World Wide Web (2022) 25:1583–1606
https://doi.org/10.1007/s11280-021-00968-2

Scaled gated networks

Ruiyuan Lu[1] · Jihua Zhu[1] · Xueming Qian[2] · Zhiqiang Tian[1] · Yi Yue[1]

Received: 2 September 2021 / Revised: 4 October 2021 / Accepted: 11 October 2021 /
Published online: 23 October 2021

Abstract
Gating transformation demonstrates great potential in recent deep convolutional neural networks design, enriching the feature representation and alleviating noisy signals by modeling the inter-channel dependencies using learnable parameters. However, the utilization of scaling approaches to reduce the redundancy of the hand-crafted attention mechanism has rarely been investigated. This paper proposes a novel scaled gated convolution that enables attention-enhanced CNNs to overcome the paradox between performance and redundancy. Our scaled gated convolution is a simple and effective alternative compared with both vanilla convolution and attention-enhanced convolutions, which can be easily applied to modern CNNs in a plug-and-play manner. Exhaustive experiments demonstrate that stacking scaled gated convolutions in baselines can significantly improve the performance in a broad range of visual recognition tasks, including image recognition, object detection, instance segmentation, keypoint detection, and panoptic segmentation, while obtaining a better trade-off between performance and attentive redundancy.

Keywords Scaling network · Gating mechanism · Visual recognition

1 Introduction

Deep convolutional neural networks (CNNs) have attracted a broad range of research interests in the computer vision field and have achieved remarkable progress in various visual recognition tasks, including image classification [20, 31, 35, 55], object detection [51, 52], semantic segmentation [5], instance segmentation [17, 28], and human keypoint detection [17, 70]. Standard convolution layers containing collections of filters express neighborhood spatial feature connectivity along input channels by a linear transformation, together with non-linear activation functions, play a central role in CNNs. Traditional CNNs

This article belongs to the Topical Collection: *Special Issue on Synthetic Media on the Web*
Guest Editors: Huimin Lu, Xing Xu, Jože Guna, and Gautam Srivastava

✉ Jihua Zhu
zhujh@xjtu.edu.cn

[1] School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China

[2] Ministry of Education Key Laboratory for Intelligent Networks and Network Security and with SMILES LAB, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China

serve as robust feature extractors by stacking convolution layers followed by activation layers; e.g., VGGNets [55] construct deep CNNs by modular designed 3x3 convolution layers with non-linear activation functions capable of capturing global context information. In order to further tap the potential of deeper architectures, modern CNNs introduce skip-connection [20] and variants [9, 21, 27] to alleviate gradient vanish.

In addition to the deeper CNNs, another category approaches [3, 22, 25, 26, 64, 68] focus on enriching feature representation according to long-range context dependencies learned by extra parameters, which present considerable potential for practical applications. Some methods [25, 26, 36] in-cooperate attention mechanism with convolution and have boosted the performance of downstream tasks. One of the representative methods is Squeeze-and-Excitation Networks (SENet) [26], combined with various network architectures, bringing consistent performance gains in a wide range of vision tasks at the cost of additional parameters. Unlike the approaches mentioned above that suffer from a heavier computational burden, this paper mainly focuses on the following question: Is it possible to tap the potential of attention-enhanced CNNs while easing the computational burden of attentive modules?

To address this issue, we first revisit the gating mechanism in SENet [26] and several variants [25, 42, 63, 68]. SE block is a micro-encoder-decoder architecture applied at module-level, which aggregates long-range spatial dependencies by non-parametric global average pooling first. It then encodes non-linear latent channel relationships by cascading fully-connected (FC) layer and ReLU [48] activation function. The decoder part models the saliency of channel information flow using another FC layer followed by the Sigmoid function. Although SENet improves the performance of CNNs, it inevitably increases unnecessary complexity compared with original models. In addition, empirical studies indicate dimensionality reduction of SE block is unnecessary and inefficient due to its side effect on cross-channel information flow [63] and increased memory access cost (MAC) [47]. GENet [25] further explores parametric sampling kernels with various fields of view, which achieve better performance at the expense of increased computational budgets for spatial conditioning than SENet. CBAM [68] enhances feature representations using a dual-attention mechanism which consists of max pooling enhanced channel attention and spatial attention captured by extra convolutional kernels. SCNet [42] proposes a conditional calibration-based parallel, heterogeneous, dual-path architecture to enlarge receptive fields and complement informative features, balancing complexity and performance. ECANet [63] presents a locality-prior-driven design that overcomes the SE block dimension reduction defect and reduces the extra computational budgets for attentive modules with 1D convolutions.

In order to further explore the potential of lightweight attentive architectures, we present a scaled gated convolution as an efficient approach to strengthen the feature representations of vanilla convolutional transformations and reduce redundancy of existing attentive modules in a plug-and-play manner. The proposed scaled gated convolution consists of a triplet of operators: ***scaling, gating***, and ***fuse***. Specifically, the scaling operator re-scales feature and kernel spaces into multiple portions for successive heterogeneous feature transformations. The gating operator aggregates global feature context to enlarge the receptive field and leverages cross-channel information flow to generate self-adaptive attentive gating representations. The fuse operator aggregates features across multiple heterogeneous feature spaces adaptively for final semantic fusion.

As an enhanced version of standard convolution, three advantages of our scaled gated convolutions can be offered. First of all, it strengthens cross-channel information flow by adaptively encoding the informative long-range context features of multiple heterogeneous feature spaces, which enlarges the receptive field and suppresses noisy signals compared to

standard convolutions. Furthermore, the design of CNNs typically requires a wide selection of complicated hyperparameters and configurations. By contrast, our scaled gated convolutions can be directly deployed in existing state-of-the-art architectures by replacing original vanilla or attentive counterparts, and the performance can be effectively boosted. Besides, the scaled gated convolutions are computationally lightweight and require less redundancy and computational burden compared to existing attentive counterparts.

To verify the effectiveness and provide evidence for these claims, we develop a series of SGNets by plugging scaled gated convolutions into baselines, and conduct a comprehensive evaluation on large scale datasets. We first evaluate the proposed approach on the large scale ImageNet [31] dataset using ResNet variants [20, 26, 72] and obtain significant improvements with comparable model complexity. We also present results on downstream tasks, including object detection, instance segmentation, keypoint detection, and panoptic segmentation, to verify the ability to generalize our models on various typical downstream vision applications. Exhaustive experiments show that, by using SGNets, baseline results can be effectively improved for all these tasks at the expense of comparable or less computational budgets, which indicates our approach's efficiency.

2 Related work

2.1 Modern architecture design

Remarkable progress has been achieved in the field of network architecture design in recent years. AlexNet [35] laid the foundation for designing modern convolutional neural networks, which dominant the image recognition field. VGGNets [55] introduce modular design and the receptive field equality principle of convolutions and construct deeper networks with fewer parameters than AlexNet [35]. NIN [38] reduces overfitting by non-parametric global average pooling (GAP). Highway network [16], ResNet [20, 21], and DenseNet [27] alleviate vanishing gradient problems by various skip connections and help deep networks convergence. DPN [9] combines residual connections and dense connections to learn robust feature representations. GoogleNet [58] and Inception series [57, 59] enhance feature representations by stacking hand-engineered inception blocks, which introduce heterogeneous multi-path convolutions. ResNeXt [72] further simplifies multi-path networks by homogeneous group convolutions. WideResNet [75] strengthens shallow networks by adjusting the width of models. Xor is utilized for efficient deep hashing [45]. ShuffleNet [47, 77] enhances feature representations of lightweight models with channel shuffle. EfficientNet [60, 61] scales width, depth, and resolution with NAS and achieves remarkable performance gains.

