Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059966

ARTICLE

A Latency-Efficient Integration of Channel Attention for ConvNets

Woongkyu Park[1], **Yeongyu Choi**[2], **Mahammad Shareef Mekala**[3], **Gyu Sang Choi**[1], **Kook-Yeol Yoo**[1] **and Ho-youl Jung**[1,*]

[1]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Republic of Korea
[2]RLRC for Autonomous Vehicle Parts and Materials Innovation, Yeungnam University, Gyeongsan, 38541, Republic of Korea
[3]School of Computing, Robert Gordon University, Aberdeen, AB10 7QB, UK

*Corresponding Author: Ho-youl Jung. Email: hoyoul@ynu.ac.kr

Received: 21 October 2024; Accepted: 15 January 2025; Published: 06 March 2025

ABSTRACT: Designing fast and accurate neural networks is becoming essential in various vision tasks. Recently, the use of attention mechanisms has increased, aimed at enhancing the vision task performance by selectively focusing on relevant parts of the input. In this paper, we concentrate on squeeze-and-excitation (SE)-based channel attention, considering the trade-off between latency and accuracy. We propose a variation of the SE module, called squeeze-and-excitation with layer normalization (SELN), in which layer normalization (LN) replaces the sigmoid activation function. This approach reduces the vanishing gradient problem while enhancing feature diversity and discriminability of channel attention. In addition, we propose a latency-efficient model named SELNeXt, where the LN typically used in the ConvNext block is replaced by SELN to minimize additional latency-impacting operations. Through classification simulations on ImageNet-1k, we show that the top-1 accuracy of the proposed SELNeXt outperforms other ConvNeXt-based models in terms of latency efficiency. SELNeXt also achieves better object detection and instance segmentation performance on COCO than Swin Transformer and ConvNeXt for small-sized models. Our results indicate that LN could be a considerable candidate for replacing the activation function in attention mechanisms. In addition, SELNeXt achieves a better accuracy-latency trade-off, making it favorable for real-time applications and edge computing. The code is available at https://github.com/oto-q/SELNeXt (accessed on 06 December 2024).

KEYWORDS: Attention mechanism; convolutional neural networks; image classification; object detection; semantic segmentation

1 Introduction

Over the past decade, significant advancements in image classification [1], object detection [2], and semantic segmentation [3] have been achieved by vision neural networks, particularly convolutional neural networks (ConvNets). Vision networks require low latency and high precision, which are often necessary for real-time applications like advanced driver assistance systems, robotics, surveillance, and security. There are several key innovations to enhance the vision network.

For example, batch normalization (BN) [4] has been incorporated into these networks to stabilizes and accelerates training by normalizing neuron inputs using the mean and variance computed from a mini-batch. This process reduces internal covariate shifts and improves gradient flow, leading to more efficient and robust training.

Recently, the usage of attention mechanisms has increased and shown enhanced performance in various vision tasks. There are several types of attention approaches, including channel and spatial attention. Such as the convolutional block attention module (CBAM) [5] utilizes channel and spatial attention simultaneously. Attention mechanisms allow models to selectively focus on relevant parts of the input, improving the model's performance.

Transformer [6] is a specific type of architecture originally designed for natural language processing. Vision transformer [7] successfully enables the use of the transformer architecture for vision tasks by utilizing a massive amount of images. Unlike traditional ConvNets, which use convolutions for feature extraction, vision transformers utilize multi-head self-attention (MHSA) mechanisms [6]. They break down the image into smaller patches, treating each patch as a sequence of tokens. MHSA is then applied to capture dependencies between these tokens. Vision transformers have shown promising results in various vision tasks, often achieving competitive performance compared to ConvNet-based models.

Modern ConvNets incorporate the benefits of vision transformers into the ConvNet architecture. Large-size kernel convolution is integrated into ConvNets to cover wide region information, similar to MHSA in transformers. Additionally, layer normalization (LN) [8] has become widely adopted in modern ConvNets.

For instance, in ConvNeXt [9], a large 7 × 7 depth-wise convolutional kernel with LN is employed, enabling the network to aggregate wider information. The depth-wise convolution reduces computational complexity while maintaining the ability to learn rich features. RepLKNet [10] uses even larger depth-wise kernels, up to 31 × 31, with BN, facilitating the learning of long-range relationships within the input data. GFNet [11] captures global spatial relationships in the Fourier domain and uses LN. MetaFormer [12] employs average pooling to aggregate spatial information, using modified LN to ensure proper normalization and stable training.

Many approaches have been developed to reduce the computational complexity, but they do not necessarily guarantee low latency. For example, depth-wise separable convolution [13] and group convolution [14] significantly reduce computational complexity, but the actual speedup is limited due to the substantial time required for memory access. HorNet [15] captures high-order spatial relationships with recursive gated convolutions, enhancing vision task performance with lower or similar complexity than ConvNeXt but resulting in longer latency.

ConvNeXt [9] does not consider the attention mechanism, while ConvNeXt V2 [16] proposes global response normalization (GRN), which reduces feature collapse. The authors measure the feature collapse by average feature cosine distance. As the average feature cosine distance approaches 0.5, the features become more diverse across the channels. GRN enhances classification performance on ImageNet-1k [17] compared to ConvNeXt without increasing computational complexity and the number of parameters. However, GRN necessitates an additional residual connection and LN in a block. Consequently, GRN increases latency, even though it does not increase the computational complexity compared to ConvNeXt. It impacts negatively the latency-accuracy trade-off. Squeeze-and-excitation(SE), CBAM and GRN are compared in terms of classification accuracy in [16], where it is reported that SE and CBAM also increase the contrast of individual channels similar to GRN. CBAM can be a considerable candidate for the attention module, but the gains in accuracy are relatively modest in latency.

Clearly, it is necessary to consider a latency-efficient attention mechanism because most vision networks are required to perform within a limited time. In this paper, we concentrate on SE-based channel attention, considering the trade-off between latency and accuracy. A latency-efficient network is developed with the following two approaches.

First, a way to enhance channel attention performance is explored. We propose a variation of the SE module, called squeeze-and-excitation with layer normalization (SELN), in which LN replaces the sigmoid activation function. The sigmoid function in the existing SE maps the scale values of each channel to a range of 0 to 1, causing the scale values to become saturated at the extremes. This results in a reduction in the discriminability of channel importance. If LN replaces the sigmoid, the scale value of each channel is normalized using the mean and standard deviation calculated across all channels and then adjusted using learnable rescale and bias parameters, which could enhance the discriminability of channel importance.

Second, redundant operations are explored to reduce latency when SELN is applied. We propose a latency-efficient ConvNeXt-based model named SELNeXt, where SELN replaces the conventional LN used in the ConvNeXt block. LN is time-consuming due to its computation of normalization across all features. By replacing LN with SELN, the computational resources originally used for LN are allocated to perform channel attention and scale normalization. Integrating channel attention into ConvNeXt is helpful for improving performance.

