

Article

EfficientMixer: A Lightweight Vision Architecture via Complementary Design Principles

Weidong Zhang , Baoxin Li , Huan Liu , Yezhou Yang and Ahmet Arda Dalyanci

School of Computing and AI, Arizona State University, Tempe, AZ 85281, USA; huanliu@asu.edu (H.L.); yz.yang@asu.edu (Y.Y.); adalyanc@asu.edu (A.A.D.)

* Correspondence: weidong.zhang@savvas.com (W.Z.); baoxin.li@asu.edu (B.L.)

Abstract

Lightweight vision models must maximize representational capacity under strict constraints on parameters and computational cost. We investigate whether principle-level architectural composition—the systematic integration of complementary design principles rather than optimization of individual components in isolation—can provide a more effective approach to lightweight model design. Motivated by this perspective, we propose EfficientMixer, a lightweight vision architecture that combines hierarchical feature scaling, isotropic spatial-channel mixing, and lightweight channel-spatial attention refinement, with each principle serving a complementary role within a unified architecture. We evaluate this composition under identical training settings and comparable parameter budgets across multiple image-classification datasets using baseline comparisons, component ablations, and computational-efficiency measurements. EfficientMixer consistently outperforms representative lightweight models while maintaining a constrained resource budget and achieves further performance gains when combined with Self-Competitive Distillation, showing that the architecture can benefit from improved training methods. These results show that systematically composing complementary architectural principles can improve representation quality and generalization in lightweight vision models, providing a practical design strategy for performance-constrained settings.

Keywords: self-competitive distillation; lightweight vision models; optimization strategy; lightweight attention; effective capacity; cross-domain generalization; multi-dataset evaluation

1. Introduction

Lightweight vision networks are increasingly important for deploying deep learning on resource-constrained devices [1,2]. Unlike large-scale models, they must satisfy strict constraints on parameters, memory, computation, and inference efficiency, limiting the applicability of conventional scaling strategies [3,4]. Consequently, a central challenge is to maximize representational capacity while maintaining a constrained resource budget [5]. Existing efforts toward this goal span several complementary directions, including lightweight convolutional architectures [6–18], efficient Transformer and hybrid architectures [19–26], neural architecture search [16,27–29], and efficient attention mechanisms [30–35].

Within lightweight convolutional networks, efficiency has been pursued through a variety of architectural mechanisms. Representative examples include depthwise separable convolutions [6], inverted bottlenecks [7], channel shuffling [9,10], and cheap feature generation [11], which reduce computation while retaining effective feature extraction. Other



Academic Editor: Guanqiu Qi

Received: 2 July 2026

Revised: 15 August 2026

Accepted: 16 August 2026

Published: 19 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

approaches employ neural architecture search to automatically identify efficient network configurations [16,27], or combine convolutional operations with Transformer-based components to improve representation capacity [19–22,36]. More recent studies have explored multi-scale feature refinement [37], pixel shuffling [38], and attention-based feature fusion [39], further demonstrating the diversity of strategies for improving representation under computational constraints.

Attention mechanisms provide another means of enhancing feature representations while avoiding the substantial computational overhead of full self-attention. Channel attention methods such as SENet and ECA [40,41], spatial and channel–spatial mechanisms such as CBAM, BAM, and Triplet Attention [42–44], coordinate attention [45], and global context modules [46–48] have therefore been widely investigated. In parallel, Transformer-based vision models have demonstrated strong representation capability through global or long-range interactions, but their computational and memory requirements can be challenging for resource-constrained deployment [1,4,25,49,50]. These developments suggest that lightweight attention can serve as a complementary feature-refinement mechanism within efficient convolutional architectures [40–42,45,51].

Despite these advances, lightweight architectural principles are often developed and evaluated independently. Limited effort has been devoted to systematically combining complementary convolutional and attention-based principles under matched parameter and computational budgets. This raises an important question: can their joint integration enhance representational capacity while preserving lightweight efficiency?

Motivated by this question, we propose EfficientMixer, a lightweight architecture that systematically composes three established but complementary design principles: (i) hierarchical feature scaling inspired by EfficientNet [14], (ii) isotropic spatial–channel mixing from ConvMixer [52], and (iii) lightweight channel–spatial attention refinement based on CBAM [42]. Its architectural contribution lies in coordinating these principles within a unified lightweight convolutional architecture for resource-constrained deployment. Specifically, hierarchical scaling allocates representational capacity across stages, spatial–channel mixing promotes efficient feature interaction within each stage, and attention-based fusion adaptively refines the resulting representations. By assigning these principles distinct and complementary roles, EfficientMixer provides a structured strategy for improving representation quality and accuracy under constrained parameter and computational budgets.

Experimental evaluation across multiple datasets under matched training settings and similar parameter budgets [5] shows that EfficientMixer consistently outperforms representative lightweight baselines. Ablation results further confirm that hierarchical scaling, spatial–channel mixing, and lightweight attention contribute complementarily to the overall performance, supporting their coordinated use for improving representation learning under constrained computational budgets.

Our major contributions can be summarized as follows:

- **Composition Strategy.** We introduce a lightweight design strategy that combines hierarchical scaling, spatial–channel mixing, and attention refinement.
- **EfficientMixer.** We develop EfficientMixer as a unified architecture that instantiates this composition strategy under a constrained resource budget.
- **Controlled Evaluation.** We evaluate EfficientMixer across multiple datasets and against representative lightweight models under matched training settings and comparable parameter budgets. Ablation studies assess component contributions.

2. Materials and Methods

2.1. Architectural Principles of Lightweight Vision Networks

Modern lightweight vision architectures improve representational efficiency under strict computational constraints through combinations of efficient convolutional processing, scaling strategies, and lightweight attention mechanisms [1,14]. Although these architectures differ substantially in implementation, many successful designs share common structural principles. To motivate the design of EfficientMixer, we examine representative principles underlying ConvMixer, EfficientNet, and MobileViT.

