International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Neural Architecture Search (NAS): A Survey of Methods, Challenges, and Future Directions

[1]A.Sindhu Devi [2] L.Godlin Atlas

[1] Research Scholar, BIHER, Chennai. [2] Associate Professor, BIHER, Chennai.

[1] sindhudevicse24@gmail [2] godlinatlas.cse@bharathuniv.ac.in

[1] ORCID iD: 0009-0003-8918-4829

ABSTRACT

Neural Architecture Search (NAS) is an emerging subfield of automated machine learning (AutoML) that seeks to automate the design of deep neural networks. Traditional manual and heuristic-based architecture design is labor-intensive, time-consuming, and dependent on expert knowledge. NAS algorithms aim to minimize human involvement by automatically discovering architectures that achieve state-of-the-art performance across diverse tasks. This survey presents a comprehensive review of NAS methodologies, including reinforcement learning-based, evolutionary algorithm-based, gradient-based, and one-shot approaches. We examine the key components of NAS frameworks—search space formulation, search strategy design, and performance estimation methods—along with their computational costs, benchmark datasets, and evaluation protocols. Furthermore, we analyze scalability challenges, generalization capabilities, and deployment considerations, particularly for resource-constrained environments. Finally, we outline open research problems and promising future directions, with emphasis on hardware-aware, explainable, and zero-shot NAS paradigms.

Keywords:
Neural Architecture Search, AutoML, Deep Learning, Meta-Learning, Reinforcement Learning, Search Space, One-Shot NAS, NASBench

1. Introduction

Deep learning models have achieved remarkable results in computer vision [15], natural language processing, and speech recognition. However, the manual design of high-performing architectures remains a resource-intensive process requiring domain expertise. Neural Architecture Search (NAS) addresses this challenge by automating architecture discovery through data-driven search strategies [16]. Recent advances in NAS have reduced search cost, improved scalability, and expanded applicability to various domains.

This survey consolidates existing NAS techniques, analyzes their strengths and limitations, and identifies open challenges. By systematically examining search space design, search strategies, and performance estimation techniques, we provide researchers and practitioners with a structured understanding of the current NAS landscape.

2. NAS Framework and Components

A general NAS framework consists of three major components [16]:

1. **Search Space** — Defines the set of possible architectures, including chain-structured, cell-based, and hierarchical designs.

2. **Search Strategy** — Determines how to explore the search space, using:

- Reinforcement Learning (e.g., NASNet [1], ENAS [3])
- Evolutionary Algorithms (e.g., AmoebaNet [4])
- Bayesian Optimization
- Gradient-Based Methods (e.g., DARTS [5])
- Differentiable NAS and One-Shot Models [10]

3. **Performance Estimation Strategy** — Predicts candidate architecture quality using:

- Full training
- Early stopping
- Weight sharing [3]
- Zero-cost proxies [12]

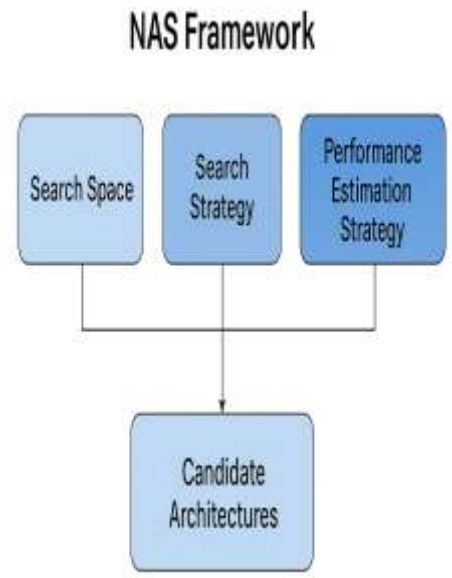

Fig 1: NAS Framework

3. NAS Search Strategies

3.1 Reinforcement Learning-Based NAS

NAS can be framed as a sequential decision process, where a controller samples architectures and updates its policy based on validation performance rewards [1], [2]. NASNet [2] demonstrated state-of-the-art results on ImageNet while reducing parameter count. ENAS [3] improved efficiency via parameter sharing, drastically cutting computational cost.