Our empirical demonstrations show that SELNeXt is more efficient regarding the accuracy-latency trade-off. We analyze the inter-channel feature independence according to the usage of SE by measuring the average feature cosine distance at each inverted residual block in pre-trained MobileNetV3 [18]. It is observed that the SE-integrated block generates feature maps that are more independent across each channel compared to the non-SE-integrated block.

The main contributions of this paper are as follows:

1. Analyze the inter-channel feature independence according to the usage of SE.
2. Introduce SELN, which further enhances vision task performance and feature independence compared to traditional SE.
3. Propose the SELNeXt model architecture, which is latency-efficient in various vision tasks compared to ConvNeXt.

2 Related Work

2.1 Squeeze-and-Excitation

SE [19] significantly enhances the representational ability of neural networks. It adaptively adjusts the scale of each channel, determining what to pay attention to [20]. For a given input activation map X with dimensions $\mathbb{R}^{C\times W\times H}$, the formulation of SE is [19].

$$\begin{aligned} G(X) &= GAP(X), \\ S &= W_2 \cdot \delta(W_1 \cdot G(X)), \\ SE(X) &= \sigma(S) \odot X, \end{aligned} \tag{1}$$

as shwon in Fig. 1. Here, $G(X) \in \mathbb{R}^{C}$ is the function of spatial aggregation. Global Average Pooling (GAP) is utilized for SE, which reduces the dimension from $\mathbb{R}^{C\times W\times H}$ to $\mathbb{R}^{C\times 1\times 1}$. S is the scale vector that is calculated from successive fully connected layers. weights of the two fully connected layers are represented by W_1 and W_2, with dimensions $\mathbb{R}^{C\times C'}$ and $\mathbb{R}^{C'\times C}$, respectively. C' is derived as $C' = C/r$, where r denotes the reduction ratio. The reduction ratio is typically set to 16 for SENet [19] 4 for MobileNetV3 [18] architectures. The symbols σ and δ represent the sigmoid and ReLU activation functions, respectively. Note that the sigmoid or hard-sigmoid function is typically applied for S to adjust the scale to a range of 0 to 1. $\odot$ represents element-wise multiplication. The vector $\sigma(S) \in \mathbb{R}^{C}$ is repeated $W \times H$ times to form an extended tensor of dimensions $\mathbb{R}^{C\times W\times H}$ and then multiplied element-wise to adaptively re-calibrate each channel. SE is suitable for developing latency-efficient attention, minimizing operations that impact latency.

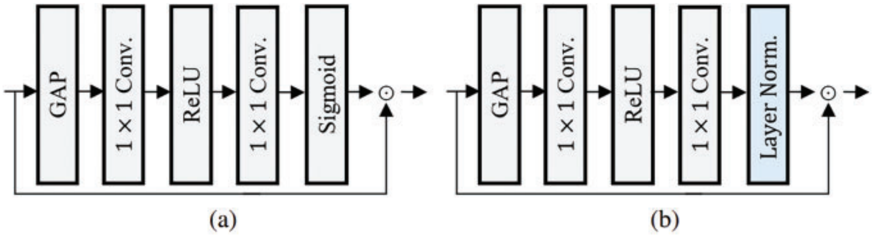


Figure 1: (a) Squeeze-and-excitation (SE) and (b) the proposed squeeze-and-excitation with layer normalization (SELN) module, where layer normalization (LN) replaces the sigmoid function

2.2 Various Attention Mechanisms

Channel Attention. Various other channel attention mechanisms have been developed. Unlike SE [19], a discrete cosine transform is utilized for spatial aggregation in FCA [21]. ECA [22] utilizes a 1D convolution instead of two fully connected layers for a lightweight attention module. They extract channel-wise statistics with a fully connected or convolution layer. Most channel attention mechanisms, including SE, FCA, and ECA, utilize the sigmoid activation function. GCT [23] and GRN [16] collect spatial global information using the l_2 norm and perform channel-wise normalization without any network processing such as fully connected layer and convolution. Taking X as the input activation map with dimensions of $\mathbb{R}^{C\times W\times H}$, GCT is formulated as [23]

$$\begin{aligned} G(X) &= \alpha \cdot \|\mathbf{X}\|_2, \\ GCT(X) &= (tanh(\gamma \cdot LRN(G(X)) + \beta) + 1) \odot X, \end{aligned} \tag{2}$$

where $\|\cdot\|_2$ is l_2-norm and α a learnable parameter. $LRN(\cdot)$ indicates local response normalization [24].

GRN is formulated as [16]

$$\begin{aligned} G(X) &= \|\mathbf{X}\|_2, \\ GRN(X) &= \gamma \cdot DN(G(X)) \odot X + \beta + X, \end{aligned} \tag{3}$$

where $DN(\cdot)$ indicates divisive normalization. GRN necessitates an additional residual connection.

Other Attention. MHSA [6] has been often used. ViT [7] demonstrates performance comparable to ConvNets in computer vision tasks using MHSA. Swin [25] employs a hierarchical approach, limiting MHSA to non-overlapping local windows and shifting them across layers to capture local and global features. However, as stated in [9] and [15], MHSA-based networks such as ViT and Swin are less latency-efficient in vision tasks compared to ConvNeXt.

Recently, hybrid networks that combine convolution and MHSA have been actively researched, particularly for small networks designed for mobile devices. MobileViT [26] aims to integrate the advantages of both ViT and ConvNet through the use of global attention blocks. EdgeNeXt [27] proposes a split depth-wise transpose attention encoder that implicitly increases the receptive field and encodes multi-scale features. MobileFormer [28] parallelizes a two-way bridge between MobileNet and Transformer. FastViT [29] applies attention only in the last stage of the network. To perform MHSA efficiently, EfficientFormer V2 [30] downsamples the query, key, and value before performing dot product, while NextViT [31] downsamples only the key and value, using a module called spatial reduction attention. In addition, more efficient MHSA is explored utilizing multi-query attention, where only a single header is generated for the key and value [32]. Hybrid networks such as MobileViT-XXS and EdgeNeXt-XXS show lower accuracy compared to the convolution-based FasterNet-T0 with longer latency on a GPU [33].

2.3 Analysis of Feature Collapse

In ConvNeXt V2 [16], they observe a "feature collapse" phenomenon in which numerous feature maps become dead or saturated. It leads to activation maps being redundant across channels. Feature collapse is analyzed by calculating the average cosine distance across the feature maps. It is shown that the attention mechanism alleviates feature collapse, emphasizing important features and diminishing less useful ones.

