JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 1, pp. 59-68, January 2026
https://doi.org/10.9708/jksci.2026.31.01.059

NA-Search: Differentiable Search of Normalization and Activation

Wangduk Seo*

***Assistant Professor, Div. of AI Computer Science and Engineering, Kyonggi University, Suwon, Korea**

[Abstract]

In this paper, we propose NA-Search, a novel differentiable neural architecture search framework that targets normalization-activation operations to achieve both efficiency and lightweight design in on-device environments. Conventional differentiable neural architecture search methods evaluate all candidate operations during search, resulting in computational and memory overhead that limits applicability on resource-constrained devices. To address this limitation, NA-Search reconstructs the search space around normalization-activation combinations and applies a k-sampling that selects only k candidates, reducing the computational and memory cost of neural network evaluation. Experiments on CIFAR-10 dataset show that the proposed model achieves 89.75% accuracy with 0.124M parameters, outperforming widely used lightweight neural network in terms of accuracy-efficiency trade-off. Additional analyses show that adjusting the value of k during search contributes to improved stability and enhanced final performance, highlighting the effectiveness of the proposed sampling-based search design.

▸ **Key words**: Neural Architecture Search, Lightweight Neural Network Design, On-device Deep Learning, Differentiable Architecture Search, Operation Sampling Strategy

[요 약]

본 연구에서는 온디바이스 환경에서 요구되는 효율성과 경량 설계를 동시에 달성하기 위해 정규화-활성화 연산을 대상으로 하는 새로운 미분 기반 신경망 구조 탐색 프레임워크인 NA-Search를 제안한다. 기존 미분 가능한 신경망 구조 탐색 방법은 탐색 과정에서 모든 후보 연산을 평가하여 계산 및 메모리 오버헤드를 발생시키며, 자원 제약이 있는 장치에서의 적용성을 제한한다. 이러한 한계 해결을 위해 NA-Search는 정규화-활성화 조합을 중심으로 탐색 공간을 재구성하고, 단 k개의 후보만 선택하는 k-샘플링 전략을 적용함으로써 신경망 평가의 계산 및 메모리 비용을 크게 줄일 수 있다. CIFAR-10 데이터셋 실험 결과, 제안된 모델은 단 0.124M 매개변수로 89.75% 정확도를 달성하여 정확도-효율성 측면에서 널리 사용되는 경량 신경망을 능가할 수 있었다. 추가적인 분석을 통해 탐색 중 k값 조정이 안정성 향상과 최종 성능 개선에 기여함을 확인하여, 제안된 샘플링 기반 탐색 설계의 효과성을 입증하였다.

▸ **주제어**: 신경망 구조 탐색, 경량 신경망 설계, 온디바이스 딥러닝, 미분 기반 구조 탐색, 연산 샘플링 기법

• First Author: Wangduk Seo, Corresponding Author: Wangduk Seo
*Wangduk Seo (wdseo@kyonggi.ac.kr), Div. of AI Computer Science and Engineering, Kyonggi University
• Received: 2025. 11. 25, Revised: 2025. 12. 06, Accepted: 2025. 12. 22.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 엣지 및 온디바이스 환경에서 합성곱 신경망(Convolutional Neural Network, CNN)의 배치가 빠르게 확산되고 있다[1]. 이러한 환경에서 배치되는 CNN들은 주로 전문가가 주어진 데이터 및 디바이스 환경에 맞추어 직접 설계한 모델이거나, 기존에 널리 사용되는 대표 구조들 중 디바이스의 제약에 부합하는 모델을 선택해 적용하는 경우가 많다[2]. 해당 접근은 가능한 신경망 구조의 조합이 기하급수적으로 많아, 모든 디바이스와 태스크에 최적화된 모델을 사람이 직접 설계하기 어렵다는 한계를 가진다[3]. 또한, 기존에 널리 사용되는 모델을 그대로 적용하는 경우, 각 디바이스의 하드웨어 제약이나 수행해야하는 데이터 특성에 최적화되지 않은 구조가 사용될 가능성이 존재한다. 이러한 한계로 인해, 최근 주어진 자원 제약과 정확도, 지연시간, 메모리 사용량 등의 목표 지표에 맞추어 최적의 신경망 구조를 자동으로 탐색하는 신경망 구조 탐색(Neural Architecture Search, NAS) 연구가 다수 보고되고 있다[4].

최적 신경망 구조를 자동으로 탐색한다는 점에서, 강화학습이나 진화 알고리즘 등의 다양한 최적화 기반의 NAS가 제안되어 왔다. 그중에서도 빠른 탐색과 모델 학습을 동시에 수행할 수 있는 미분 기반 NAS가 많은 주목을 받고 있으며, 대표적인 예로 Differentiable Neural Architecture Search(DARTS)가 있다[5]. DARTS는 신경망의 셀 내부에서 노드 간 모든 후보 연산을 병렬로 연결한 Supernet을 구성한 뒤, 연속적인 이완을 통해 각 연결에 배치될 최적의 연산을 선택하는 방식으로 최적화를 수행한다.

DARTS는 다양한 데이터셋과 응용 도메인에서 기존 NAS 기법들 보다 우수한 성능을 보이며 널리 활용되고 있다. 그러나 이러한 DARTS 기반 방법을 온디바이스 환경에 직접 적용하기에는 두 가지 근본적인 한계가 존재한다. 첫째, 모든 셀과 노드 간 연결에 다양한 합성곱 연산을 포함하는 Supernet은 모델 파라미터 수가 급격히 증가하여, 메모리 용량이 제한된 디바이스에서는 탐색 과정 자체가 어려워질 수 있다[6]. 둘째, 탐색 과정에서 모든 후보 연산이 병렬로 동시에 활성화되므로, 후보 연산의 개수 M이 증가함에 따라 순전파 시 요구되는 메모리와 연산량이 선형적으로 증가한다[12]. 즉, 정확도를 향상시키면서도 탐색 중 메모리 요구량을 최소화해야 하는 온디바이스 환경의 제약과는 부합하지 않는다. 이러한 한계를 해결하기 위해서는 탐색 비용을 근본적으로 줄이면서도 모델 구조를 효율적으로 최적화할 수 있는 새로운 접근이 필요하다.