2.2 Attention and gating mechanisms

In addition to plain architectures, effective attention and gating mechanisms design also attract more and more research interests. Attention and gating mechanisms can be interpreted as self-adaptive content-aware computational resource reallocation mechanisms based on informative components, demonstrating their utility across various tasks. SENet [26] firstly adopts Squeeze-and-Excitation blocks among channel dimensions. Beyond channel, GENet [25] leverages extra 2D convolutions to generate spatial region-aware attention weights. SKNet [36] further extends attention and gating mechanisms on kernels, which adjusts the receptive field of convolutions dynamically. CBAM [68]

combines spatial and channels attention mechanisms to recalibrate feature representations. SCNet [42] proposes an hourglass-style calibration-based operator to enhance the standard convolution. NLNet [64] models long-range dependency using self-attention. GCNet [3] further simplifies the NL block and proposes a lightweight GC block compatible with the SE block. DANet [12] aggregates dual-path heterogeneous attention to capture large-scale feature dependency. Fuzzy attention [44] is utilized to extract robust features. CCNet [29] proposes a lightweight recurrent criss-cross attention block to reduce the computational budget for large resolution scene parsing. ECANet [63] generates attentive weights based on locality prior to overcome the paradox of performance and complexity trade-off.

2.3 Dynamic neural networks

Different from static neural networks, which recognize visual objects by utilizing static and content-independent filters, dynamic neural networks construct sample-aware architectures using parametric components. CondConv [74] and DyConv [7] generate dynamic kernels conditioned on input samples. WeightNet [46] further introduces SE block and sparse block diagonal matrix to save computational budget. SkipNet [65] learns specific components with reinforcement learning. DRSS [37] builds a gating mechanism to adjust feature scales according to input samples. DyReLU [8] and APReLU [78] are capable of adaptive rectified factor correction using a learnable parametric rectified linear unit to boost performance.

2.4 Neural network scaling

In order to overcome the paradox of complexity and performance, scaling deep neural networks are widely explored in both hand-crafted and automated-searched neural network architectures. After the modular design principle introduced by VGGNets [55] is applied widespread, the ResNet [20, 21] series further tap the potential of depth scaling of models using residual connections and achieve remarkable gains with less complexity compared to VGGNets. MobileNets [23, 24, 53] scale the width of bottleneck structure to enhance feature representations. WideResNet [75] proposes depth-width scaled shallow-wide architectures and reaches comparable performance compared to deep-narrow counterparts [20, 21]. EfficientNet [61] introduces a neural architecture search-based compound search approach to scale input resolution, depth and width automatically and achieves a better balance between complexity and performance. RegNet [50] proposes a statistical information-based principle to adjust design spaces to sample a series of compound scaled ResNeXt-style networks with neural architecture search.

2.5 Transfer learning on vision tasks

Extracting informative and robust feature representations is of great importance to a wide range of modern deep transfer-learning-driven vision recognition tasks, including object detection, instance segmentation, human skeleton keypoint detection, and panoptic segmentation. Plenty of previous architectures demonstrate the ability to generalize on multiple aforementioned transfer-learning tasks.

2.5.1 Object detection

Recognizing and locating various objects in a scene requires backbone networks to balance the collision of feature representations between classification and localization and overcome

the aliasing effect caused by the uncertainty of localization. ImageNet [31] pre-trained networks, e.g., VGGNets [55], are firstly utilized as feature extractors for R-CNN families [13, 14, 52]. In order to generate fixed-sized feature representation, SPPNet [19] proposes spatial pyramid pooling to bridge the gap between convolution and fully connected layers. Fast R-CNN [13] extends the SPPNet and proposes ROI pooling to ease the difficulty of the learning process. The feature pyramid network (FPN) [39] further enhances multi-scale feature representations extracted by backbone networks at different stages with pyramid structure and alleviates the feature aliasing effect using a lateral connection.

2.5.2 Instance segmentation

In order to segment instances both accurately and precisely, segmentation networks [1, 2, 17] extract instance-aware representative context features and alleviate irrelevant noise with various backbones [20, 55]. Mask R-CNN [17] alleviates feature misalignment by bi-linear sampling-based RoI align operation, which reduces quantization error compared to RoI pooling [13] and introduces an extra mask prediction head for high-resolution dense prediction. Inspired by a single-stage object detector [40], YOLACT networks [1, 2] regard instance segmentation as a mask coefficients prediction task based on fully convolutional networks (FCNs). PolarMask [71] constructs a unified framework in polar coordinate space with center-guided classification and dense distance regression, which unites the coarse-grained bounding box localization and fine-grained edge prediction with the same representations. SOLO [66, 67] proposes a fast and straightforward FCN framework to segment objects by different locations. BlendMask [4] further fuses the instance feature representations and dense segmentation features using Blender and achieves higher performance.

2.5.3 Keypoint detection

In recent years, deep CNNs have significantly advanced keypoint detection, and various networks have been proposed to extract instance-aware skeleton features. Mask R-CNN [17] introduces a joint training scheme of keypoint and object detection based on the ResNet [20] backbone. HRNet [56] splits the main single-scale branch into multiple branches with different scales to enhance features with multi-scale representations. CPN [6] proposes a cascade pyramid refinement network together with online hard keypoint mining loss to extract keypoint features from coarse to fine. In contrast to complicated keypoint detection models, SimpleBaseline [70] constructs a simple and effective keypoint detection benchmark. DarkPose [76] designs a novel and model-agnostic encoding-decoding-based coordinate representation to boost the performance of keypoint detection.

2.5.4 Panoptic segmentation

Different from instance segmentation [4, 17] and semantic segmentation [5, 43, 49, 79], which focus on stuff/thing segmentation tasks in isolation, panoptic segmentation [34] requires a reconciliation between these tasks and recently attracted increasing research interests. Panoptic FPN [33] combines FPN [39] with mask R-CNN [17] and semantic segmentation head to generate a robust panoptic prediction. UPSNet [73] alleviates feature conflicts between semantic and instance segmentation by utilizing deformable convolution [11, 80] and a parametric-free panoptic segmentation head in a unified framework.

introduces attentive structure to alleviate feature noise and improve the performance of panoptic segmentation.

3 Methodology

In this section, we present the details of the proposed scaled gated convolutions for image recognition. It is a lightweight module based on transformation $\widetilde{\mathcal{F}}$, capable of mapping an input tensor $\mathbf{X} = [x_1, x_2, \ldots, x_{\widetilde{C}}] \in \mathbb{R}^{\widetilde{C}\times\widetilde{H}\times\widetilde{W}}$ to feature representation $\mathbf{U} = [u_1, u_2, \ldots, u_C] \in \mathbb{R}^{C\times H\times W}$. Conventional convolution transformation $\mathcal{F}$ consists of homogeneous filters $\mathbf{V} = [v_1, v_2, \ldots, v_C]$ and learns local representations with a fixed receptive field. Given the above notations, the transformed feature representations at the i-th channel can be written as:

$$u_i = v_i * \mathbf{X} = \sum\nolimits_{j=1}^{\widetilde{C}} v_i^j * x^j \tag{1}$$

where $*$ denotes convolution operator, $u_i \in \mathbb{R}^{H\times W}$, $v_i = [v_i^1, v_i^2, \ldots, v_i^{\widetilde{C}}]$ v_i^j is a 2D single-channel convolutional filter with a fixed kernel size that acts on the corresponding channel of $\mathbf{X}$. Spatial dimension and bias terms are omitted for neat notations. As can be seen in (1), traditional convolutions extract local information by sliding windows with predefined kernel sizes. The channel correlation of feature representations is built by the inherent weighted summation of convolutions. We expect discriminative convolutional feature transformation learning to be strengthened by explicitly exploring long-range spatial semantic information and combining cross-channel correspondence with lightweight and powerful computational components. To this end, we propose scaled gated convolution.

3.1 Scaled gated convolutions

In SENet, Squeeze-and-Excitation modules are cascaded after the main branch, which constructs cross-channel information flow as auxiliary branches and imposes post-gating transformation to enhance feature representations. Different from SENet-style design cascaded after the main branch. Our scaled gated convolutions can apply gating mechanisms in a parallel paradigm, which combines both convolutional and gating transformation. Similar to group convolutions, the input feature representations are parted at the beginning and merged to generate final outputs. However, differently, group convolutions perform homogeneous transformation for groups, while ours introduce heterogeneous transformation to construct relationships among groups so that they may complement each other.

