This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

SCSC: Spatial Cross-scale Convolution Module to Strengthen both CNNs and Transformers

Xijun Wang[1]*, Xiaojie Chu[2]*, Chunrui Han[2], Xiangyu Zhang[2]
[1]The University of Maryland, College Park, Maryland, USA. [2]MEGVII Technology, Beijing, China
xijun@umd.edu, {chuxiaojie, hanchunrui, zhangxiangyu}@megvii.com

Abstract

*This paper presents a module, Spatial Cross-scale Convolution (SCSC), which is verified to be effective in improving both CNNs and Transformers. Nowadays, CNNs and Transformers have been successful in a variety of tasks. Especially for Transformers, increasing works achieve state-of-the-art performance in the computer vision community. Therefore, researchers start to explore the mechanism of those architectures. Large receptive fields, sparse connections, weight sharing, and dynamic weight have been considered keys to designing effective base models [39, 24, 63, 44]. However, there are still some issues to be addressed: large dense kernels and self-attention are inefficient, and large receptive fields make it hard to capture local features. Inspired by the above analyses and to solve the mentioned problems, in this paper, we design a general module taking in these design keys to enhance both CNNs and Transformers. SCSC introduces an efficient spatial cross-scale encoder and spatial embed module to capture assorted features in one layer. On the face recognition task, FaceResNet with SCSC can improve **2.7%** with **68%** fewer FLOPs[1] and **79%** fewer parameters. On the ImageNet classification task, Swin Transformer with SCSC can achieve even better performance with **22%** fewer FLOPs, and ResNet with CSCS can improve **5.3%** with similar complexity. Furthermore, a traditional network (e.g., ResNet) embedded with SCSC can match Swin Transformer's performance.*

1. Introduction

Vision transformers have achieved impressive breakthroughs in a variety of tasks, including classification, object detection, segmentation, video action recognition, and *etc*. [19, 49, 39], making transformers promising backbones for vision applications. Transformers have a strong representation ability for their special designs, supporting a variety of data forms (tensor, set, sequence, graph, and *etc*.), and being robust to blocks and noise [12, 18, 57, 42]. As a result, more and more researchers are trying to figure out what makes transformers so powerful. Han et al. [24] and [63] analyze vision transformers from the respective of receptive fields, sparse connectivity, weight sharing, and dynamic weight, which has been considered as desired properties in model architecture design.

For receptive field, [19, 39, 17, 46] have shown that large kernels can efficiently introduce large receptive fields and can partially avoid the optimization problem caused by the increase of model depth. [46, 13, 8, 43, 6] has proved that large kernel (kernel size $\geq 5\times5$) applied to CNN can obtain competitive performance, especially for downstream tasks. However, the intensive computational cost of large dense kernels makes it hard to be widely used in practice. For Transformer, multi-head self-attention layer can simulate a convolutional layer by linear projection operations [12], to some extent, transformer acts like a global receptive field CNN network. But the high computational cost of self-attention still hinder transformers to be applied in practice.

To alleviate the intensive computational complexity, [39] and [24] have explored the sparse connectivity property, [39] proposed shifted windowing scheme by limiting self-attention computation to local windows. And [24] utilizes 7×7 depth-wise convolution[9, 28] to simulate the local window scheme. However, the efficiency and spatial modeling ability still can be improved. Moreover, the receptive field in one layer could be more diverse.

For weight sharing, both CNN and Swin-Transformer [39] have applied weight sharing mechanism to obtain computational efficiency. The difference is CNNs share weights across spatial dimension while Swin-Transformer shares weight across channel dimension.

For dynamic weight, there are many works [30, 32, 59, 3] illustrate its effectiveness on CNN. And transformers' self-attention structure uses dynamic weight for each input instance so that model capacity is increased.

To conclude, all the four properties listed above in various architectures have potential to improve performance or efficiency. Convolution is efficient because of its spa-

[1] The number of multiply-adds operations. *Equal contribution. This work was completed when Xijun Wang was an intern at MEGVII.