2.1.1. ConvMixer Principles

ConvMixer [52] exhibits three main design patterns: (A) *patch-based tokenization*, where input images are converted into patch-level representations using strided convolutions, mimicking Transformer-style tokenization while remaining fully convolutional (Equation (2)) [50]; (B) *isotropic processing*, where a constant feature dimensionality is maintained across layers and representations are refined through repeated application of identical blocks (Equation (1)) [53]; and (C) *decoupled spatial–channel mixing*, where spatial and channel interactions are separated via depthwise and pointwise convolutions, respectively (Equation (3)), enabling efficient feature transformation [6,7].

$$z^{(0)} = \phi(x), \quad z^{(l+1)} = z^{(l)} + g_{\theta}(z^{(l)}), \quad (1)$$

$$\phi(x) = \text{Conv}_{k,s}(x), \quad (2)$$

$$g_{\theta}(z) = W_p * (W_d * z), \quad (3)$$

where W_d and W_p denote depthwise and pointwise convolutions, respectively.

$$g_{\theta} = g_{\text{channel}} \circ g_{\text{spatial}}, \quad (4)$$

where $g_{\text{spatial}}(z) = W_d * z$ and $g_{\text{channel}}(z) = W_p * z$.

$$z^{(l)} \in \mathbb{R}^{H' \times W' \times C}, \quad \forall l. \quad (5)$$

2.1.2. EfficientNet Principles

From EfficientNet [14], we identify three design patterns: (A) *stem embedding*, where a convolutional stem maps RGB inputs into feature space, as defined in Equation (6); (B) *MBConv-based feature extraction*, where inverted bottleneck blocks combine channel expansion, depthwise convolution, and squeeze-and-excitation operations [7,40] (Equation (7)); and (C) *hierarchical scaling*, where spatial resolution is progressively reduced while channel capacity increases via compound scaling rules [14] (Equation (8)).

$$x^{(i+1)} = h_i(x^{(i)}), \quad (6)$$

$$h_i(\cdot) = \text{MBConv}(\cdot), \quad \text{MBConv} = \text{PWConv} \circ \text{DWConv} \circ \text{SE}, \quad (7)$$

$$H_{i+1} = \frac{H_i}{s_i}, \quad W_{i+1} = \frac{W_i}{s_i}, \quad C_{i+1} > C_i. \quad (8)$$

2.1.3. MobileViT Principles

MobileViT [19] introduces a global aggregation operator $\mathcal{A}(\cdot)$ that unfolds feature maps into tokens, applies self-attention, and projects them back to spatial form. This enables long-range dependencies through (A) convolution–attention coupling via QKV pro-

jections (Equation (9)) and (B) global contextual modeling via scaled dot-product attention, originally introduced in Vision Transformers [50] (Equation (10)).

$$Q, K, V = \text{Unfold}(x) W_{Q,K,V}, \quad (9)$$

$$\mathcal{A}(x) = \text{Fold}\left(\text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V\right). \quad (10)$$

2.2. EfficientMixer Architecture

As illustrated in Figure 1, rather than introducing an entirely new primitive operation, EfficientMixer investigates whether complementary inductive biases from existing lightweight architectures can be systematically composed to improve representation efficiency under strict computational constraints. Its primary contribution lies in this architectural composition strategy and the empirical validation of the resulting design, rather than in claiming individual components as fundamentally novel. EfficientMixer integrates design principles inspired by EfficientNet [14], ConvMixer [52], and MobileViT [19], organizing them according to complementary functions: hierarchical feature scaling for multi-resolution feature abstraction, isotropic feature mixing for efficient spatial–channel interaction, and lightweight attention refinement for adaptive feature enhancement. By jointly incorporating these principles within a unified parameter-constrained architecture, EfficientMixer provides a structured approach to improving the accuracy–efficiency trade-off of lightweight vision models under comparable computational budgets.

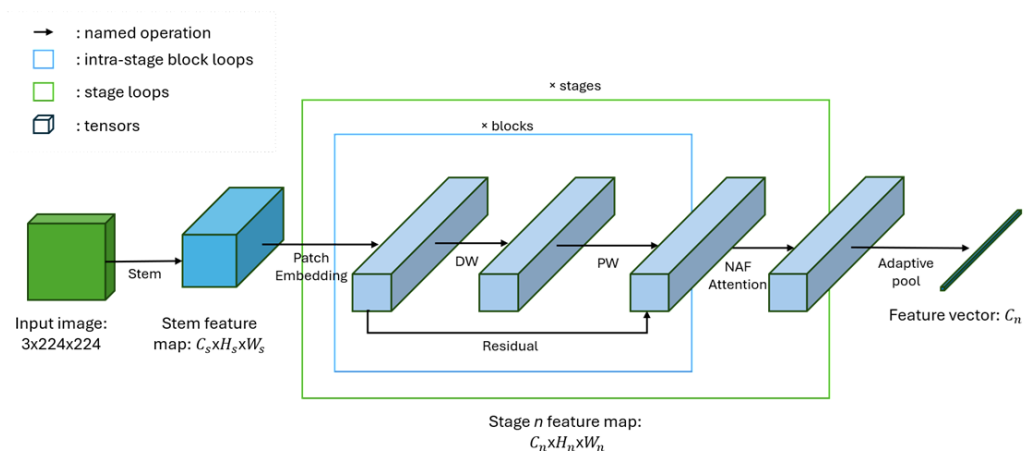


Figure 1. EfficientMixer Schematic Overview. DW and PW denote depthwise and pointwise convolutions, respectively. The architecture consists of repeated stages, each comprising repeated intra-stage blocks with residual connections, followed by NAF attention. Further details of the overall architecture and individual operations are provided in Tables 1–3.

- (A) **Hierarchical stem embedding.** A convolutional stem maps the input RGB image into a feature representation while preserving local spatial structure prior to downsampling, following EfficientNet-style design principles.
- (B) **Hierarchical spatial–channel scaling.** The network adopts EfficientNet’s stage-wise scaling, progressively reducing spatial resolution while increasing channel capacity (e.g., $H \times W \rightarrow \frac{H}{2} \times \frac{H}{2} \rightarrow \frac{H}{4} \times \frac{H}{4}$, $C : 64 \rightarrow 128 \rightarrow 256 \rightarrow 512$), enabling a transition from local spatial detail to high-level semantic features.
- (C) **Intra-stage isotropic mixing.** Following the ConvMixer design, features are refined at a fixed spatial resolution within each stage using repeated depthwise–pointwise convolutional blocks with residual connections. Depthwise convolutions capture local

spatial dependencies, pointwise convolutions enable channel mixing, and residual connections improve optimization stability.