본 연구에서는 온디바이스 환경의 제약을 고려하면서도 DARTS의 탐색 효율성을 유지할 수 있는 새로운 미분 기반 탐색 기법인 Differentiable Search of Normalization and Activation(NA-Search)을 제안한다. 기존의 NAS 방법들이 대부분 합성곱 연산의 종류나 연결 구조를 탐색하는 데 초점을 맞춘 반면, 본 연구는 상대적으로 간과되어 온 정규화(Normalization)와 활성화(Activation) 연산의 조합 공간을 새로운 탐색 대상으로 설정하였다. 이로써, 파라미터 수가 많고 연산량이 큰 합성곱 연산 대신 비교적 경량의 연산을 탐색 대상으로 삼아 탐색 중 요구되는 메모리와 연산 비용을 크게 줄이면서도 모델 성능을 향상시킬 수 있다. 또한 탐색 중 각 셀에서 활성화되는 후보 연산의 수를 제한하는 k-샘플링 전략을 도입하여, 탐색 효율성과 자원 사용량 간의 균형을 조절할 수 있도록 설계하였다.

II. Related Works

1. Efficient CNNs based on manual design

NAS를 통한 자동화된 모델 구조 탐색 이전까지는, CNN의 구조는 주로 전문가의 설계에 의존하였다. 특히 온디바이스 환경에서도 CNN을 포함한 심층 신경망을 효율적으로 배치하기 위해 모델의 경량화 전략이 많이 연구되어 왔다. 대표적으로 MobileNet 계열은 합성곱 연산을 Depthwise Separable Convolution과 Pointwise Convolution으로 분리하여 연산량과 파라미터 수를 획기적으로 감소시켰다[7]. 이어서 MobileNetV2에서는 Inverted Residual Block과 Linear Bottleneck 구조를 도입해 특징 추출의 손실을 줄이면서도 동시에 효율적인 표현 학습이 가능하게 하였다[8].

ShuffleNet은 Group Convolution 기반의 병렬 구조 합성곱 연산에서 발생하는 채널 간 정보 단절 문제를 해결하기 위해 Channel Shuffle 연산을 제안하여, 적은 연산량으로도 높은 정확도를 달성하였다[9]. 합성곱 연산 자체에 대한 경량화 외에도 신경망의 깊이, 너비, 해상도를 균형 있게 조절하는 EfficientNet 또한 기존 모델보다 우수한 정확도-효율성 균형을 달성하였다[10].

이 외에도 CNN의 구조적 혁신으로, SqueezeNet, ResNet, DenseNet 등이 제안되었으며, 이들은 Residual Connection, 병렬적 구조를 활용한 설계를 통해 효율성을 크게 향상시켰다[11].

그러나 이러한 수동 설계 기반 접근은 전문가의 경험과 직관에 크게 의존하기 때문에, 모델이 특정 하드웨어 환경이나 데이터셋에 최적화되어 있는 경우가 많다. 따라서 이러한 구조를 엣지 디바이스나 메모리 제약이 심한 온디바이스 환경에 그대로 적용할 경우, 지연시간, 메모리 사용량, 정확도 간 균형이 최적화되지 않을 수 있다. 또한 가능한 네트워크 구조의 조합이 기하급수적으로 증가함에 따라, 모든 태스크와 디바이스 환경에 맞는 최적 모델을 전문가가 직접 설계하는 것은 조합 최적화 관점에서 매우 어려운 문제로 볼 수 있다. 이러한 한계로 인해, 디바이스 환경, 데이터 특성에 최적화 된 모델을 자동으로 탐색하는 NAS 기술이 새로운 패러다임으로 주목받게 되었다.

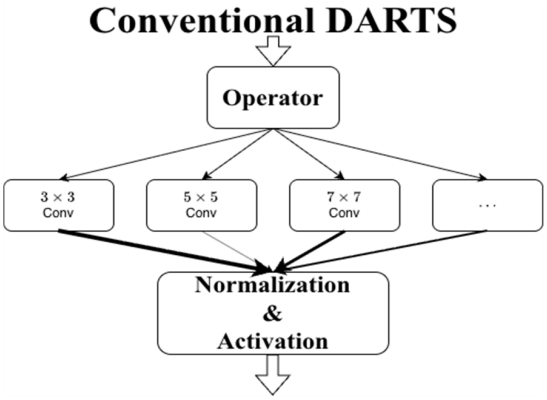

Fig. 1. Parallel operator structure of conventional DARTS

2. Differentiable neural architecture search

NAS는 사람이 직접 설계하던 모델 구조 탐색을 자동화하고, 주어진 정확도, 지연시간, 메모리 사용량 등의 복합 제약을 만족하는 최적의 구조를 알고리즘적으로 찾는 기술이다. 초기 NAS 기법들은 강화학습이나 진화 알고리즘을 활용해 후보 구조들을 반복해서 평가하는 방식이었으나, 방대한 탐색 공간으로 인해 탐색에 필요한 연산량과 시간 비용이 매우 커지는 문제가 있었다.

이러한 문제를 해결하기 위해 등장한 것이 DARTS다[5]. DARTS는 탐색 공간 내 후보 연산들을 연속 매개 변수화하고, 구조 파라미터 α와 모델 가중치 w를 교대로 최적화하는 방식으로 탐색 효율을 크게 향상시켰다. 셀 내부에 다수의 후보 연산 $o_1(x)$, $o_2(x)$, $\cdots$, $o_M(x)$를 병렬로 배치하고 각 연산에 대응하는 가중치 α_i를 할당하여 다음과 같이 출력값을 정의한다:

$$y = \sum_{i=1}^{M} \frac{e^{\alpha_i}}{\sum_{j=1}^{M} e^{\alpha_j}} \cdot o_i(x) \tag{1}$$

DARTS는 이러한 연속적 매개변수화를 통해 구조 탐색을 이중 레벨 최적화 문제로 재구성한다. 하위 수준에서는 훈련 데이터셋을 사용해 모델의 가중치 w를 학습하고, 상위 수준에서는 검증 데이터셋을 사용해 구조 파라미터 α를 최적화한다. 이러한 최적화 문제는 다음과 같은 식으로 표현할 수 있다:

$$\begin{aligned} &\min_{\alpha} L_{val}(w(\alpha),\ \alpha) \\ &w(\alpha) = \arg\min_{w} L_{train}(w,\ \alpha) \end{aligned} \tag{2}$$

이와 같이 DARTS는 두 데이터셋을 교대로 사용하여 모델의 가중치와 구조를 동시에 학습함으로써 효율적인 최적화가 가능하다. 그러나 Fig. 1에서 확인할 수 있듯이, 기존 DARTS는 하나의 edge에 여러 합성곱 기반 후보 연산을 병렬로 활성화하는 Supernet 구조를 사용하므로 탐색 중 메모리 및 연산 요구량이 최적 연산 후보 수 M에 비례해 증가하는 근본적 한계를 가진다. 또한 구조 파라미터 α의 경쟁 구조에서 특정 연산이 과도하게 선택되는 편향이나 일반화 성능 저하 등의 문제도 발생할 수 있다[6].