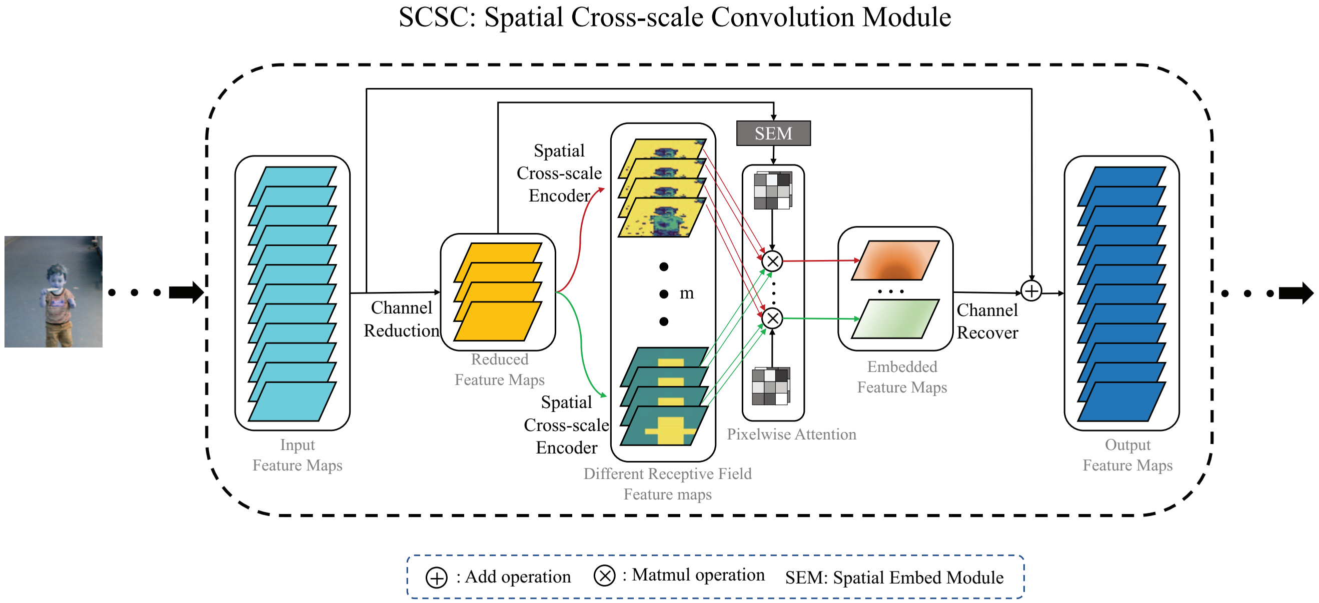

Figure 1. Illustration of Spatial Cross-scale Convolution Module (SCSC) with different receptive fields in one layer. First, we decrease the input feature maps' channel to get the channel reduced features maps, which can save the channel-wise convolution computational cost. Second, we apply m different Spatial encoder to get m different receptive fields feature maps. Third, we use Spatial Embed Module to merge the m feature maps to get the embedded feature maps. Finally, we recover the channel number as same as the input feature maps to get the output feature maps.

tial sharing pattern, and transformers have a large capacity because of their self-attention's large receptive field and dynamic scheme. Therefore, convolution and self-attention have complementary qualities, and a well-designed module that combines all desirable properties is possible.

To deal with all the above limitations and exploit complementary qualities of Transformer and CNN once for all, in this work, we propose a Spatial Cross-scale Convolution Module (SCSC), which can capture microscopic and macroscopic feature representation synchronously without intensive computational cost. As shown in Figure 1, for SCSC, different from the mainstream CNN's small kernel size or the Transformer's global receptive field, we design an intermediate expression for the receptive field and spatial modeling ability by using a wide range of kernel sizes from 3×3 to 13×13, depth-wise convolution is applied for its efficiency. Small kernels (e.g., 3×3) have advantage in modeling low-level and detailed information as shown in the upper red path in Figure 1, large kernels (e.g., 13×13) can handle the semantic dependence in a large receptive field as shown in the lower green path in Figure 1. Therefore, keeping both of them can obtain diverse spatial representations. Furthermore, we design an efficient spatial embed module to aggregate the different spatial representation features, which can capably integrate different levels of information. As a result, in our proposed SCSC, we can acquire different receptive fields in one layer, share weight in both spatial-wise and channel-wise, exploit depth-wise convolution to hold the sparse connectivity property, and apply the spatial embed module to bring dynamic connection across the channel dimension.

Moreover, we have evaluated the proposed SCSC in different tasks. On ImageNet classification, Swin-T embedded with SCSC obtains 81.6% (VS Swin-T: 81.3%) Top-1 accuracy with 22% (1G) fewer FLOPs. ConvNet embedded with SCSC obtains 82.3% Top-1 accuracy. On MS1M face recognition, FaceResNet embedded with SCSC achieves 95.6% (VS FaceResNet: 92.6%) rank-1 face identification accuracy with 68% fewer FLOPs and 79% fewer parameters . On COCO detection, Swin-SCSC with Mask R-CNN gains 43.2% (VS Swin-T: 42.7) box AP with 23G fewer FLOPs. On ADE20K segmentation, ResNet-SCSC achieve 45.7% mIoU (VS Swin-T: 44.4%). These experiments demonstrate the effectiveness of our proposed SCSC.

To summarize, we make the following contributions:
1) Present a high capacity and effective convolution module SCSC, which can dynamically combine a large range of receptive fields (**pixel-wise**) in one layer to enhance the presentation ability.
2) Architectures applied with SCSC can obtain better performance with fewer computational cost and parameters. Furthermore, SCSC is a general module and can be applied to strengthen both CNNs and Transformers.
3) SCSC module can power the classical neural networks (e.g., ResNet50) to achieve comparable performance with strong Transformers (e.g., Swin).

2. Related Work

2.1. Vision Transformers

Recently, increasing Vision Transformers obtain state-of-the-art performance on visual tasks [34, 23]. To improve the original vision Transformer (ViT) [19], [10] offers a conditional positional encoding (CPE) technique. Unlike prior fixed or learnable positional encodings, which are predefined and independent of input tokens, CPE is dynamically produced and conditioned by the immediate neighborhood of the input tokens. For self-attention, [14] provides gated positional self-attention (GPSA), a kind of positional self-attention that includes a ”soft” convolutional inductive bias. And [21] presents a new attention mechanism called external attention, which substitutes self-attention in existing popular architectures. External attention is linear in complexity and implicitly considers all data samples' correlations. By preserving encoder branches at various scales while engaging attention across scales, [55] introduces a co-scale mechanism to image Transformers. [62] creates a progressive tokenization system in order to overcome the restriction of ViT when training from scratch on a medium-sized dataset such as ImageNet.