3.1.1 Overview

The overall pipeline of our scaled gated convolution is illustrated in Figure 1. The given input feature representation is denoted as $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$. The output feature map $\mathbf{Y} \in \mathbb{R}^{C\times H\times W}$ is designed to keep the same dimension as input $\mathbf{X}$ so that the scaled gated convolution can be applied to existing architectures in a plug-and-play manner. In order to reduce the redundancy of scaled gated convolution, the given input features $\mathbf{X}$ are scaled by λ and divided into two branches using ***scaling*** operator, i.e., $\mathbf{X_1}$, $\mathbf{X_2}$ for lightweight heterogeneous transformation. The overall framework of our proposed scaled gated convolution is formulated as:

$$\mathbf{Y} = \{\mathbf{X_1}; \mathbf{X_2}\} = \{\mathbf{X}_{1-\lambda}; \widetilde{\mathcal{F}}(\mathbf{X}_\lambda)\} \tag{2}$$

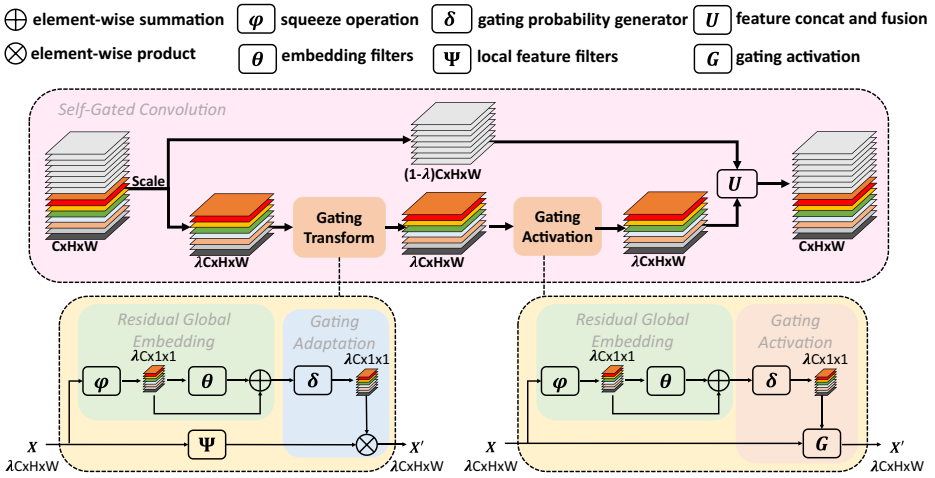


Figure 1 The pipeline of our proposed scaled gated convolution. The given input features representations are divided into gating branch and identity branch for heterogeneous processing. The heterogeneous branches are scaled by λ so as to reduce redundancy and improve performance. The gating mechanism is composed of a scaled gated transformation module that is succeeded by a scaled gated activation using lightweight filters. More details can be found in Section 3.1. Best viewed in color

where $\{\cdot;\cdot\}$ denotes feature fusion and $\widetilde{\mathcal{F}}$ denoted scaled gated transformation. Inspired by [77], the first branch uses identity mapping to generate the identity intermediate feature representation $\mathbf{Y_1}$, i.e., $\mathbf{Y_1} = \mathbf{X_1} \in \mathbb{R}^{(1-\lambda)C\times H\times W}$, which preserves spatial context based on high-resolution feature representation, and avoids loss of informative details caused by introducing extra learnable parameters. More importantly, the identity branch also preserves an auxiliary constant gradient flow where $\frac{\partial y}{\partial x} = \mathbf{1}$ so as to accelerate model convergence. The second branch adaptively adjusts input feature $\mathbf{X_2} \in \mathbb{R}^{\lambda C\times H\times W}$ using global context embedding guided scaled gated operation to obtain the other intermediate feature representation $\mathbf{Y_2}$. The scaled gated operation consists of a two-stage gating mechanism, more specifically, a scaled gated transformation module succeeded by a scaled gated activation. The final output feature representation $\mathbf{Y}$ is obtained by concatenating and fusing $\mathbf{Y_1}, \mathbf{Y_2}$. The details of our gating mechanism will be described in the following sections.

3.1.2 Scaled gated transformation

In order to effectively tackle the issue of exploiting the input information flow of scaled gated convolutions, we propose a scaled gated transformation module based on cross-channel information flow. Specifically, the channel-wise statistics $\boldsymbol{\mu} \in \mathbb{R}^{\lambda C}$ are created using non-parametric global average pooling. Formally, given the input feature representations of second branch $\mathbf{X_2} \in \mathbb{R}^{\lambda C\times H\times W}$, the channel-wise statistics $\boldsymbol{\mu} \in \mathbb{R}^{\lambda C}$ are generated by operation φ, which shrink the spatial dimension of $\mathbf{X_2} \in \mathbb{R}^{\lambda C\times H\times W}$. Thus, the c-th channel μ_c of channel-wise statistics $\boldsymbol{\mu}$ is calculated by:

$$\mu_c = \varphi(\mathbf{X_2^c}) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X_2^c}(i, j) \tag{3}$$

Furthermore, in order to construct a self-adaptive selection and adjustment mechanism, we utilize lightweight linear projection to model cross-channel information flow. A lightweight learnable parameter matrix $\mathbf{W} \in \mathbb{R}^{\lambda C \times \lambda C}$ which contains only around 1.3% parameters of the whole model (e.g., 0.3M vs. 22.3M for SGNet-50), is introduced to build cross-channel information flow among arbitrary pairs (μ_i, μ_j) of channel-wise statistics $\boldsymbol{\mu}$, so as to enhance the robustness of features. Formally, $\mathbf{W}$ is defined as:

$$\mathbf{W} = \begin{bmatrix} w^{1,1} & \cdots & w^{1,\lambda C} \\ \vdots & \ddots & \vdots \\ w^{\lambda C,1} & \cdots & w^{\lambda C,\lambda C} \end{bmatrix} \tag{4}$$

However, the lightweight linear projection $\mathbf{W} \in \mathbb{R}^{\lambda C \times \lambda C}$ limits the capability of modeling cross-channel information flow. On the one hand, such a linear parameter matrix limits the capability of non-linear projection; On the other hand, the introduced parameters might increase the optimization difficulty and potential risk of overfitting. Thus, residual global embeddings $\mathbf{e} \in \mathbb{R}^{\lambda C}$ are generated by combining both linear projection and residual connection to accelerate convergence, mitigate the risk of overfitting, and suppress the vanishing gradient problem. Formally, $\mathbf{e} \in \mathbb{R}^{\lambda C}$ is calculated as follows:

$$\mathbf{e} = f(\boldsymbol{\mu}) = \theta(\boldsymbol{\mu}) + \boldsymbol{\mu} = \mathbf{W} \circ \boldsymbol{\mu} + \boldsymbol{\mu} \tag{5}$$

where $\circ$ denotes matrix multiplication, $+$ denotes element-wise summation. The residual global embeddings $\mathbf{e} \in \mathbb{R}^{\lambda C}$ are then soft-gated by gating operation δ, and applied on large scale feature representations to construct powerful feature representation $\mathbf{z} \in \mathbb{R}^{\lambda C}$. As aforementioned, vanilla convolutions are able to enhance informative local details with fixed receptive fields. Thus, we introduce convolutional filters ψ to model local feature patterns of large-scale input feature representations $\mathbf{X_2}$. Formally, the c-th channel of the output of scaled gated transformation z_c can be calculated by:

$$z_c = (\psi * \mathbf{X_2^c}) \odot \delta(e_c) \tag{6}$$

where $*$ denotes convolution, and $\odot$ denotes element-wise multiplication. The soft-gating selection operation δ adaptively selects large scale feature representation based on residual global embeddings $\mathbf{e}$ using a sigmoid function.

3.1.3 Scaled gated activation

Conventional rectified linear unit [48] provides sparsity and non-linear fitting ability by suppressing negative feature representations yet limiting the robustness of negative feature representations. Parametric alternatives, e.g., PReLU [18] and ELU [10], introduce extra hyperparameters, which require parameter tuning for various downstream tasks.

