Improving Convolutional Networks with Self-Calibrated Convolutions

Jiang-Jiang Liu[1*] Qibin Hou[2*] Ming-Ming Cheng[1] Changhu Wang[3] Jiashi Feng[2]
[1]CS, Nankai University [2]NUS [3]ByteDance AI Lab
https://mmcheng.net/scconv/

Abstract

Recent advances on CNNs are mostly devoted to designing more complex architectures to enhance their representation learning capacity. In this paper, we consider improving the basic convolutional feature transformation process of CNNs without tuning the model architectures. To this end, we present a novel self-calibrated convolution that explicitly expands fields-of-view of each convolutional layer through internal communications and hence enriches the output features. In particular, unlike the standard convolutions that fuse spatial and channel-wise information using small kernels (e.g., 3×3), our self-calibrated convolution adaptively builds long-range spatial and inter-channel dependencies around each spatial location through a novel self-calibration operation. Thus, it can help CNNs generate more discriminative representations by explicitly incorporating richer information. Our self-calibrated convolution design is simple and generic, and can be easily applied to augment standard convolutional layers without introducing extra parameters and complexity. Extensive experiments demonstrate that when applying our self-calibrated convolution into different backbones, the baseline models can be significantly improved in a variety of vision tasks, including image recognition, object detection, instance segmentation, and keypoint detection, with no need to change network architectures. We hope this work could provide future research with a promising way of designing novel convolutional feature transformation for improving convolutional networks. Code is available on the project page.

1. Introduction

Deep neural networks trained on large-scale image classification datasets (*e.g.,* ImageNet [30]) are usually adopted as backbones to extract strong representative features for down-stream tasks, such as object detection [23, 29, 2, 8], segmentation [45, 11], and human keypoint detection [11, 39]. A good classification network often has strong feature transformation capability and therefore provides powerful representations to benefit down-stream tasks [20, 10, 27]. Hence, it is highly desired to enhance the feature transformation capability of convolutional networks.

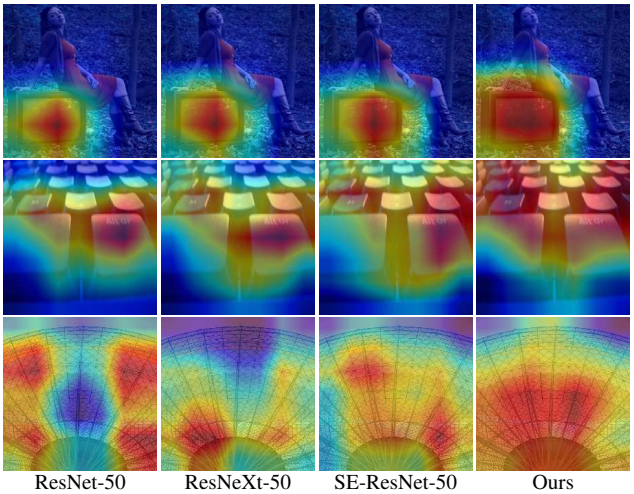

Figure 1. Visualizations of feature activation maps learned by different networks through Grad-CAM [31]. All the networks are trained on ImageNet [30]. Our results are obtained from ResNet-50 with the proposed self-calibrated convolution. From the activation maps, one can observe residual networks [12, 40] with conventional (grouped) convolutions and even SE blocks [16] fail to capture the whole discriminative regions, due to limited receptive fields of their convolution layers. In contrast, calibrated-convolutions help our model well capture the whole discriminative regions.

In the literature, an effective way to generate rich representations is using powerful hand-designed network architectures, such as residual networks (ResNets) [12] as well as their diverse variants [40, 43, 34, 7] or designing networks based on AutoML techniques [47, 26]. Recently, some methods attempt to do so by incorporating either attention mechanisms [38, 48, 16, 15] or non-local blocks [37, 3] into mature networks to model the interdependencies among spatial locations or channels or both. The common idea behind the above methods is focused on adjusting the network architectures for producing rich fea-

*Authors contributed equally.

ture representations, which needs too much human labors.

In this paper, rather than designing complex network architectures to strengthen feature representations, we introduce *self-calibrated convolution* as an efficient way to help convolutional networks learn discriminative representations by augmenting the basic convolutional transformation per layer. Similar to grouped convolutions, it separates the convolutional filters of a specific layer into multiple portions but unevenly, the filters within each portion are leveraged in a heterogeneous way. Specifically, instead of performing all the convolutions over the input in the original space homogeneously, Self-calibrated convolutions transform the inputs to low-dimensional embeddings through down-sampling at first. The low-dimensional embeddings transformed by one filter portion are adopted to calibrate the convolutional transformations of the filters within another portion. Benefiting from such heterogeneous convolutions and between-filter communication, the receptive field for each spatial location can be effectively enlarged.

As an augmented version of the standard convolution, our self-calibrated convolution offers two advantages. First, it enables each spatial location to adaptively encode informative context from a long-range region, breaking the tradition of convolution operating within small regions (*e.g.,* 3×3). This makes the feature representations produced by our self-calibrated convolution more discriminative. In Figure 1, we visualize the feature activation maps produced by ResNets with different types of convolutions [12, 40]. As can be seen, ResNet with self-calibrated convolutions can more accurately and integrally locate the target objects. Second, the proposed self-calibrated convolution is generic and can be easily applied to standard convolutional layers without introducing any parameters and complexity overhead or changing the hyper-parameters.

