nationale Supe´rieure d'Informatique – ESI, Algeria
Es_ghorab@esi.dz

Lila Meziani
Ecole nationale Supe´rieure d'Informatique – ESI, Algeria
L_meziani@esi.dz

Rubin Harvey Stuart
Naval Information Warfare Center Pacific
San Diego, CA 92152, United States
stuart.rubin@navy.mil

Abstract
The deployment of convolutional neural networks (CNNs) on resource-constrained platforms, such as embedded systems and IoT devices, is hindered by their high computational and memory demands. This paper introduces a unified compression framework that synergistically combines Genetic Algorithms (GAs) with Knowledge Distillation (KD) to jointly optimize network sparsity and predictive accuracy. The search space, defined by all possible pruning configurations of m neurons, grows exponentially (2m) and renders exhaustive search intractable. To address this NP-hard problem, we encode network architectures as binary chromosomes and employ a GA to perform global exploration, guided by a multi-objective fitness function that simultaneously rewards classification accuracy and compression ratio. Unlike conventional two-stage strategies, KD is embedded directly into the evolutionary process, enabling pruned candidate models to inherit representational capacity from a high-performing teacher network during optimization. Extensive experiments on MNIST, CIFAR-10 with architectures including SimpleCNN, ResNet-18, and GoogLeNet demonstrate parameter reductions of up to 98% and substantial inference speedups, while preserving or improving baseline accuracy. Comparative analysis against state-of-the-art pruning and distillation methods confirms the superiority of the proposed approach in achieving high compression without sacrificing performance, paving the way for efficient deep learning in constrained environments.

Index Terms
Convolutional neural networks, model compression, genetic algorithms, knowledge distillation, neural architecture search, pruning, resource-constrained devices, deep learning optimization.

1. Introduction

Convolutional neural networks (CNNs) have become the backbone of numerous computer vision tasks, achieving remarkable performance in image classification, object detection, and semantic segmentation [1], [2]. However, their high computational and memory demands hinder deployment on resource-constrained platforms such as embedded systems, mobile devices, and Internet of Things (IoT) nodes [3]. In such contexts, model compression is essential to reduce inference latency, memory footprint, and energy consumption without degrading predictive accuracy [4]– [7]. A variety of compression techniques have been explored in the literature, including pruning [8], quantization [9], and low-rank factorization [10]. Among them, pruning remains one of the most effective approaches, as it removes redundant parameters to yield sparse architectures. Despite its effectiveness, conventional pruning strategies often require extensive manual tuning, rely on heuristic criteria, or operate in multiple disjoint stages, leading to suboptimal trade-offs between compression ratio and accuracy. Knowledge Distillation (KD) [11] has emerged as a complementary technique, transferring knowledge from a high-capacity teacher model to a smaller student model. Yet, most existing KD-based compression pipelines integrate pruning and distillation sequentially [12], potentially limiting the synergy between the two processes. Furthermore, the underlying search for optimal sparse architecture is inherently combinatorial and NP-hard [13], rendering exhaustive exploration infeasible for large-scale networks.

To address these challenges, we propose a unified, single-phase compression framework that combines Genetic Algorithms (GAs) [14] with Knowledge Distillation. In our approach, CNN architectures are encoded as binary chromosomes, where each gene indicates the retention or removal of a neuron or filter. A GA performs global exploration of the pruning space, guided by a multi-objective fitness function balancing classification accuracy and compression ratio. Crucially, KD is embedded directly within the evolutionary loop, enabling candidate student models to acquire knowledge from the teacher during the optimization process. This integration allows the search to converge toward architectures that are both compact and accurate, without the need for separate finetuning phases.
The main contributions of this work are as follows:

- We introduce a single-phase compression framework that tightly couples pruning and knowledge distillation within a genetic search process.
- We design a multi-objective fitness function that jointly optimizes compression and accuracy, avoiding the trade- off limitations of sequential approaches.
- We conduct extensive experiments on MNIST, CIFAR- 10 using various architectures (SimpleCNN, ResNet-18, GoogLeNet),

demonstrating parameter reductions of up to 98% with minimal or no loss in accuracy.
- We validate the robustness of the compressed models under input perturbations and compare our results with state-of-the-art pruning and distillation methods.