Inspired by the success of applying APReLU [78] for fault diagnosis, we hypothesize that negative embeddings surpassed by ReLU [48] encode noise disturbed class-distribution-aware information whose potential has not been fully explored for visual recognition. Thus, to adaptively enhance the non-linear fitting ability of feature representations and inhibit noise, we propose a hyper-parameter-free module called scaled gated activation to tap the potential of class-aware negative embeddings and ease the learning process for our model. Formally, given the output $\mathbf{z} \in \mathbb{R}^{\lambda C}$ of (6) as input, the activated amplitude $\mathrm{m} \in \mathbb{R}^{\lambda C}$ can be calculated as:

$$\mathbf{m} = \delta(f(\varphi(\mathbf{z})) \tag{7}$$

where δ, f, φ are defined in Section 3.1.2. After the activated amplitude is obtained, the output feature representation of the second branch $\mathbf{Y_2} \in \mathbb{R}^{\lambda C \times H \times W}$ can be calculated as follows:

$$\mathbf{Y_2} = G(\mathbf{z}) = \max(\mathbf{z}, \mathbf{0}) + \min(\mathbf{z}, \mathbf{0}) \odot \mathbf{m} \tag{8}$$

Different from dual-branch APReLU [78], which constructs statistical correlations for positive/negative embeddings separately using cascaded fully-connected layers, which are computational costly. Our scaled gated activation module extracts non-linear representations based on our lightweight scaled gating mechanism. Thus, our scaled gated activation module can be plugged into our scaled gated convolution as an auxiliary non-linear feature extractor with a bit of additional computational budget, while APReLU [78] can not.

3.1.4 Post fusion

Inspired by the success of linear bottleneck in MobileNet-V2 [53] and calibration operator in SCNet [42], we propose a post-fusion module as the post-process of our scaled gated convolution to gather local feature context and fuse heterogeneous output feature representations. The intermediate feature representations $\mathbf{Y_1}, \mathbf{Y_2}$ are concatenated and fused to generate final output feature representations. Formally, given the output of heterogeneous output $\mathbf{Y_1}, \mathbf{Y_2} \in \mathbb{R}^{\lambda C \times H \times W}$, and post fuse convolutional filters $\mathbf{U}$, the final output feature representation is fused by:

$$\mathbf{Y} = F(\mathbf{Y_1}, \mathbf{Y_2}) = \mathbf{U} * [\mathbf{Y_1}; \mathbf{Y_2}] \tag{9}$$

where $\mathbf{U}$ denotes group convolutions which are divided into $\mathbf{K}$ groups. $*$ denotes convolution operation and $[\cdot; \cdot]$ represents feature concatenation.

3.2 Network architecture

Based on the scaled gated convolution, the overall architecture of our proposed SGNet is listed in Table 1. The reasons why our scaled gated convolution is applied to ResNet [20] are as follows. First of all, the design choices of ResNet follow modular design principles introduced by VGGNets [55], which are easy to extend to various downstream tasks, e.g., object detection and pose estimation, and compatible with existing methods like feature pyramid network [39]. In other words, plenty of tasks can benefit from replacing original convolutions with scaled gated convolutions in a plug-and-play manner. Moreover, the application of scaled gated convolution can benefit from residual connections for deep models, which avoids vanishing gradient problems. Last but not least, due to the design of the efficient bottleneck modules, ResNet is one of the state-of-the-art architectures with a low computational budget and model complexity. Specifically, our proposed SGNet consists of multiple bottlenecks containing scaled gated convolutions, termed "SG Bottleneck". Each SG bottleneck is composed of stacking 1×1 convolution, scaled gated convolution, and 1×1 convolution sequentially. By replacing large-size convolutions with our scaled gated convolutions, our SGNet is able to enhance cross-channel information flow, suppress feature noise, and strengthen the robustness of representations. The detailed configurations of SGNet-50 is shown in Table 1. Similar to ResNet-50, SGNet-50 contains four stages which consists of $\{3, 4, 6, 3\}$ SG bottlenecks respectively. Different SGNet architectures can be obtained by varying the number of bottleneck blocks of each stage. Compared with ResNet-50, our SGNet-50 is capable of maintaining comparable performance while saving around 8.6% parameters and 10.1% computational budget. Furthermore, our SGNet-50 can reduce 16.8% parameters and save 10.3% computational budget compared with SENet-50.

Table 1 Architectures comparison among ResNet-50, SENet-50 and our proposed SGNet-50

Output	ResNet-50	SENet-50	SGNet-50
112×112	7×7, 64, stride 2		
56×56	3×3 max pool, stride 2		
56×56	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \\ fc, [16, 256] \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ SG[3 \times 3, K = 2], 128 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
28×28	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 256 \\ fc, [32, 512] \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ SG[3 \times 3, K = 2], 256 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
14×14	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 512 \\ fc, [64, 1024] \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 512 \\ SG[3 \times 3, K = 2], 512 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
7×7	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 1024 \\ fc, [128, 2048] \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 1024 \\ SG[3 \times 3, K = 2], 1024 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
1×1	7×7 global average pool, 1000-d fc, softmax		
Params	24.4M	26.8 M	22.3M
FLOPs	3.86G	3.87G	3.62G

The building blocks are presented in brackets, including receptive fields and the number of feature channels, with the stacked blocks of each stage outside the brackets. The fc together with inner brackets indicates the fully connected layers integrated into the SE module. "$K = 2$" denotes the number of groups of convolutions **U** defined in Section 3.1.4. Params denotes the number of parameters and the FLOPs represents the number of multiply-adds

3.2.1 Relation to attentive architectures

Our proposed SGNet is quite different from existing attentive architectures. SENet [26] applies homogeneous attention to all channels, which leads to a lack of feature diversity. Differently, we apply heterogeneous operations, i.e., identity mapping and scaled gated design, to enhance the diversity and preserve the informative features and improve efficiency. Also, SENet [26] applies channel-wise reduction to reduce the complexity, introducing information loss, while ours scales the gating path to achieve a similar purpose without channel reduction. Furthermore, SENet [26] inserts channel-wise attention module after convolution as an individual operator, while ours integrates gating module with convolution as a whole to replace the original convolution operator in a plug-and-play manner. In order to generate heterogeneous features, SCNet [42] applies filters in hourglass-style, which inevitably requires extra computational budgets due to reserved large spatial scale, while ours is based on identity mapping together with scaled gating transformation, which is more lightweight. CBAM [68] also preserves large spatial resolutions and applies an attention mechanism to improve performance. However, our work demonstrates that even without the help of large spatial scale reservation, competitive performance can also be achieved, and computational

budgets might be reduced a little in terms of flops. Besides, heterogeneous filters utilized in SCNet [42] might introduce bias and lead to information loss, while ours may not. Moreover, our scaled gated activation is capable of enhancing non-linear fitting ability, while SCNet cannot. ECANet [63] explores a locality-based attention mechanism to reduce attention redundancy using 1D convolution, while ours scale the gating branch to achieve such purpose. In short, Table 2 shows the relationship to existing attentive architectures.

3.3 Complexity analysis

Given scaling coefficient $0 < \lambda \leq 1$, fusion groups $\mathbf{K} \in \mathbb{N}^+$, kernel size $S \times S$ and input feature $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, the complexity of our scaled gated convolution can be formulated as:

$$\begin{cases} \#\mathbf{P} = (\lambda^2 + S^2/\mathbf{K})C^2 \\ \#\mathbf{F} = [\lambda^2(HW + 1) + S^2HW/\mathbf{K}]C^2 \end{cases} \tag{10}$$

where $\#\mathbf{P}$ and $\#\mathbf{F}$ denote the number of parameters and flops of our scaled gated convolution, respectively. Compared to vanilla convolutions, the saved computational budget can be calculated as follows. Note that $\Delta\#\mathbf{P}$ and $\Delta\#\mathbf{F}$ denote the number of saved parameters and flops, respectively.

$$\begin{cases} \Delta\#\mathbf{P} = [(\mathbf{K} - 1)S^2/\mathbf{K} - \lambda^2]C^2 \\ \Delta\#\mathbf{F} = [(\mathbf{K} - 1)S^2HW/\mathbf{K} - \lambda^2(HW + 1)]C^2 \end{cases} \tag{11}$$

Note the complexity might increase in some cases, e.g., when $\mathbf{K} = 1$ and $\lambda = 1$. Yet, as presented in Section 4.1.2, we typically set $\mathbf{K} = 2$ and $\lambda = 0.5$ to overcome the paradox between redundancy and performance if not otherwise noted.