To make Transformer more efficient, [55] introduces Convolutional Vision Transformer (CvT), which enhances the performance and efficiency of Vision Transformer (ViT) by incorporating convolutions into ViT to provide the best of both designs. [39] proposes a hierarchical Transformer named Swin Transformer that computes its representation using Shifted windows. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection, making Swin an efficient and effective Vision Transformer architecture. To go further, [17] creates CSWin Self-Attention, which divides multi-heads into parallel groups and conducts self-attention in horizontal and vertical stripes.

For different transformers design and learning strategies, [49] constructs competitive convolution free transformers DeiTs, which can compete with the state-of-the-art on ImageNet without using any external data at that time. They also present a transformer-specific teacher-student method. [51] offers Pyramid Vision Transformer (PVT), which addresses the challenges of applying Transformer to a variety of dense prediction applications. Moreover, CrossFormer [52] uses cross-scale convolution as a downsample (embedding) layer which blends each embedding with multiple patches of different scales for self-attention module.

Instead of proposing algorithms only to complement the shortcomings of ViTs, our method absorbs the advantages of vision transformer (e.g. large receptive field) and go a further step to benefit both transformer and CNN.

2.2. Large Kernel and Spatial Modeling

In the exploration of large kernels, DetNAS [8] chooses large-kernel blocks in low-level layers and deep blocks in high-level layers. [43] discovers that the large kernel (and effective receptive field) plays a crucial role when we have to do classification and localization tasks at the same time (e.g., semantic segmentation). To solve both classification and localization challenges in semantic segmentation, [43] presents a Global Convolutional Network. Deeplab [6] applies ”atrous convolution” with upsampled filters for dense feature extraction for semantic segmentation and expands it even further to atrous spatial pyramid pooling, which stores objects and visual information at several scales. ConvNet [40] revisits the use of large kernel-sized (7×7) convolutions and RepLKNet [16] further scale up receptive fields using 31×31re-parameterized large depth-wise convolutions. However, RepLKNet [16] focus on large models with a large number of parameters (≥ 79M).

To better model spatial information, [13] presents deformable ConvNets for modeling dense spatial transformation to learn receptive fields adaptively. Inception family (e.g., GoogLeNet [47] and Inception-V4 [46]) extract multi-scale features by different convolutional kernels and fuse them statically using concatenation. DRConv [3] learns a guided mask to assign different customized weights to different spatial regions for better spatial representation. SKNet [36] proposes a dynamic selection technique in CNNs that allows each neuron to modify its receptive field size adaptively based on different scales of input.

Compared with these methods, we additionally introduce large range of receptive field (especially large receptive field), and consider weight sharing and dynamic mechanism at the same time.

2.3. Dynamic Mechanism

With the prevalence of data dependency mechanism [1, 31, 50] , which emphasizes to extract more customized feature for diverse representation [41]. Benefited from the data dependency mechanism, networks can flexibly adjust themselves, including the structure and parameters, to automatically fit the fickle information to improve the representation ability of neural networks. [5, 54] indicate that different regions in the spatial dimension are not equally important in representation learning and should be processed differently. For instance, activation in important regions needs to be amplified to play a dominant role in the forward propagation. SKNet [36] designs a dynamic module to channel-wisely select suitable receptive fields based on channel attention and achieves better performance. It dynamically restructures the networks for the sake of different receptive fields with dilated convolutions [60, 61].

From the aspect of dynamic weights, CondConv [59] obtains dynamic weights by the dynamical linear combination of several weights. And the specialized convolution kernels for each sample are learned in a way similar to the mixture of experts. In the spatial domain, to handle object deformations, Deformable Kernels [20] directly resamples the original kernel space to adapt the effective receptive field (ERF) while leaving the receptive field untouched. Local Relation Networks [29] adaptively determine aggregation weights for spatial dimension based on the compositional relationship of local pixel pairs. Non-local [53] operation computes the response at each position as the weighted sum of the features at all positions, which can make it capture long-range dependencies. Different from above dynamic methods, apart from dynamic filters, DRConv achieves a dynamic guided mask to automatically determine the distribution of multiple filters so that it can process variable distribution of spatial semantics. However, these methods have high computational and memory complexity which significantly limits the efficiency of the model.

2.4. CNN VS Transformers

Apart from designing new Vision Transformers, some works focus on exploring the relationship between Transformer and CNN. [63] performs empirical research on various DNN frameworks (e.g., CNN, Transformer, and MLP) in order to grasp their benefits and drawbacks better. [12] shows that self-attention layers can learn to behave similar to convolutional layers. [48] discovers that CNN-based pretrained models are competitive and outperform their Transformer counterparts in some NLP settings. [24] recasts local attention as a channel-wise locally-connected layer and empirically find that the models based on depth-wise convolution with lower processing complexity perform on par with or somewhat better than Swin Transformer. In this paper, we propose an effective general convolution module, which narrows the gap between CNNs and Transformers.

3. Method

For CNNs and Transformers, CNNs can exploit large kernels to obtain large receptive fields, and Transformers can get global receptive fields through self-attention. Nevertheless, both large dense kernels and self-attention mechanisms cannot avoid the high computational cost. Moreover, large kernels may hinder the model to capture local features. Therefore, find an efficient way to obtain large effective receptive fields and keep the feature diversity meanwhile attracts increasing attention.

Since depth-wise convolution is widely used in modern CNNs for its efficiency and effectiveness, it becomes feasible solution to introduce large receptive fields without intensive computational cost. As analyzed in [24], depth-wise convolution can resemble local attention (Local Vision Transformer, e.g., Swin-Transformer [39]) in sparse connectivity. But the fixed large windows or kernels may not be a good manner to capture local features and still have space to be improved.