For a given activation map $X \in \mathbb{R}^{C\times W\times H}$, average feature cosine distance $\overline{dist}(X)$ is calculated using cosine distance of the pair-wise i-th and j-th channel feature maps, X_i and $X_j \in \mathbb{R}^{W\times H}$ as [16]

$$\overline{dist}(X) = \frac{1}{C^2}\sum_{i=0}^{C-1}\sum_{j=0}^{C-1}\frac{1 - cos(X_i, X_j)}{2}, \tag{4}$$

where

$$cos(X_i, X_j) = \frac{X_i \cdot X_j}{\|X_i\|\|X_j\|}. \tag{5}$$

To analyze the feature collapse according to the usage of SE and SELN, we evaluate the average feature cosine distance in three cases such as with SE, without SE, and with the proposed SELN, using MobileNetV3 as a backbone. The average feature cosine distance is measured at each inverted residual block in MobileNetV3 [18], testing on 50,000 validation images of the ImageNet-1k dataset [17]. Fig. 2 shows the average feature cosine distance along the number of network blocks. In the blocks 2 to 4 and 9 to 13, indicated by a red dashed box, SE or SELN is applied. SE and SELN integrated blocks tend to have average distances closer to 0.5. In particular, it shows the average distance closest to 0.5 in the SELN integrated block. In the case of SE, the sigmoid-activated scale vector $\sigma(S)$ always has non-negative elements. Since both the input$X$ and the output $SE(X)$ have the same direction, their cosine distance remains unchanged. Nevertheless, blocks integrated with SE tend to have an average cosine distance closer to 0.5 compared to those without SE. This indicates that the overall network is trained to have more diverse features by integrating SE. Note that SELN will be proposed and discussed in detail in the following section.

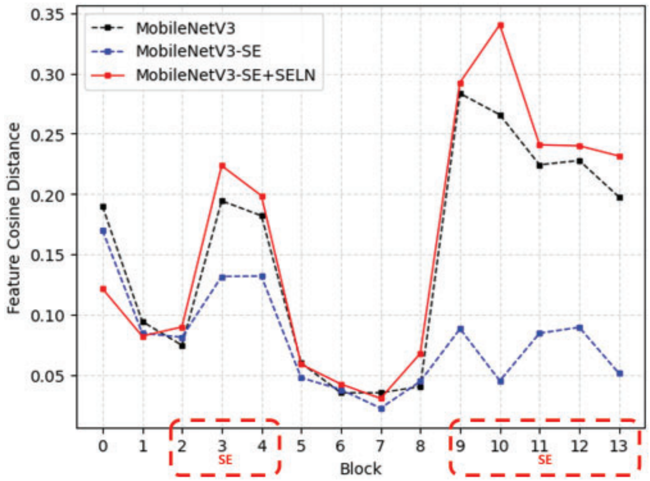


Figure 2: Average feature cosine distance after the expansion layer along the number of network blocks, where three cases such as with SE (denoted as MobileNetV3), without SE (MobileNetV3-SE), and with the proposed SELN (MobileNetV3-SE+SELN), are compared

3 Proposed Work

3.1 Squeeze-and-Excitation with Layer Normalization

Most channel attention mechanisms apply the sigmoid or hard-sigmoid activation function to the scale vector, as mentioned in Section 2. The sigmoid function in existing SE models plays a role in normalizing scale values to a specific range, as mentioned in [20,34]. Considering that both the sigmoid and LN functions are used for normalization, we are exploring the replacement of the sigmoid function with LN.

The sigmoid function can lead to the vanishing gradient problem, as discussed in [35]. The sigmoid function maps the scale vector coming from the fully connected layers to a fixed value between 0 and 1, regardless of the values of other channels. The sigmoid function approaches 1 asymptotically when the scale value becomes greater than a certain threshold and approaches 0 asymptotically when the scale value is smaller than a certain threshold. This saturation causes a reduction in the discriminability of the scale vector, as described in Fig. 3a, where the sigmoid function causes the scale values to flatten at 0 or 1.

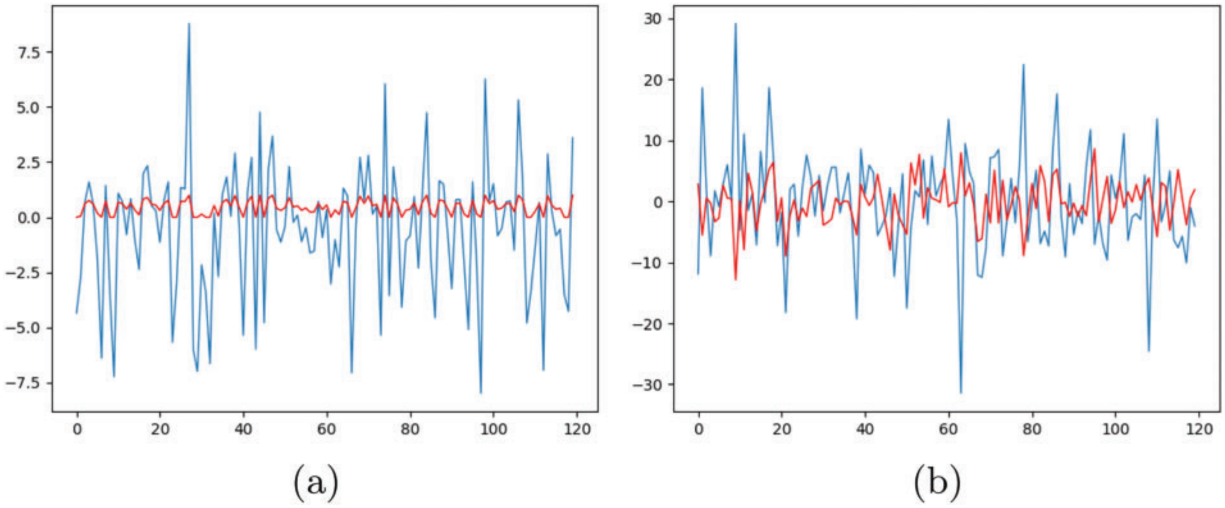


Figure 3: Examples of channel scales before (blue) and after (red) applying (a) the sigmoid function and (b) layer normalization (LN)

If the sigmoid in SE is replaced by LN, which involves channel normalization and adjustment steps, the discriminability of the scale vector is enhanced. This change alleviates the vanishing gradient problem by controlling gradient flow. In the normalization step, the scale value of each channel is adjusted relative to the distribution of other channels. The adjustment step learns the inherent importance of the channel through trainable parameters, enabling a better reflection of the channel's significance. It is observed that LN enriches the values of the scale vector compared to the sigmoid function, as shown in Fig. 3b. This is the reason why the proposed SELN integrates LN instead of the activation function for the scale vector, as shown in Fig. 1b. SELN is formulated as

$$
\begin{aligned}
G(X) &= GAP(X),\\
S &= W_2 \cdot \delta(W_1 \cdot G(X)),\\
SELN(X) &= LN(S) \odot X,
\end{aligned}
\tag{6}
$$

where the scale vector $S = (s_1, s_2, \ldots, s_C)$ obtained from two fully connected layers is normalized by $LN(\cdot)$ as given by [8].

$$\mu = \frac{1}{C}\sum_{i=1}^{C} s_i, \quad \sigma^2 = \frac{1}{C}\sum_{i=1}^{C}(s_i - \mu)^2, \\ \hat{s}_i = \frac{s_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \quad s_i' = \gamma \cdot \hat{s}_i + \beta, \tag{7}$$

where μ is the mean of S, σ^2 is the variance of S, ε is a small constant, γ and β are learnable parameters, and $s_i' \in LN(S)$ is the output of the layer normalization for i-th channel. $LN(\cdot)$ differs from typical LN because it does not use a feature map of dimension $\mathbb{R}^{C\times W\times H}$ as input, but rather a scale vector S of dimension $\mathbb{R}^{C}$.