4 Experiments

We evaluate the performance on large-scale datasets on various tasks, including ImageNet [31] classification, object detection, instance segmentation, and keypoint detection

Table 2 Relation to attentive architectures

Model	CA.	CR.	SA.	SR.	Hetero.	Pos.
ResNet [20] (baseline)	✗	✗	✗	✗	✗	✗
SENet [26]	✓	bottleneck-style	✗	GAP	✗	after conv
CBAM [68]	✓	bottleneck-style	✓	GAP+GMP	✗	after conv
ECANet [63]	✓	conv 1D	✗	GAP	✗	after conv
SCNet [42]	✗	✗	✓	hourglass-style	✓	in conv
SGNet (ours)	✓	scaling factor	✗	GAP	✓	in conv

"CA." denotes channel-wise attention, "CR." denotes channel reduction approaches, "SA." denotes spatial-wise attention. "SR." denotes spatial reduction methods. "Hetero." denotes if models are capable of generating heterogeneous features. "Pos." denotes the position to apply attentive modules. "GAP" and "GMP" denote global average pooling and global max pooling, respectively

on COCO [41]. Specifically, classification performance is evaluated on ImageNet [31], and SGNet is adopted as the backbones for these tasks. Faster/Mask R-CNN [17, 52] and [70] are utilized as code bases for object detection, instance segmentation and keypoint detection, respectively. SGNet is also applied to the keypoint detection task to verify the transferability.

4.1 ImageNet classification

ImageNet [31] contains 1.28 million training images and 50k validation images of 1k classes. Models are trained on the training set, and accuracy is reported on the validation set. We adopt the official code based on the widely used Pytorch framework to run our experiments.

4.1.1 Implementation details

The standard data augmentation is applied as done in [20]. Specifically, the training images are randomly cropped to 224×224 with random horizontal flipping. All models are trained on 8 GPUs with batch size 256 for 100 epochs, and parameters are optimized by stochastic gradient descent (SGD) with a weight decay of 0.00005 and momentum of 0.9. The initial learning rate is set to 0.2, and we utilize a cosine learning rate schedule [30] with a linear warmup [15] for the first five epochs. Note that all experiments share the same environment and experimental settings using the same code base.

4.1.2 Ablation study

Fair comparison To compare the effectiveness of SGNet with other counterparts, the original large fields-of-view convolutions used in ResNet [20] are replaced by our scaled gated convolutions. We consider ResNet [20], ResNeXt [72], and attentive architectures [26, 42, 63, 68] with different depths and evaluate performance on the large-scale ImageNet [31] dataset. Specifically, for single-branch gating, SGNet-50 and SGNet-101 are obtained by replacing correspond convolutions of ResNet-50 and ResNet-101, respectively. For multi-branch gating, we follow [72] to simplify multiple gating branches as a grouped dual-branch scaled gated convolution containing a grouped gating branch and an identity branch. We adjust cardinality settings to SGNeXt-8x14d for these ResNeXt-style models to fit our scaled gated convolution instead of the default 32x4d settings in ResNeXt [72], i.e., ResNeXt-8x14d and SENeXt-8x14d models are utilized as baselines when compared with our SGNeXt-8x14d models for fair comparison if not otherwise specified. Speed is evaluated on 8 GTX-1080Ti.

As shown in Table 3, our SGNet-50 improves 1.7% top-1 accuracy (76.8% vs. **78.5**%) and 0.9% top-5 accuracy (93.4% vs. **94.3**%). Compared to ResNet-101, our SGNet-101 improves 1.0% top-1 accuracy (78.6% vs. **79.6**%) and 0.5% top-5 accuracy (94.3% vs. **94.8**%). Similar improvements can be seen for other counterparts. Note that our SGNet achieves comparable performance using our lightweight scaled gated convolution under same experimental settings compared to both attentive and vanilla architectures.

Pooling choices The global embeddings are channel-wise statistics obtained by pooling operations. Thus, we consider the influence of global average pooling (GAP) and global max pooling (GMP). As shown in Table 4, using GAP improves 0.6% top-1 accuracy (77.9% vs. **78.5**%) and 0.4% top-5 accuracy (93.9% vs. **94.3**%)). This might due to the fact

Table 3 Fair comparison on ImageNet [31]

Model	Params	FLOPs	Top-1	Top-5	FPS T.	FPS I.
ResNet-50 [20]	24.4M	3.86G	76.8	93.4	1249	4987
SENet-50 [26]	26.8M	3.87G	77.9	94.0	1062	4166
CBAM-50 [68]	26.8M	3.87G	77.9	94.0	800	2743
ECANet-50 [63]	24.4M	3.81G	77.9	94.0	1094	4454
SCNet-50 [42]	24.4M	4.29G	78.3	94.2	957	4008
SGNet-50	22.3M	3.62G	**78.5**	**94.3**	1053	3831
ResNeXt-50 [72]	24.4M	3.97G	77.9	93.9	750	4246
SENeXt-50 [26]	26.8M	3.98G	78.2	94.1	788	3378
SGNeXt-50	25.2M	4.10G	**79.0**	**94.4**	731	2743
ResNet-101 [20]	42.5M	7.27G	78.6	94.3	773	2967
SENet-101 [26]	47.0M	7.27G	79.4	94.7	788	3378
CBAM-101 [68]	47.0M	7.28G	79.1	94.6	455	1612
ECANet-101 [63]	42.5M	7.27G	79.2	94.7	640	2604
SCNet-101 [42]	42.5M	7.75G	79.1	94.6	610	2478
SGNet-101	38.5M	6.57G	**79.6**	**94.8**	644	2301
ResNeXt-101 [72]	43.0M	7.52G	79.1	94.5	554	2169
SENeXt-101 [26]	47.6M	7.53G	79.4	**94.8**	478	1972
SGNeXt-101	44.6M	7.79G	**79.5**	**94.8**	366	1289

"T." and "I." denote training and inference, respectively

that GMP captures global maximum as statistics while GAP constructs connections among arbitrary spatial positions so as to generate more powerful representations.

Residual global embedding To evaluate the residual connections in global embeddings, we compare global embeddings w/wo. residual connections. As presented in Table 4, introducing residual connections improves 0.4% top-1 accuracy and 0.2% top-5 accuracy. As can be seen, residual connections accelerate model convergence.

Scaling factor In order to verify the parameter sensitivity of scaling factor λ and fusion factor $\mathbf{K}$ for our scaled gated mechanism, we scale gating branch with different λ and $\mathbf{K}$ to balance the redundancy and performance. As presented in Table 5, empirically, $\lambda = 0.5$ and

Table 4 Pooling methods and residual connection of embeddings

Model	Max Pool	Avg Pool	Residual	Top-1	Top-5
ResNet [20]				76.8	93.4
SGNet-50	✓		✓	77.9	93.9
		✓		78.1	94.1
		✓	✓	78.5	94.3

Table 5 Parameter sensitivity of λ and $\mathbf{K}$ using SGNet-50

λ	0.1	0.3	0.5
Top-1/Top-5	78.1/94.1	78.3/94.2	78.5/94.3
MParams/GFlops	19.1/2.97	20.2/3.19	22.3/3.62
λ	0.7	0.9	1.0
Top-1/Top-5	78.6/94.3	78.5/94.3	78.5/94.3
MParams/GFlops	25.4/4.25	29.6/5.12	32.2/5.64
$\mathbf{K}$	1	2	4
Top-1/Top-5	78.6/94.4	78.5/94.3	78.1/94.2
MParams/GFlops	27.7/4.48	22.3/3.62	19.6/3.19

$\mathbf{K} = 2$ overcomes the paradox between computational budget and performance. Note that λ is fixed when varying $\mathbf{K}$ and vice versa.

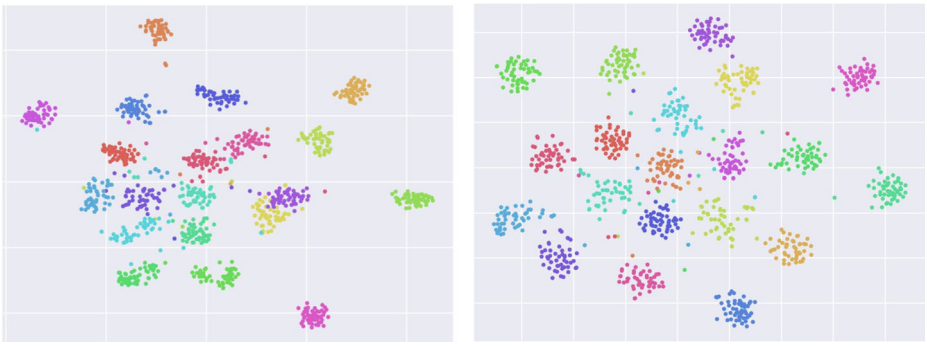
Ablation study of scaled gated modules To evaluate the influence of each scaled gated module, we remove scaled gated transformation, scaled gated activation as well as post fusion and validate the performance. As can be seen in Table 6, removing post fusion module leads to 1.8% top-1 accuracy and 0.9% top-5 accuracy drop. A similar trend can be observed when removing scaled gated transformation and scaled gated activation.