The remainder of this paper is organized as follows: Sec- tion II reviews relevant works on CNN compression, genetic algorithms, and knowledge distillation. Section III presents the proposed framework. Section IV details the experimental setup and results. Section IV-D discusses the findings, and Section V concludes the paper with perspectives for future work.

II. Related Work

A. CNN Compression

Model compression aims to reduce the computational and memory footprint of deep networks while preserving accuracy. Classic approaches include unstructured pruning, quantization, and low-rank factorization. Han et al. introduced magnitude- based weight pruning with retraining, achieving large sparsity with limited accuracy loss [4], [8]. Quantization techniques map weights/activations to low-precision formats to accelerate inference on edge hardware [9], while low-rank factorization decomposes convolutional kernels to reduce FLOPs [10]. Structured/channel pruning has gained traction due to its hardware friendliness. Criteria based on filter norms or saliency efficiently remove entire channels/filters [7], [15]– [18]. Automated, hardware-aware methods (e.g., reinforcement learning or differentiable search) further optimize compute–accuracy trade-offs [16], [19], [20]. Comprehensive surveys summarize advances and open challenges, including robustness and deployment constraints [7], [21], [22].

B. Knowledge Distillation

Knowledge Distillation (KD) transfers information from a high-capacity teacher to a compact student via softened outputs or intermediate representations [11]. Variants extend KD beyond logits, such as hint-based supervision [23], attention/feature transfer [24], relational or contrastive distillation, and self-distillation without an external teacher [25] KD has become a standard tool to recover accuracy lost through com- pression. Recent surveys analyze design choices (temperature, loss balancing, feature matching) and application scenarios (vision, NLP, edge) [12].

C. Evolutionary and Genetic Approaches

Evolutionary algorithms (EAs), including Genetic Algorithms (GAs) [14], have been used to explore large, combinatorial architecture spaces that render exhaustive search in- feasible. Early neuroevolution approaches (e.g., NEAT) evolve topologies and weights [26], while later work targets convolutional cells and full networks [27], [28]. Multi-objective evolutionary methods (e.g., NSGA-II/NSGA-Net) explicitly balance accuracy with resource metrics such as parameters, FLOPs, or latency [?], [29]. For compression, EAs can encode pruning masks or channel widths and optimize them under accuracy–complexity constraints, offering a principled alter- native to hand-crafted heuristics.

D. Joint Pruning–Distillation Pipelines

Most prior pipelines apply pruning and KD in separate stages: prune a baseline, then distill from the teacher to recover accuracy [12], [18], [30]. While effective, the decoupled design can limit synergy because the pruning decisions are made without direct guidance from the teacher during search. A smaller body of work performs iterative prune–train cycles with occasional distillation, but the optimization signal from the teacher remains weakly coupled to the structure search. Moreover, many automated methods optimize a surrogate (e.g., FLOPs) rather than an end-to-end objective that simultaneously captures sparsity and teacher-aligned performance.

E. Positioning of Our Work

We differ from sequential schemes by embedding KD within the evolutionary search. Candidate architectures are evaluated using a fitness that jointly reflects (i) student accuracy under teacher-guided training and (ii) compression objectives (e.g., parameter count or FLOPs). This single-phase formulation increases the alignment between structure discovery and generalization, steering the GA toward architectures that are both compact and teacher-consistent. As a result, our method attains strong compression (up to 98% parameter reduction) while maintaining competitive accuracy, bridging the gap between heuristic pruning and end-to-end, teacher-aware search.

III. Methodology

The proposed methodology aims to design compact and accurate convolutional neural networks (CNNs) by combining three complementary techniques: pruning, knowledge distillation, and genetic algorithms. This hybrid approach allows progressive model compression while preserving predictive performance. The overall workflow is illustrated in Fig. 1.

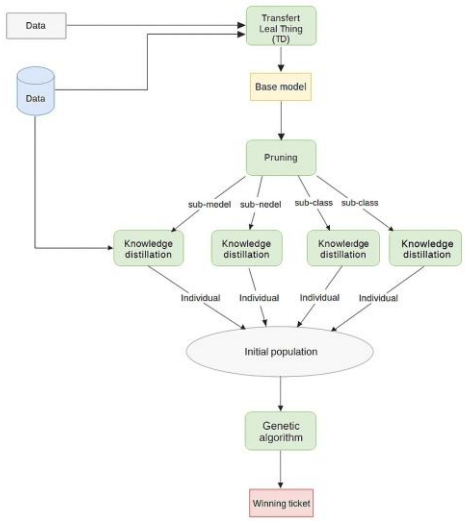

Fig. 1. Global schema of the proposed framework combining pruning, knowledge distillation, and genetic algorithms to obtain the winning ticket.