이러한 한계를 극복하기 위해 미분 기반 구조 탐색 과정에서 다양한 개선 방법들이 제안되었다. 먼저, 얕은 신경망으로 탐색을 시작해 점차 깊이를 늘리는 방식으로 탐색-평가간 간극을 줄이는 방법이 제안되었으며, 입력 채널 일부만을 샘플링함으로써 메모리 사용량을 낮추는 연구도 이루어졌다[12]. 최근에는 저메모리 구조를 도입하여 탐색 시간과 메모리 요구량을 함께 감소시킨 연구가 있다. 또한 후보 연산 간 경쟁 강도를 조절하거나 연산의 중요도를 동적으로 재측정하는 방식으로 탐색 안정성을 향상시키는 연구들도 활발히 수행되고 있다[13].

이러한 접근 방식은 탐색 효율성과 메모리 사용량의 개선을 통해 DARTS의 한계를 일정 부분 완화할 수 있으나, 온디바이스 환경과 같이 연산 자원과 메모리 용량이 극도로 제한된 상황에서는 여전히 근본적인 제약이 존재한다. 대부분의 기존 연구는 합성곱 연산 중심의 탐색 공간을 유지하고 있어서, 탐색 과정에서 요구되는 연산량과 메모리 사용량이 구조적으로 크다는 한계를 가진다. 또한 탐색 효율을 높이기 위해 후보 연산의 수를 줄이거나 입력 채널을 샘플링하는 방식은 일시적인 자원 절감 효과를 보이지만, 탐색 공간의 다양성을 축소시켜 최적 구조의 발견 가능성을 저하시킬 수 있다.

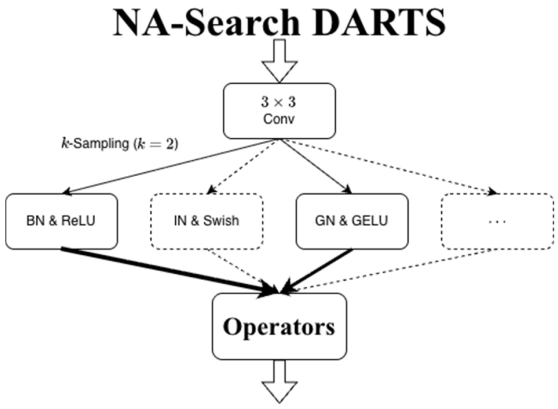

Fig. 2. Overview of the proposed NA-Search, illustrating k-sampling with $k = 2$

III. The Proposed Method

1. Motivation

기존의 미분 기반 신경망 구조 탐색 기법들은 탐색 효율성을 높이고 메모리 사용량을 줄이기 위해 다양한 개선 방안들을 제시해 왔다. 이러한 방법들은 DARTS의 연산 병렬성 문제와 과도한 자원 소모를 부분적으로 완화했지만, 온디바이스 환경과 같이 연산 자원과 메모리 용량이 극도로 제한된 상황에서는 여전히 근본적인 한계를 갖는다. 이는 대부분의 기존 연구들은 여전히 합성곱 연산 중심의 탐색 공간을 유지하고 있으며, 탐색 과정에서 모든 후보 연산이 동시에 활성화되는 구조적 한계를 벗어나지 못하고 있기 때문이다. 이로 인해 후보 연산 수가 증가할수록 순전파와 역전파에 필요한 메모리 및 연산 비용이 선형적으로 증가하며, 결국 메모리 용량이 제한된 환경에서는 탐색 자체가 불가능해지는 문제가 발생한다. 이에 본 연구는 합성곱 연산 중심의 복잡하면서도 고비용인 탐색 공간에서 벗어나, 정규화와 활성화 연산이라는 구조적으로 가벼우면서도 모델 표현력에 큰 영향을 미치는 요소를 새로운 탐색 대상으로 설정함으로써, 온디바이스 환경에서도 실행 가능한 경량형 미분 기반 구조 탐색을 제안하고자 한다.

더 나아가, 탐색 과정에서 모든 후보 연산을 동시에 활성화하는 기존 DARTS의 병렬 구조는 연산량과 메모리 사용량을 구조적으로 증가시키는 한계가 있다. 이를 해결하기 위해 본 연구에서는 후보 연산 전체 중 일부 연산만을 활성화하여 탐색에 참여시키는 k-샘플링 전략을 도입한다. 샘플링 과정은 단순 자원 절감 효과를 넘어 기존 DARTS가 가지는 문제인 과도한 특정 연산 쏠림 현상이나 탐색 불안전성 문제를 완화하여 탐색 품질을 적은 메모리 사용량으로도 유지할 수 있는 효과를 가져올 수 있다. 이에 본 연구에서는 위 문제들을 해결하기 위한 NA-Search의 구체적 탐색 구조와 k-샘플링 메커니즘을 다음 절에서 상세히 기술한다.

2. Proposed Method

제안하는 NA-Search는 Fig. 2에서 묘사된 것처럼 합성곱 기반 연산 중심의 기존 NAS 탐색 공간에서 벗어나, 모델 표현력과 학습 안정성에 중요한 영향을 미치는 정규화-활성화 연산의 조합을 새로운 탐색 대상으로 설정한다. 이러한 조합 기반 탐색 공간은 연산 비용이 매우 낮고 메모리 사용량이 작아, 온디바이스 환경에서도 실질적으로 탐색을 수행할 수 있는 경량 구조를 제공한다. 정규화 연산은 Batch Normalization(BN), Instance Normalization(IN), Layer Normalization(LN), Group Normalization(GN)으로 정규화 탐색 공간(N)을 설정하였으며, 활성화 함수 탐색 공간(A)은 ReLU, Swish, GELU, Mish로 구성하였다[14]. 따라서 NA-Search의 탐색 공간 O는 다음과 같이 정의된다:

$$O = \{(n_i,\ a_j) \mid n_i \in N,\ a_j \in A\} \tag{3}$$

정규화-활성화 조합은 총 16개의 후보 연산으로 이루어져 있으며, 이는 합성곱 기반 탐색에 비해 매우 가벼운 탐색 공간을 제공하면서도 모델의 성능에 직접적인 영향을 미치는 핵심 연산을 탐색 대상으로 포함한다는 점에서 충분한 탐색적 유의미성을 갖는다.

