Comparative Paper Review: NASNet and MnasNet

Benjamin Park

January 16, 2026

Abstract

This review compares two seminal papers in Neural Architecture Search (NAS): *MnasNet: Platform-Aware Neural Architecture Search for Mobile* (Tan et al., 2019) and *Learning Transferable Architectures for Scalable Image Recognition* (Zoph et al., 2018). Both papers address the challenge of automating neural network design but with different objectives and methodologies. MnasNet focuses on mobile deployment with real-world latency constraints, while NASNet emphasizes transferability and scalability across diverse image recognition tasks. We analyze their contributions, methodologies, results, and impact on the field of automated machine learning.

1 Introduction

The design of convolutional neural networks (CNNs) has traditionally required significant manual architecture engineering. Neural Architecture Search (NAS) emerged as a paradigm shift, enabling automated discovery of high-performing architectures. The two papers under review represent significant milestones in this evolution:

- **NASNet** (2018) introduced a transferable search space where architectures discovered on small datasets (CIFAR-10) could be scaled to larger datasets (ImageNet).
- **MnasNet** (2019) pioneered platform-aware NAS, directly optimizing for real-world mobile latency alongside accuracy.

Both papers build upon the original NAS framework using reinforcement learning but introduce novel search spaces and optimization objectives that have profoundly influenced subsequent research.

2 Background and Motivation

2.1 NASNet: The Transferability Challenge

NASNet addresses the computational expense of performing architecture search directly on large datasets like ImageNet. The key insight is that searching for reusable building blocks (cells) on smaller datasets can yield architectures that transfer effectively to larger tasks. This approach reduces search time by approximately 7× compared to previous methods.

Key Motivation:

- Direct search on ImageNet is prohibitively expensive
- Manual architecture engineering requires significant expertise
- Need for architectures that generalize across different scales

2.2 MnasNet: The Mobile Deployment Gap

MnasNet recognizes that existing NAS methods optimize for indirect metrics (FLOPs) that poorly correlate with actual inference latency on mobile devices. For example, MobileNet and NASNet have similar FLOPs (575M vs. 564M) but vastly different latencies (113ms vs. 183ms on Pixel phones).

Key Motivation:

- FLOPs are inaccurate proxies for real-world latency
- Mobile deployment requires explicit latency constraints
- Hardware heterogeneity demands platform-aware optimization

3 Methodology Comparison

3.1 Search Space Design

3.1.1 NASNet Search Space

NASNet introduces a *cell-based* search space with two types of cells:

1. **Normal Cell:** Maintains spatial dimensions
2. **Reduction Cell:** Reduces spatial dimensions by 2×

Each cell is constructed from $B = 5$ blocks, where each block involves:

- Selecting two hidden states
- Choosing operations for each (13 options including separable convolutions, pooling, identity)
- Combining outputs via addition or concatenation

The search space size is approximately 10^{13} compared to 10^{39} for per-layer search.

3.1.2 MnasNet Search Space

MnasNet employs a *factorized hierarchical* search space that partitions the network into predefined blocks based on input resolution and filter size. For each block, the search determines:

- Convolutional operations (regular, depthwise, mobile inverted bottleneck)
- Kernel size (3×3, 5×5)
- Squeeze-and-Excitation ratio (0, 0.25)
- Skip operations (pooling, identity, none)
- Filter size and number of layer repetitions

This factorization balances layer diversity with tractable search space size ($\sim 10^{13}$).

3.2 Optimization Objectives

3.2.1 NASNet: Accuracy Maximization

NASNet optimizes solely for validation accuracy on CIFAR-10:

$$\max_m \mathrm{ACC}(m) \tag{1}$$

The assumption is that architectures performing well on CIFAR-10 will transfer to ImageNet when scaled appropriately.

3.2.2 MnasNet: Multi-Objective Optimization

MnasNet formulates a multi-objective problem balancing accuracy and latency:

$$\max_m \mathrm{ACC}(m) \times \left[\frac{\mathrm{LAT}(m)}{T}\right]^w \tag{2}$$

where T is the target latency and w is defined as:

$$w = \begin{cases} \alpha, & \text{if } \mathrm{LAT}(m) \leq T \\ \beta, & \text{otherwise} \end{cases} \tag{3}$$

The values $\alpha = \beta = -0.07$ are empirically chosen to ensure Pareto-optimal solutions have similar rewards under different accuracy-latency trade-offs.

3.3 Search Algorithm

Both papers use **reinforcement learning** with an RNN controller:

- **NASNet:** Uses Proximal Policy Optimization (PPO) with 500 GPUs over 4 days (2,000 GPU-hours)
- **MnasNet:** Uses PPO with 500 GPUs over 4.5 days, sampling 8,000 models

NASNet demonstrates that reinforcement learning outperforms random search, though the gap is smaller than in original NAS work, suggesting well-designed search spaces are crucial.