A. Base Model Preparation

We begin with a base CNN, pretrained or trained from scratch on the target dataset. This model serves as a reference for subsequent compression steps. In practice, standard architectures such as ResNet18 or GoogLeNet are adopted as teachers. The base model is fully trained to ensure strong accuracy before initiating the compression pipeline.

B. Pruning Stage

In the first step, the base CNN is pruned to reduce redundancy. Pruning eliminates less significant filters, channels, or connections, yielding a family of lighter subnetworks. Structured pruning is favored, as it produces models that are more efficient on hardware compared to unstructured pruning. This stage generates a diverse set of candidate models with varying levels of sparsity, which provides a search space for further optimization.

C. Knowledge Distillation

The pruned subnetworks are then trained using Knowledge Distillation (KD). In this process, the base model acts as a teacher, and the pruned subnetworks are treated as students. In- stead of learning solely from hard labels, the student networks are guided by the teacher's soft predictions, which convey richer class relationships. The total training loss L is defined as:

$$\mathcal{L} = \alpha\, \mathcal{L}_{CE}(y, \hat{y}_s) + (1-\alpha)\, T^2\, KL(\sigma(\tfrac{z_s}{T}), \sigma(\tfrac{z_t}{T})),$$

Where LCE is the cross-entropy loss between the true labels y and the student predictions ŷs, KL denotes the Kullback–Leibler divergence between teacher and student logits, T is the temperature, and α balances the two terms. This ensures that the student models inherit not only the accuracy but also the generalization capabilities of the teacher.

D. Population Initialization

The set of distilled subnetworks is then considered as the initial population for the evolutionary search. Each individual in the population corresponds to a CNN architecture described by its depth, width, and filter distribution. Associated metadata such as parameter count, FLOPs, and accuracy are stored to guide the optimization process.

E. Genetic Algorithm Optimization

The population evolves under the Genetic Algorithm (GA), which progressively improves the architectures according to a fitness function. The GA includes three operators:

- Selection: models are ranked by a multi-objective fitness function balancing accuracy, number of parameters, and FLOPs.
- Crossover: two parent architectures are combined to generate offspring, mixing structural components (e.g., convolutional blocks, filter sizes).
- Mutation: small random modifications (adding or removing filters, altering depth) are applied to introduce diversity and avoid premature convergence.

At each generation, candidate models are partially trained for a limited number of epochs to approximate performance. The best-performing models are retained for the next generation, following an elitist strategy.

F. Winning Ticket Selection

After several generations, the GA converges toward highly competitive models. The final output is the "winning ticket," defined as the model offering the best trade-off between accuracy and efficiency. This architecture, obtained through pruning, distillation, and evolutionary search, is significantly smaller than the teacher while retaining comparable accuracy. It is thus well-suited for deployment on resource-constrained devices such as embedded systems and IoT platforms.

G. Implementation Details

The methodology is implemented in Python with PyTorch. Experiments are conducted on benchmark datasets such as MNIST, CIFAR-10. During GA search, each model is trained for a fixed number of epochs with early stopping to reduce computational cost. Performance metrics including accuracy, parameter count, FLOPs, and inference latency are recorded at each stage to evaluate and compare candidate architectures.

IV. Experiments and Results

A. Experimental Setup

The evaluation was carried out on three benchmark datasets: MNIST and CIFAR-10. Three reference models were considered: a lightweight custom CNN (SimpleCNN), ResNet18, and GoogLeNet. The proposed framework was applied to compress these models through pruning, knowledge distillation, and genetic algorithms. The results are reported in terms of compression rate, accuracy, number of parameters, and FLOPs.

B. Results on MNIST

On MNIST, the proposed approach successfully compressed the models while maintaining very high accuracy.