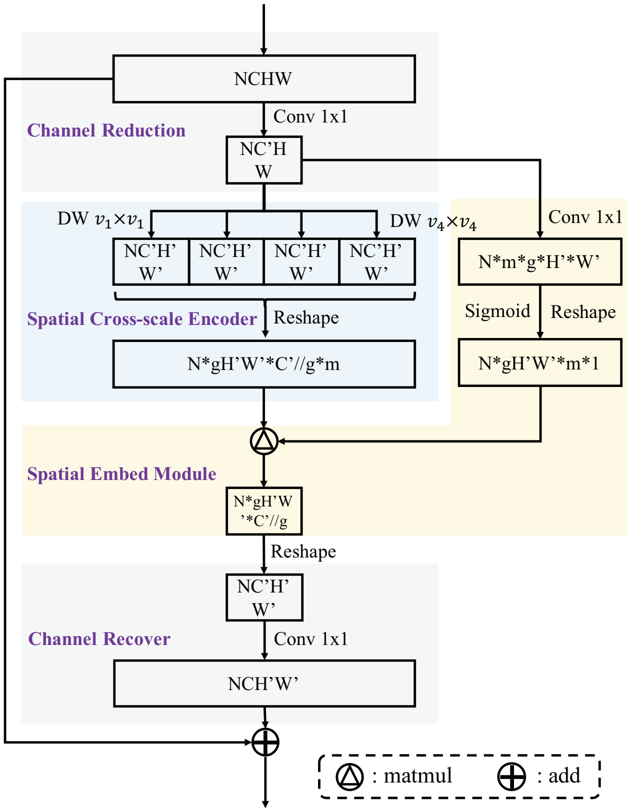

Figure 2. Design details of Spatial Cross-scale Convolution Module (SCSC).

Therefore, based on depth-wise convolution, we applied a wide range of kernel size from 3 × 3 to 13 × 13 to obtain effective receptive fields, so as to capture microscopic and macroscopic feature representation. This way can capture different receptive fields in one layer and strengthen the spatial modeling ability. Furthermore, we put forward an efficient spatial embed module to combine different level spatial features, which can further enhance the overall presentation ability.

3.1. SCSC: Spatial Cross-scale Convolution Module

For clear illustration, we split SCSC into four steps: Channel Reduction, Spatial Cross-scale Encoder, Spatial Combination, and Channel Recover. Taking a K layers CNN for example, the input of the k_{th} layer can be denoted as $X^k \in \mathbb{R}^{N \times C \times H \times W}$, where N, C, H, W are batchsize, channel, height, and width respectively. As shown in Figure 2, for Channel Reduction, we use conv 1×1 as the mapping function,

$$X_d^k = W_d^k \otimes X^k, \tag{1}$$

where $\otimes$ denotes convolutional operation, $X_d^k \in \mathbb{R}^{N \times C//m \times H \times W}$, m is a constant and means how many

	downsp. rate (output size)	Swin	Swin-SCSC	ResNet	ResNet-SCSC
stage1	4× (56×56)	concat 4×4, 96-d, LN $\begin{bmatrix}\text{win. sz. } 7\times7 \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	concat 4×4, 96-d, LN SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,11]\end{bmatrix} \times 2$	bottleneck block $\begin{bmatrix}\text{kernel sz.} \\ [1,3]\end{bmatrix} \times 3$	SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,9,13]\end{bmatrix} \times 3$
stage2	8× (28×28)	concat 2×2, 192-d, LN $\begin{bmatrix}\text{win. sz. } 7\times7 \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	concat 2×2, 192-d, LN SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,9]\end{bmatrix} \times 2$	bottleneck block $\begin{bmatrix}\text{kernel sz.} \\ [1,3]\end{bmatrix} \times 4$	SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,7,11]\end{bmatrix} \times 5$
stage3	16× (14×14)	concat 2×2, 384-d, LN $\begin{bmatrix}\text{win. sz. } 7\times7 \\ \text{dim 96, head 3}\end{bmatrix} \times 6$	concat 2×2, 384-d, LN SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,7]\end{bmatrix} \times 6$	bottleneck block $\begin{bmatrix}\text{kernel sz.} \\ [1,3]\end{bmatrix} \times 6$	SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,5,7]\end{bmatrix} \times 12$
stage4	32× (7×7)	concat 2×2, 768-d, LN $\begin{bmatrix}\text{win. sz. } 7\times7 \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	concat 2×2, 768-d, LN SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,5]\end{bmatrix} \times 2$	bottleneck block $\begin{bmatrix}\text{kernel sz.} \\ [1,3]\end{bmatrix} \times 3$	SCSC block $\begin{bmatrix}\text{kernel sz.} \\ [3,5]\end{bmatrix} \times 3$

Table 1. Detailed architecture specifications. The kernel size (sz.) of our SCSC can be any combination for the specific tasks, here we just give some examples.

different kernels in Spatial Cross-scale Encoder. Then we calculate the Spatial Cross-scale Encoder by

$$X_s^k = \left\{X_{s_1}^k, ..., X_{s_i}^k, ..., X_{s_m}^k\right\}, \tag{2}$$

$$X_{s_i}^k = W_{s_i}^k \otimes X_d^k \ , \ W_{s_i}^k \in \mathbb{R}^{C//m\times C//m\times v\times v}, \tag{3}$$

in which $i \in [1, m], v \in [3, 13], X_{s_i}^k \in \mathbb{R}^{N\times C//m\times H'\times W'}$. After getting different level spatial features, we design a Spatial Embed Module to dynamically fuse multi-scale features at each spatial position (i.e., pixel level).