As the layer normalization vector $LN(S)$ consists of signed elements and multiplies to the activation map$X$, $SELN(X)$ also has signed values, which alter the direction of feature maps. It influences the average feature cosine distance value to make feature maps more diverse across the channels. On the other hand, $SE(X)$ only has positive values. This leads to an average cosine distance being closest to 0.5 in the SELN integrated network, MobileNetV3-SE+SELN, as shown in Fig. 2.

3.2 SELNeXt

Using SELN, we propose a new family of neural networks, SELNeXt, that is more efficient in terms of the accuracy latency trade-off compared to ConvNeXt [9]. The basic architecture for SELNeXt consists of four hierarchical stages, like in ConvNeXt. The stem block consists of 4 × 4 convolution with stride 4 and LN. Each downsampling block is designed LN and 2 × 2 convolution with stride 2 as in ConvNeXt. The main difference between SELNeXt and ConvNeXt is that LN is removed after depth-wise convolution and replaced with SELN after the GELU activation function, as described in Fig. 4. As LN is contained in SELN, it is the redundant operation if the LN after the depth-wise convolution remains when inserting SELN. In addition, LN in the SELN is performed on one-dimensional scale vectors, so it is advantageous in terms of fast processing compared to general LN. Three SELNeXt variants are designed with the following configurations:

- SELNeXt-T: **C** = (96, 192, 384, 768), **B** = (3, 3, 9, 3).
- SELNeXt-S: **C** = (96, 192, 384, 768), **B** = (3, 3, 27, 3).
- SELNeXt-B: **C** = (128, 256, 512, 1024), **B** = (3, 3, 27, 3).

C represents the number of input and output channels, while **B** indicates the number of blocks within each stage.

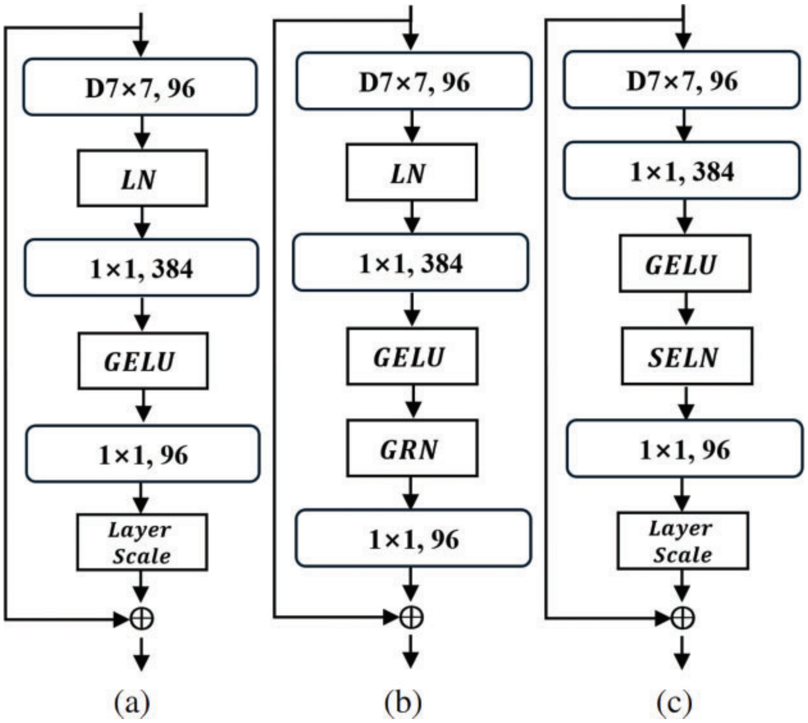


Figure 4: Block designs for (a) ConvNeXt, (b) ConvNeXt V2, and (c) the proposed SELNeXt

4 Experiment

4.1 Image Classification on ImageNet-1k Dataset

Setup. ImageNet-1k [17] containing around 1.3 million images in the training set and 50 thousand in the validation set is used for image classification performance evaluation of SELNeXt. The performance is compared with Swin [25], ConvNeXt [9], HorNet [9], and ConvNeXt V2 [16], which have similar architecture and computational complexity.We use the same training hyperparameters as those in ConvNeXt [9], which are listed in Table 1.

Table 1: Hyperparameter settings for image classification

	MobileNetV3	**SELNeXt-T/S/B**	**Transfer learning**
Dataset	ImageNet-1k	ImageNet-1k	etc.
Weight init	trunc. normal (0.2)	trunc. normal (0.2)	Pre-trained
Optimizer	RMSprop	AdamW	AdamW
Base learning rate	0.064	4e-3	1e-4
Weight decay	1e-5	0.05	1e-4
Optimizer momentum	0.9	$\beta_1, \beta_2 = 0.9, 0.999$	$\beta_1, \beta_2 = 0.9, 0.999$
Batch size	1024	4096	256
Training epochs	600	300	40
Learning rate schedule	Step decay	Cosine decay	Cosine decay
Warmup epochs	5	20	10

(Continued)

Table 1 (continued)