- (D) **Nonlinear Attention Fusion (NAF).** Inspired by lightweight attention-based architectures [19], EfficientMixer adopts a learnable Nonlinear Attention Fusion (NAF) to extend CBAM [42]. Given $x \in \mathbb{R}^{C \times H \times W}$, CBAM produces $x_{cbam} = \text{CBAM}(x)$, which is fused with x as

$$\text{NAF}(x, x_{cbam}) = \alpha x + \beta x_{cbam} + \gamma(x \odot x_{cbam}), \quad (11)$$

where α , β , and γ are learned coefficients. This formulation combines identity preservation, attention refinement, and multiplicative feature interaction without introducing a new attention operator.

Together, these complementary architectural patterns provide a unified framework that balances hierarchical feature abstraction, spatial–channel interaction, and adaptive feature refinement. The effectiveness of this composition strategy is evaluated experimentally under controlled parameter and computational budgets. The resulting architectural patterns are summarized in Table 1.

Table 1. EfficientMixer design pattern mapping. Each component integrates architectural principles from EfficientNet, ConvMixer, and MobileViT into a unified hierarchical convolutional framework.

Component	EfficientMixer Design	Source Design Patterns
Stem	Convolutional stem for initial feature embedding	EfficientNet: stem convolution for low-level feature extraction
Hierarchical stages	Progressive spatial downsampling with channel expansion	EfficientNet: stage-wise scaling and hierarchical feature abstraction
Intra-stage isotropic blocks	Repeated depthwise–pointwise mixing blocks with residual connections for isotropic feature refinement	ConvMixer: isotropic architecture; decoupled spatial–channel mixing
Stage-level refinement	CBAM-based attention with nonlinear attention fusion (NAF)	MobileViT: lightweight attention for global contextual modeling; convolution–attention complementarity

Table 2. Overall architecture of the baseline EfficientMixer model. Each EfficientMixer stage consists of one patch embedding layer, three Intra-Stage Isotropic Blocks, and one NAF attention module.

Stage	Operator	Output	Channels	Repeat
Stem	Conv 3×3 + GELU + BN	224×224	32	1
Stem	Conv 3×3 + GELU + BN	224×224	64	1
Stage 1	EfficientMixer Stage	112×112	128	3
Stage 2	EfficientMixer Stage	56×56	256	3
Stage 3	EfficientMixer Stage	28×28	512	3
Stage 4	EfficientMixer Stage	14×14	256	3
Head	Adaptive GAP + FC	1×1	256	1

Table 3. Definition of the EfficientMixer building blocks.

Component	Structure
Stem	Conv $3 \times 3 \rightarrow$ GELU $\rightarrow$ BN $\rightarrow$ Conv $3 \times 3 \rightarrow$ GELU $\rightarrow$ BN
EfficientMixer Stage	Overall structure: Patch Embedding $\rightarrow$ Intra-Stage Isotropic Block $\times 3 \rightarrow$ NAF Attention
Patch Embedding	Conv 2×2 , stride 2 $\rightarrow$ GELU $\rightarrow$ BN
Intra-Stage Isotropic Block	DWConv $3 \times 3 \rightarrow$ GELU $\rightarrow$ BN $\rightarrow$ PWConv $1 \times 1 \rightarrow$ GELU $\rightarrow$ BN $\rightarrow$ Residual Addition
NAF Attention	Channel Attention $\rightarrow$ Spatial Attention
Classification Head	Adaptive Global Average Pool $\rightarrow$ Dropout ($p = 0.2$) $\rightarrow$ Fully Connected

2.3. Baseline EfficientMixer Model

To evaluate the proposed architecture under realistic mobile constraints, we instantiate a baseline EfficientMixer model containing approximately 2.5 million parameters. Matching parameter budgets across models ensures that performance differences primarily reflect architectural design rather than model scale or hyperparameter tuning. This setting reflects practical edge-deployment scenarios, where memory, computation, and latency constraints favor compact models. Maintaining a fixed parameter budget also enables a fair comparison of architectural efficiency, ensuring that observed performance differences arise from design choices rather than increased model capacity.

Tables 2 and 3 together provide a complete specification of the proposed EfficientMixer architecture. Table 2 summarizes the hierarchical network organization, including stage-wise resolutions, channel dimensions, and configurations, while Table 3 details the internal structure of each building block.

The EfficientMixer architecture follows a hierarchical design consisting of a lightweight convolutional stem, multiple EfficientMixer stages, and a compact classification head. The stem preserves the input resolution of 224×224 through two consecutive 3×3 convolutional layers, allowing the network to capture richer low-level representations before spatial reduction. This design avoids aggressive early downsampling commonly used in lightweight models, which can lead to information loss.

Each EfficientMixer stage performs progressive feature abstraction through three sequential operations: patch embedding, Intra-Stage Isotropic Blocks, and NAF attention. The patch embedding layer uses a 2×2 convolution with stride 2 to reduce spatial resolution while increasing channel capacity, enabling efficient hierarchical representation learning. The following Intra-Stage Isotropic Blocks maintain the spatial resolution and channel dimensionality while performing efficient feature mixing through depthwise and pointwise convolutions. Residual connections are incorporated to improve gradient propagation and optimization stability.

At the end of each stage, the NAF attention module performs lightweight feature recalibration by sequentially applying channel and spatial attention. This placement allows attention to operate on already aggregated features, improving discriminative representation learning with limited computational overhead.

After the final EfficientMixer stage, the resulting $14 \times 14 \times 256$ feature maps are compressed through adaptive global average pooling into a compact feature vector. A dropout layer is then applied for regularization before the fully connected classifier produces the final predictions. This design maintains a low-complexity prediction head while preserving the discriminative capability of the learned representation.

More generally, the proposed EfficientMixer architecture can be parameterized by the tuple

$$\mathcal{A} = \left\{ S, L_{\text{stem}}, \mathbf{C}_{\text{stem}}, p, \{C_i, H_i, W_i, L_i\}_{i=1}^S \right\}, \quad (12)$$

where S denotes the number of hierarchical stages, L_{stem} is the number of stem convolutional layers, C_{stem} denotes the stem channel dimensions, p is the patch size used for spatial downsampling, C_i is the output channel dimension, $H_i \times W_i$ is the feature-map resolution, and L_i is the number of Intra-Stage Isotropic Blocks in stage i . The baseline model presented in Table 2 corresponds to one specific instantiation of EfficientMixer, while the ablation studies systematically vary these architectural hyperparameters to quantify their individual contributions to model accuracy and computational efficiency.