기존 DARTS에서는 모든 후보 연산이 동시에 활성화되며, 구조 파라미터 α에 기반한 연속 이완 방식으로 식(1)과 같이 계산한다. 이때 모든 후보 연산이 매 스텝마다 병렬로 수행되므로, 메모리 사용량과 연산량이 M에 선형적으로 증가하는 문제가 발생한다. 이러한 점은 온디바이스 환경에서는 탐색 자체를 불가능하게 만드는 주요 병목으로 작용한다.

이 문제를 해결하기 위해 NA-Search는 전체 후보 연산 O 중에서 매 스텝 k개의 연산만을 선택하여 계산하는 k-샘플링 전략을 도입한다. 구조 파라미터 α에 Gumbel 잡음을 더한 후 점수가 높은 k개의 연산만을 선택하는 방식이며, 선택된 연산 집합은 다음과 같이 정의된다:

$$S(k) = \arg\max_{S \subseteq 1,\cdots,\mathrm{M},\ |\mathrm{S}| = \mathrm{k}} \sum_{\mathrm{i} \in \mathrm{S}} (\alpha_\mathrm{i} + \mathrm{g_i}), \qquad g_i \sim Gumbel(0,1) \tag{4}$$

선택된 연산 집합 $S(k)$이 결정되면, NA-Search는 기존 DARTS와 동일하게 선택된 연산들의 출력을 연속적으

로 이완하여 가중합 형태로 계산한다. 단, 기존처럼 전체 후보 연산 M개를 모두 평가하는 대신, 선택된 k개의 연산만을 이용하여 출력을 계산하므로 연산식은 다음과 같이 단순화된다:

$$y = \sum_{i \in S(k)} w_i \cdot o_i,$$
$$w_i = \frac{e^{\alpha_i / \tau}}{\sum_{j \in S(k)} e^{\alpha_j / \tau}} \quad (5)$$

여기서 τ는 Gumbel-Softmax의 온도 파라미터로, 탐색 초기에는 상대적으로 큰 값을 사용하고 후반부로 갈수록 점진적으로 감소시키는 스케줄을 적용한다[15]. τ가 클 때에는 연산간의 확률 분포가 평탄해져 다양한 후보 연산이 고르게 탐색되며, 이는 탐색 초기 단계에서 구조 파라미터가 특정 연산에 과도하게 치우치는 DARTS의 문제를 완화하는 역할을 한다. 반대로 탐색 후반부에 τ를 낮추면 구조 파라미터가 상대적으로 우세한 연산에 보다 집중된 확률을 부여할 수 있게 된다. 이와 같이 출력 계산이 전체 후보 연산이 아닌 부분 집합 $S(k)$ 기반으로 이루어지면서, DARTS의 연속 이완 방식은 유지하되 실제 순전파에서 계산되는 연산 수가 줄어든다. 기존 DARTS의 연산 및 메모리 복잡도가 후보 연산 수 M에 비례하는 $O(M)$ 규모였다면, NA-Search에서는 k개의 연산만을 평가하므로 전체 복잡도가 $O(k)$로 감소한다. 이때 $k \ll M$인 상황에서는 탐색 비용이 근본적으로 축소되는 효과를 갖는다.

IV. Experimental Results

1. Experimental settings

본 연구에서는 제안한 NA-Search의 성능을 검증하기 위해 CIFAR-10 데이터셋을 사용하였다[5]. CIFAR-10은 열 개의 클래스로 구성된 32×32 크기의 컬러 이미지 데이터셋으로, 50,000장의 학습 이미지와 10,000장의 테스트 이미지로 이루어져 있다. 학습 데이터는 탐색 과정에서 훈련용 데이터와 검증 데이터로 분리하였으며, 데이터 증강 없이 정규화만 수행하였다. 실험은 Google Colab 환경에서 수행되었으며, 연산 가속을 위해 NVIDIA L4 GPU를 사용하였다. 또한 탐색 및 학습 과정의 재현성을 확보하기 위해 난수 시드를 고정하였다.

탐색 단계에서 네트워크 구조는 기존 DARTS와 동일하게 세 개의 계층으로 구성되며, 각 계층은 [16, 32, 64]의 출력 채널 수를 갖고 각 계층마다 3개의 셀을 배치하였다. 각 셀 내부에는 정규화-활성화 연산 후보들로 이루어진 연산 집합이 배치되며, NA-Search는 이 후보들 중에서 학습 과정에서 선택된 연산만을 활성화하여 해당 셀의 출력을 계산함으로써 최적의 연산 조합을 탐색한다. 정규화 및 활성화 연산 후보 집합은 3장에서 제시한 탐색 공간을 따른다. 탐색 과정에서 각 셀에서 샘플링되는 연산의 개수 k는 초기값을 $k = 2$, 최종값을 $k = 4$로 설정하였으며, 전체 에폭 진행에 따라 선형적으로 증가시키는 Linear Increase 스케줄을 적용하였다. 이와 같은 설정은 탐색 초기에는 적은 수의 후보 연산만을 평가하여 연산 비용과 메모리 사용량을 최소화하고, 에폭이 진행될수록 샘플링 범위를 점차 넓혀 더 많은 후보 연산을 고려함으로 구조 파라미터가 안정적으로 수렴할 수 있도록 설계한 것이다. 이러한 방법의 효과성을 확인하기 위해, 본 연구에서는 심층 분석 절에서 k값을 고정, 감소, 증가시키는 다양한 설정에 대해 탐색 결과와 최종 정확도를 비교하였다.

탐색 단계에서는 CIFAR-10 데이터셋을 사용하여 NA-Search 기반 구조 탐색을 수행하였다. 전체 학습 데이터의 90%를 탐색 과정의 학습용 데이터로, 나머지 10%를 구조 파라미터 최적화를 위한 검증용 데이터로 분리하였으며, 모델 가중치와 구조 파라미터는 총 50 에폭 동안 교대로 갱신되었다. 도출된 최종 구조는 테스트 데이터셋을 사용하여 평가하기 전에, 동일한 CIFAR-10 학습 데이터 전체를 이용해 200 에폭 동안 재학습하였다. 재학습 단계 및 탐색 단계 모두에서 배치 크기는 128, 입력 해상도는 32×32로 설정하였다. 옵티마이저는 Adam을 사용하였으며 초기 학습률은 0.001로 설정하였다. 비교 대상 모델로는 MobileNet[8], ShuffleNet[9], EfficientNet[10], ResNet-18[17] 등 널리 사용되는 경량 및 표준 CNN 모델들을 선정하였으며, 모든 비교 모델은 동일한 데이터 전처리와 학습 설정을 적용하여 제안 모델과 공정한 비교가 가능하도록 구성하였다.