To demonstrate the effectiveness of the proposed self-calibrated convolution, we first apply it to the large-scale image classification problem. We take the residual network [12] and its variants [40, 16] as baselines, which get large improvements in top-1 accuracy with comparable model parameters and computational capacity. In addition to image classification, we also conduct extensive experiments to demonstrate the generalization capability of the proposed self-calibrated convolution in several vision applications, including object detection, instance segmentation, and keypoint detection. Experiments show that the baseline results can be greatly improved by using the proposed self-calibrated convolutions for all three tasks.

2. Related Work

In this section, we briefly review the recent representative work on architecture design and long-range dependency building of convolutional networks.

Architecture Design: In recent years, remarkable progress has been made in the field of novel architecture design [33, 35, 32, 44]. As an early work, VGGNet [33] builds deeper networks using convolutional filters with smaller kernel size (3×3) compared to AlexNet [19], yielding better performance while using fewer parameters. ResNets [12, 13] improve the sequential structure by introducing residual connections and using batch normalization [18], making it possible to build very deep networks. ResNeXt [40] and Wide ResNet [43] extend ResNet by grouping 3×3 convolutional layers or increasing their widths. GoogLeNet [35] and Inceptions [36, 34] utilize carefully designed Inception modules with multiple parallel paths of sets of specialized filters (3×3, *etc.*) for feature transformations. NASNet [48] learns to construct model architectures by exploring a predefined search space, enabling transferability. DenseNet [17] and DLA [42] aggregate features through complicated bottom-up skip connections. Dual Path Networks (DPNs) [7] exploit both residual and dense connections to build strong feature representations. SENet [16] introduces a squeeze-and-excitation operation to explicitly model the interdependencies between channels.

Long-Range Dependency Modeling: Building long-range dependencies is helpful in most computer vision tasks. One of the successful examples is the SENet [16], which adopts Squeeze-and-Excitation blocks to build interdependencies among the channel dimensions. Later work, like GENet [15], CBAM [38], GCNet [3], GALA [25], AA [1], and NLNet [37] further extend this idea by introducing spatial attention mechanisms or designing advanced attention blocks. Another way to model long-range dependency is to exploit spatial pooling or convolutional operators with large kernel windows. Some typical examples like PSPNet [45] adopt multiple spatial pooling operators with different sizes to capture multi-scale context. There are also many work [28, 14, 41, 5, 22] that leverage large convolutional kernels or dilated convolutions for long-range context aggregation. Our work is also different from Octave convolution [6], which aims at reducing spatial redundancy and computation cost.

Different from all above-mentioned approaches that focus on tuning network architectures or adding additional hand-designed blocks to improve convolutional networks, our approach considers more efficiently exploiting the convolutional filters in convolutional layers and designing powerful feature transformations to generate more expressive feature representations.

3. Method

A conventional 2D convolutional layer $\mathcal{F}$ is associated with a group of filter sets $\mathbf{K} = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_{\hat{C}}]$, where $\mathbf{k}_i$ denotes the i-th set of filters with size C, and transforms

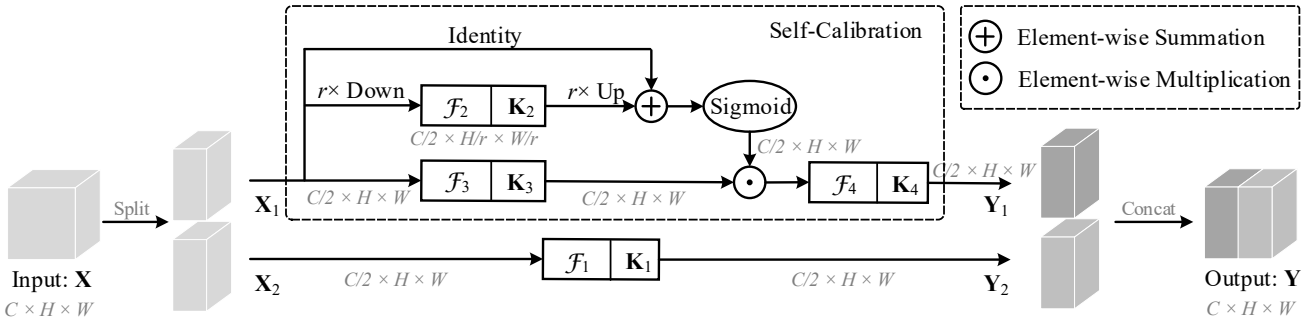

Figure 2. Schematic illustration of the proposed self-calibrated convolutions. As can be seen, in self-calibrated convolutions, the original filters are separated into four portions, each of which is in charge of a different functionality. This makes self-calibrated convolutions quiet different from traditional convolutions or grouped convolutions that are performed in a homogeneous way. More details about the self-calibration operation can be found in Sec. 3.1.