We compare EfficientMixer against representative lightweight architectures with similar parameter counts. EfficientNet [14] and ConvMixer [52] are included because EfficientMixer is derived from architectural principles extracted from these models and because they achieved the strongest cross-domain performance among lightweight convolutional architectures in our previous evaluations [5,54]. MobileNet [6–8] serves as a representative baseline based on depthwise separable convolutions, while MobileViT [19] represents hybrid convolution–attention architectures employing lightweight Transformer blocks. In contrast, EfficientMixer combines hierarchical scaling, isotropic feature mixing, and lightweight CBAM-based attention within a fully convolutional framework.

We focus on representative lightweight architectures with comparable parameter budgets for controlled efficiency evaluation. Table 4 summarizes the model complexity and inference efficiency of the evaluated lightweight architectures using four complementary metrics: parameter count, floating-point operations (FLOPs), GPU inference latency, and average inference memory. Parameter count reflects model size, FLOPs estimate theoretical computational complexity, GPU inference latency measures practical execution speed, and average inference memory quantifies the runtime memory footprint during inference. Latency was measured using PyTorch CUDA events with a batch size of 1 and an input resolution of 224×224 , averaged over 1,000 forward passes following 100 warm-up iterations on an NVIDIA GeForce RTX 3090. Despite requiring the highest FLOPs, EfficientMixer achieves GPU inference latency (6.85 ms) and memory consumption (19.29 MB) comparable to those of the other lightweight architectures, demonstrating an effective balance between representational capacity and practical inference efficiency.

Table 4. Comparison of model complexity and inference efficiency for lightweight vision models at an input resolution of 224×224 . Parameter count, FLOPs, GPU inference latency, and average inference memory are reported.

Model	Params (M)	FLOPs (G)	GPU Latency (ms)	Avg. Memory (MB)
EfficientMixer	2.58	4.33	6.85	19.29
EfficientNet	2.57	0.54	6.61	18.94
ConvMixer	2.53	0.65	1.75	15.43
MobileNet	2.24	0.31	5.74	17.73
MobileViT	2.81	2.10	6.19	26.89

2.4. Dataset Selection

We evaluate all models on four image classification benchmarks with complementary characteristics: CIFAR-10 [55], Stanford Dogs [56], HAM10000 [57], and MIT Indoor-67 [58]. These datasets were identified in previous work [54] as a representative proxy benchmark whose rankings closely correlate with evaluations on a broader collection of image classification datasets, including ImageNet-1K. Together, they encompass general object recognition, fine-grained classification, medical image analysis, and scene recognition, providing a compact yet diverse benchmark for assessing cross-domain generalization while substantially reducing the computational cost of large-scale benchmarking. Following standard transfer learning practice, all input images, including CIFAR-10, were resized to

224×224 to provide a consistent input resolution across all models. Dataset statistics are summarized in Table 5.

Table 5. Summary of the four datasets used for SCD evaluation.

Dataset	Number of Classes	Train Size	Test Size	Train Images per Class
CIFAR-10	10	50,000	10,000	5000
Stanford Dogs	120	12,000	8580	100
HAM10000	7	7000	3000	~1000
MIT Indoor-67	67	5360	1340	~80

CIFAR-10 [55] consists of low-resolution images from 10 object categories, challenging models to learn discriminative features under limited spatial detail and strong visual overlap between classes. For consistency, we preprocess this dataset to 224×224 px.

Stanford Dogs [56] is a fine-grained recognition benchmark with 120 visually similar dog breeds, testing a model's ability to capture subtle inter-class differences from limited training data.

HAM10000 [57] contains dermatoscopic images from seven diagnostic categories and exhibits severe class imbalance, challenging models to represent rare classes without overfitting dominant ones.

MIT Indoor-67 [58] focuses on indoor scene classification, emphasizing global spatial layout and contextual reasoning amid high intra-class variability.

2.5. Implementation Details

All architectures were trained using identical training configurations, including the optimizer, learning rate schedule, batch size, loss function, and data augmentation strategy, to eliminate model-specific hyperparameter tuning effects. The complete training configuration is summarized in Table 6.

Table 6. Standardized training hyperparameters used across all models.

Hyperparameter	Value
Data augmentations	CutMix, RandomFlip, ColorJitter
Image size	224×224 , normalized per dataset
Model parameter count	~2.5 Million
Maximum learning rate	10^{-3}
Minimum learning rate	10^{-5}
Warm-up	5 epochs of linear increase from 0 to lr_{max}
Learning rate decay	Cosine annealing
Optimizer	Adam (without weight decay)
Loss function	Cross-entropy with label smoothing
Batch size	32
Training Epochs	100

All models were trained for 100 epochs using the Adam optimizer with zero weight decay (weight decay = 0). The Adam momentum coefficients were set to the default values ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). The learning rate was initialized at zero and linearly increased to lr_{max} (1×10^{-3}) during the first 5 warm-up epochs. Subsequently, the learning rate was reduced from lr_{max} to lr_{min} (1×10^{-5}) using cosine annealing for the remaining epochs. No gradient clipping was applied.

All models were trained using PyTorch 2.2.1 with CUDA acceleration on an NVIDIA RTX 3090 GPU. Unless otherwise specified, experiments were conducted using the default random initialization of the training framework. Random seeds were not fixed across runs, and each experiment used a randomly initialized training state. Models were initialized using the default PyTorch initialization schemes and were trained from scratch without ImageNet pretraining.

The input images were resized to 224×224 and normalized using dataset-specific statistics. During training, data augmentation included random horizontal flipping, color jittering, and CutMix. At inference time, classification accuracy was computed using the original test images without stochastic augmentation. The reported accuracy corresponds to the final model after 100 training epochs. The effective batch size was 32 for all experiments, without gradient accumulation.

The complete source code framework, including data preprocessing pipelines, training scripts, and evaluation procedures, is publicly available in the project repository [59] (commit f7f8899).

We report both baseline and Self-Competitive Distillation (SCD) results in terms of average accuracy. Under matched parameter budgets and training conditions, differences in average performance reflect how effectively each architecture converts representational capacity into cross-domain performance, while improvements under SCD indicate more complete utilization of that capacity.

3. Experiments and Results

3.1. Baseline Evaluation of EfficientMixer

We hypothesize that combining these complementary design patterns will provide stronger representations than any individual architectural motif alone, resulting in improved performance across diverse visual domains without substantially increasing model complexity. Because reported state-of-the-art efficiency results often involve substantially larger models with more parameters, as well as different training conditions, hardware, and measurement protocols, directly comparing EfficientMixer with such results may not provide a reasonable basis for comparison. We therefore evaluate EfficientMixer alongside representative lightweight architectures with comparable parameter budgets under a unified training-from-scratch protocol, reporting parameter count, FLOPs, latency, memory consumption, and test accuracy.