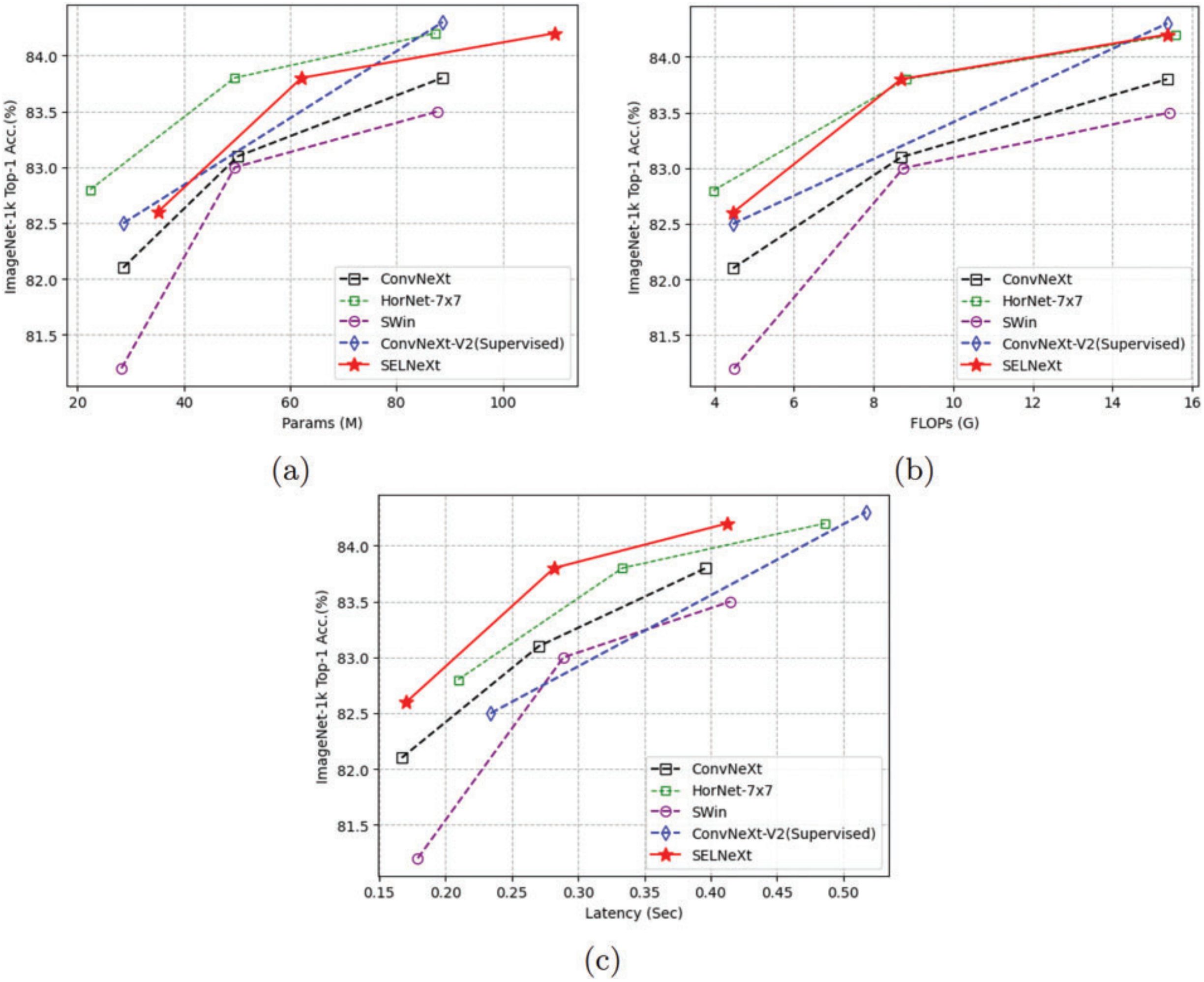
	MobileNetV3	SELNeXt-T/S/B	Transfer learning
Warmup schedule	Linear	Linear	Linear
Randaugment [36]	(9, 0.5)	(9, 0.5)	(9, 0.5)
Mixup [37]	None	0.8	0.8
Cutmix [38]	None	1.0	1.0
Random erasing [39]	0.2	0.25	0.0
Label smoothing [40]	0.1	0.1	0.1
Stochastic depth [41]	0.2	0.1/0.4/0.5	0.0
Layer scale [42]	None	1e-6	Pre-trained
Exponential moving average (EMA) [43]	0.9999	0.9999	None

Results. Table 2 shows the validation results of SELNeXt, including transformer-based and other ConvNeXt-based networks. It reports the top-1 accuracy on the validation set of ImageNet-1k, as well as the number of parameters, computational complexity (FLOPs), and latency. Latency is measured except for model loading time and data preprocessing, which are conducted in the same manner as in [33]. The experiment is performed iteratively five times for consistent and reliable measurement. The average latency with its standard deviation is included in the table. Inference latency is measured on a single RTX 2080 Ti with batch size 96 and full precision (FP32).

Table 2: Classification results on ImageNet-1k

Network	Image size	Params. (M)	FLOPs (G)	Latency (s/it)	Acc.
Swin-T [25]	224^2	28	4.5	0.167 ± 0.001	81.2
ConvNeXt-T [9]	224^2	29	4.5	0.157 ± 0.000	82.1
HorNet-T$_{7\times7}$ [15]	224^2	22	4.0	0.192 ± 0.001	82.8
HorNet-T$_{GF}$ [15]	224^2	23	3.9	0.239 ± 0.000	83.0
ConvNeXt V2-T [16]	224^2	29	4.5	0.212 ± 0.001	82.5
SELNeXt-T	224^2	35	4.5	0.160 ± 0.001	82.6
Swin-S [25]	224^2	50	8.7	0.284 ± 0.001	83.0
ConvNeXt-S [9]	224^2	50	8.7	0.264 ± 0.002	83.1
HorNet-S$_{7\times7}$ [15]	224^2	50	8.8	0.333 ± 0.003	83.8
HorNet-S$_{GF}$ [15]	224^2	50	8.7	0.397 ± 0.004	84.0
SELNeXt-S	224^2	62	8.7	0.276 ± 0.001	83.8
Swin-B [25]	224^2	89	15.4	0.413 ± 0.002	83.5
ConvNeXt-B [9]	224^2	88	15.4	0.394 ± 0.002	83.8
HorNet-B$_{7\times7}$ [15]	224^2	87	15.6	0.489 ± 0.002	84.2
HorNet-B$_{GF}$ [15]	224^2	88	15.5	0.570 ± 0.006	84.3
ConvNeXt V2-B [16]	224^2	89	15.4	0.515 ± 0.001	84.3
SELNeXt-B	224^2	110	15.4	0.413 ± 0.003	84.2

Fig. 5 illustrates the accuracy comparison with other models in terms of the number of parameters, FLOPs, and latency. Our SELNeXt requires approximately 23% more parameters than ConvNeXt. The parameter increase comes from adding two non-shared weights of size $\mathbb{R}^{C\times\frac{C}{r}}$ for each block repetition. SELNeXt is expected to achieve slightly higher classification accuracy with a similar number of parameters compared to ConvNeXt. However, it is less parameter-efficient than ConvNeXt V2 and HorNet.



(a)

(b)

(c)

Figure 5: Top-1 accuracy of ImageNet-1k classification in terms of (a) the number of parameters, (b) FLOPs, and (c) latency

Our small and base models demonstrate comparable accuracy to HorNet and ConvNeXt V2 regarding computational complexity. The fully connected layers in SELN are applied after GAP, which reduces the dimension from $\mathbb{R}^{C\times W\times H}$ to $\mathbb{R}^{C\times 1\times 1}$. The computational cost for performing a single fully connected layer is $C \times C'$. SELNeXt has a negligible increase in computational complexity compared to ConvNeXt, as shown in Table 2. Our tiny model, SELNeXt-T, shows slightly lower accuracy compared to HorNet-T due to its shallower depth and wider channels.