an input $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_C] \in \mathbb{R}^{C\times H\times W}$ to an output $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{\hat{C}}] \in \mathbb{R}^{\hat{C}\times \hat{H}\times \hat{W}}$. Note that we omit the spatial size of the filters and the bias term for notational convenience. Given the above notations, the output feature map at channel i can be written as

$$\mathbf{y}_i = \mathbf{k}_i * \mathbf{X} = \sum_{j=1}^{C} \mathbf{k}_i^j * \mathbf{x}_j, \tag{1}$$

where '*' denotes convolution and $\mathbf{k}_i = [\mathbf{k}_i^1, \mathbf{k}_i^2, \ldots, \mathbf{k}_i^C]$. As can be seen above, each output feature map is computed by summation through all channels and all of them are produced uniformly by repeating Eqn. 1 multiple times. In this way, the convolutional filters can learn similar patterns. Moreover, the fields-of-view for each spatial location in the convolutional feature transformation is mainly controlled by the predefined kernel size and networks composed of a stack of such convolutional layers are also short of large receptive fields to capture enough high-level semantics [46, 45]. Both above shortcomings may lead to feature maps that are less discriminative. To alleviate the above issues, we propose *self-calibrated convolution*, which is elaborated below.

3.1. Self-Calibrated Convolutions

In grouped convolutions, the feature transformation process is homogeneously and individually performed in multiple parallel branches and the outputs from each branch are concatenated as the final output. Similar to grouped convolutions, the proposed self-calibrated convolutions also split the learnable convolutional filters into multiple portions, yet *differently*, each portion of filters is not equally treated but responsible for a special functionality.

3.1.1 Overview

The workflow of the proposed design is illustrated in Figure 2. In our approach, we consider a simple case where the input channel number C is identical to the output channel number $\hat{C}$, *i.e.*, $\hat{C} = C$. Thus, in the following, we use C to replace $\hat{C}$ for notational convenience. Given a group of filter sets $\mathbf{K}$ with shape (C, C, k_h, k_w) where k_h and k_w are respectively the spatial height and width, we first uniformly separate it into four portions, each of which is in charge of a different functionality. Without loss of generality, suppose C can be divided by 2. After separation, we have four portions of filters denoted by $\{\mathbf{K}_i\}_{i=1}^4$, each of which is with shape $(\frac{C}{2}, \frac{C}{2}, k_h, k_w)$, respectively.

Given the four portions of filters, we then uniformly split the input $\mathbf{X}$ into two portions $\{\mathbf{X}_1, \mathbf{X}_2\}$, each of which is then sent into a special pathway for collecting different types of contextual information. In the first pathway, we utilize $\{\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3\}$ to perform the self-calibration operation upon $\mathbf{X}_1$, yielding $\mathbf{Y}_1$. In the second pathway, we perform a simple convolution operation: $\mathbf{Y}_2 = \mathcal{F}_1(\mathbf{X}_2) = \mathbf{X}_2 * \mathbf{K}_1$, which targets at retaining the original spatial context. Both the intermediate outputs $\{\mathbf{Y}_1, \mathbf{Y}_2\}$ are then concatenated together as the output $\mathbf{Y}$. In what follows, we detailedly describe how to perform the self-calibration operation in the first pathway.

3.1.2 Self-Calibration

To efficiently and effectively gather informative contextual information for each spatial location, we propose to conduct convolutional feature transformation in two different scale spaces: an original scale space in which feature maps share the same resolution with the input and a small latent space after down-sampling. The embeddings after transformation in the small latent space are used as references to guide the feature transformation process in the original feature space because of their large fields-of-view.

Self-Calibration: Given the input $\mathbf{X}_1$, we adopt the average pooling with filter size $r \times r$ and stride r as follows:

$$\mathbf{T}_1 = \text{AvgPool}_r(\mathbf{X}_1). \tag{2}$$

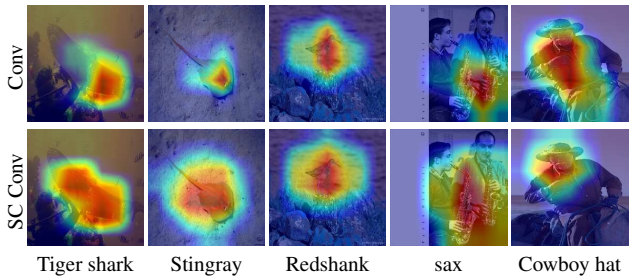

Figure 3. Visual comparisons of the intermediate feature maps produced by different settings of ResNet-50. The feature maps are selected from the 3×3 convolutional layer in the last building block. For the top row, we use the traditional convolutions while for the bottom row, we use the proposed self-calibrated convolutions (SC-Conv). It is obvious that ResNet-50 with self-calibrated convolutions can capture richer context information.