$$\mathcal{M}^k = SEM(X_d^k), \tag{4}$$

in $SEM()$, we use 1×1 conv to reduce the channel number to $m * g$, then apply sigmoid function, the result noted as $\mathcal{M}^k \in \mathbb{R}^{N\times m*g\times H'\times W'}$, g is a constant and means the combination times, which can bring more dynamic combinations. We reshape the X_s^k to $N\times g*H'*W'\times C//m//g\times m$, and $\mathcal{M}^k$ to $N \times g * H' * W' \times m \times 1$. and then we can get the embedded feature,

$$X_e^k = Matmul(X_s^k, \mathcal{M}^k), \tag{5}$$

where $X_e^k \in \mathbb{R}^{N\times C//m\times H'\times W'}$.

Finally, we product Channel Recover operation by

$$X_o^k = W_r^k \otimes X_e^k, \tag{6}$$

and $X_o^k \in \mathbb{R}^{N\times C\times H'\times W'}$. All the operations are differentiable, so our objective function is

$$\min_{W_d^k, W_s^k, W_{\mathcal{M}^k}, W_r^k|_{k=1}^K} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(Y_i; \hat{Y}|X_i), \tag{7}$$

in which W is the learnable weights, $(X_i, Y_i), i \in [1, n]$ is the input image and label, $\hat{Y}$ is the predicted label. $\mathcal{L}()$ is the loss function.

3.2. Architecture

The proposed SCSC module can be easily embedded in any existing architecture. As shown in Table 1, taking the most used CNN ResNet for example, we directly replace the original bottleneck block of ResNet-50 with our SCSC block. For Transformer, we replace the self-attention with the proposed SCSC module, in which the pre-linear-projections and post-linear-projections over the values can be regarded as the 1×1 convolutions in our design.

3.3. Discussion and Limitation

Some existing methods have contributed to exploring the effectiveness of multi-scale features, but they are suboptimal in terms of efficiency. For example, PSP [26] and ASPP [6] use sampling operators (e.g., pooling or atrous convolution) to efficiently extract multi-scale features, but this comes with the expense of losing spatial information. CrossFormer and GoogleNet use resource-intensive operations, such as regular convolution with large kernels, to extract multi-scale features, and they fuse them through static concatenation. However, this approach significantly increases model parameters.

In contrast, our SCSC approach not only introduces a wide range of receptive fields for effective representation learning but also incorporates weight sharing and dynamic mechanisms for greater efficiency. This allows our model to achieve high performance while keeping the model size and computation cost low.

In details, first, our SCSC is compact and effective, the Channel Reduction step decreases the size of the input feature maps, and then inputs feature maps into the Spatial Cross-scale Encoder, which consists of depth-wise convo-

Model	Image Resolution	Kernel Size Range	Params	FLOPs	Top-1 Acc. (%)
ResNet-50 [27]	224×224	1×1 to 3×3	26M	4.1G	76.2
ResNet-101 [27]	224×224	1×1 to 3×3	45M	7.9G	77.4
ResNet-152 [27]	224×224	1×1 to 3×3	60M	11.6G	78.3
SKNet-50 [36]	224×224	1×1 to 5×5	27M	4.5G	79.3
SKNet-100 [36]	224×224	1×1 to 5×5	49M	8.5G	79.8
SENet [30]	224×224	1×1 to 3×3	22M	3.9G	79.9
ResNeXt [58]	224×224	1×1 to 3×3	25M	4.2G	80.1
ResNet-SCSC-V1 (Ours)	224×224	1×1 to 13×13	**10M**	**1.7G**	**79.4**
ResNet-SCSC-V2 (Ours)	224×224	1×1 to 13×13	**12M**	**2.2G**	**80.3**
ResNet-SCSC-V3 (Ours)	224×224	1×1 to 13×13	**25M**	**4.5G**	**81.5**
ViT-B/16 [19]	384×384	Global Receptive Field	86M	55.4G	77.9
ViT-L/16 [19]	384×384	Global Receptive Field	307M	190.7G	76.5
DeiT-S [49]	224×224	Global Receptive Field	22M	4.6G	79.8
DW-Conv.-T [24]	224×224	7×7	24M	3.8G	81.3
Swin-T [39]	224×224	Local Window 7×7	28M	4.5G	81.3
ConvNet-T [40]	224×224	7×7	28M	4.5G	82.1
Swin-T-SCSC (Ours)	224×224	1×1 to 11×11	**22M**	**3.5G**	**81.6**
ConvNet-T-SCSC (Ours)	224×224	1×1 to 11×11	28M	4.5G	**82.2**

Table 2. Comparison of Top-1 classification accuracy with different architectures (CNNs and Transformers) and some state-of-the-art backbones on ImageNet.

lution with a wide range of kernel size to model the spatial information. This manner can effectively avoid the intensive computational cost. In the meanwhile, the wide range of kernel size provides different receptive fields in one layer, small kernels for the detailed local information, large kernels for the semantic dependence. Furthermore, we design a dynamic Spatial Embed Module to merge the different spatial information. The proposed SCSC naturally aggregates the advantage of CNN and Transformer, small kernels and weight sharing from the CNN, large receptive fields and dynamic from the Transformer. However, it still has its limitation: depth-wise convolution may need exceptional acceleration in practice.