Our models outperform the others in terms of latency. SELNeXt is designed to minimize additional latency-impacting operations and exclude conventional LN. In contrast, both HorNet and ConvNeXt require longer latency because HorNet contains more depth-wise convolution and high-order element-wise multiplication, and ConvNeXt V2 requires conventional LN and additional element-wise additions.

Fig. 6 shows the attention maps of ConvNeXt-B and SELNeXt-B, generated using Ablation CAM [48]. SELNeXt-B produces heat maps that are more focused on the ground truth objects, such as the stopwatch and basketball.

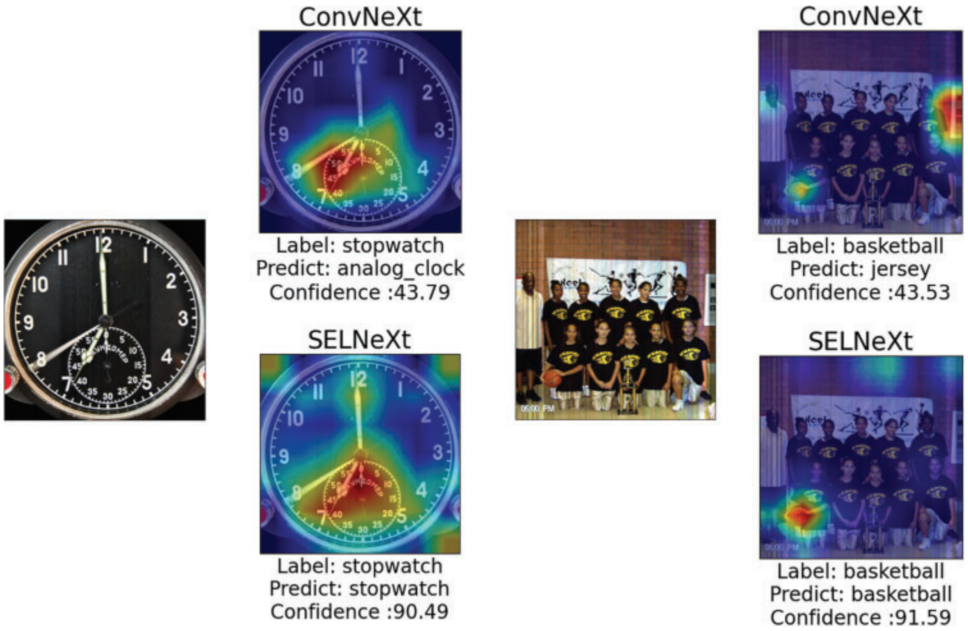


Figure 6: Attention maps for the input image (left), showing ConvNeXt-B (top right) and SELNeXt-B (bottom right)

4.2 *Transfer Learning*

Transfer learning is conducted for both image classification and object detection/segmentation.

4.2.1 Image Classification

CIFAR-100 [44], Oxford 102 Flowers [45], Stanford Cars [46], and FGVC-Aircraft [47] datasets are tested to validate generalized performance. These datasets comprise 60,000 images across 100 classes, 8189 images across 102 classes, 16,185 images across 196 classes, and 10,000 images across 100 classes, respectively. Table 1 outlines the detailed settings for the training hyperparameters. Table 3 shows that SELNeXt is more accurate than ConvNeXt across all datasets when comparing models of similar sizes.

Table 3: Classification accuracy on CIFAR-100, Oxford 102 Flowers, Stanford Cars, and FGVC-Aircraft datasets

	ConvNeXt-T [9]	**SELNeXt-T**	**ConvNeXt-S [9]**	**SELNeXt-S**	**ConvNeXt-B [9]**	**SELNeXt-B**
CIFAR-100 [44]	89.1	89.9	90.6	90.7	90.8	91.5
Oxford 102 Flowers [45]	90.0	92.0	91.4	93.9	94.1	94.5
Stanford Cars [46]	86.0	87.2	87.0	89.9	89.6	90.2
FGVC Aircraft [47]	77.7	79.8	79.9	80.8	81.5	82.7

4.2.2 Object Detection and Segmentation

Setup. COCO [49] dataset, which contains 118 and 5 k images for training and validation, is used to evaluate the object detection and instance segmentation performances. We finetune Mask R-CNN [50] and Cascade Mask R-CNN [51] on the COCO dataset with SELNeXt as a backbone, following Swin Transformer [25] and ConvNeXt [9] for a fair comparison. We use multi-scale training, AdamW [52] optimizer, and a 3× schedule. FPS is measured on a single RTX 2080 Ti. FLOPs are calculated with image size (1280 × 800). Table 4 offers detailed summary of the training hyperparameter settings.

Table 4: Hyperparameter settings for object detection and segmentation

	MobileNetV3	**SELNeXt-T/S/B**
Resolution	(1333,800)	(1333,800)
Batch size	16	16
Optimizer	AdamW	AdamW
Train schedule	1×	1×/3×
Weight decay	1e-4	0.05
Learning rate schedule	StepLR	StepLR
Learning rate	2e-4	1e-4 (1×)/2e-4 (3×)
Stochastic depth [41]	0.15	0.4 (T)/0.6 (S)/0.7 (B)

Results. Table 5 shows the object detection and instance segmentation performance of SELNeXt and also lists the performance of Swin Transformer [25] and ConvNeXt [9] for comparison. Across different model complexities, SELNeXt achieves better performance than Swin Transformer and ConvNeXt in many cases. However, SELNeXt shows limited performance for base-sized models in downstream tasks.

Table 5: Object detection and segmentation results on COCO

Backbone	**#Param.**	**FLOPs**	**FPS**	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Mask-RCNN 3 × schedule									
Swin-T [25]	48 M	264 G	11.8	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T [9]	48 M	262 G	12.5	46.2	67.9	50.8	41.7	65.0	45.0
SELNeXt-T	55 M	262 G	12.5	46.7	68.6	51.6	42.4	66.0	45.9
Cascade Mask-RCNN 3 × schedule									
Swin-T [25]	86 M	745 G	4.8	50.4	69.2	54.7	43.7	66.6	47.3
ConvNeXt-T [9]	86 M	741 G	4.9	50.4	69.1	54.8	43.7	66.5	47.3
SELNeXt-T	92 M	741 G	5.0	51.0	69.9	55.4	44.4	67.5	48.1
Swin-S [25]	107 M	838 G	4.4	51.9	70.7	56.3	45.0	68.2	48.8
ConvNeXt-S [9]	108 M	827 G	4.4	51.9	70.8	56.5	45.0	68.4	49.1
SELNeXt-S	119 M	827 G	4.4	52.3	71.4	56.7	45.4	69.0	49.3
Swin-B [25]	145 M	982 G	3.9	51.9	70.5	56.4	45.0	68.1	48.9
ConvNeXt-B [9]	146 M	964 G	3.9	52.7	71.3	57.2	45.6	68.9	49.5
SELNeXt-B	167 M	964 G	3.9	52.5	71.5	56.8	45.6	69.0	49.6

4.3 Ablation Studies

In this sub-section, detailed ablation studies for SELN and SELNeXt are provided. Tables 6–8 show ablation studies for SELN. MobileNetV3 is utilized as the baseline with the same hyperparameters for classification in [53]. Table 1 provides detailed training hyperparameter settings. For object detection and segmentation, Mask RCNN is fine-tuned using baseline, and SELN integrated models are used as a backbone with the hyperparameter configurations of Table 4.