Feature transformations on $\mathbf{T}_1$ is performed based on $\mathbf{K}_2$:

$$\mathbf{X}'_1 = \mathrm{Up}(\mathcal{F}_2(\mathbf{T}_1)) = \mathrm{Up}(\mathbf{T}_1 * \mathbf{K}_2), \tag{3}$$

where $\mathrm{Up}(\cdot)$ is a bilinear interpolation operator that maps the intermediate references from the small scale space to the original feature space. Now, the calibration operation can be formulated as follows:

$$\mathbf{Y}'_1 = \mathcal{F}_3(\mathbf{X}_1) \cdot \sigma(\mathbf{X}_1 + \mathbf{X}'_1), \tag{4}$$

where $\mathcal{F}_3(\mathbf{X}_1) = \mathbf{X}_1 * \mathbf{K}_3$, σ is the sigmoid function, and '$\cdot$' denotes element-wise multiplication. As shown in Eqn. 4, we use $\mathbf{X}'_1$ as residuals to form the weights for calibration, which is found beneficial. The final output after calibration can be written as follows:

$$\mathbf{Y}_1 = \mathcal{F}_4(\mathbf{Y}'_1) = \mathbf{Y}'_1 * \mathbf{K}_4. \tag{5}$$

Advantages: The advantages of the proposed self-calibration operation are three-fold. First of all, compared to conventional convolutions, by employing the calibration operation as shown in Eqn. 4, each spatial location is allowed to not only adaptively consider its surrounding informative context as embeddings from the latent space functioning as scalars in the responses from the original scale space, but also model inter-channel dependencies. Thus, the fields-of-view for convolutional layer with self-calibration can be effectively enlarged. As shown in Figure 3, convolutional layers with self-calibration encode larger but more accurate discriminative regions. Second, instead of collecting global context, the self-calibration operation only considers the context around each spatial location, avoiding some contaminating information from irrelevant regions to some extent. As can be seen in the right two columns of Figure 6, convolutions with self-calibration can accurately locate the target objects when visualizing the final score layer. Third, the self-calibration operation encodes multi-scale information, which is highly desired by object detection related tasks. We will give more experimental analysis in Sec. 4.

3.2. Instantiations

To demonstrate the performance of the proposed self-calibrated convolutions, we take several variants of the residual networks [12, 40, 16] as exemplars. Both 50- and 101-layer bottleneck structures are considered. For simplicity, we only replace the convolutional operation in the 3×3 convolutional layer in each building block with our self-calibrated convolutions and keep all relevant hyper-parameters unchanged. By default, the down-sampling rate r in self-calibrated convolutions is set to 4.

Relation to Grouped Convolutions: Grouped convolutions adopt the split-transform-merge strategy, in which individual convolutional transformations are conducted homogeneously in multiple parallel branches [40] or in a hierarchical way [9]. Unlike grouped convolutions, our self-calibrated convolutions can exploit different portions of convolutional filters in a heterogeneous way. Thus, each spatial location during transformation can fuse information from two different spatial scale spaces through the self-calibration operation, which largely increases the fields-of-view when applied to convolutional layers and hence results in more discriminative feature representations.

Relation to Attention-Based Modules: Our work is also quiet different from the existing methods relying on add-on attention blocks, such as the SE block [16], GE [15] block, or the CBAM [38]. Those methods require additional learnable parameters, while our self-calibrated convolutions internally change the way of exploiting convolutional filters of convolutional layers, and hence require no additional learnable parameters. Moreover, though the GE block [15] encodes spatial information in a lower-dimension space as we do, it does not explicitly preserve the spatial information from the original scale space. In the following experiment section, we will show without any extra learnable parameters, our self-calibrated convolutions can yield significant improvements over baselines and other attention-based approaches on image classification. Furthermore, our self-calibrated convolutions are complementary to attention and thus can also benefit from the add-on attention modules.

4. Experiments

4.1. Implementation Details

We implement our approach using the publicly available PyTorch framework[1]. For fair comparison, we adopt the official classification framework to perform all classification experiments unless specially declared. We report re-

[1] https://pytorch.org

Network	Params	MAdds	FLOPs	Top-1	Top-5
50-layer					
ResNet [12]	25.6M	4.1G	8.2G	76.4	93.0
SCNet	25.6M	4.0G	7.9G	77.8	93.9
ResNeXt [40]	25.0M	4.3G	8.5G	77.4	93.4
ResNeXt 2x40d	25.4M	4.2G	8.3G	76.8	93.3
SCNeXt	25.0M	4.3G	8.5G	78.3	94.0
SE-ResNet[16]	28.1M	4.1G	8.2G	77.2	93.4
SE-SCNet	28.1M	4.0G	7.9G	78.2	93.9
101-layer					
ResNet [12]	44.5M	7.8G	15.7G	78.0	93.9
SCNet	44.6M	7.2G	14.4G	78.9	94.3
ResNeXt [40]	44.2M	8.0G	16.0G	78.5	94.2
SCNeXt	44.2M	8.0G	15.9G	79.2	94.4
SE-ResNet[16]	49.3M	7.9G	15.7G	78.4	94.2
SE-SCNet	49.3M	7.2G	14.4G	78.9	94.3

Table 1. Comparisons on ImageNet-1K dataset when the proposed structure is utilized in different classification frameworks. We report single-crop accuracy rates (%).