4. Experiments

In this section, we evaluate the effectiveness of our proposed SCSC module by embedding it into the classical CNN backbones (ResNet-50 [27], MobileFaceNet [7] and FaceResNet [15]), and state-of-the-art Swin-Transformer [39] respectively. We conduct experiments for SCSC-based architectures on ImageNet [45], MS1M-V2 [22], COCO 2017 [38] and ADE20K[64] in terms of image classification, face recognition, object detection and segmentation.

4.1. Classification

We take classical ResNet-50 [27] and state-of-the-art Swin-Transformer [39] as the backbone to evaluate SCSC by replacing their original components respectively.

Settings: The ImageNet 2012 dataset [45] is a well-known image classification dataset includes 1.28 million training images and 50k validation images from 1000 classes. All our models are trained on the whole training dataset and validated using the single-crop top-1 validation accuracy. The training settings follow [39], all models in our experiments are trained for 300 epochs with AdamW [35] optimizer and a cosine decay learning rate scheduler starts from 0.001/0.0005. We set batch size as 1024 and weight decay as 0.05.

Resnet50 with SCSC. We directly replace the original bottleneck block with our SCSC block. Following the mainstream works, we refer to the four residual stages of ResNet-50 as c2, c3, c4, c5, respectively. And the block number for the four stages are 3, 4, 6, 3, respectively. For ResNet-SCSC-V1, the four stages' input/output channel number are 96, 192, 384, 512, expansion is set as 2. The kernel sets for the four stages are [3,9,13], [3,7,11], [3,5,7] and [3,5] respectively. We increase the numbers of blocks in c2,c3,c4,c5 from 3,4,6,3 to 3,4,8,3 so that the FLOPs can match the original network. For ResNet-SCSC-V2, most setting keep the same as ResNet-SCSC-V1, except increasing the numbers of blocks in c2,c3,c4,c5 from 3,4,8,3 to 3,5,12,3. For ResNet-SCSC-V3, most settings keep the same as ResNet-SCSC-V2, except the expansion is set as 3. We list the architecture of ResNet-SCSC-V3 in Table 1.

Swin-Transformer with SCSC. We replace local self-attention with our SCSC module. The pre-linear-projections and post-linear-projections over the values can be regarded as the 1×1 convolutions in our design.

The results are shown in Table 2. As can be seen, for ResNet-SCSC-V1 and ResNet-SCSC-V2, we use much less computational cost and parameters to obtain competitive results. For ResNet-SCSC-V3, with comparable model com-

Model	Params	FLOPs	Acc.(%)
MobileFaceNet [7]	0.98M	162M	90.9
MobileFaceNet-SCSC	**0.89M**	**146M**	**92.0**
FaceResNet [15]	40.3M	1050M	92.9
FaceResNet-SCSC	**8.3M**	**330M**	**95.6**

Table 3. Results of SCSC on Megaface. "Acc." refers to the rank-1 face identification accuracy with 1M distractors.

plexity, we successfully make the performance of a classical CNN architecture (ResNet) match the Transformers' performance. For Swin-Transformer, with our SCSC module, it achieves a higher accuracy with 1G (22%) FLOPs fewer computational cost. These results show that SCSC-based architectures not only have a considerable improvement, but also saving computational cost and parameters, demonstrating the effectiveness of our method.

4.2. Face Recognition

We further evaluate the effectiveness of our SCSC on Face Recognition task. MobileFaceNet [7] and FaceResNet [15] are applied as our backbone with input size 96×96.
Settings: MS1M-V2 dataset is a large-scale face dataset with 5.8M images from 85k celebrities. We use a refined semi-automatic version of the MS-Celeb-1M dataset [22] which consists of 1M photos from 100k identities for training. The dataset we use for validation is MegaFace [33], which includes 1M images of 60k identities as the gallery set and 100k images of 530 different individuals. We use SGD with a momentum of 0.9 to optimize the model, and the batch size is 512. We train all the models for 420k iterations. The learning rate begins with 0.1 and is divided by 10 at 252k, 364k, and 406k iterations. For evaluation, we use face identification metric which refers to the rank-1 accuracy on MegaFace as the evaluation indicator.
MobileFaceNet with SCSC. We directly replace the original bottleneck block with our SCSC block. In [7], there are 5 stages, we denote them as s1, s2, s3, s4, s5, respectively. And the block number for the five stages are 5, 1, 6, 1,2, respectively. For MobileFaceNet-SCSC, the kernel sets for the five stages are [3,9], [3,7], [3,7], [3,5] and [3,5] respectively. The kernel size is relatively small because of the input image size is small. We set expansion as 3 so that not reducing too many FLOPs since MobileFaceNet is already an efficient network.
FaceResNet with SCSC. As in FaceResNet [15], we denote the four residual stages of FaceResNet as c2, c3, c4, c5, respectively. And the block number for the four stages are 3, 2, 2, 2. For FaceResNet-SCSC, We set the numbers of blocks in c2,c3,c4,c5 as 6,6,6,4. The kernel sets for the four stages are [5,11], [3,9], [3,5] and [3,3] respectively.

As Table 3 shows, given that MobileFaceNet is already an efficient network, MobileFaceNet-SCSC still outperforms the baseline by 1.1% gain even with fewer FLOPs and parameters. Moreover, FaceResNet-SCSC surpasses FaceResNet by a large margin of 2.7% accuracy with 68% fewer computational cost and 79% fewer parameters, further indicating the superiority of our proposed SCSC.