4 Key Technical Contributions

4.1 NASNet Innovations

4.1.1 Transferable Cell Design

The Normal and Reduction cell paradigm enables architecture transfer across datasets by:

- Decoupling cell complexity from network depth
- Allowing flexible stacking (varying N and filter counts)
- Maintaining consistent computational patterns across scales

4.1.2 ScheduledDropPath Regularization

A novel regularization technique that linearly increases path dropout probability during training. Unlike standard DropPath, this scheduled approach significantly improves generalization for NASNet architectures on both CIFAR-10 and ImageNet.

4.2 MnasNet Innovations

4.2.1 Platform-Aware Real-World Latency Measurement

MnasNet directly measures inference latency by executing models on actual mobile devices (Pixel phones) rather than using FLOPs approximations. This addresses hardware-specific optimizations and variability.

4.2.2 Factorized Hierarchical Search Space

The search space enables layer diversity across the network while maintaining tractability. Early network layers (processing larger feature maps) can have different architectures than later layers, optimizing the accuracy-latency trade-off at each stage.

4.2.3 Multi-Objective Reward Function

The weighted product formulation allows discovering multiple Pareto-optimal solutions in a single search, providing models across different computational budgets.

5 Experimental Results

5.1 CIFAR-10 Performance

Model	Params	Error (%)
Shake-Shake + cutout	26.2M	2.56
NASNet-A + cutout	3.3M	2.65
NASNet-A (large) + cutout	27.6M	**2.40**
MnasNet-A + cutout	3.3M	2.65

Table 1: CIFAR-10 results showing state-of-the-art performance for both architectures.

Both achieve competitive CIFAR-10 results, with NASNet-A achieving 2.40% error (state-of-the-art at publication).

5.2 ImageNet Classification

Model	MAdds	Top-1	Top-5
Inception-v3	5.72B	78.8	94.4
Inception-ResNet-v2	13.2B	80.1	95.1
NASNet-A (6@4032)	23.8B	**82.7**	**96.2**
MobileNetV2	300M	72.0	91.0
MnasNet-A1	312M	**75.2**	**92.5**

Table 2: ImageNet results comparing NASNet and MnasNet against baselines.

Key Findings:

- **NASNet:** Achieves 82.7% top-1 accuracy, matching SENet but with 28% fewer FLOPs
- **MnasNet:** Runs 1.8× faster than MobileNetV2 with 0.5% higher accuracy

5.3 Mobile-Optimized Models

MnasNet particularly excels in the mobile regime:

Model	Latency	Top-1	Params
MobileNet-224	113ms	70.6	4.2M
ShuffleNet (2x)	-	70.9	5M
NASNet-A (4@1056)	-	74.0	5.3M
MnasNet-A1	78ms	75.2	3.9M

Table 3: Mobile-optimized models on ImageNet (224×224 images).

MnasNet-A1 achieves 3.1% higher accuracy than MobileNet with fewer parameters.

5.4 Object Detection Transfer

Both architectures demonstrate strong transfer learning capabilities:

Backbone	Resolution	mAP
Inception-ResNet-v2	600×600	35.7
NASNet-A (6@4032)	1200×1200	**43.1**
MnasNet-A1	600×600	29.6

Table 4: COCO object detection results using Faster-RCNN.

NASNet achieves 43.1% mAP on COCO, surpassing previous state-of-the-art by 4.0%.

6 Architectural Analysis

6.1 NASNet Architecture Characteristics

The discovered NASNet cells exhibit:

- Extensive use of separable convolutions
- Multiple parallel branches (5 blocks with diverse operations)
- Skip connections learned automatically
- Both 3×3 and 5×5 kernel sizes

6.2 MnasNet Architecture Characteristics

MnasNet architectures show:

- Layer diversity throughout the network
- Heavy use of mobile inverted bottleneck convolutions
- Squeeze-and-Excitation modules in later layers
- Mixture of 3×3 and 5×5 kernels
- Adaptive layer depth per block

The factorized search space allows MnasNet to optimize different parts of the network independently, leading to heterogeneous architectures better suited for efficiency.

7 Search Efficiency Comparison

Method	GPU-hours	Models Sampled
Original NAS	22,400	-
NASNet	2,000	20,000
MnasNet	2,160	8,000

Table 5: Search efficiency comparison (approximate values).