2. Experimental results

본 절에서는 제안한 NA-Search의 탐색 결과와 최종 성능을 다양한 관점에서 분석한다. 먼저 NA-Search가 탐색 과정에서 도출한 최종 정규화-활성화 조합을 테이블로 정리하고, 이어서 기존 수동 설계 기반 CNN 모델과의 성능을 비교하였다. 이후 정확도-파라미터 수, 정확도-레이턴시, 정확도-메모리의 세 가지 좌표계에서 제안 방법의 위치를 시각적으로 확인함으로써, 경량성, 효율성, 정확도 측면에서의 우수성을 종합적으로 검증하였다.

Table 1은 NA-Search가 CIFAR-10 데이터셋에 대해

수행한 구조 탐색 결과를 요약한 것으로, 총 9개의 셀 각각에서 최종적으로 선택된 정규화-활성화 연산 조합을 제시한다. 각 셀은 탐색 과정 동안 구조 파라미터의 학습을 통해 16개의 후보 연산 중 하나를 선택하게 되며, 표에는 해당 선택 결과만을 간결하게 나타내었다. 첫 번째 열은 네트워크 내에서의 셀 순서를 의미하며, 두 번째 열은 선택된 최종 연산 조합을 보여준다.

탐색 결과를 보면, 모든 셀에서 동일한 연산 조합이 선택된 것이 아니라 셀의 위치에 따라 서로 다른 정규화-활성화 구성이 선택되었음을 확인할 수 있다. 이는 제안된 NA-Search가 네트워크 전반에 걸쳐 단일 연산을 일괄적으로 적용하는 기존 방식과 달리, 각 셀의 역할과 입력 특징 분포에 따라 구체적인 연산을 선택할 수 있도록 설계되어 있음을 의미한다. 이러한 셀 단위의 비균일한 구조는 표현 학습의 유연성을 높이는 요인으로 작용하며, 다양한 조합이 성능에 미치는 영향은 이후 심층 분석 절에서 보다 상세히 논의한다.

Table 2는 제안한 NA-Search 모델과 대표적인 경량 CNN 모델들 간의 분류 정확도(Top-1 Accuracy) 및 모델 파라미터 수를 비교한 결과를 제시한다. 표는 모델 이름, CIFAR-10 테스트 정확도, 그리고 파라미터 수를 포함하며, 가장 우수한 수치는 볼드체로 강조 표시하였다.

제안 모델은 단 0.124M이라는 매우 작은 파라미터 규모로도 89.75%의 테스트 정확도를 기록하였다. 이는 ResNet-18과 EfficientNet-B0 등 상대적으로 규모가 큰 기존 모델들과 비교해 파라미터 수는 수십 배 이상 적으면서도 경쟁력 있는 정확도를 제공한다는 점에서 의미가 있다. 또한 제안 모델은 MobileNetV2, ShuffleNetV2 등 기존 경량 모델들 보다 더 높은 정확도를 보이면서도 오히려 파라미터 수는 훨씬 작아, 정확도-모델 크기 측면에서 매우 높은 효율성을 보여준다.

Table 3은 제안한 NA-Search 모델과 비교모델들에 대해, 단일 배치 기준 추론 지연시간 및 학습 중 피크 메모리 사용량을 정량적으로 비교한 결과를 나타낸다. 비교 결과, 제안 모델은 0.99ms의 가장 낮은 추론 지연시간을 기록하여, 모든 비교 대상 중 가장 빠른 실행 속도를 보였다. 피크 메모리 사용량은 제안 모델이 ShuffleNetV2, MobileNetV3[16]에 이어 가장 낮은 메모리 사용량을 보이고 있으며, 이는 경량 모델들과 유사한 수준의 메모리 요구량을 갖는다는 것을 의미한다.

Table 1. Normalization-activation combinations selected by NA-Search for each cell in the final architecture.

Cell Index	Selected Operation
1	BN + GELU
2	GN + GELU
3	GN + GELU
4	LN + GELU
5	BN + Swish
6	BN + Swish
7	BN + Swish
8	BN + Swish
9	GN + GELU

Table 2. Comparison of classification accuracy and model size between the proposed NA-Search architecture and baseline CNN models.

Model	Top-1 Accuracy (%)	Params (M)
Proposed	**89.75**	**0.124**
ResNet-18	89.14	11.182
EfficientNet	87.53	4.020
MobileNetV2	85.27	2.237
ShuffleNetV2	80.40	1.264
MobileNetV3	73.86	1.528

Table 3. Comparison of inference latency and peak training memory between the proposed NA-Search architecture and baseline CNN models.

Model	Latency (ms)	Peak Train Memory (MB)
Proposed	**0.99**	115.23
ResNet-18	2.60	201.13
EfficientNet	8.07	188.00
MobileNetV2	5.25	139.81
ShuffleNetV2	6.48	90.44
MobileNetV3	5.29	**75.30**

본 연구에서는 제안한 NA-Search가 모델 구조 효율성, 연산 효율성, 메모리 효율성 측면에서 기존 모델들과 어떠한 차이를 보이는지 시각적으로 분석하기 위해 여러 평가 지표에서 산점도를 제시한다. 각 그래프는 모델의 정확도와 파라미터 수, 지연시간, 메모리 사용량 간의 관계를 나타내며, 이를 통해 NA-Search가 다양한 자원 제약 조건에서 어떤 위치에 놓이는지 명확하게 확인할 수 있다.

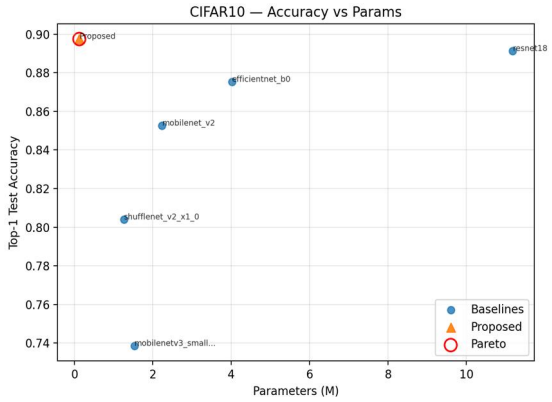

Fig. 3. Comparison of Top-1 accuracy and parameter size among NA-Search(Proposed) and baseline models on CIFAR-10.