Table 6: Ablation study for SELN with classification accuracy on ImageNet-1k. Where $\overline{dist}$ indicates the mean of average feature cosine distance

Activation func. in SE	$\overline{dist}$	Acc.
Baseline [13]	0.1501	75.78
tanh	0.1579	75.57
Identity	0.1601	75.91
LN (SELN)	0.1612	76.02

Table 7: Classification accuracy comparison on ImageNet-1k according to attention mechanisms, where − and + indicate eliminating and adding, respectively

Attention method	Params. (M)	FLOPs (G)	Latency (s/it)	$\overline{dist}$	Acc.
Baseline [13]	5.48	0.23	0.027 ± 0.000	0.1501	75.78
Baseline - SE	3.97	0.22	0.025 ± 0.001	0.0791	74.32
Baseline - SE + GRN	3.97	0.22	0.030 ± 0.000	0.0711	74.37
Baseline - SE + ECA	3.97	0.24	0.027 ± 0.000	0.1044	74.82
Baseline - SE + SELN	5.49	0.22	0.027 ± 0.000	0.1612	76.02
MobileViT-XS [26]	2.31	1.05	0.102 ± 0.001	–	74.8
EdgeNeXt-XS [27]	2.33	0.54	0.064 ± 0.001	–	75.0

Table 8: Ablation study for SELN with object detection and segmentation performance on COCO

Mask-RCNN	AP^{box}	AP^{mask}
SE (Baseline) [13]	30.5	29.6
SELN	31.1	29.9

Table 6 shows classification accuracy on ImageNet-1k according to activation or normalization methods just after two fully connected layers of SE. H-sigmoid, tanh, none (indicated as identity), and LN(SELN) are compared, considering their influences on the average feature cosine distance. In this table, the SELN integrated model shows better performance over the baseline in classification.

Table 7 shows the classification accuracy across various attention mechanisms with the baseline of MobileNetV3, where SE is replaced with GRN, ECA, and SELN in the baseline. Other hybrid mobile networks, such as MobileViT-XS and EdgeNeXt-XS, are also compared. Baseline-SE+SELN shows the highest accuracy. Baseline-SE+GRN yields nearly identical performance to Baseline-SE because the channel attention effect is bypassed by the shortcut of GRN. Baseline-SE+ECA does not show any improvement compared to the baseline. Hybrid mobile networks, such as MobileViT-XS and EdgeNeXt-XS, are less accurate while requiring longer latency, as explained in [33]. In object detection and segmentation performance on COCO, the SELN case shows improved performance compared to the baseline, as shown in Table 7.

We also present an ablation study on the block design of our SELNeXt network to verify the design of a latency-efficient network as an alternative to LN, as illustrated in Table 9. We conduct ConvNeXt-T as the baseline network. Inference latency is measured on a single RTX 2080 Ti GPU with batch size 96 and full precision. The result of the Baseline+GRN case, equivalent to ConvNeXt V2, is borrowed from [16].

Table 9: Ablation study for SELNeXt on ImageNet-1k classification, where − and + indicate eliminating and adding, respectively

Model	Params. (M)	FLOPs (G)	Latency (s/it)	Acc.
Baseline [9]	28.6	4.5	0.157 ± 0.001	82.1
Baseline + GRN [16]	28.6	4.5	0.212 ± 0.001	82.5
Baseline + SE	35.1	4.5	0.174 ± 0.001	82.41
Baseline + SELN	35.1	4.5	0.175 ± 0.001	82.39
Baseline + SE - LN	35.1	4.5	0.160 ± 0.001	82.45
Baseline + SELN - LN	35.1	4.5	0.160 ± 0.001	82.55

We analyze the cases in which SE or SELN is directly added to the baseline. Both baseline+SE and baseline+SELN showed better accuracy over the baseline model while consuming longer inference time because of the additional operations. The cases for adding SE or SELN to the baseline and eliminating LN are also tested for ablations. Baseline+SELN-LN shows further improvement in accuracy within shorter latency compared to SE or SELN-added cases. Therefore, the block of SELNeXt is designed by adding SELN and removing LN from the baseline.

5 Conclusion

In this paper, we propose SELN, an enhancement to SE-based channel attention mechanisms, by replacing the sigmoid activation function with LN. Building on SELN, we also introduce SELNeXt, a latency-efficient model that incorporates SELN into the ConvNeXt block and substitutes the conventional LN. SELNeXt achieves a superior accuracy-latency trade-off across benchmarks, including ImageNet-1k for classification and COCO for detection/segmentation tasks. To further validate its generalization capabilities, we test on various datasets with fine-tuning, and the experimental results demonstrate that SELNeXt consistently achieves higher accuracy than ConvNeXt.

SELNeXt is limited in terms of the number of model parameters. SELNeXt requires approximately 23% more parameters than ConvNeXt. In addition, the performance improvement of SELNeXt-B appears to be limited on tasks like object detection and segmentation.

Future research on SELN and SELNeXt could investigate integrating self-supervised learning approaches. The increased feature diversity shown by the SELN-integrated model in supervised learning shows its potential for effectively capturing diverse input characteristics, especially during the reconstruction of masked inputs, as reported in ConvNeXt V2 [16].

Consequently, LN could serve as a viable alternative to the activation function in attention mechanisms. It could help alleviate the vanishing gradient problem and feature collapse while enhancing the discriminability of channel attention. SELNeXt achieves a superior accuracy-latency trade-off, making it particularly valuable for real-time applications and edge computing.

Acknowledgement: None.

Funding Statement: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2021R1A6A1A03039493.