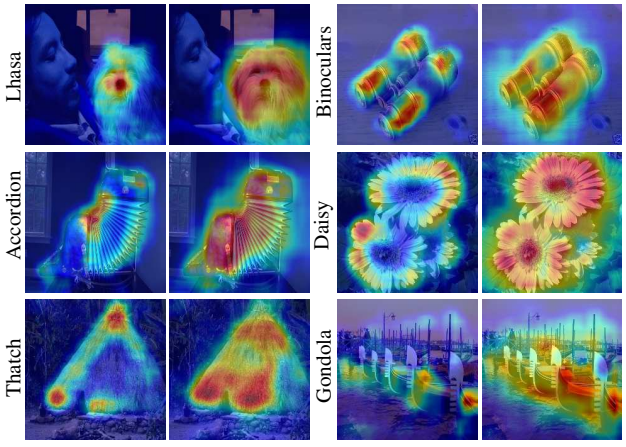

Figure 4. Visualizations of feature maps from the side outputs at `res3` of different networks (ResNet *v.s.* SCNet). We use 50-layer settings for both networks.

sults on the ImageNet dataset [30]. The size of input images is 224×224 which are randomly cropped from resized images as done in [40]. We use SGD to optimize all models. The weight decay and momentum are set to 0.0001 and 0.9, respectively. Four Tesla V100 GPUs are used and the mini-batch size is set to 256 (64 per GPU). By default, we train all models for 100 epochs with an initial learning rate 0.1, which is divided by 10 after every 30 epochs. In testing, we report the accuracy results on the single 224×224 center crop from an image with shorter side resized to 256 as in [40]. Note that models in all ablation comparisons share the same running environment and hyper-parameters except for the network structures themselves. All models in Table 1 are trained under the same strategy and tested under the same settings.

4.2. Results on ImageNet

We conduct ablation experiments to verify the importance of each component in our proposed architecture and compare with existing attention-based approaches on the ImageNet-1K classification dataset [30].

4.2.1 Ablation Analysis

Generalization Ability: To demonstrate the generalization ability of the proposed structure, we consider three widely used classification architectures as baselines, including ResNet [12], ResNeXt [40], and SE-ResNet [16]. The corresponding networks with self-calibrated convolutions are named as SCNet, SCNeXt, and SE-SCNet, respectively. Following the default version of ResNeXt [40] (32×4d), we set the bottleneck width to 4 in SCNeXt. We also adjust the cardinality of each group convolution according to our structure to ensure that the capacity of SCNeXt is close to ResNeXt. For SE-SCNet, we apply the SE module to SCNet in the same way as [16].

In Table 1, we show the results produced by both 50- and 101-layer versions of each model. Compared to the original ResNet-50 architecture, SCNet-50 has an improvement of 1.4% in accuracy (77.8% *vs.* 76.4%). Moreover, the improvement by SCNet-50 (1.4%) is also higher than that by ResNeXt-50 (1.0%) and SE-ResNet-50 (0.8%). This demonstrates that self-calibrated convolutions perform much better than increasing cardinality or introducing the SE module [16]. When the networks go deeper, a similar phenomenon can also be observed.

Another way to investigate the generalization ability of the proposed structure is to see its behaviors on other vision tasks as backbones, such as object detection and instance segmentation. We will give more experiment comparisons in the next subsection.

Self-Calibrated Convolution *v.s.* Vanilla Convolution: To further investigate the effectiveness of the proposed self-calibrated convolutions compared to the vanilla convolutions, we add side supervisions (auxiliary losses) as done in [21] to both ResNet-50 and SCNet-50 after one intermediate stage, namely `res3`. Results from side outputs can reflect how a network performs when the depth varies and how strong the feature representations at different levels are. The top-1 accuracy results from the side supervision at `res3` have been depicted in Figure 5. It is obvious that the side results from SCNet-50 are much better than those from ResNet-50. This phenomenon indirectly indicates that networks with the proposed self-calibrated convolutions can generate richer and more discriminative feature representations than the vanilla convolutions. To further demonstrate

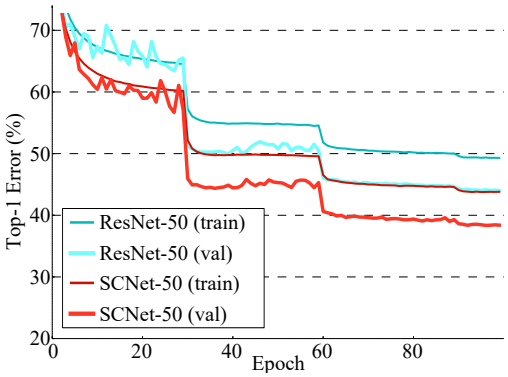

Figure 5. Auxiliary loss curves for both ResNet-50 and SCNet-50. We add auxiliary loss after `res3`. As can be seen, SCNet (red lines) works much better than ResNet (cyan lines). This demonstrates that self-calibrated convolutions work better for networks with lower depth.

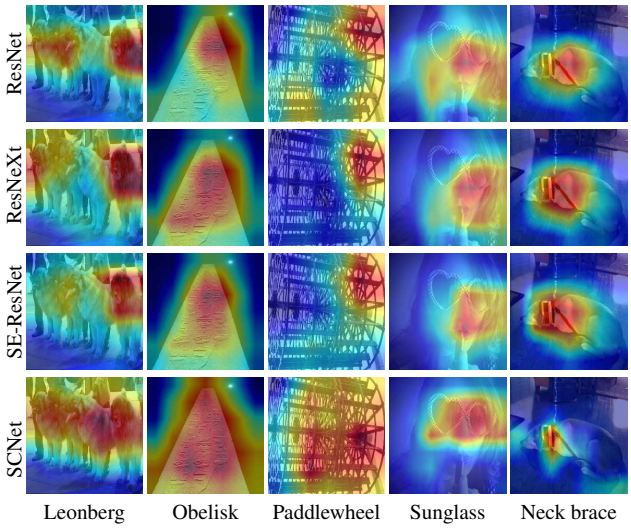

Figure 6. Visualization of attention maps generated by Grad-CAM [31]. It is obvious that our SCNet can more precisely locate the foreground objects than other networks no matter how large and what shape they are. This heavily relies on our self-calibration operation which benefits adaptively capturing rich context information. We use 50-layer settings for all networks.

this, we show some visualizations from the score layers of the side outputs in Figure 4. Apparently, SCNet can more precisely and integrally locate the target objects even at a lower depth of the network. In Sec. 4.3, we will give more demonstrations on this by applying both convolutions to different vision tasks.