Table 7 summarizes the test accuracy of each model on each dataset with and without SCD, together with each model's average accuracy across datasets. The statistical significance of these performance differences is reported in Table 8. Since HAM10000 is class-imbalanced, weighted F1 scores are additionally reported. The weighted F1 scores exhibit the same trend as accuracy; therefore, only accuracy is reported in subsequent ablation studies. EfficientMixer achieves the highest average accuracy among all evaluated models both without SCD (75.78%) and with SCD (77.20%), outperforming the corresponding averages of EfficientNet, ConvMixer, MobileNet, and MobileViT in both settings. Moreover, EfficientMixer benefits substantially from SCD, achieving an average accuracy gain of approximately 1.42 percentage points across the four datasets. The improvement is consistent across all datasets, with gains of +0.27% on CIFAR-10, +1.33% on HAM10000, +4.00% on Stanford Dogs, and +0.10% on Indoor-67. The gains on CIFAR-10, HAM10000, and Stanford Dogs are statistically significant ($p = 0.017$, $p = 0.003$, and $p < 0.001$, respectively), while the small improvement on Indoor-67 is not statistically significant ($p = 0.578$).

Table 7. Test accuracy (%) with and without Self-Competitive Distillation (SCD) over 100 epochs. Results are mean $\pm$ standard deviation over 5 runs. Mean denotes the average across datasets; HAM10000 also reports weighted F1 below accuracy.

Model	SCD	CIFAR-10	HAM10000 (Weighted F1)	Dogs	Indoor-67	Average $\pm$ Std
EfficientMixer	N	94.98 $\pm$ 0.14	80.36 $\pm$ 0.54 (0.78)	65.30 $\pm$ 0.55	62.48 $\pm$ 0.86	75.78 $\pm$ 0.27
	Y	95.25 $\pm$ 0.17	81.69 $\pm$ 0.24 (0.80)	69.30 $\pm$ 0.42	62.58 $\pm$ 0.68	77.20 $\pm$ 0.38
EfficientNet	N	95.35 $\pm$ 0.16	80.84 $\pm$ 0.92 (0.78)	64.33 $\pm$ 0.64	59.51 $\pm$ 0.61	75.01 $\pm$ 0.58
	Y	95.32 $\pm$ 0.17	82.32 $\pm$ 0.34 (0.81)	64.82 $\pm$ 0.59	60.60 $\pm$ 0.74	75.77 $\pm$ 0.46
ConvMixer	N	94.48 $\pm$ 0.08	80.56 $\pm$ 0.84 (0.78)	55.66 $\pm$ 0.71	57.93 $\pm$ 0.49	72.16 $\pm$ 0.53
	Y	94.93 $\pm$ 0.12	82.29 $\pm$ 0.57 (0.81)	63.23 $\pm$ 0.32	61.15 $\pm$ 0.63	75.40 $\pm$ 0.41
MobileNet	N	94.41 $\pm$ 0.17	77.81 $\pm$ 0.86 (0.74)	56.23 $\pm$ 1.38	50.61 $\pm$ 0.68	69.77 $\pm$ 0.77
	Y	94.52 $\pm$ 0.12	79.65 $\pm$ 0.67 (0.76)	56.87 $\pm$ 1.23	51.15 $\pm$ 0.87	70.55 $\pm$ 0.72
MobileViT	N	95.19 $\pm$ 0.32	77.38 $\pm$ 0.41 (0.74)	59.18 $\pm$ 0.36	54.78 $\pm$ 0.52	71.63 $\pm$ 0.39
	Y	94.97 $\pm$ 0.28	74.32 $\pm$ 0.60 (0.72)	63.78 $\pm$ 0.59	55.67 $\pm$ 0.91	72.19 $\pm$ 0.59

Table 8. Statistical significance of SCD gains over non-SCD baselines across models and datasets. Values are p -values from a two-sided Welch's t -test over five runs. Red, blue, and green denote $p \geq 0.05$, $0.01 \leq p < 0.05$, and $p < 0.01$, respectively.

Model	CIFAR-10	HAM10000	Dogs	Indoor-67
EfficientMixer	0.017	0.003	0.000	0.578
EfficientNet	0.780	0.030	0.243	0.036
ConvMixer	0.000	0.007	0.000	0.000
MobileNet	0.277	0.009	0.609	0.307
MobileViT	0.281	0.000	0.000	0.026

Figure 2 visualizes how EfficientMixer achieves strong performance under both training settings, with larger and more consistent SCD gains than the other architectures. The similar gains observed for ConvMixer suggest that the isotropic feature mixing adopted in EfficientMixer may contribute to more effective use of the additional supervisory signal.

3.2. Ablation Study

To quantify the contribution of individual architectural components, we conduct ablation studies using EfficientMixer variants with approximately matched parameter counts. Specifically, we analyze the effects of the stem design, hierarchical staging, attention, and dropout. By modifying one component at a time while maintaining a comparable computational budget, we isolate the impact of each design choice and evaluate whether performance gains arise from individual components or their interactions. We compare a baseline EfficientMixer model with eight architectural variants, as detailed in Table 9.

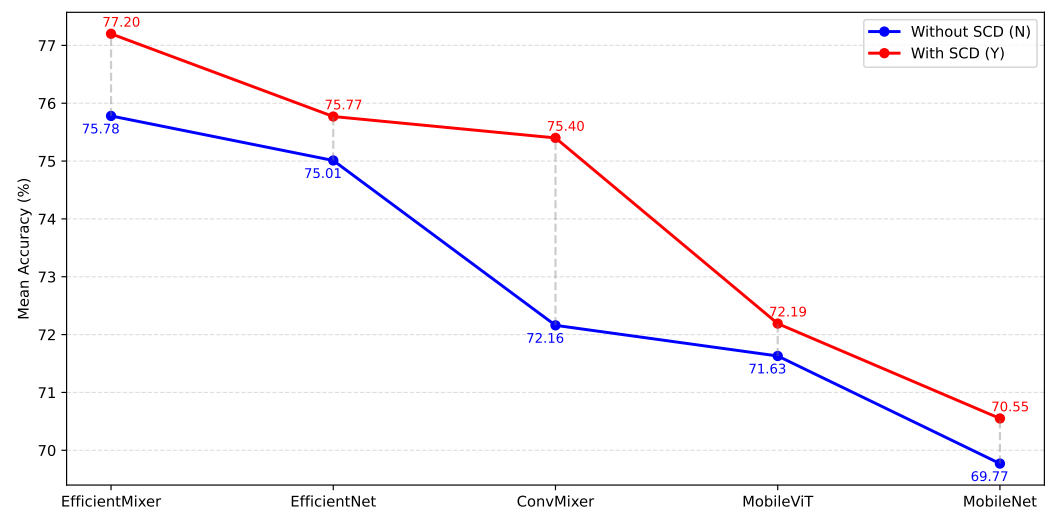


Figure 2. Performance of EfficientMixer and other lightweight architectures with and without Self-Competitive Distillation (SCD). Blue and red curves represent the mean test accuracy across the four benchmark datasets (CIFAR-10, HAM10000, Stanford Dogs, and MIT Indoor-67) for models trained without and with SCD, respectively.