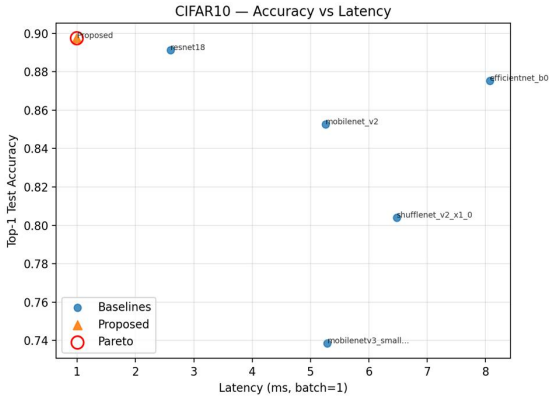

Fig. 4. Comparison of Top-1 accuracy and single-batch latency among NA-Search(Proposed) and baseline models on CIFAR-10.

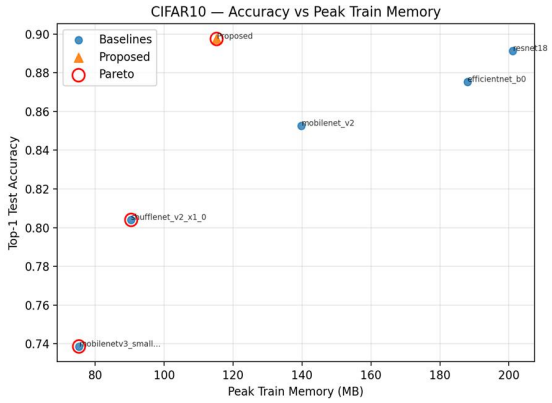

Fig. 5. Comparison of Top-1 accuracy and peak training memory usage among NA-Search(Proposed) and baseline models on CIFAR-10.

Table 4. Comparison of classification accuracy and inference latency on CIFAR-100 dataset.

Model	Top-1 Accuracy (%)	Latency (ms)
Proposed	**65.74**	**1.32**
ResNet-18	60.47	2.59
EfficientNet	57.86	8.38
MobileNetV2	58.50	5.33
ShuffleNetV2	53.14	6.91
MobileNetV3	45.57	5.65

Fig. 3은 제안한 NA-Search를 통해 탐색된 최종 구조가 기존 CNN 모델들과 비교하여 어떤 정확도-파라미터 특성을 갖는지를 시각적으로 보여준다. 해당 산점도는 모델 크기 대비 성능 효율성을 한눈에 파악하기 위해 제시되었다. 그래프에서 확인할 수 있듯이, 제안 모델은 0.124M이라는 매우 작은 파라미터 수에도 불구하고 높은 정확도를 유지하여, 정확도-파라미터 측면에서 비교 모델들 대비 우수한 효율성을 나타낸다.

Fig. 4는 NA-Search로 탐색된 제안 모델의 정확도와 추론 시간 간 관계를 기존 모델과 함께 비교한 결과이다. Fig. 3에서의 결과와 유사한 양상을 보이며, 마찬가지로 정확도-지연시간 측면에서 높은 효율성을 보인다. 전체적인 분포를 살펴보면, 제안 모델은 가장 작은 지연시간을 유지하면서도 높은 정확도를 기록하여, 두 지표 모두 다른 모델 대비 우수한 위치에 자리하고 있음을 확인할 수 있다.

Fig. 5는 제안 모델의 정확도-피크 학습 메모리 사용량 간 관계를 나타내며, 메모리 효율성 관점에서 기존 모델과의 비교를 제공한다. 제안 모델은 일부 경량 모델에 비해 약간 높은 피크 메모리를 사용하지만, ResNet-18과 EfficientNet 등 상대적으로 중대형 모델 대비 현저히 적은 메모리를 학습 중에 사용하는 것을 확인할 수 있다. ResNet-18이 온디바이스에서 활용되는 대표적인 딥러닝 모델임을 고려하면, 제안 모델은 동일한 환경에서 훨씬 적은 메모리로 유사한 수준의 정확도를 달성할 수 있어 실제 배치 관점에서도 높은 실용성을 갖는다. 또한 연산 비용이 큰 합성곱 연산 중심의 구조 대신 정규화-활성화 조합을 탐색 대상으로 사용함으로써 모델 크기와 메모리 사용량을 동시에 줄이면서도 정확도 저하 없이 효율적인 구조를 확보했음을 Fig. 5를 통해 확인할 수 있다.

추가적으로, 제안한 NA-Search의 일반화 성능을 평가하기 위해 보다 복잡한 데이터 분포를 갖는 CIFAR-100 데이터셋에 대한 실험을 수행하였다. CIFAR-100은 CIFAR-10과 동일한 해상도를 가지지만, 100개의 클래스와 더 높은 시각적 다양성을 포함하고 있어 모델의 표현력

및 일반화 능력을 더욱 엄격하게 평가할 수 있는 벤치마크로 활용될 수 있다. Table 4는 CIFAR-100에서의 분류 정확도와 단일 배치 기준 평균 추론 시간을 요약한 결과이다. 제안 모델은 65.74%의 정확도와 1.32ms의 추론 시간을 기록하여, 모든 비교 대상 모델보다 높은 정확도를 달성함과 동시에 빠른 추론 속도를 유지하였다. 이러한 결과는 정규화-활성화 조합 탐색이 데이터 복잡도가 증가하더라도 구조적 적응성을 확보할 수 있음을 보여준다.

3. In-depth analysis

본 절에서는 제안한 NA-Search의 구조적 특성이 최종 모델 성능에 어떤 영향을 미치는지 보다 정밀하게 분석한다. 이를 위해 세 가지 관점에서 실험을 수행하였다. 첫째, 정규화-활성화 조합 탐색의 효과성을 확인하기 위해 NA-Search로 탐색된 구조와 기존에 흔히 사용되는 고정 조합(BN+ReLU)의 구조를 동일한 학습 설정에서 비교하였다. 또한 백본(Backbone) 구조와 연산 조합 탐색의 분리된 효과를 검증하기 위해, ResNet-18의 합성곱 구조를 그대로 유지하되, 정규화-활성화 연산만을 NA-Search로 탐색하는 실험을 수행하였다. 이를 통해 성능 향상이 연산 조합 탐색 자체에 의해 발생한 것인지, 혹은 백본 구조 차이에서 기인한 것인지 평가하였다. 둘째, k-샘플링 전략의 동작 특성을 명확히 평가하기 위해 k값을 고정, 증가, 감소시키는 세 가지 시나리오에 대해 탐색 성능을 비교 분석하였다. 셋째, 제안 모델의 온디바이스 적용 가능성을 검증하기 위해 스마트폰 환경에서 CoreML 기반의 실측 추론 지연시간을 측정하여, 기존 경량 CNN 모델들과의 비교를 통해 실제 환경에서의 실행 효율성을 분석하였다.