Attention Comparisons: To show why the proposed self-calibrated convolution is helpful for classification networks, we adopt the Grad-CAM [31] as an attention extraction tool to visualize the attentions produced by ResNet-50, ResNeXt-50, SE-ResNet-50, and SCNet-50, as shown in Figure 6. It can be clearly seen that the attentions produced by SCNet-50 can more precisely locate the target objects and do not expand to the background areas too much. When the target objects are small, the attentions by our network are also better confined to the semantic regions compared to those produced by other three networks. This suggests that our self-calibrated convolution is helpful for discovering more integral target objects even though their sizes are small.

Design Choices: As demonstrated in Sec. 3.1, we introduce the down-sampling operation to achieve self-calibration, which has been proven useful for improving CNNs. Here, we investigate how the down-sampling rate in self-calibrated convolutions influences the classification performance. In Table 2, we show the performance with different down-sampling rates used in self-calibrated convolutions. As can be seen, when no down-sampling operation is adopted ($r = 1$), the result is already much better than the original ResNet-50 (77.38% *v.s.* 76.40%). As the down-sampling rate increases, better performance can be achieved. Specially, when the down-sampling rate is set to 4, we have a top-1 accuracy of 77.81%. Note that we do not use larger down-sampling rates as the resolution of the last residual blocks is already very small (e.g., 7×7). Furthermore, we find taking the feature maps at lower resolution (after $\mathcal{F}_2$) as residuals by adding an identity connection as shown in Figure 2 is also helpful for better performance. Discarding the extra identity connection leads to a decrease of performance to 77.48%.

Average Pooling *vs.* Max Pooling: In addition to the above design choices, we also investigate the influence of different pooling types on the performance. In our experiments, we attempt to replace all the average pooling operators in self-calibrated convolutions with the max pooling operators and see the performance difference. With all other configurations unchanged, as shown in Table 2, using the max pooling operator yields a performance decrease of about 0.3% in top-1 accuracy (77.81 *vs.* 77.53). We argue that this may be due to the fact that, unlike max pooling, average pooling builds connections among locations within the whole pooling window, which can better capture local contextual information.

Discussion: According to the above ablation experiments, introducing self-calibrated convolutions is helpful for classification networks, like ResNet and ResNeXt. However, note that exploring the optimal architecture setting is beyond the scope of this paper. This paper just provides a preliminary study about how to improve the vanilla convolutions. We encourage readers to further investigate more effective structures. In the next subsection, we will show how our approach behaves as pretrained backbones when applied to popular vision tasks.

Model	DS Rate (r)	Identity	Pooling	Top-1 Accuracy
ResNet	-	-	-	76.40%
ResNeXt	-	-	-	77.40%
SE-ResNet	-	-	AVG	77.20%
SCNet	1	✓	-	77.38%
SCNet	2	✓	AVG	77.48%
SCNet	4	✗	AVG	77.48%
SCNet	4	✓	MAX	77.53%
SCNet	4	✓	AVG	77.81%
SCNeXt	4	✓	AVG	78.30%

Table 2. Ablation experiments about the design choices of SCNet. 'Identity' refers to the corresponding component with the same name as in Figure 2. 'DS Rate' is the down-sampling rate in Eqn. 2. We also show results under two types of pooling operations: average pooling (AVG) and max pooling (MAX).

4.2.2 Comparisons with Attention-Based Approaches

Here, we benchmark the proposed SCNet against existing attention-based approaches, including CBAM [38], SENet [16], GALA [25], AA [1], and GE [15], on the ResNet-50 architecture. The comparison results can be found in Table 3. It can be easily found that most attention or non-local based approaches require additional learnable parameters to build their corresponding modules and then plug them into building blocks. Quite differently, our approach does not rely on any extra learnable parameters, but only heterogeneously exploits the convolutional filters. Out results are obviously better than those of all other approaches. It should also be mentioned that the proposed self-calibrated convolutions are also compatible with the above mentioned attention-based approaches. For example, when adding GE blocks to each building block of SCNet as done in [15], we can further gain another 0.5% boost in accuracy. This also indicates that our approach is different from this kind of add-on modules.

4.3. Applications

In this subsection, we investigate the generalization capability of the proposed approach by applying it to popular vision tasks as backbones, including object detection, instance segmentation, and human keypoint detection.

4.3.1 Object Detection

Network Settings: In the object detection task, we take the widely used Faster R-CNN architecture [29] with feature pyramid networks (FPNs) [23] as baselines. We adopt the widely used `mmdetection framework`[2] [4] to run all our experiments. As done in previous work [23, 11], we train each model using the union of 80k COCO train images and 35k images from the validation set (`trainval35k`) [24] and report results on the rest 5k validation images (`minival`).