Challenges in RL-based NAS include high GPU resource demands, sparse rewards, and large discrete action spaces. Recent work addresses these with multi-objective rewards [9], early stopping, and transfer learning.

Workflow: Controller initialization → Architecture sampling → Model instantiation → Proxy/full training → Reward computation → Policy update → Iteration until budget exhaustion.

3.2 Evolutionary Algorithms (EAs)

EAs evolve populations of architectures through mutation and crossover [4], [18]. They are robust, parallelizable, and effective for large search spaces but require careful tuning of fitness functions.

3.3 Differentiable NAS

Methods such as DARTS [5] convert discrete search spaces into continuous relaxations, enabling gradient descent optimization. While faster, they may converge to suboptimal architectures [11].

3.4 One-Shot NAS

One-shot methods train a supernet encompassing all possible subnets [10], allowing rapid architecture evaluation through weight sharing. However, this can introduce performance estimation bias [13].

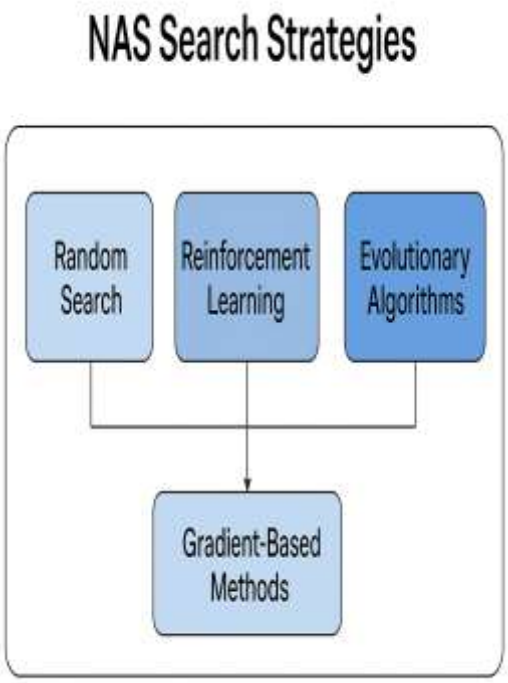

Fig 2: NAS Search Strategies

4. Benchmarking and Evaluation

4.1 Evaluation Metrics

- **Accuracy / Error Rate** [15]
- **Search Cost** (GPU-days, FLOPs) [3]
- **Model Complexity** (parameters, FLOPs) [17]
- **Latency & Energy Efficiency** (measured on target hardware) [8], [20]
- **Multi-Objective Trade-offs** via Pareto analysis [9]

4.2 Benchmark Datasets

- **Image Classification**: CIFAR-10/100 [14], ImageNet [15]
- **Object Detection**: COCO, Pascal VOC
- **NLP**: Penn Treebank, Wikitext-2 [6]
- **Speech**: Librispeech, TIMIT

4.3 Evaluation Protocols

1. Proxy evaluation [3]
2. Weight sharing [10]
3. Full retraining [2]
4. Cross-dataset validation [7]
5. Hardware-aware evaluation [8], [20]

4.4 NAS Benchmarks

- NAS-Bench-101 [7]
- NAS-Bench-201 [6]
- NAS-Bench-301
- NAS-Bench-NLP
- ProxylessNAS benchmark [8]

4.5 Challenges in NAS Evaluation

Despite advancements, benchmarking in NAS faces challenges:

- **Reproducibility Issues**: Variations in training pipelines can lead to inconsistent results.
- **Overfitting to Benchmarks**: Excessive tuning on fixed datasets may reduce real-world applicability.
- **Search Space Bias**: Results depend heavily on the predefined search space.
- **Hardware-Specific Optimizations**: An architecture optimal for one device may underperform on another.