Fig. 6는 NA-Search로 탐색된 최종 구조와 모든 셀에 BN+ReLU 조합을 고정 적용한 구조를 비교한 결과를 나타낸다. 두 모델은 동일한 네트워크 토폴로지와 동일한 학습 설정에서 훈련하였으며, 차이는 정규화-활성화 조합을 셀마다 탐색했는지 여부이다. 그림에서 확인할 수 있듯이, NA-Search는 고정 조합 모델 대비 더 높은 정확도를 보여주고 있다. 이는 네트워크 셀마다 입력 분포나 역할이 다르기 때문에, 단일 정규화-활성화 조합을 강제하는 기존 방식보다 셀 단위의 맞춤형 연산 조합이 특성 추출에 더 적합함을 의미한다. 또한 추론 지연 시간 면에서도 양쪽 모델이 모두 경량 구조를 기반으로 하기 때문에 유사한 수준을 유지하여, 제안된 연산 조합 탐색이 추론 효율성을 저해하지 않으면서 성능을 향상시킨다는 점을 확인할 수 있다.

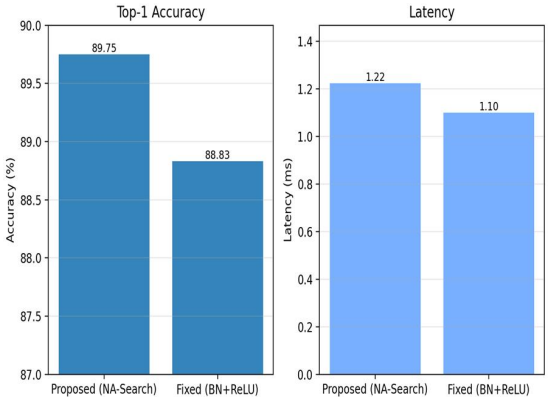

Fig. 6. Comparison between NA-Search architecture and a fixed BN+ReLU architecture in terms of Top-1 accuracy and inference latency.

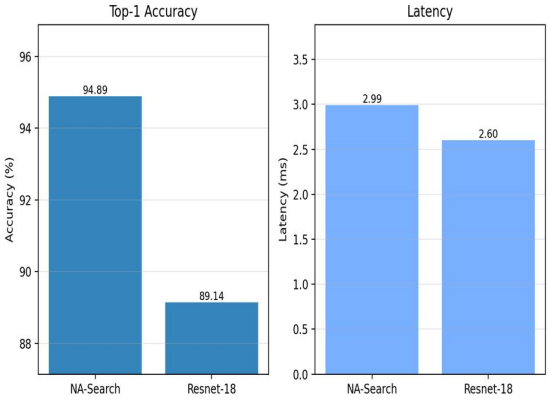

Fig. 7. Comparison of Top-1 accuracy and inference latency between ResNet-18 and ResNet-18 enhanced with NA-Search.

Fig. 7은 보다 복잡한 백본 구조인 ResNet-18을 대상으로 정규화-활성화 조합 탐색의 독립적 효과를 분석한 결과이다. 실험 결과, NA-Search를 적용한 모델은 기본 ResNet-18 대비 유의미한 정확도 향상을 기록하였으며, 추론 지연 시간은 소폭 증가하는 수준에서 유지되었다. 이러한 결과는 정규화-활성화 연산 조합의 선택이 합성곱 백본과 독립적으로 성능 향상에 기여하는 구조적 요소임을 뒷받침한다. 두 실험 결과는 서로 다른 모델 규모와 조건에서 일관된 경향을 보이며, 정규화-활성화 조합 탐색이 단일 백본 또는 특정 네트워크 구성에만 국한되지 않고 일반적인 신경망 구조 최적화 요소로 기능함을 확인할 수 있다.

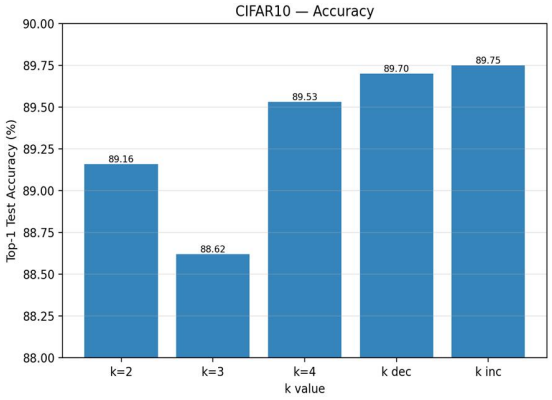

Fig. 8. Effect of different k-sampling strategies on Top-1 accuracy during architecture search.

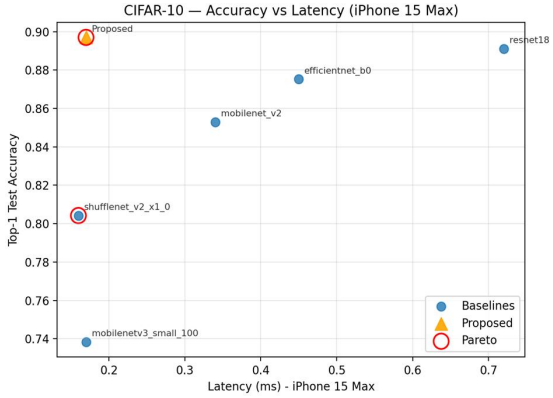

Fig. 9. Accuracy-latency comparison on CIFAR-10, evaluated through real-world inference measurement on an iPhone 15 Pro Max.

Fig. 8은 탐색 과정에서 샘플링 되는 연산의 개수 k를 2, 3, 4로 고정하거나, 4에서 2로 선형 감소시키거나, 2에서 4로 선형 증가시키는 등 서로 다른 설정을 적용했을 때 탐색 성능이 어떻게 달라지는지를 비교한 결과이다. 실험 결과 k를 선형적으로 증가시키는 전략이 가장 높은 정확도를 달성하였다. 이는 탐색 초기에는 상대적으로 작은 k 값을 유지함으로써 연산량과 메모리 부담을 최소화하고, 소수 후보 연산만을 활용해 구조 파라미터 간의 초기 분별력을 확보할 수 있도록 하며, 탐색이 진행될수록 k를 점진적으로 증가시켜 더 많은 연산 조합을 평가함으로써 부분적으로 형성된 최적화 방향을 확장하고 보강할 수 있게 한다. 또한 탐색 후반부에 보다 안정적으로 수렴하도록 도와, 최종적으로 높은 정확도를 달성하는데 기여한다.