Table 9. EfficientMixer architectural variants evaluated in the ablation study. All stages use patch size $p = 2$ unless otherwise noted. Stage notation $C[L]$ denotes output channels C and number of ConvMixer-style blocks L within each stage.

Variant	Stem	Stages	Attention	Dropout
Baseline	3–32–64	128[3]–256[3]–512[3]–256[3]	NAF-attention	0.20
Stem 1	3–64	128[3]–256[3]–512[3]–256[3]	NAF-attention	0.20
Stem 2	3–32	64[8]–128[8]–256[7]–512[4]	NAF-attention	0.20
Stage 1	3–32–64	128[4]–256[3]–512[3]–256[2]	NAF-attention	0.20
Stage 2	3–32–64	128[4]–192[4]–256[4]–512[4]	NAF-attention	0.20
Linear Attention	3–32–64	128[3]–256[3]–512[3]–256[3]	CBAM	0.20
No Attention	3–32–64	128[3]–256[3]–512[3]–256[3]	None	0.20
Dropout 0.25	3–32–64	128[3]–256[3]–512[3]–256[3]	NAF-attention	0.25
Dropout 0.30	3–32–64	128[3]–256[3]–512[3]–256[3]	NAF-attention	0.30

Each configuration was trained using three independent random seeds, with mean accuracy and standard deviation reported to characterize variability. Due to the limited sample size ($n = 3$), formal significance testing was not performed. We acknowledge this limitation and will increase the number of runs in future work for more rigorous statistical evaluation. The best-performing values are highlighted in bold in Table 10.

Table 10. Ablation study of EfficientMixer. Results are mean test accuracy (%) $\pm$ standard deviation over three runs with different random seeds. Best-performing results for each dataset and the overall average are shown in bold.

Variant	CIFAR-10	HAM10000	Stanford Dogs	Indoor-67	Mean $\pm$ Std
Baseline	94.98 $\pm$ 0.14	80.36 $\pm$ 0.54	65.30 $\pm$ 0.55	62.48 $\pm$ 0.86	75.78 $\pm$ 0.27
Stem 1	94.93 $\pm$ 0.07	79.48 $\pm$ 0.53	65.12 $\pm$ 0.41	60.38 $\pm$ 0.88	74.98 $\pm$ 0.28
Stem 2	95.47 $\pm$ 0.13	78.38 $\pm$ 0.87	62.14 $\pm$ 0.51	60.35 $\pm$ 0.76	74.09 $\pm$ 0.32
Stage 1	94.91 $\pm$ 0.07	79.95 $\pm$ 0.08	64.41 $\pm$ 0.49	62.54 $\pm$ 1.32	75.45 $\pm$ 0.35
Stage 2	95.50 $\pm$ 0.11	80.01 $\pm$ 0.44	63.77 $\pm$ 0.11	61.63 $\pm$ 1.29	75.23 $\pm$ 0.34

Table 10. Cont.

Variant	CIFAR-10	HAM10000	Stanford Dogs	Indoor-67	Mean $\pm$ Std
Linear Attention	95.04 $\pm$ 0.11	79.22 $\pm$ 0.32	64.52 $\pm$ 0.32	62.73 $\pm$ 0.34	75.38 $\pm$ 0.14
No Attention	94.66 $\pm$ 0.19	78.01 $\pm$ 0.73	62.05 $\pm$ 0.54	62.93 $\pm$ 0.83	74.41 $\pm$ 0.31
Dropout 0.25	94.85 $\pm$ 0.07	79.82 $\pm$ 0.57	64.40 $\pm$ 0.81	62.84 $\pm$ 0.74	75.48 $\pm$ 0.28
Dropout 0.30	94.96 $\pm$ 0.07	80.13 $\pm$ 0.49	64.74 $\pm$ 0.36	62.79 $\pm$ 0.43	75.66 $\pm$ 0.17

Figure 3 shows the per-component deltas from the ablation study, reported as deviations in mean test accuracy relative to the baseline EfficientMixer configuration (75.78%). This enables a direct comparison of how each architectural modification influences performance. The ablation variants are grouped into four categories—stem design, stage depth, attention mechanisms, and dropout configurations—facilitating a structured analysis of each architectural component.

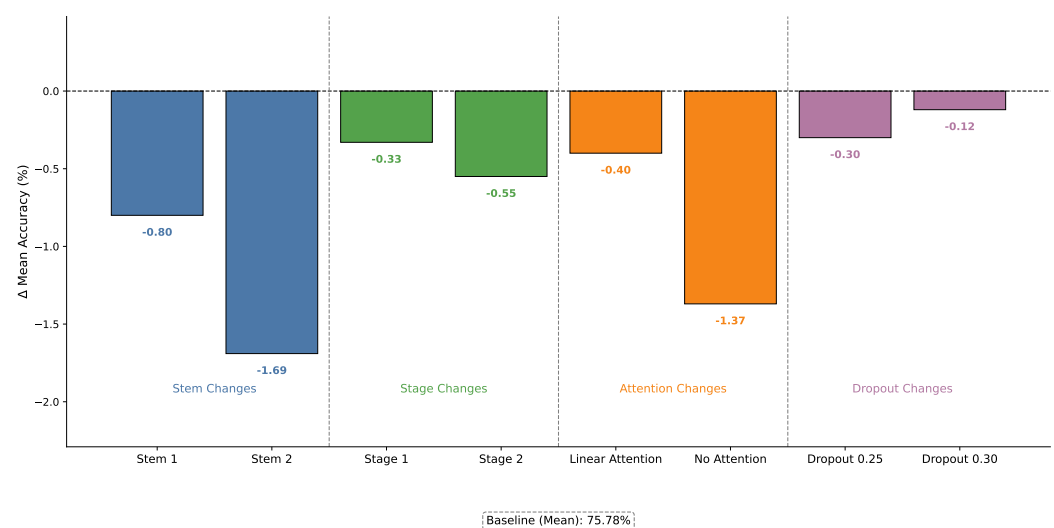


Figure 3. Ablation analysis of EfficientMixer components. The plot reports the impact of individual architectural modifications on mean test accuracy (%), measured as deviation from the baseline model (75.78%). Variants are grouped into stem changes, stage depth modifications, attention mechanism variations, and dropout settings, highlighting the relative sensitivity of each design component.