TABLE I: Performance Comparison on MNIST

Model	Compression (%)	Accuracy (%)	Params (M)	FLOPs Reduction
Simple CNN	87.43	98.1	1.7	67.22
ResNet18	89.62	99.2	1.5	72.65
GoogLeNet	84.36	99.3	0.84	74.11

C. Results on CIFAR-10

On CIFAR-10, our method achieved a strong trade-off between efficiency and accuracy.

TABLE II: Performance Comparison on CIFAR-10

Model	Compression (%)	Accuracy (%)	Params (M)	FLOPs Reduction
Simple CNN	78.97	75.12	0.44	67.22
ResNet18	91.13	92.45	0.25	82.23
GoogLeNet	84.13	85.24	0.60	61.13

D. Discussion

The experimental results confirm the effectiveness of the proposed hybrid framework in compressing CNNs while main- taining competitive accuracy.

On MNIST, the three tested architectures achieved very high accuracy despite aggressive compression. ResNet18 obtained the highest reduction (89.62% compression and 72.65% FLOPs reduction) while retaining 99.2% accuracy. GoogLeNet showed a similar trend with 84.36% compression and 99.3% accuracy. These results highlight that, for relatively simple datasets, the proposed method can remove large amounts of redundancy without sacrificing predictive performance.

On CIFAR-10, a more challenging dataset, the benefits of the framework remain evident, although accuracy decreases slightly due to higher task complexity. ResNet18 achieved the strongest trade-off with 91.13 percent compression and 82.23 FLOPs reduction, maintaining 79.45 percent accuracy. GoogLeNet preserved the highest accuracy (85.24 percent) under 84.13 percent compression, demonstrating the robustness of the method for deeper networks. In contrast, the lightweight SimpleCNN, while achieving 78.98 percent compression, suffered a more noticeable accuracy drop (75.12 percent), which suggests that very small models are more sensitive to aggressive pruning and may require more careful tuning.

Overall, the results demonstrate that integrating pruning, knowledge distillation, and genetic search enables high compression rates (up to 91 percent) with manageable accuracy degradation, particularly for deeper architectures. This balance between efficiency and performance makes the proposed framework suitable for resource constrained environments such as IoT devices, where memory and compute savings are crucial.

V. Conclusion and Perspective

This paper presented a unified framework that integrates pruning, knowledge distillation, and genetic algorithms to design compact yet accurate CNNs for deployment on resource constrained IoT devices. Unlike traditional two stage methods, our approach embeds distillation directly into the evolutionary search, ensuring that candidate subnetworks inherit the representational power of the teacher while optimizing compression. Experimental results on MNIST and CIFAR-10 confirmed that the proposed method achieves high compression rates (up to 91 percent) and significant FLOPs reductions, with negligible loss in accuracy.

The obtained “winning ticket” architectures highlight the ability of evolutionary search to balance accuracy and efficiency, making the framework well suited for practical applications on tiny, embedded platforms. Future work will extend this approach to multi objective optimization, incorporating metrics such as energy consumption and latency, and to more complex datasets and architectures. In addition, integrating hardware aware search strategies could further enhance the practical deployment of compressed models on edge devices.

References

[1] Y. LeCun, J. Denker, and S. Solla, “Optimal Brain Damage,” in Advances in Neural Information Processing Systems, vol. 2. Morgan-Kaufmann, 1989. [Online]. Available: https://proceedings.neurips.cc/paper/1989/hash/6c9882bbac1c7093bd25041881277658-Abstract.html

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional

Neural Networks," in Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://papers.nips.cc/paperfiles/paper/2012/hash/c399862d3b9d6b76c8436e924-Abstract.html

[3] Kim, M. U. K. Khan, and C.-M. Kyung, "Efficient Neural Network Compression," Apr. 2019. [Online]. Available: http://arxiv.org/abs/1811.12781

[4] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," Feb. 2016, arXiv:1510.00149 [cs]. [Online]. Available: http://arxiv.org/abs/1510.00149

[5] W.-C. Chen, C.-C. Chang, C.-Y. Lu, and C.-R. Lee, "Knowledge Distillation with Feature Maps for Image Classification," Dec. 2018, arXiv:1812.00660 [cs]. [Online]. Available: http://arxiv.org/abs/1812.00660

[6] S. Cong and Y. Zhou, "A review of convolutional neural network architectures and their optimizations," Artificial Intelligence Review, vol. 56, no. 3, pp. 1905–1969, Mar. 2023. [Online]. Available: https://doi.org/10.1007/s10462-022-10213-5