Backbone	box AP	mask AP	FLOPs	Params
R50 [27]	38.2	34.7	260.1G	44.2M
R101 [27]	40.0	36.1	336.2G	63.2M
X101-32x4d [58]	41.9	37.5	340.0G	62.8M
X101-64x4d [58]	42.8	38.4	493.4G	101.9M
Swin-T [39]	42.7	39.3	263.8G	47.8M
ResNet-SCSC	**44.0**	**40.4**	270.8G	44.9M
Swin-SCSC	**43.2**	**39.6**	**240.5G**	**41.8M**

Table 4. Results of object detection and instance segmentation on the COCO *mini-val* with Mask R-CNN (1x schedule). FLOPs are measured on an 800 × 1280 image.

Backbone	box AP	mask AP	FLOPs	Params
R50 [27]	46.3	43.4	739G	82M
DeiT-S [49]	48.0	41.4	889G	80M
Swin-T [39]	50.4	43.7	742G	86M
DW Conv.-T [24]	49.9	43.4	730G	82M
ResNet-SCSC	50.3	**43.7**	749G	83M
Swin-SCSC	49.9	43.2	**719G**	**80M**

Table 5. Results of object detection and instance segmentation performance on the COCO *mini-val* with Cascade Mask R-CNN (3x schedule). FLOPs are measured on an 800 × 1280 image.

4.3. Detection

To evaluate the effectiveness of our SCSC on object detection, we utilize the COCO 2017 dataset [38] which consists of 80k train images and 40k val images.
Settings: we exploit Mask R-CNN [25] and Cascade Mask R-CNN [2] framework with FPN [37]. For Mask-RCNN, the implementation is based on MMDetection [4]. We train the detection networks on 8 GPU with mini-batch 2 per GPU for 1x schedule (12 epochs). We use AdamW optimizer, and the initial learning rate is 0.0001, started after 500 iteration warmup and decayed by 0.1 times at the 8th and 11th epoch. We use stochastic drop path regularization of 0.2 and weight decay of 0.05. All other hyper-parameters follow the default settings in MMDetection [4]. For Cascade Mask-RCNN, we follow the implementation, training, and test settings from Swin Transformer [39]. Backbone weights are initialized by the parameters of Swin-SCSC and ResNet-SCSC respectively.

As shown in Table 4, we compare Swin-SCSC and ResNet-SCSC with standard ConvNets, i.e., ResNe(X)t, and previous Transformer networks, e.g., Swin-T with Mask R-CNN. The comparisons are conducted by changing only the backbones with other settings unchanged. Our ResNet-SCSC architecture achieves 5.0% improvement over baseline ResNet-50. Compared with strong baseline Swin-T, ResNet-SCSC improves 1.3% box AP and 1.1% mask AP over Swin-T. Swin-SCSC achieves better

Backbone	mIoU	FLOPs	Params
R50 [27]	42.1	952G	67M
R101 [27]	43.8	1029G	86M
DeiT-S [49]	42.9	1099G	52M
DW Conv.-T [24]	45.5	928G	56M
Swin-T [39]	44.4	941G	60M
ResNet-SCSC	**45.7**	956G	64M
Swin-SCSC	44.0	**916G**	**54M**

Table 6. Results of semantic segmentation on the ADE20K *val* set with UperNet. FLOPs are measured on an 512 × 2048 image.

performance even with $23G$ FLOPs fewer computational cost and $6M$ fewer parameters, illustrating that SCSC also works on downstream tasks.

4.4. Semantic Segmentation

Settings: ADE20K[64] is a widely used semantic segmentation dataset, covering a broad range of 150 semantic categories. It has 25K images, with 20K for training, 2K for validation, and another 3K for testing. We utilize the implementation of UperNet [56] in MMSegmentation [11] as our base framework for its high efficiency. We use the same setting as the Swin Transformer [39]. Models are trained on 8 GPUs with mini-batch 2 per GPU for 160k iterations. We employ the AdamW optimizer with an initial learning rate of 0.00006, a weight decay of 0.01, a scheduler that uses linear learning rate decay, and a linear warmup of 1,500 iterations. SyncBN and stochastic depth with the ratio of 0.3 is applied for Swin-SCSC. The experimental results are reported as single-scale testing. All other hyper-parameters follow the default settings in MMSegmentation [11].

Table 6 lists the mIoU, FLOPs and model size (number of parameters) for different backbones. Compared with Swin-T, our ResNet-SCSC achieves +1.3 mIoU higher (45.7 *vs.* 44.4) than Swin-T with slightly larger model size and computation cost. Our Swin-SCSC achieves comparable performance with 25G less computation cost and 6M fewer parameters.

4.5. Ablation Study

We conduct a series of experiments to evaluate the impact of specific design choices in our spatial Cross-scale Convolution Module. We use the ImageNet classification dataset [45] for our experiments, and follow the same experimental settings as described in section 4.1 by default.

The effectiveness of combination number g in Spatial Embed Module: We use ResNet-SCSC to explore which combination number g in Spatial Embed Module is better. We set g as 2, 4, 8, respectively, and the Top-1 accuracy is 81.1%, 81.5%, and 81.2% respectively. Therefore, we set $g = 4$ by default.

The Effect of Spatial Embed Module: We design the Spatial Embed Module (SEM) for fusing multi-scale features dynamically. For ResNet-SCSC, if we only use 7x7 kernel size, we will get 80.6% top-1 accuracy. Adding Spatial Cross-scale Encoder, the accuracy can reach 81.0%. Further adding Spatial Embed Module, we can obtain 81.6% accuracy. This shows the effectiveness of dynamic fusing by using our SEM.