Among the eight variants, only Stem 2 and No Attention consistently underperform the EfficientNet baseline, indicating that EfficientMixer’s hierarchical staging and isotropic feature mixing form a strong architectural backbone. Overall, performance is most sensitive to changes in the stem and attention modules, highlighting the importance of early feature embedding and global context modeling. In contrast, variations in stage depth and dropout induce more moderate fluctuations around the baseline, suggesting that EfficientMixer is relatively robust to these design and regularization choices. This structured comparison clarifies the relative influence of each component on final model performance.

The convolutional stem was highly influential: the hierarchical 3–32–64 stem achieved the highest xScore (75.78), outperforming both 3–64 and 3–32 variants. Although 3–32 performed best on CIFAR-10, its weaker cross-domain performance reduced its overall xScore. These results suggest that progressive channel expansion improves low-level representations and cross-dataset generalization.

Alternative stage width and depth configurations produce only modest performance changes, with Stage 1 and Stage 2 reducing accuracy by 0.33 and 0.55 points, respectively. Although Stage 2 achieves the highest CIFAR-10 accuracy (95.50%), its cross-dataset performance is lower than the baseline. These results suggest that Efficient-

Mixer performance depends more on its overall architectural design than on specific width or depth configurations.

The attention module consistently improves performance: replacing NAF attention with linear attention reduces xScore from 75.78 to 75.38, while removing attention lowers it further to 74.41. The largest drop occurs on Stanford Dogs, from 65.30 to 62.05, highlighting the importance of contextual refinement for fine-grained recognition. NAF attention therefore provides a useful performance benefit with modest computational overhead.

The impact of dropout is small. Dropout rates of 0.25 and 0.30 achieve 75.48 and 75.66, respectively, close to the 75.78 baseline. Differences across datasets are generally within one standard deviation, suggesting that EfficientMixer already benefits from strong implicit regularization through its architecture, augmentation, and label smoothing. Thus, additional dropout provides only marginal gains and little effect on cross-domain robustness.

Overall, the ablation study shows that the hierarchical convolutional stem and lightweight attention provide additional contributions to EfficientMixer's cross-domain performance, beyond its core effective complementary architecture, while variations in stage width/depth and dropout have only marginal effects. This suggests that feature extraction and contextual refinement matter more than capacity scaling alone.

4. Conclusions

This work investigated how complementary architectural principles can improve representation learning in lightweight vision models under strict parameter constraints. We proposed EfficientMixer by combining hierarchical feature scaling, isotropic spatial-channel mixing, and lightweight attention within a unified architecture.

Experiments across multiple visual domains show that EfficientMixer achieves the highest average accuracy among the evaluated lightweight architectures, including EfficientNet, ConvMixer, MobileNet and MobileViT, under matched parameter budgets and standardized training conditions, while maintaining competitive parameter and inference costs. EfficientMixer also benefits substantially from Self-Competitive Distillation (SCD), with consistent improvements across the evaluated datasets and statistically significant gains on three of the four datasets.

Ablation studies further indicate that the hierarchical convolutional stem and lightweight attention contribute to improved cross-domain performance, whereas changes in stage configuration and dropout have comparatively smaller effects. These findings suggest that effective lightweight model design can benefit from systematically combining complementary feature hierarchy, spatial-channel interaction, and attention mechanisms rather than relying primarily on increased depth, width, or isolated architectural innovations.

A limitation of this study is that CIFAR-10 was resized from its native 32×32 resolution to 224×224 for standardized evaluation, which may alter its spatial characteristics. In addition, the strict parameter constraint limits direct comparison with state-of-the-art models that typically employ substantially larger model sizes and different training settings. Future work will investigate resolution-specific evaluation and scale EfficientMixer to larger parameter budgets, enabling a more direct comparison with state-of-the-art architectures under appropriately matched training and evaluation protocols. We will also further explore how the proposed complementary architectural principles can be composed and scaled to improve the accuracy-efficiency trade-off across a broader range of model sizes and visual domains.

Author Contributions: Conceptualization, W.Z.; methodology, W.Z.; software, W.Z.; validation, W.Z., B.L. and H.L.; formal analysis, W.Z. and A.A.D.; investigation, W.Z.; resources, W.Z. and A.A.D.; data curation, W.Z. and A.A.D; writing—original draft preparation, W.Z.; writing—review and editing,

W.Z. and H.L.; supervision, B.L., Y.Y. and H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article; the raw data and source code that generates all the tables and graphs in this paper are listed at https://github.com/javawormer/beyond_imagenet (accessed on 30 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Menghani, G. Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. *ACM Comput. Surv.* **2023**, *55*, 259. [CrossRef]
2. Liu, H.I.; Galindo, M.; Xie, H.; Wong, L.K.; Shuai, H.H.; Li, Y.H.; Cheng, W.H. Lightweight Deep Learning for Resource-Constrained Environments: A Survey. *ACM Comput. Surv.* **2024**, *56*, 267. [CrossRef]
3. Wang, X.; Tang, Z.; Guo, J.; Meng, T.; Wang, C.; Wang, T.; Jia, W. Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. *ACM Comput. Surv.* **2025**, *57*, 228. [CrossRef]
4. Lin, J.; Zhu, L.; Chen, W.M.; Wang, W.C.; Han, S. Tiny Machine Learning: Progress and Futures. *IEEE Circuits Syst. Mag.* **2023**, *23*, 8–34. [CrossRef]
5. Zhang, W.; Li, B.; Liu, H.; Ding, P.L.K.; Dalyanci, A.A. Accelerating Realization of Effective Capacity in Lightweight Vision Models via Self-Competitive Distillation. *Algorithms* **2026**, *19*, 262. [CrossRef]
6. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2017.
7. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; pp. 4510–4520.
8. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, R. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2019; pp. 1314–1324.
9. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; pp. 6848–6856.
10. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2018; pp. 116–131.
11. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 1580–1589.
12. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetV2: Enhance Cheap Operation with Long-Range Attention. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 9969–9982.
13. Vasu, P.K.A.; Gabriel, J.; Zhu, J.; Tuzel, O.; Ranjan, A. MobileOne: An Improved One Millisecond Mobile Backbone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 7907–7917.
14. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the International Conference on Machine Learning*; PMLR: New York, NY, USA, 2019; pp. 6105–6114.
15. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. In *Proceedings of the International Conference on Machine Learning*; PMLR: New York, NY, USA, 2021; pp. 10096–10106.
16. Wu, B.; Dai, X.; Zhang, P.; Wang, Y.; Sun, F.; Wu, Y.; Tian, Y.; Vajda, P.; Jia, Y.; Keutzer, K. FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 10734–10742.
17. Wan, A.; Dai, X.; Zhang, P.; He, Z.; Tian, Y.; Xie, S. FBNetV2: Differentiable Neural Architecture Search for Spatial and Channel Dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 12965–12974.
18. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. ResNeSt: Split-Attention Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*; Curran Associates Inc.: Red Hook, NY, USA, 2020.