Visualization analysis To explore the class-specific information encoded by scaled gated activation, we uniformly sample 1k images from 20 randomly chosen classes and then project the extracted high-level semantic features using t-SNE [62] to verify the discriminability of class-specific distribution information before/after scaled gated activation. As presented in Figure 2, scaled gated activation enables features in the same class closer to each other and far from samples of other classes. Thus, the proposed scaled gated activation module is capable of encoding class-specific information.

In order to demonstrate the intuitive insight of our heterogeneous design, we visualize features at different positions and heatmaps [54] in Figure 3. Some filters focus on local details of textures and edges, which is darker, while others pay more attention to overall semantic information, which is brighter. The heterogeneous design is capable of making different modules extract heterogeneous features that complement each other.

Table 6 Ablation study of each scaled gated module, where "T.", "A." and "F." represent scaled gated transformation, scaled gated activation, and post fusion defined in Section 3.1, respectively

Model	T.	A.	F.	Params	Flops	Top-1	Top-5
ResNet-50 [20]				24.4M	3.86G	76.8	93.4
SGNet-50	✓	✓		16.9M	2.61G	76.7	93.4
		✓	✓	19.3M	2.99G	76.8	93.3
	✓		✓	22.0M	3.43G	78.3	94.1
	✓	✓	✓	22.3M	3.62G	**78.5**	**94.3**



(a) before scaled gated activation (b) after scaled gated activation

Figure 2 Class-aware features before/ after scaled gated activation

4.2 Object detection

To evaluate the transferability of our scaled gated convolutions, our SGNet models serve as backbones of object detectors and are trained on the COCO dataset [41].

4.2.1 Experimental settings

The widely used Faster R-CNN [52] with FPN [39] is utilized based on the Detectron2 [69] benchmark to run our detection experiments. All models are trained on the COCO-2017 training set, and we report COCO-style metrics (AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L) on the COCO-2017 validation set. Images are resized so that the edges are not longer than 1333 pixels. We use 8 GPUs to train each model for 90000 iterations, with batch size set to 16. The initial learning rate is set to 0.02 and divided by 10 after 60000 and 80000 iterations. SGD is utilized to optimize parameters. The weight decay and momentum are set to 0.0001

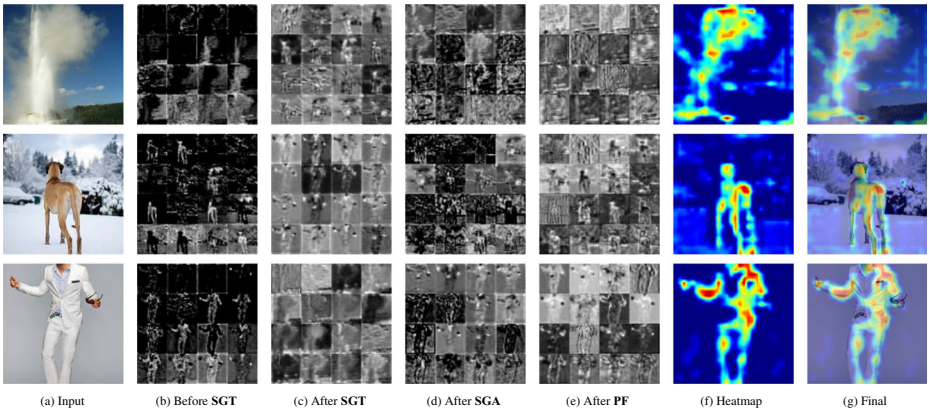


Figure 3 Visualization of selected features at different positions and heatmaps of the proposed scaled gated convolution. **SGT:** scaled gated transformation; **SGA:** scaled gated activation; **PF:** post fusion. Filters that generate darker visualizations focus more on textures and edges than those generate brighter visualization, which focus more on overall semantic information

and 0.9, respectively. For a fair comparison, multi-scale training and synchronized batch normalization are enabled for all models.

4.3 Object detection

To evaluate the transferability of our scaled gated convolutions, our SGNet models serve as backbones of object detectors and are trained on the COCO dataset.

4.3.1 Object detection results

As can be seen in Table 7, our SGNet-50 based detector outer-performs ResNet-50 based one around 3.5% AP (38.9% vs. 42.4%). Besides, our SGNet-50 brings 4.2%, 3.7%, and 3.4% for AP_S, AP_M, and AP_L, respectively. The same phenomena can be observed for other configurations in Table 7. Thus, our scaled gated convolution is capable of generating scale-robust feature representations compared to ResNet-50. Moreover, our SGNeXt based Faster R-CNN models bring large performance gaps (37.8% vs. **42.8**% for 50 layers and 39.6% vs. **44.4**% for 101 layers) compared to ResNeXt baselines, which indicates the heterogeneous gating transformation is able to generate much more powerful representations than homogeneous baselines using group convolutions. Besides, our model also achieves promising performance compared to attentive approaches [26, 42, 63]. In order to verify our scaled gated convolution is capable of overcoming the paradox of complexity and performance, we also evaluate the model complexity in terms of parameters and flops using the same code base. As can be seen in Table 7, our lightweight scaled gated convolution can be plugged into modern architectures and can achieve comparable performance with promising computational budgets.

Table 7 Fair comparison of object detection results on COCO [41]

Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params	Flops
ResNet-50 [20]	38.9	59.1	42.3	22.7	42.2	50.8	43.1M	294.2G
SENet-50 [26]	40.8	61.9	44.2	25.4	44.9	52.0	45.6M	298.3G
ECANet-50 [63]	40.6	61.5	44.3	26.8	44.7	51.2	43.1 M	298.3G
SCNet-50 [42]	42.2	63.4	45.7	26.7	**46.1**	54.3	43.1M	295.5G
SGNet-50	**42.4**	**63.7**	**46.2**	**26.9**	**45.9**	**54.2**	40.9M	294.7G
ResNeXt-50 [72]	37.8	58.4	41.0	22.3	41.2	49.6	43.1M	294.3G
SENeXt-50 [26]	41.4	62.9	44.7	26.7	45.1	52.9	45.6M	301.6G
SGNeXt-50	**42.8**	**64.0**	**46.7**	**27.5**	**46.4**	**55.4**	44.0M	300.1G
ResNet-101 [20]	40.4	60.8	44.0	24.6	44.5	51.7	62.1M	361.0G
SENet-101 [26]	43.1	64.3	47.0	26.3	47.5	54.8	66.8M	365.2G
ECANet-101 [63]	42.8	64.0	46.6	26.7	46.7	55.1	62.1M	365.2G
SCNet-101 [42]	43.6	64.6	47.7	27.1	47.6	**56.6**	62.1M	354.0G
SGNet-101	**43.8**	**65.3**	**47.9**	**28.3**	**47.8**	55.6	57.9M	352.7G
ResNeXt-101 [72]	39.6	59.8	43.2	22.8	43.3	51.3	62.6M	362.9G
SENeXt-101 [26]	43.5	65.0	47.3	26.2	47.4	56.0	67.4M	370.2G
SGNeXt-101	**44.4**	**66.0**	**48.5**	**27.4**	**48.3**	**56.7**	64.2M	365.4G

4.4 Instance segmentation

4.4.1 Instance segmentation results

In addition to objection detection, we apply our scaled gated convolution to Mask R-CNN [17] and share the same settings as aforementioned in Section 4.2.1. As shown in Table 8, our SGNet-50 based Mask R-CNN outer-performs ResNet-50 based Mask R-CNN by 3.2% (34.9% vs. **38.1**%), while SGNeXt-50 version outer-performs ResNeXt-50 one by 4.2% (34.5% vs. **38.7**%). For deeper configurations, our SGNet-101 version brings 2.4% absolute improvement compared to ResNet-101 based model, and SGNeXt-101 version introduce additional 3.7% AP increase compared to ResNeXt-101 based model. As can be seen, our scaled gated convolution can boost the performance of instance segmentation. Compared to SCNet [42] and ECANet [63], our approach also achieves comparable or better performance using a less computational budget. We also observe a slight performance gap in terms of AP_M and AP_L compared to SCNet-50 based Mask R-CNN. It will be our future work to further improve the capability of modeling large-scale instances.