심층 분석 실험의 결과는 다음 두 가지 주요 결론을 제시한다. 첫째, 정규화-활성화 조합 탐색은 단일 조합을 사용하는 전통적 방식보다 모델 성능의 향상을 가져올 수 있다. 둘째, k-샘플링 전략은 탐색 품질과 비용 사이의 균형을 제어하는 핵심 요소이며, 특히 선형 증가 전략은 가장 큰 정확도 향상을 보였다. 이러한 결과는 제안 방법이 연산 비용을 크게 증가시키지 않으면서도 탐색 구조 자체의 효율성을 높일 수 있음을 실험적으로 보여준다.

Fig. 9는 제안 모델과 기존 경량 CNN 모델들을 iPhone 15 Pro Max 환경에서 직접 실행하여 얻은 실측 정확도와 실측 추론 지연시간의 관계를 비교한 결과이다. 제안 모델은 단일 배치당 평균 0.17ms로 두 번째로 낮은 지연시간을 기록하면서도, 가장 높은 89.70%의 정확도를 달성하였다. 반면, ShuffleNetV2는 가장 낮은 지연시간을 보였으나 정확도가 크게 감소하는 경향을 나타냈다. 이러한 결과는 제안 모델이 온디바이스 환경에서 요구되는 실시간성 및 정확도 간 균형을 달성하고 있음을 보여준다.

V. Conclusions

본 연구에서는 온디바이스 환경에서 요구되는 연산 효율성과 메모리 제약을 동시에 만족시키기 위해, 정규화-활성화 연산 조합을 탐색 대상으로 설정한 경량 미분 기반 신경망 구조 탐색 기법인 NA-Search를 제안하였다. 또한 매 스텝에서 소수의 후보 연산만을 활성화 하는 k-샘플링 전략을 도입함으로써 탐색 과정의 메모리 및 연산 부담을 크게 완화하였다. CIFAR-10 및 CIFAR-100 데이터셋에서의 실험을 통해 제안 방법의 효과성을 검증하였으며, 제안 모델은 각각 0.124M 파라미터로 89.75%, 0.50M 파라미터로 65.74%의 정확도를 달성하여 기존 경량 CNN 모델 대비 우수한 정확도-효율성 균형을 보였다.

그럼에도 불구하고 본 연구에서는 다음과 같은 한계를 지닌다. 첫째, 실험이 주로 CIFAR 계열의 소규모 이미지 데이터셋에 기반하므로, 보다 복잡하고 고해상도의 실제 응용 환경에서의 일반화 성능 검증이 필요하다. 둘째, 스마트폰에서의 제한적인 실측 결과 외에도, 다양한 모바일 SoC나 임베디드 보드에서의 실측 성능 분석이 추가적으로 요구된다. 셋째, 최신 온디바이스 NAS 기법과의 직접적인 비교가 수행되지 않아, 제안 방법의 상대적 위치를 보다 명확하게 평가할 필요가 있다. 따라서 향후 연구에서는 이러한 한계점을 보완하여 NA-Search의 범용성과 실용성을 강화할 계획이다. 또한 정규화-활성화 탐색 공간을 확장하거나, 합성곱 이외의 연산과 결합한 하이브리드 탐색 전략 등으로 제안 방법의 발전 가능성을 더욱 넓힐 수 있을 것으로 기대된다.

ACKNOWLEDGEMENT

This work was supported by Kyonggi University Research Grant 2025.

REFERENCES

[1] K. Sun, X. Wang, X. Miao, and Q. Zhao, "A review of AI edge devices and lightweight CNN and LLM deployment," Neurocomputing, Vol. 614, pp. 128791, 2025.

[2] J. Wang, C. Chen, S. Li, C. Wang, X. Cao, and L. Yang, "Researching the CNN collaborative inference for heterogeneous edge devices," Sensors, Vol. 24, No. 13, pp. 4176, 2024.

[3] Q. Li, C. Ma, H. Chen, X. Chen, and X. Yang, "Combinatorial progressive architecture search for crowd counting," Displays, Vol. 83, pp. 102686, 2024.

[4] M. Lupion, N. C. Cruz, E. M. Ortigosa, and P. M. Ortigosa, "A holistic approach for resource-constrained neural network architecture search," Applied Soft Computing, Vol. 172, pp. 112832, 2025.

[5] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," arXiv preprint arXiv:1806.09055, 2018.

[6] H. Wei, F. Lee, L. Xie, L. Liu, H. Yu, and Q. Chen, "CSC-DARTS: Efficient differentiable neural architecture search using channel splitting connections," Information Sciences, pp.122538, 2025.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520, Salt Lake City, USA, June 2018. DOI:10.1109/CVPR.2018.00474

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6848-6856, Salt Lake City, USA, June 2018. DOI:10.1109/CVPR.2018.00716

[10] M. Tan, and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," Proceedings of the 36th International Conference on Machine Learning (ICML), pp. 6105-6114, Long Beach, USA, June 2019.

[11] K. Kanwal, K. T. Ahmad, A. Shabir, L. Jing, H. Garay, L. E. P. Gonzalez, H. Karamti, and I. Ashraf, "Efficient CNN architecture with image sensing and algorithmic channeling for dataset harmonization," Scientific Reports, Vol. 15, No. 1, pp. 7552, 2025.

[12] Y. Xu, L. Xie, X. Zhang, X. Chen, G. -J. Qi, Q. Tian, and H. Xiong, "PC-DARTS: Partial channel connections for memory-efficient architecture search," arXiv preprint arXiv:1907.05737, 2019.

[13] C. Jin, J. Huang, and Y. Chen, "Neural architecture search via progressive partial connection with attention mechanism," Scientific Reports, Vol. 14, No. 1, pp. 6462, 2024.

[14] H. Liu, A. Brock, K. Simonyan, and Q. Le, "Evolving normalization-activation layers," Advances in Neural Information Processing Systems, Vol. 33, pp. 13539-13550, 2020.

[15] B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia, and K. Keutzer, "FBNet: Hardware-aware efficient Convnet design via differentiable neural architecture search," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10734-10742, June 2019.

[16] A. Howard, M. Sandler, G. Chu, L. -C. Chen, et al., "Searching for MobileNetV3," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314-1324, Seoul, Korea, Oct. 2019. DOI:10.1109/ICCV.2019.00140

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 770-778, June 2016.

Authors

Wangduk Seo is currently an assistant professor in the Division of AI Computer Science and Engineering, Kyonggi Univ. (KGU) in Suwon, Korea. Prior to joining KGU, he completed his postdoctoral research and received his Ph.D., M.S., and B.S. degrees from Chung-Ang Univ., Korea. His research interests include neural architecture search, feature selection, and efficient model learning.