[2]https://github.com/open-mmlab/mmdetection

Network	Params	MAdds	Top-1	Top-5
ResNet [12]	25.6M	4.1G	76.4	93.0
ResNeXt [40]	25.0M	4.3G	77.4	93.4
SE-ResNet [16]	28.1M	4.1G	77.2	93.4
ResNet + CBAM [38]	28.1M	4.1G	77.3	93.6
GCNet [3]	28.1M	4.1G	77.7	93.7
ResNet + GALA [25]	29.4M	4.1G	77.3	93.6
ResNet + AA [1]	28.1M	4.1G	77.7	93.6
ResNet + GE [15]†	31.2M	4.1G	78.0	93.6
SCNet	25.6M	4.0G	77.8	93.9
SCNet†	25.6M	4.0G	78.2	94.0
SE-SCNet	28.1M	4.0G	78.2	93.9
GE-SCNet	31.1M	4.0G	78.3	94.0

Table 3. Comparisons with prior attention-based approaches on the ImageNet-1K dataset. All approaches are based on the ResNet-50 baseline. We report single-crop accuracy rate (%) and show complexity comparisons as well. '†' means models trained with 300 epochs.

We set hyper-parameters strictly following the Faster R-CNN work [29] and its FPN version [23]. Images are all resized so that their shorter edges are with 800 pixels. We use 8 Tesla V100 GPUs to train each model and the mini-batch is set to 16, *i.e.,* , 2 images on each GPU. The initial learning rate is set to 0.02 and we use the $2\times$ training schedule to train each model. Weight decay and momentum are set to 0.0001 and 0.9, respectively. We report the results using the standard COCO metrics, including AP (averaged mean Average Precision over different IoU thresholds), $AP_{0.5}$, $AP_{0.75}$ and AP_S, AP_M, AP_L (AP at different scales). Both 50-layer and 101-layer backbones are adopted.

Detection Results: In the top part of Table 4, we show experimental results on object detection when different classification backbones are used. When taking Faster R-CNN [29] as an example, adopting ResNet-50-FPN as the backbone gives an AP score of 37.6 while replacing ResNet-50 with SCNet-50 yields a large improvement of 3.2 (40.8 *v.s.* 37.6). More interestingly, Faster R-CNN with SCNet-50 backbone performs even better than that with ResNeXt-50 (40.8 *v.s.* 38.2). This indicates the proposed way of leveraging convolutional filters is much more efficient than directly grouping the filters. This may be because the proposed self-calibrated convolutions contain the adaptive response calibration operation, which help more precisely locate the exact positions of target objects as shown in Figure 6. In addition, from Table 4, we can observe that using deeper backbones leads to a similar phenomenon as above (ResNet-101-FPN: 39.9 $\rightarrow$ SCNet-101-FPN: 42.0).

Backbone	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
Object Detection (Faster R-CNN)						
ResNet-50-FPN	37.6	59.4	40.4	21.9	41.2	48.4
SCNet-50-FPN	40.8	62.7	44.5	24.4	44.8	53.1
ResNeXt-50-FPN	38.2	60.1	41.4	22.2	41.7	49.2
SCNeXt-50-FPN	40.4	62.8	43.7	23.4	43.5	52.8
ResNet-101-FPN	39.9	61.2	43.5	23.5	43.9	51.7
SCNet-101-FPN	42.0	63.7	45.5	24.4	46.3	54.6
ResNeXt-101-FPN	40.5	62.1	44.2	23.2	44.4	52.9
SCNeXt-101-FPN	42.0	64.1	45.7	25.5	46.1	54.2
Instance Segmentation (Mask R-CNN)						
ResNet-50-FPN	35.0	56.5	37.4	18.3	38.2	48.3
SCNet-50-FPN	37.2	59.9	39.5	17.8	40.3	54.2
ResNeXt-50-FPN	35.5	57.6	37.6	18.6	38.7	48.7
SCNeXt-50-FPN	37.5	60.3	40.0	18.2	40.5	55.0
ResNet-101-FPN	36.7	58.6	39.3	19.3	40.3	50.9
SCNet-101-FPN	38.4	61.0	41.0	18.2	41.6	56.6
ResNeXt-101-FPN	37.3	59.5	39.8	19.9	40.6	51.2
SCNeXt-101-FPN	38.2	61.2	40.8	18.8	41.4	56.1

Table 4. Comparisons with state-of-the-art approaches on COCO `minival` dataset. All results are based on single-model test and the same hyper-parameters. For object detection, AP refers to box IoU while for instance segmentation AP refers to mask IoU.

4.3.2 Instance Segmentation

For instance segmentation, we use the same hyper-parameters and datasets as in Mask R-CNN [11] for a fair comparison. The results are based on the `mmdetection framework` [4] for all experiments performed in this part.

We compare the SCNet version Mask R-CNN to the ResNet version at the bottom of Table 4. Because we have introduced object detection results in details, here we only report the results using mask APs. As can be seen, the ResNet-50-FPN version and the ResNeXt-50-FPN version Mask R-CNNs have 35.0 and 35.5 mask APs, respectively. However, when taking SCNet into account, the corresponding results are respectively improved by 2.2 and 2.0 in mask AP. Similar results can also be observed when adopting deeper backbones. This suggests our self-calibrated convolutions are also helpful for instance segmentation.