4.5 Panoptic segmentation

In order to evaluate the capabilities to generalize on dense mask prediction task of our scaled gated convolution, we adopt SGNet as the backbone network for Panoptic FPN [33] and compare it with several modern counterparts [20, 26, 42, 63] using the same code base.

4.5.1 Experimental settings

Following the experimental settings in previous work [33], we utilize COCO-2017 [41] data splits with 118k images for training, 5k images for validation with 80 thing classes

Table 8 Mask R-CNN [17] based instance segmentation results

Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params	Flops
ResNet-50 [20]	34.9	56.1	37.3	17.1	37.5	50.3	45.7M	326.6G
SENet-50 [26]	37.5	59.5	40.0	19.1	40.4	56.7	48.2M	324.1G
ECANet-50 [63]	36.8	58.5	39.2	18.5	39.5	52.4	45.7M	324.5G
SCNet-50 [42]	**38.3**	60.6	**40.9**	19.7	**41.2**	**54.0**	45.7M	321.0G
SGNet-50	38.0	**60.7**	40.5	**19.9**	40.7	53.8	43.5M	296.9G
ResNeXt-50 [72]	34.5	55.7	36.7	16.7	36.8	49.4	45.8M	321.5G
SENeXt-50 [26]	37.5	59.6	40.0	19.2	40.0	53.8	48.3M	326.9G
SGNeXt-50	**38.6**	**61.6**	**41.0**	**20.2**	**41.4**	**54.9**	46.6M	322.8G
ResNet-101 [20]	37.2	58.6	40.0	18.6	40.0	53.3	64.7M	386.5G
SENet-101 [26]	39.2	61.4	42.2	20.5	42.5	54.9	69.5M	387.4G
ECANet-101 [63]	38.7	60.9	41.3	20.0	42.1	54.4	64.7M	389.0G
SCNet-101 [42]	39.4	61.9	42.0	20.3	42.8	56.2	64.7M	377.2G
SGNet-101	**39.6**	**62.6**	**42.7**	**21.0**	**42.5**	**56.7**	60.6M	352.7G
ResNeXt-101 [72]	36.3	57.4	38.8	17.3	39.1	52.5	65.3M	390.2G
SENeXt-101 [26]	39.7	62.7	42.2	**21.0**	42.9	55.8	70.0M	391.3G
SGNeXt-101	**40.0**	**63.0**	**42.8**	**20.9**	**43.4**	**56.9**	66.8M	365.4G

for instance segmentation. We also use COCO-2017 [41] stuff data including 40k training images and 5k validation images with 92 stuff classes. The panoptic segmentation is trained by all images containing 80 thing and 53 stuff classes as in [33]. For fair comparison, all models are trained for 90000 iterations using the same code base, and the scale jitter is also adopted as described in [33]. To evaluate the performance of both panoptic segmentation and semantic segmentation, we report different metrics for these tasks. Specifically, the mIoU,fwIoU,mACC and pACC are reported for semantic segmentation, and PQ, SQ, and RQ related metrics are reported for panoptic segmentation, respectively.

4.5.2 Panoptic Segmentation Results

The panoptic segmentation results are listed in Table 9. As can be seen, our proposed scaled gated convolution achieves comparable or better panoptic segmentation results compared to both vanilla architectures [20, 72] and advanced attentive architectures. Compared to ResNet-50 [20] based Panoptic FPN, our SGNet achieves 3.0% performance gains (39.4% vs. **42.4**%). For deep models, our SGNet boosts the performance of ResNet-101 [20] from 41.6% to **44.3**%. Furthermore, our SGNet also achieves superior performance compared to SENet [26] and other attentive counterparts [42, 63], as shown in Table 9. Thus, our proposed scaled gated convolution can generalize on dense prediction tasks.

4.6 Keypoint Detection

To evaluate the ability to generalize on keypoint detection tasks, we also apply our scaled gated convolution to the human keypoint-detection-based pose estimation pipeline and report the OKS-based mAP on the COCO-2017 validation set.

4.6.1 Experimental settings

We strictly follow the default settings in [70], the initial learning rate is set to 0.001 and divided by 10 after 90 and 120 epochs. For fair comparison, all models are trained with batch size 32 for 140 epochs using Adam optimizer [32]. A Faster R-CNN detector with 56.4 mAP detection results of "person" category is used during inference as done in [70]. We consider input sizes of 256×192 and 384×288 as in [70].

4.6.2 Keypoint detection results

The keypoint detection results are shown in Table 10. We omit the complexity metrics in Table 10 since all the keypoint detection results are based on the same Faster R-CNN [52] code base whose complexity has been discussed in Section 4.3.1, and the running speed of our SGNet backbones has been discussed in Section 4.1.2. As can be seen, our proposed scaled gated convolution outer-performs previous work [42, 70] in terms of OKS-based AP. Specifically, given 256×192 input images, our SGNet-50 outer-performs ResNet-50 and SCNet-50 by 2.9% and 1.2%, respectively. Our SGNet-101 outer-performs ResNet-101 and SCNet-101 by 3.2% and 2.0%, respectively. Similar phenomena can be observed when larger images are given, as shown in Table 10. However, there is a performance gap between SCNet [42] and our SGNet in terms of AP_L and AP_M, which indicates that SGNet and SCNet complement each other. More specifically, SCNet might be helpful for large-scale keypoint detection, while our SGNet is complementary to SCNet in terms of AP_{50},

Table 9 Panoptic FPN [33] based panoptic segmentation results using the same code base

Backbone	PQ	SQ	RQ	PQ_{th}	SQ_{th}	RQ_{th}	PQ_{st}	SQ_{st}	RQ_{st}	mIoU	fwIoU	mAcc	pAcc	Params	Flops
ResNet-50 [20]	39.4	77.8	48.3	46.4	81.0	55.8	28.9	73.0	37.0	41.1	68.2	52.7	79.9	47.3M	354.7G
SENet-50 [26]	41.1	78.3	50.2	48.1	81.1	57.8	30.5	74.0	38.7	42.6	69.3	54.2	80.9	49.9M	352.1G
ECANet-50 [63]	40.7	77.9	49.9	47.7	81.7	57.6	30.1	72.1	38.3	42.3	69.3	54.1	80.8	47.3M	352.6G
SCNet-50 [42]	41.8	78.1	51.0	41.5	81.3	58.2	31.7	73.3	40.2	43.8	70.0	55.7	81.3	47.3M	349.2G
SGNet-50	**42.4**	**79.4**	**51.7**	**49.4**	**82.2**	**59.3**	**31.9**	**75.2**	**40.2**	**44.3**	**70.5**	**56.4**	**81.7**	45.1M	346.6G
ResNeXt-50 [72]	40.5	78.1	49.5	47.9	81.0	57.7	29.3	73.8	37.3	41.7	68.6	53.7	80.3	47.4M	356.4G
SENeXt-50 [26]	41.8	78.8	51.2	48.8	81.8	58.8	31.3	74.2	39.7	43.5	69.7	55.3	81.1	49.9M	353.7G
SGNeXt-50	**43.2**	**79.4**	**52.6**	**49.9**	**82.3**	**59.9**	**33.0**	**75.1**	**41.6**	**45.0**	**70.8**	**57.3**	**81.9**	48.2M	350.7G
ResNet-101 [20]	41.6	78.4	50.7	48.6	81.3	58.3	31.0	74.0	39.3	43.1	69.6	54.8	81.0	66.3M	417.9G
SENet-101 [26]	42.9	78.6	52.2	50.1	81.6	59.9	32.1	74.1	40.7	43.9	70.3	56.1	81.5	71.1M	416.3G
ECANet-101 [63]	40.9	77.9	49.9	48.4	81.5	57.9	29.7	72.6	37.8	41.6	68.6	54.6	80.2	66.3M	417.8G
SCNet-101 [42]	42.8	79.2	51.8	49.4	81.7	60.0	32.8	75.4	41.1	44.8	70.7	56.8	81.8	66.3M	405.3G
SGNet-101	**44.3**	**79.9**	**53.8**	**51.2**	**82.6**	**61.3**	**33.9**	**75.9**	**42.4**	**45.5**	**71.3**	**58.0**	**82.2**	62.2M	403.2G
ResNeXt-101 [72]	42.0	78.7	51.1	49.1	82.3	58.8	31.2	73.2	39.6	43.2	69.3	55.6	80.9	66.9M	421.7G
SENeXt-101 [26]	43.5	79.2	52.9	50.6	82.4	60.6	32.7	74.5	41.4	44.8	70.5	57.4	81.6	71.7M	419.8G
SGNeXt-101	**44.7**	**80.3**	**54.3**	**51.4**	**82.3**	**61.6**	**34.6**	**77.3**	**43.2**	**46.3**	**71.7**	**59.1**	**82.4**	68.5M	414.4G