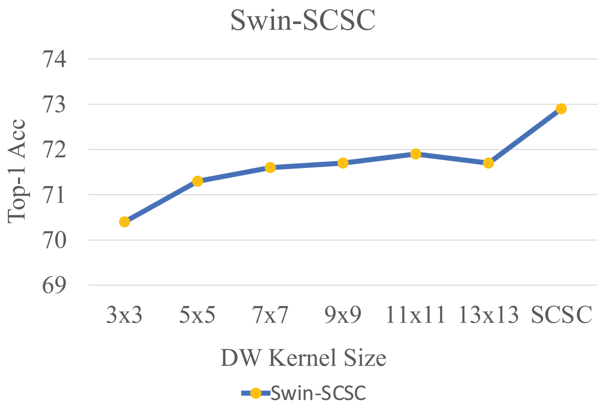

Figure 3. Influence of DW kernel size in Spatial Embed Module. We set DW kernel size as 3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, 13×13, SCSC setting, respectively. Larger kernel will result better performance, but when the kernel size increase too large, the accuracy begin to decline. Our SCSC achieves a large improvement by using different receptive fields in one layer

Kernel size: We train Swin-SCSC for 100 epochs to explore the choices of kernel sizes. We set the size of depthwise (DW) kernels with 3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, 13 × 13, as well as our SCSC setting. As shown in Figure 3, larger kernels results in better accuracy but when the kernel size exceeds 11 × 11, accuracy begins to decline. Our results demonstrate that both excessively small and excessively large kernel sizes are suboptimal. Furthermore, using different receptive fields in one layer achieves the highest accuracy and significantly outperforms other methods. This suggests that utilizing multiple receptive fields can effectively enhance the performance of model.

5. Conclusion and Future Work

Different from the mainstream CNN's small kernel size and the Transformer's global receptive field, we increase the spatial modeling ability by using a wide range of kernel sizes. Furthermore, we design an efficient spatial embedding module to merge the different spatial representation features. As a result, the proposed SCSC naturally combines the advantage of CNNs and Transformers, small kernel and weight sharing from the CNN, large receptive field and dynamic from the Transformer. Intensive experiments illustrate the effectiveness and generalization of our SCSC. In the future, we will explore more about the relationship between CNNs and Transformers.

References

[1] Alan Allport. Visual attention. The MIT Press, 1989. 3

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6154–6162, 2018. 7

[3] Jin Chen, Xijun Wang, Zichao Guo, Xiangyu Zhang, and Jian Sun. Dynamic region-aware convolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8064–8073, 2021. 1, 3

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 7

[5] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5659–5667, 2017. 3

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017. 1, 3, 5

[7] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition (CCBR)*, pages 428–438, 2018. 6, 7

[8] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. pages 6642–6652, 2019. 1, 3

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1258, 2017. 1

[10] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 3

[11] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 8

[12] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019. 1, 4

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 1, 3

[14] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021. 3

[15] Xiaohan Ding, Chunlong Xia, Xiangyu Zhang, Xiaojie Chu, Jungong Han, and Guiguang Ding. Repmlp: re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv preprint arXiv:2105.01883*, 2021. 6, 7

[16] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11963–11975, 2022. 3

[17] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021. 1, 3

[18] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *arXiv preprint arXiv:2103.03404*, 2021. 1

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 6

[20] Hang Gao, Xizhou Zhu, Steve Lin, and Jifeng Dai. Deformable kernels: Adapting effective receptive fields for object deformation. *arXiv preprint arXiv:1910.02940*, 2019. 4

[21] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint arXiv:2105.02358*, 2021. 3

[22] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision (ECCV)*, pages 87–102, 2016. 6, 7

[23] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*, 2020. 3

[24] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *International Conference on Learning Representations*, 2021. 1, 4, 6, 7, 8

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 7

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 5

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 7, 8

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1

[29] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3464–3473, 2019. 4

[30] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018. 1, 6

[31] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2017–2025, 2015. 3

[32] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 667–675, 2016. 1

[33] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4873–4882, 2016. 7

[34] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021. 3

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[36] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–519, 2019. 3, 6

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 7

[38] 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 *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 6, 7

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 4, 6, 7, 8

[40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 3, 6

[41] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 3

[42] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021. 1

[43] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2017. 1, 3

[44] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[45] 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. *International Journal of Computer Vision (IJCV)*, pages 211–252, 2015. 6, 8

[46] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017. 1, 3

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[48] Yi Tay, Mostafa Dehghani, Jai Gupta, Dara Bahri, Vamsi Aribandi, Zhen Qin, and Donald Metzler. Are pre-trained convolutions better than pre-trained transformers? *arXiv preprint arXiv:2105.03322*, 2021. 4

[49] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pages 10347–10357. PMLR, 2021. 1, 3, 6, 7, 8

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 3

[51] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 3

[52] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 7794–7803, 2018. 4

[54] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[55] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 3

[56] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 8

[57] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–509, 2019. 1

[58] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017. 6, 7

[59] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3

[60] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[61] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 472–480, 2017. 3

[62] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 3

[63] Yucheng Zhao, Guangting Wang, Chuanxin Tang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. A battle of network structures: An empirical study of cnn, transformer, and mlp. *arXiv preprint arXiv:2108.13002*, 2021. 1, 4

[64] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision (IJCV)*, 127(3):302–321, 2019. 6, 8