19. Mehta, S.; Rastegari, M. MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 25–29 April 2022.
20. Chen, Y.; Dai, X.; Chen, D.; Liu, M.; Dong, X.; Yuan, L.; Liu, Z. MobileFormer: Bridging MobileNet and Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; pp. 5270–5279.
21. Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; Yuan, Y. EfficientViT: Memory Efficient Vision Transformer With Cascaded Group Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2023; pp. 14420–14430. [[CrossRef](#)]
22. Shaker, A.; Maaz, M.; Rasheed, H.; Khan, S.; Yang, M.H.; Khan, F.S. SwiftFormer: Efficient Additive Attention for Transformer-based Real-time Mobile Vision Applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2023; pp. 17379–17390. [[CrossRef](#)]
23. Yu, W.; Luo, M.; Zhou, P.; Si, C.; Zhou, Y.; Wang, X.; Feng, J.; Yan, S. MetaFormer is Actually What You Need for Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; pp. 10819–10829.
24. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 568–578.
25. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 10012–10022.
26. Li, K.; Wang, Y.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unified Transformer for Efficient Spatiotemporal Representation Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 25–29 April 2022.
27. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 2820–2828.
28. Cai, H.; Gan, C.; Wang, T.; Zhang, Z.; Han, S. Once-for-All: Train One Network and Specialize it for Efficient Deployment. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
29. Guo, Z.; Zhang, X.; Mu, H.; Heng, W.; Liu, Z.; Wei, Y.; Sun, J. Single Path One-Shot Neural Architecture Search with Uniform Sampling. In *Proceedings of the Computer Vision—ECCV 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12361, pp. 544–560.
30. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
31. Wang, S.; Li, B.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-Attention with Linear Complexity. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
32. Choromanski, K.; Likhoshesterov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J.; Mohiuddin, A.; Kaiser, L.; et al. Rethinking Attention with Performers. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 3–7 May 2021.
33. Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; Li, H. Efficient Attention: Attention with Linear Complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 3531–3539.
34. Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; Hsieh, C.J. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 13937–13949.
35. Liang, Y.; Chongjian, G.; Tong, Z.; Song, Y.; Wang, J.; Xie, P. EViT: Expediting Vision Transformers via Token Reorganizations. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 3–7 May 2021.
36. Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Ren, J. EfficientFormer: Vision Transformers at MobileNet Speed. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 12934–12949.
37. Al-Sameai, H.; Saleh, R.A.A.; de Moura, J.; Akay, R. Dual-encoder Multi-Scale Refinement network for robust crack segmentation across diverse domains. *Autom. Constr.* **2026**, *188*, 107004. [[CrossRef](#)]
38. Ibrahim, H.; Salem, A.; Kang, H.S. Pixel Shuffling Is All You Need: Spatially Aware ConvMixer for Dense Prediction Tasks. *Pattern Recognit.* **2025**, *158*, 111068. [[CrossRef](#)]
39. Saleh, R.A.A.; Ertunç, H.M. Attention-based deep learning for tire defect detection: Fusing local and global features in an industrial case study. *Expert Syst. Appl.* **2025**, *269*, 126473. [[CrossRef](#)]
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018.

41. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 11531–11539.
42. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2018; pp. 3–19.
43. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. BAM: Bottleneck Attention Module. In *Proceedings of the British Machine Vision Conference (BMVC)*, Newcastle, UK, 3–6 September 2018.
44. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to Attend: Convolutional Triplet Attention Module. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 3138–3147.
45. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2021; pp. 13708–13717.
46. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*; IEEE: New York, NY, USA, 2019.
47. Yang, Z.; Zhu, L.; Wu, Y.; Yang, Y. Gated Channel Transformation for Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 11791–11800.
48. Zhang, Q.; Yang, Y. SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*; IEEE: New York, NY, USA, 2021; pp. 2235–2239.
49. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017.
50. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*, Virtual Event, 3–7 May 2021.
51. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning*; PMLR: New York, NY, USA, 2021; pp. 11863–11874.
52. Trockman, A.; Kolter, J.Z. Patches Are All You Need? In *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual Event, 25–29 April 2022.
53. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. MLP-Mixer: An All-MLP Architecture for Vision. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual Event, 3–7 May 2021.
54. Zhang, W.; Ding, P.L.K.; Li, B.; Liu, H. xScore: A Simple Metric for Cross-Domain Robustness in Lightweight Vision Models. *Algorithms* **2026**, *19*, 14. [[CrossRef](#)]
55. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
56. Khosla, A.; Jayadevaprakash, N.; Yao, B.; Fei-Fei, L. Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs. In *Proceedings of the CVPR Workshop on Fine-Grained Visual Categorization*, Colorado Springs, CO, USA, 20–25 June 2011.
57. Tschandl, P.; Rosendahl, C.; Kittler, H. The HAM10000 Dataset: A Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions. *Sci. Data* **2018**, *5*, 180161. [[CrossRef](#)] [[PubMed](#)]
58. Quattoni, A.; Torralba, A. Recognizing Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2009; pp. 413–420.
59. Zhang, W. xScore: Evaluation Framework with Training Wrapper and Dataset Loader. GitHub Repository. 2025. Available online: https://github.com/javawormer/beyond_imagenet (accessed on 31 October 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.