5. Challenges in NAS

Although Neural Architecture Search has matured significantly over the past few years, several critical challenges hinder its scalability, reproducibility, and real-world applicability. This section discusses these challenges in depth.

5.1 Scalability and Computational Cost

A major barrier to adopting NAS in both academia and industry is its extremely high computational demand. Early approaches such as NASNet [2] required over 1,800 GPU-days to complete a single search, making them impractical for most researchers and organizations.

Even with advancements like weight sharing in ENAS [3] and differentiable search in DARTS [5], the cost of exploring large search spaces remains significant.
The problem is exacerbated when:

- The search space contains billions of possible architectures.
- Multiple objectives (accuracy, latency, energy) are optimized simultaneously [9], [20].
- Search must be repeated for different datasets or hardware targets.

This challenge motivates research into zero-cost proxies [12], surrogate models, and progressive search space pruning to reduce evaluation overhead without sacrificing accuracy.

5.2 Generalization Across Tasks and Datasets

Many NAS-discovered architectures perform well on the dataset they were optimized for (e.g., CIFAR-10) but fail to generalize to different datasets or modalities [16].
This is often due to:

- Overfitting to specific benchmark datasets.
- Search space bias toward architectures that exploit dataset-specific characteristics.
- Lack of cross-domain evaluation during search.

Potential remedies include cross-dataset validation [7], meta-learning approaches for NAS, and incorporating transferability constraints into the search objective.

5.3 Search Space Bias and Limitations

The design of the search space heavily influences NAS performance. A poorly designed search space may exclude optimal architectures entirely. Examples of bias include:

- Restricting to only convolutional operators, ignoring attention-based or graph-based modules.
- Imbalanced operator choices leading to convergence toward certain patterns (e.g., skip connections in DARTS [5] dominating results [11]).

Mitigation strategies involve:

- **Expanding the diversity** of search operators (convolutions, transformers, attention blocks).
- **Hierarchical search spaces** to explore macro- and micro-architectural variations.
- Regularization techniques to encourage fair operator selection [13].

5.4 Reproducibility and Benchmarking Issues

NAS reproducibility remains a significant concern [16]. Different implementations of the same algorithm can yield vastly different results due to:

- Variations in training pipelines and hyperparameters.
- Non-deterministic GPU computations.
- Omission of crucial details in published works.

NASBench datasets [6], [7], [8] have improved reproducibility by providing standardized search spaces and pre-computed performance data, but many NAS methods still lack open-source code or detailed experimental protocols.

5.5 Multi-Objective Optimization

Real-world deployments rarely optimize for accuracy alone. Constraints such as inference latency, energy consumption, and model size must also be considered [8], [9], [20].
Challenges include:

- Defining meaningful and balanced objective functions.
- Handling trade-offs via Pareto-optimal search.
- Dynamically adapting objectives for different deployment environments (e.g., edge devices vs. cloud).

Emerging work in hardware-aware NAS integrates real device measurements into the search loop [8], but this increases search time and complexity.

5.6 Hardware Dependency and Deployment Constraints

Architectures optimized for one hardware platform (e.g., GPU) may underperform on another (e.g., ARM CPU, FPGA) [20]. This is due to:

- Different compute-memory bottlenecks.
- Hardware-specific acceleration features.
- Variation in parallelization efficiency.

Solutions include:

- Platform-specific NAS [9].
- Multi-platform profiling during search.
- Use of hardware simulators to predict cross-platform performance.

5.7 Interpretability and Explainability

Most NAS methods treat architectures as **black boxes**, making it difficult to explain **why** certain architectures perform better than others. This lack of interpretability:

- Hinders trust in automatically discovered models.
- Limits adoption in domains with safety-critical requirements (e.g., healthcare, autonomous driving).

Research into explainable NAS [16] seeks to identify key architectural features contributing to performance and to visualize decision boundaries in search space exploration.