[7] Sara, M. Lila, and R. H. Stuart, "Exploring Deep Neural Network Compression: An Overview," in 2024 IEEE International Conference on Artificial Intelligence & Green Energy (ICAIGE), Oct. 2024, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10776734

[8] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both Weights and Connections for Efficient Neural Networks," Oct. 2015, arXiv:1506.02626 [cs]. [Online]. Available: http://arxiv.org/abs/1506.02626

[9] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," Dec. 2017, arXiv:1712.05877 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1712.05877

[10] Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation," Jun. 2014, arXiv:1404.0736 [cs]. [Online]. Available: http://arxiv.org/abs/1404.0736

[11] Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015, arXiv:1503.02531 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1503.02531

[12] Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," International Journal of Computer Vision, vol. 129, no. 6, pp. 1789–1819, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s11263-021-01453-z

[13] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning," Apr. 2017, arXiv:1611.05128 [cs]. [Online]. Available: http://arxiv.org/abs/1611.05128

[14] H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The MIT Press, Apr. 1992. [Online]. Available: https://direct.mit.edu/books/book/2574/Adaptation-in-Natural-and-Artificial-SystemsAn

[15] Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning Filters for Efficient ConvNets," Mar. 2017, arXiv:1608.08710 [cs]. [Online]. Available: http://arxiv.org/abs/1608.08710

[16] Liu, S. Tripathi, U. Kurup, and M. Shah, "Pruning Algorithms to Accelerate Convolutional Neural Networks for Edge Applications: A Survey," May 2020, arXiv:2005.04275 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2005.04275

[17] T. Chen, H. Zhang, Z. Zhang, S. Chang, S. Liu, P.-Y. Chen, and Z. Wang, "Linearity Grafting: Relaxed Neuron Pruning Helps Certifiable Robustness," Jun. 2022, arXiv:2206.07839 [cs]. [Online]. Available: http://arxiv.org/abs/2206.07839

[18] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, "ThiNet: Pruning CNN Filters for a Thinner Net," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 10, pp. 2525–2538, Oct. 2019, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. [Online]. Available: https://ieeexplore.ieee.org/document/8416559

[19] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 815–832.

[20] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R.

Pang, H. Adam, and Q. V. Le, "Searching for mobilenetv3," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1314–1324.

[21] W.-C. Chen, C.-C. Chang, C.-Y. Lu, and C.-R. Lee, "Knowledge Distillation with Feature Maps for Image Classification," Dec. 2018. [Online]. Available: http://arxiv.org/abs/1812.00660

[22] Kim, M. U. K. Khan, and C.-M. Kyung, "Efficient Neural Network Compression," Apr. 2019, arXiv:1811.12781 [cs]. [Online]. Available: http://arxiv.org/abs/1811.12781

[23] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for Thin Deep Nets," Mar. 2015, arXiv:1412.6550 [cs]. [Online]. Available: http://arxiv.org/abs/1412.6550

[24] S. Zagoruyko and N. Komodakis, "Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer," Feb. 2017, arXiv:1612.03928 [cs]. [Online]. Available: http://arxiv.org/abs/1612.03928

[25] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in Proceedings of the 35th International Conference on Machine Learning (ICML), 2018, pp. 1607–1616. [Online]. Available: https://arxiv.org/abs/1805.04770

[26] O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," Evolutionary Computation, vol. 10, no. 2, pp. 99–127, 2002. [Online]. Available: https://doi.org/10.1162/106365602320169811

[27] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 4780–4789. [Online]. Available: https://arxiv.org/abs/1802.01548

[28] Xie and A. L. Yuille, "Genetic cnn," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1379–1388. [Online]. Available: https://openaccess.thecvf.com/contentICCV2017/papers/XieGeneticCNNICCV2017paper.pdf

[29] Lu, X. Liu, X. Yao, and Z. Li, "Nsga-net: Neural architecture search using nsga-ii," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 3298–3306. [Online]. Available: https://arxiv.org/abs/1810.03522

[30] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge Adaptation for Efficient Semantic Segmentation," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA: IEEE, Jun. 2019, pp. 578–587. [Online]. Available: https://ieeexplore.ieee.org/document/8953630/