Table 10 Pose estimation results based on the same code base [70]

Backbone	Input	AP	AP_{50}	AP_{75}	AP_M	AP_L
ResNet-50 [70]	256×192	70.4	88.6	78.3	67.1	77.2
SCNet-50 [42]		72.1	89.4	79.8	69.0	**78.7**
SGNet-50		**73.4**	**92.4**	**81.4**	**70.4**	78.2
ResNet-50 [70]	384×288	72.2	89.3	78.9	68.1	79.7
SCNet-50 [42]		74.4	89.7	81.4	70.7	**81.7**
SGNet-50		**75.8**	**92.5**	**82.7**	**72.5**	80.7
ResNet-101 [70]	256×192	71.4	89.3	79.3	68.1	78.1
SCNet-101 [42]		72.6	89.4	80.4	69.4	**79.4**
SGNet-101		**74.6**	**93.5**	**82.3**	**71.7**	78.8
ResNet-101 [70]	384×288	73.6	89.6	80.3	69.9	81.1
SCNet-101 [42]		74.8	89.6	81.8	71.2	**81.9**
SGNet-101		**76.4**	**93.5**	**83.5**	**73.1**	81.0

AP_{75} and AP_M. Thus, we believe combining SGNet with other approaches might boost the performance of models.

5 Conclusion

We propose a lightweight scaled gated convolution that introduces scaled heterogeneous gating to generate powerful features and reduce redundancy and can be plugged into modern architectures in a plug-and-play manner. The gating mechanisms consist of gating transformation, gating activation, and post fusion. Experiments on large-scale datasets verify the effectiveness of our scaled gated convolution, and it can also be applied to downstream tasks to boost performance. We hope this work might inspire the study of efficient convolution design in the future.

Acknowledgment This work is supported in part by the National Key R&D Program of China under Grant [2020AAA0109602]; by the National Natural Science Foundation of China under Grant [61573273]; by the Key Research and Development Program of Shaanxi, China under Grant [2021GY-025].

Declarations

Conflict of Interest The authors wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

1. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: Real-time instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
2. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact++: Better real-time instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
3. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the International Conference on Computer Vision (2019)

4. Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., Yan, Y.: Blendmask: Top-down meets bottom-up for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
5. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: Proceedings of the International Conference on Learning Representations (2015)
6. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
7. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2019)
8. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic relu. In: Proceedings of the European Conference Computer Vision (ECCV) (2020)
9. Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J.: Dual path networks. In: Proceedings of the Neural Information Processing Systems (2017)
10. Clevert, D.-A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv:1511.07289 (2015)
11. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp. 764–773 (2017)
12. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
13. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448 (2015)
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587 (2014)
15. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv:1706.02677 (2017)
16. Greff, K., Srivastava, R.K., Schmidhuber, J.: Highway and residual networks learn unrolled iterative estimation. arXiv: Neural and Evolutionary Computing (2016)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.s.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034 (2015)
19. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell **37**(9), 1904–1916 (2015)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
21. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Proceedings of the European conference on computer vision, pp. 630–645 (2016)
22. Hou, Q., Zhang, L., Cheng, M.-M., Feng, J.: Strip Pooling: Rethinking spatial pooling for scene parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
23. Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)
24. 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. arXiv:1704.04861 (2017)
25. Hu, J., Shen, L., Albanie, S., Sun, G., Vedaldi, A.: Gather-excite: Exploiting feature context in convolutional neural networks. In: Proceedings of the Neural Information Processing Systems, pp. 9401–9411 (2018)
26. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
27. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269 (2017)
28. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6409–6418 (2019)

29. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the International Conference on Computer Vision (2019)
30. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
31. Jia, D., Wei, D., Richard, S., Li, L.-J., Kai, L., Li, F.-F.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255 (2009)
32. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference for Learning Representations (ICLR) (2015)
33. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6399–6408 (2019)
34. Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P.: Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
35. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Neural Inf. Process. Syst., 1097–1105 (2012)
36. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–519 (2019)
37. Li, Y., Lin, S., Chen, Y., Li, Z., Zhang, X., Wang, X., Sun, J.: Learning dynamic routing for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
38. Lin, M., Chen, Q., Yan, S.: Network in network. In: Proceedings of the International Conference on Learning Representations (ICLR) (2014)
39. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125 (2017)
40. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988 (2017)
41. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, pp. 740–755, Springer (2014)
42. Liu, J.-J., Hou, Q., Cheng, M.-M., Wang, C., Feng, J.: Improving convolutional networks with self-calibrated convolutions. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2020)
43. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440 (2015)
44. Lu, H., Yang, R., Deng, Z., Zhang, Y., Gao, G., Lan, R.: Chinese image captioning via fuzzy attention-based densenet-bilstm. ACM Trans. Multimedia Comput. Commun. Appl. (2021)
45. Lu, H., Zhang, M., Xu, X., Li, Y., Shen, H.T.: Deep fuzzy hashing network for efficient image retrieval. IEEE Trans. Fuzzy Syst. (1), 166–176 (2021)
46. Ma, N., Zhang, X., Huang, J., Sun, J.: Weightnet: Revisiting the design space of weight networks. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
47. Ma, N., Zhang, X., Zheng, H., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 116–131 (2018)
48. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the International Conference on Machine Learning (ICML) (2010)
49. Nakayama, Y., Lu, H., Li, Y., Kamiya, T.: Widesegnext: Semantic image segmentation using wide residual network and next dilated unit. IEEE Sens. J., 11427–11434 (2021)
50. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10428–10436 (2020)
51. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788 (2016)
52. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015)
53. 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, pp. 4510–4520 (2018)

54. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp. 618–626 (2017)
55. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
56. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
57. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the National Conference on Artificial Intelligence, pp. 4278–4284 (2016)
58. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015)
59. 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 (CVPR), pp. 2818–2826 (2016)
60. Tan, M., Pang, R., Le, QV.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10778–10787 (2020)
61. Tan, M., Le, QV.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML) (2019)
62. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(11) (2008)
63. 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 Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
64. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
65. Wang, X., Yu, F., Dou, Z., Darrell, T., Gonzalez, J.E.: Skipnet: Learning dynamic routing in convolutional networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 420–436 (2018)
66. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L.: Solo: Segmenting objects by locations. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
67. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic and fast instance segmentation. In: Proceedings of the Advances in Neural Information Processing Systems (2020)
68. Woo, S., Park, J., Lee, J., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
69. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
70. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 472–487 (2018)
71. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
72. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
73. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
74. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: Conditionally parameterized convolutions for efficient inference. In: Proceedings of the Neural Information Processing Systems, pp. 1305–1316 (2019)
75. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC), pp. 87.1–87.12 (2016)
76. Zhang, F., Zhu, X., Dai, H., Ye, M., Zhu, C.: Distribution-aware coordinate representation for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
77. 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, pp. 6848–6856 (2018)

78. Zhao, M., Zhong, S., Fu, X., Tang, B., Dong, S., Pecht, M.: Deep residual networks with adaptively parametric rectifier linear units for fault diagnosis. IEEE Transactions on Industrial Electronics (2020)
79. Zhou, Q., Yang, W., Gao, G., Ou, W., Lu, H., Chen, J., Latecki, L.J.: Multi-scale deep context convolutional neural networks for semantic segmentation. World Wide Web, 555–570 (2019)
80. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9308–9316 (2019)