5.8 Evaluation Bias and Overfitting to Benchmarks

NAS research heavily relies on a small set of benchmarks (CIFAR-10, ImageNet, NASBench). While convenient, this can lead to:

- Over-tuning algorithms to fixed datasets.
- Reduced applicability to real-world, noisy, or imbalanced datasets.
- Underestimation of robustness in unseen scenarios.

Adopting diverse evaluation datasets, real-world benchmarks, and task-specific validation protocols can mitigate this risk.

6. Future Directions

Key future research directions in NAS include:

Hardware-Aware NAS

- Integrate real-time latency and energy profiling during search to optimize architectures for specific deployment platforms (e.g., mobile, embedded, FPGA).

Zero-Shot NAS

- Develop predictive models that estimate architecture performance without training, significantly reducing search cost.

Neuro-Symbolic NAS

- Combine symbolic reasoning with neural search methods to incorporate prior knowledge and improve interpretability.

Explainable NAS

- Design tools and frameworks that provide human-interpretable insights into why certain architectures are selected and how they achieve high performance.

Meta-NAS

- Create NAS algorithms that can automatically adapt their own search strategies, enabling cross-task and cross-domain generalization.

Continual and Lifelong NAS

- Enable architectures to evolve incrementally in dynamic environments without complete retraining.

Multi-Modal NAS

- Extend NAS techniques to jointly optimize architectures for multi-modal tasks, such as vision–language or sensor fusion applications.

7. Conclusion

Neural Architecture Search (NAS) has emerged as a transformative approach in Automated Machine Learning, offering the potential to automate deep neural network design and reduce reliance on expert-driven engineering. This survey has reviewed major NAS paradigms—reinforcement learning, evolutionary algorithms, differentiable methods, and one-shot approaches—highlighting their search strategies, performance estimation techniques, and benchmark datasets. We also discussed key challenges, including scalability, generalization, search space bias, reproducibility, and multi-objective optimization. While significant progress has been made toward efficiency and hardware-awareness, further advances are required for robust cross-domain generalization and deployment in real-world, resource-constrained environments. Future research will likely focus on hardware-aware optimization, zero-shot performance prediction, explainable NAS, and adaptive search strategies. By addressing these challenges, NAS can accelerate the democratization of deep learning, enabling practical, efficient, and widely deployable AI solutions across diverse applications and computational platforms.

8. References

[1] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2017.

[2] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8697–8710.

[3] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 4095–4104.

[4] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, 2019, pp. 4780–4789.

[5] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2019.

[6] X. Dong and Y. Yang, "NAS-Bench-201: Extending the scope of reproducible neural architecture search," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2020.

[7] C. Ying, A. Klein, E. Real, et al., "NAS-Bench-101: Towards reproducible neural architecture search," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7105–7114.

[8] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2019.

[9] M. Tan, B. Chen, R. Pang, et al., "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2820–2828.

[10] X. Bender, R. A. Coelho, and T. Elsken, "One-shot neural architecture search: Maximising reuse for efficiency," *J. Mach. Learn. Res.*, vol. 21, no. 108, pp. 1–31, 2020.

[11] Y. Xu, L. Xie, X. Zhang, et al., "PC-DARTS: Partial channel connections for memory-efficient architecture search," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2020.

[12] C. Liu, L. Chen, Y. He, et al., "Zero-cost proxies for lightweight NAS," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021.

[13] X. Dong, A. A. Ahmed, and Y. Yang, "FairNAS: Rethinking evaluation fairness of weight sharing NAS," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020, pp. 1466–1475.

[14] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., University of Toronto, 2009.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[16] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 55, pp. 1–21, 2019.

[17] A. Howard, M. Sandler, G. Chu, et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.

[18] E. Real, S. Moore, A. Selle, et al., "Large-scale evolution of image classifiers," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 2902–2911.

[19] X. Dong and Y. Yang, "Searching for a robust neural architecture in four GPU hours," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1761–1770.

[20] Y. Dong, A. Gholami, and K. Keutzer, "HNAS: Hardware-aware neural architecture search," *IEEE Access*, vol. 8, pp. 149–160, 2020.