4.3.3 Keypoint Detection

At last, we apply SCNet to human keypoint detection and report results on the COCO keypoint detection dataset [24]. We take the state-of-the-art method [39] as our baseline. We only replace the backbone ResNet in [39] with SCNet and all other train and test settings[3] are kept unchanged. We evaluate the results on the COCO `val2017` set using the standard OKS-based mAP, where OKS (object keypoints similarity) defines the similarity between different human poses. A Faster R-CNN object detector [29] with detection AP of 56.4 for the 'person' category on COCO `val2017` set is adopted for detection in the test phase as in [39].

[3]https://github.com/Microsoft/human-pose-estimation.pytorch

Backbone	Scale	AP	$AP_{.5}$	$AP_{.75}$	AP_m	AP_l
ResNet-50	256×192	70.6	88.9	78.2	67.2	77.4
SCNet-50	256×192	72.1	89.4	79.8	69.0	78.7
ResNet-50	384×288	71.9	89.2	78.6	67.7	79.6
SCNet-50	384×288	74.4	89.7	81.4	70.7	81.7
ResNet-101	256×192	71.6	88.9	79.3	68.5	78.2
SCNet-101	256×192	72.6	89.4	80.4	69.4	79.4
ResNet-101	384×288	73.9	89.6	80.5	70.3	81.1
SCNet-101	384×288	74.8	89.6	81.8	71.2	81.9

Table 5. Experiments on keypoint detection [24]. We report results on the COCO `val2017` set using the OKS-based mAP and take the state-of-the-art method [39] as our baseline. Two different input sizes (256×192 and 384×288) are considered as in [39].

Table 5 shows the comparisons. As can be seen, simply replacing ResNet-50 with SCNet-50 improves the AP score by 1.5% for 256×192 input size and 2.5% for 384×288 input size. These results demonstrate that introducing the proposed self-calibration operation in convolutional layers benefits human keypoint detection. When using deeper networks as backbones, we also have more than 1% performance gain in AP as shown in Table 5.

5. Conclusions and Future Work

This paper presents a new self-calibrated convolution, which is able to heterogeneously exploit the convolutional filters nested in a convolutional layer. To promote the filters to be of diverse patterns, we introduce the adaptive response calibration operation. The proposed self-calibrated convolutions can be easily embedded into modern classification networks. Experiments on large-scale image classification dataset demonstrate that building multi-scale feature representations in building blocks greatly improves the prediction accuracy. To investigate the generalization ability of our approach, we apply it to multiple popular vision tasks and find substantial improvements over the baseline models. We hope the thought of heterogeneously exploiting convolutional filters can provide the vision community a different perspective on network architecture design.

Acknowledgement. This research was partly supported by Major Project for New Generation of AI under Grant No. 2018AAA01004, NSFC (61620106008), the national youth talent support program, and Tianjin Natural Science Foundation (18ZXZNGX00110). Part of this work was done when Jiang-Jiang Liu interned at ByteDance AI Lab.

References

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. *arXiv preprint arXiv:1904.09925*, 2019. 2, 7

[2] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, 5(2):117–150, 2019. 1

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*, 2019. 1, 2, 7

[4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. mmdetection. https://github.com/open-mmlab/mmdetection, 2018. 7, 8

[5] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NeurIPS*, pages 8699–8710, 2018. 2

[6] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *ICCV*, pages 3435–3444, 2019. 2

[7] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4467–4475, 2017. 1, 2

[8] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*, pages 186–202, 2018. 1

[9] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*, pages 1–1, 2020. 4

[10] Shiming Ge, Xin Jin, Qiting Ye, Zhao Luo, and Qiang Li. Image editing by object-aware optimal boundary searching and mixed-domain composition. *Computational Visual Media*, 4(1):71–82, 2018. 1

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017. 1, 7, 8

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4, 5, 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016. 2

[14] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019. 2

[15] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS*, pages 9401–9411, 2018. 1, 2, 4, 7

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 1, 2, 4, 5, 7

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 2

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 2

[20] Thuc Trinh Le, Andrés Almansa, Yann Gousseau, and Simon Masnou. Object removal from complex videos using a few annotations. *Computational Visual Media*, 5(3):267–291, 2019. 1

[21] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570, 2015. 5

[22] Yi Li, Zhanghui Kuang, Yimin Chen, and Wayne Zhang. Data-driven neuron allocation for scale aggregation networks. In *CVPR*, pages 11526–11534, 2019. 2

[23] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 7

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 7, 8

[25] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. In *ICLR*, 2019. 2, 7

[26] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 1

[27] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, pages 3917–3926, 2019. 1

[28] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, pages 4353–4361, 2017. 2

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2016. 1, 7, 8

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 5

[31] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 1, 6

[32] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 2

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 1, 2

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 2

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 2

[37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 1, 2

[38] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. 1, 2, 4, 7

[39] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 1, 8

[40] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 1, 2, 4, 5, 7

[41] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[42] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412, 2018. 2

[43] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1, 2

[44] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *CVPR*, pages 718–726, 2017. 2

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2, 3

[46] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 3

[47] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 1

[48] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018. 1, 2