Author Contributions: The authors confirm their contribution to the paper as follows: conceptualization, methodology: Woongkyu Park; investigation: Woongkyu Park, Yeongyu Choi, and Mahammad Shareef Mekala; experiment: Woongkyu Park, and Yeongyu Choi; analysis: Woongkyu Park, Kook-Yeol Yoo, Gyu Sang Choi, and Ho-youl Jung; writing: Woongkyu Park, Mahammad Shareef Mekala, and Ho-youl Jung; supervision: Ho-youl Jung. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data and materials used in this paper are derived from publicly accessible, which are cited throughout the text. References to these sources are provided in the bibliography.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Chen R, Pan L, Li C, Zhou Y, Chen A, Beckman E. An improved deep fusion CNN for image recognition. Comput Mater Contin. 2020;65(2):1691–706. doi:10.32604/cmc.2020.011706.
2. Wu H, Liu Q, Liu X. A review on deep learning approaches to image classification and object segmentation. Comput Mater Contin. 2019;60(2):575–97. doi:10.32604/cmc.2019.03595.
3. Song W, Dong L, Zhao X, Xia J, Liu T, Shi Y. Review of nodule mineral image segmentation algorithms for deep-sea mineral resource assessment. Comput Mater Contin. 2022;73(1):1649–69. doi:10.32604/cmc.2022.027214.
4. Bjorck N, Gomes CP, Selman B, Weinberger KQ. Understanding batch normalization. Adv Neural Inf Process Syst. 2018;31:7705–16.
5. Woo S, Park J, Lee JY, Kweon IS. Cbam: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). Cham, Switzerland: Springer; 2018. p. 3–19.
6. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:6000–10.
7. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: transformers for image recognition at scale. In: International Conference on Learning Representations. Appleton, WI, USA: ICLR; 2020.
8. Ba JL, Kiros JR, Hinton GE. Layer normalization. arXiv:160706450. 2016.
9. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S. A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2022. p. 11976–86.
10. Ding X, Zhang X, Han J, Ding G. Scaling up your kernels to 31 × 31: revisiting large kernel design in CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2022. p. 11963–75.
11. Rao Y, Zhao W, Zhu Z, Zhou J, Lu J. GFNet: global filter networks for visual recognition. IEEE Transact Patt Anal Mach Intell. 2023;45(9):10960–73. doi:10.1109/TPAMI.2023.3263824.
12. Yu W, Luo M, Zhou P, Si C, Zhou Y, Wang X, et al. Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2022. p. 10819–29.

13. Howard AG. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:170404861. 2017.
14. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2017. p. 1492–500.
15. Rao Y, Zhao W, Tang Y, Zhou J, Lim SN, Lu J. Hornet: efficient high-order spatial interactions with recursive gated convolutions. Adv Neural Inform Process Syst. 2022;35:10353–66.
16. Woo S, Debnath S, Hu R, Chen X, Liu Z, Kweon IS, et al. Convnext v2: co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2023. p. 16133–42.
17. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2009. p. 248–55.
18. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2019. p. 1314–24.
19. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2018. p. 7132–41.
20. Guo MH, Xu TX, Liu JJ, Liu ZN, Jiang PT, Mu TJ, et al. Attention mechanisms in computer vision: a survey. Computat Vis Media. 2022;8(3):331–68. doi:10.1007/s41095-022-0271-y.
21. Qin Z, Zhang P, Wu F, Li X. Fcanet: frequency channel attention networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2021. p. 783–92.
22. 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. Piscataway, NJ, USA: IEEE; 2020. p. 11534–42.
23. 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. Piscataway, NJ, USA: IEEE; 2020. p. 11794–803.
24. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25:84–90.
25. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2021. p. 10012–22.
26. Mehta S, Rastegari M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. arXiv:211002178. 2021.
27. Maaz M, Shaker A, Cholakkal H, Khan S, Zamir SW, Anwer RM, et al. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2022. p. 3–20.
28. Chen Y, Dai X, Chen D, Liu M, Dong X, Yuan L, et al. Mobile-former: bridging mobilenet and transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2022. p. 5270–9.
29. Vasu PKA, Gabriel J, Zhu J, Tuzel O, Ranjan A. FastViT: a fast hybrid vision transformer using structural reparameterization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2023. p. 5785–95.
30. Li Y, Hu J, Wen Y, Evangelidis G, Salahi K, Wang Y, et al. Rethinking vision transformers for mobilenet size and speed. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2023. p. 16889–900.
31. Li J, Xia X, Li W, Li H, Wang X, Xiao X, et al. Next-vit: next generation vision transformer for efficient deployment in realistic industrial scenarios. arXiv:220705501. 2022.
32. Shazeer N. Fast transformer decoding: one write-head is all you need. arXiv:191102150. 2019.

33. Chen J, Sh K, He H, Zhuo W, Wen S, Lee CH, et al. Run, don't walk: chasing higher FLOPS for faster neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2023. p. 12021–31.
34. Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z. Dynamic relu. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2020, p. 351–67.
35. Mulindwa DB, Du S. An n-sigmoid activation function to improve the squeeze-and-excitation for 2D and 3D deep networks. Electronics. 2023;12(4):911. doi:10.3390/electronics12040911.
36. Cubuk ED, Zoph B, Shlens J, Le QV. Randaugment: practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Piscataway, NJ, USA: IEEE; 2020. p. 702–3.
37. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. Mixup: beyond empirical risk minimization. In: International Conference on Learning Representations. Appleton, WI, USA: ICLR; 2018.
38. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y. Cutmix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2019. p. 6023–32.
39. Zhong Z, Zheng L, Kang G, Li S, Yang Y. Random erasing data augmentation. Proc AAAI Conf Artif Intell. 2020;34(7):13001–8. doi:10.1609/aaai.v34i07.7000.
40. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2016. p. 2818–26.
41. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ. Deep networks with stochastic depth. In: Computer Vision–ECCV 2016: 14th European Conference; 2016 Oct 11–14; Amsterdam, The Netherlands: Springer. p. 646–61.
42. Touvron H, Cord M, Sablayrolles A, Synnaeve G, Jégou H. Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2021. p. 32–42.
43. Polyak BT, Juditsky AB. Acceleration of stochastic approximation by averaging. SIAM J Cont Optimiz. 1992;30(4):838–55. doi:10.1137/0330046.
44. Krizhevsky A. Learning multiple layers of features from tiny images. In: Handbook of systemic autoimmune diseases. Toronto, ON, Canada: University of Toronto; 2009.
45. Nilsback ME, Zisserman A. Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. Piscataway, NJ, USA: IEEE; 2008. p. 722–9.
46. Krause J, Stark M, Deng J, Fei-Fei L. 3D object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops. Piscataway, NJ, USA: IEEE; 2013.
47. Maji S, Kannala J, Rahtu E, Blaschko M, Vedaldi A. Fine-grained visual classification of aircraft. arXiv:1306.5151. 2013.
48. Desai S, Ramaswamy HG. Ablation-cam: visual explanations for deep convolutional network via gradient-free localization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Piscataway, NJ, USA: IEEE; 2020. p. 983–91.
49. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference; 2014 Sep 6–12; Zurich, Switzerland. p. 740–55.
50. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ, USA: IEEE; 2017. p. 2961–9.
51. Cai Z, Vasconcelos N. Cascade R-CNN: delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE; 2018. p. 6154–62.
52. Loshchilov I, Hutter F. Decoupled weight decay regularization. In: International Conference on Learning Representations. Appleton, WI, USA: ICLR. 2018.
53. Wightman R. PyTorch Image Models. GitHub [Internet]; 2019. [cited 2025 Jan 14]. Available from: https://github.com/rwightman/pytorch-image-models.