Both approaches are significantly more efficient than original NAS due to:

- Searching on smaller proxy tasks (CIFAR-10)
- Efficient search space design
- Improved search algorithms (PPO)

8 Critical Analysis

8.1 Strengths

8.1.1 NASNet

- **Transferability:** Successfully demonstrates cross-dataset transfer
- **State-of-the-art accuracy:** Achieves top performance on ImageNet
- **Scalability:** Works across computational budgets by varying N and filters
- **Generalizability:** Strong transfer to object detection

8.1.2 MnasNet

- **Platform awareness:** Directly optimizes real-world latency
- **Multi-objective optimization:** Provides Pareto-optimal solutions
- **Layer diversity:** Enables heterogeneous architectures
- **Mobile excellence:** Outstanding performance in resource-constrained settings

8.2 Limitations

8.2.1 NASNet

- **Latency mismatch:** CIFAR-10 search doesn't account for mobile deployment
- **Uniform cells:** All Normal cells identical, limiting flexibility
- **Computational cost:** Still requires 2,000 GPU-hours
- **Proxy task assumption:** Transfer effectiveness depends on task similarity

8.2.2 MnasNet

- **Platform-specific:** Architectures optimized for specific hardware (Pixel phones)
- **CIFAR search:** Still uses proxy task despite platform awareness
- **Hyperparameter sensitivity:** Choice of α, β requires tuning
- **Limited theoretical justification:** Empirical approach to multi-objective formulation

9 Impact and Follow-up Work

9.1 Influence on NAS Research

Both papers have profoundly influenced subsequent work:

NASNet's legacy:

- Cell-based search spaces became standard (ENAS, DARTS, ProxylessNAS)
- Transfer learning paradigm widely adopted
- Inspired differentiable NAS methods

MnasNet's legacy:

- Hardware-aware NAS became essential (ProxylessNAS, FBNet, Once-for-All)
- Multi-objective optimization for NAS standardized
- Mobile inverted bottleneck convolutions popularized
- Direct influence on EfficientNet family

9.2 Complementary Approaches

While addressing different primary objectives, the papers are complementary:

- NASNet focuses on *what* architectures transfer
- MnasNet focuses on *how* to optimize for deployment

An ideal approach would combine NASNet's transferability with MnasNet's platform awareness.

10 Methodological Insights

10.1 Search Space Design Principles

Both papers demonstrate that **search space design is critical**:

1. **Modularity:** Cell-based approaches enable tractable search
2. **Diversity:** MnasNet's factorization allows layer heterogeneity
3. **Expressiveness:** Sufficient operations to discover novel patterns
4. **Constraints:** Factorization balances flexibility and search cost

10.2 Optimization Strategy

Single vs. Multi-Objective:

- NASNet's single-objective approach is simpler but requires post-hoc latency optimization
- MnasNet's multi-objective formulation directly addresses deployment constraints

Proxy Tasks: Both use CIFAR-10 as a proxy, raising questions about:

- Generalization to very different tasks
- Optimal proxy dataset selection
- Trade-offs between proxy accuracy and search efficiency

11 Reproducibility and Practicality

11.1 Implementation Accessibility

- **NASNet:** Code released on TensorFlow TPU repository
- **MnasNet:** Code available on TensorFlow TPU models

Both provide sufficient detail for reproduction, though computational requirements remain high.

11.2 Practical Deployment

NASNet: Best suited for scenarios prioritizing accuracy over latency.

MnasNet: Explicitly designed for mobile deployment, making it more practical for resource-constrained applications.

12 Future Directions

Based on these papers, promising research directions include:

1. **Unified frameworks:** Combining transferability with multi-objective hardware awareness
2. **Reduced search cost:** Weight sharing (ENAS), differentiable methods (DARTS)
3. **Broader objectives:** Energy consumption, memory footprint, privacy
4. **Task-specific search:** Moving beyond image classification
5. **Theoretical understanding:** Why certain architectures transfer well

13 Conclusion

NASNet and MnasNet represent complementary advances in Neural Architecture Search:

NASNet pioneered the transferable cell-based search paradigm, demonstrating that architectures discovered on small datasets can achieve state-of-the-art performance on ImageNet. Its key contribution is the NASNet search space that decouples architecture complexity from network scale.

MnasNet introduced platform-aware NAS with multi-objective optimization, directly addressing the critical gap between theoretical performance (FLOPs) and real-world deployment (latency). Its factorized hierarchical search space enables layer diversity while maintaining search tractability.

Together, these papers established foundational principles that continue to guide NAS research:

- Search space design is paramount
- Modularity enables transferability
- Real-world constraints must be explicit
- Multi-objective formulations provide practical flexibility

While both require substantial computational resources, they demonstrate the viability of automated architecture design and have catalyzed subsequent work in efficient NAS methods. The combination of NASNet's transferability insights and MnasNet's platform awareness represents an ideal target for future unified NAS frameworks.

Acknowledgments

This review synthesizes the contributions of both research teams at Google Brain and Google Inc., whose work has significantly advanced the field of automated machine learning and mobile computer vision.