LOCAL ATTENTION LAYERS FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention layers in transformer networks have contributed to state-of-the-art results on many vision tasks. Still, attention layers leave room for improvement because relative position information is not learned, and locality constraints are typically not enforced. To mitigate both issues, we propose a convolution-style attention layer, LA-layer, as a replacement for traditional attention layers. LA-layers implicitly learn the position information in a convolutional manner. Given an input feature map, keys in the kernel region deform in a designated constrained region, which results in a larger receptive field with locality constraints. Query and keys are processed by a novel aggregation function that outputs attention weights for the values. The final result is an aggregation of the attention weights and values. In our experiments, we replace ResNet's convolutional layers with LA-layers and address image recognition, object detection and instance segmentation tasks. We consistently demonstrate performance gains with LA-layers over the state-of-the-art, despite having fewer floating point operations and training parameters. These results suggest that LA-layers more effectively and efficiently extract features. They can replace convolutional and attention layers across a range of networks.

1 INTRODUCTION

Convolutional neural networks (CNNs) have become the backbone for many computer vision tasks, including object detection (He et al., 2016; Redmon et al., 2016), instance segmentation (Chen et al., 2018; Iglovikov et al., 2018), and image generation (Han et al., 2018).

One challenge with CNNs is to achieve long-range interactions between pixels. The receptive field of a single layer is bound by the small kernel size. Long-range interactions can typically only be modeled through many layers. To overcome this issue, recent approaches introduce self-attention (SA) into computer vision tasks (Dosovitskiy et al., 2021; Vaswani et al., 2017). Building on SA layers, researchers have proposed different types of vision transformer models (Dong et al., 2022; Liu et al., 2021; Han et al., 2021; Zhou et al., 2021; Liu et al., 2022) that typically outperform CNNs in terms of accuracy by integrating transformer-style modules into CNN-based architectures. For example, ViT (Dosovitskiy et al., 2021) directly processes image patches of CNN outputs through self-attention. Ramachandran et al. (2019) and Hu et al. (2019) present a stand-alone design of local self-attention modules to even fully replace the spatial convolutions in ResNet architectures.

Despite the success of self-attention in video transformers, two issues remain. The first is that the position information is embedded through a manually designed, fixed position encoder. The performance could deteriorate when the designed position encoder is sub-optimal for the given task or image domain. The second issue is that most transformers neglect the performance improvement from the locality constraint and the extraction of useful tokens, while focusing on enlarging the receptive field to achieve long-range dependencies. Consequently, this leads to excessive memory use and computational cost for the attention layer.

To address these issues, researchers have proposed convolution-style local attention layers to implicitly learn the position information. For example, Contextual Transformers (Li et al., 2022) have demonstrated impressive results on many challenging vision tasks, but the performance remains limited by the kernel size. Other works have begun to use deformable kernels to extract useful tokens so that they can enlarge the receptive field while keeping the kernel small. For example, inspired by the deformable convolution (Dai et al., 2017), DAT (Xia et al., 2022) utilizes a deformable self-attention

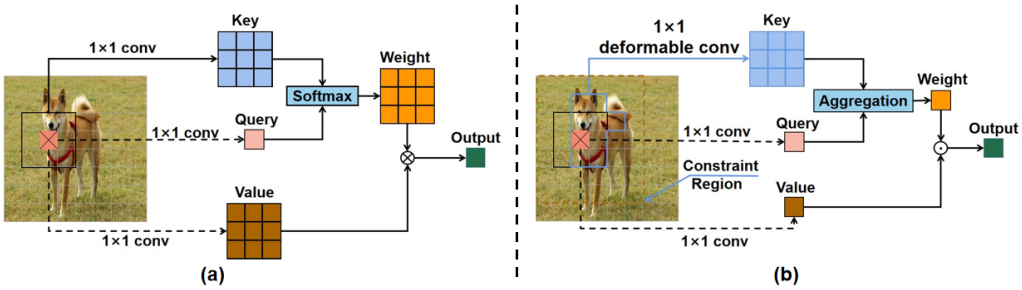

Figure 1: Typical local attention (a) compared to LA-layer (b). The red box denotes the anchor of the self-attention. (a) Local attention extracts a fixed region to compute the key and attention weight. (b) LA-layer learns the deformable region to extract the key, and employs an efficient aggregation function to produce the attention weight. The attention result is the multiplication of weight and value.

module to extract region of interests. However, the method can generate only a few groups of position offsets for attention layers due to the high computation and memory cost of the self-attention module.

In this work, we propose a novel convolution-style local attention layer, termed LA-layer and shown in Figure 1(b). Compared to a typical local attention module (Hu et al., 2019; Ramachandran et al., 2019; Zhao et al., 2020), our LA-layer not only implicitly learns the position information but also inherits the ability of extracting deformable regions of interests with locality constraints. In the LA-layer, we first extract the single query within a kernel region-like convolution operation. Keys are added with position offsets in a constrained region to expand the receptive field while observing the locality constraint. The single query and keys pass a novel attention aggregation function to produce the attention weight with lower computation and memory cost. The attention weight is further utilized to aggregate all input values and return the output with implicitly encoded position information and useful regions of interest with proper locality constraint.

By integrating the proposed LA-layer into common CNN models, full attention models of the same architecture can be achieved with fewer parameters and FLOPs. Our two main contributions:

- We introduce LA-layer, a local attention layer that can serve as an effective replacement for convolution layers. The LA-layer uses an efficient novel attention aggregation function.
- We use ResNets with LA-layers as full attention networks. Extensive experiments on image classification, object detection and instance segmentation demonstrate that LA-layers outperform several state-of-the-art backbones.

We first discuss related work on convolution layers, self-attention and methods to combine the advantages of both. We then introduce the LA-layer in Section 3. In Section 4, we demonstrate the performance of the LA-layer on various core vision tasks. We conclude in Section 5.

2 Related Work

Convolution Layers and Extensions: Convolutional neural networks (CNNs) have shown great performance on many vision tasks. The convolution layer is the basic building block for CNNs. Its aim is to encourage the network to learn local correlation structures in the input. Due to the specific way of extracting feature maps, convolution layers require convolution kernels to have a fixed size. Although larger kernels result in larger receptive fields, they come with higher computation and memory cost. Moreover, large kernels in CNNs tend to harm the performance on vision tasks (Ding et al., 2022). One way to avoid large kernels while increasing the receptive field is to model the interactions between spatially distant regions with a limited receptive field in the image through successive convolution layers (Howard et al., 2017; Huang et al., 2017; Radosavovic et al., 2020).

While typically increasing accuracy, this approach reduces the efficiency of CNNs. To overcome the short-range problem for small kernels, several extensions to the regular convolution layer have been proposed. Group convolutions (Krizhevsky et al., 2012) and depth-wise convolutions (Chollet, 2017)

are examples of such efforts. Another option is to modify the spatial scope for aggregation, which can enlarge the receptive field. One popular implementation of this idea is the dilated convolution (Yu & Koltun, 2015), which increases the spacing inside the kernel to increase the receptive field.

Self-Attention: Researchers have begun to apply self-attention to computer vision tasks (Vaswani et al., 2017; Dosovitskiy et al., 2021; Dong et al., 2022). This is achieved by employing self-attention over feature vectors across different spatial locations within an image, such that a large content-based receptive field is obtained. The process is mathematically presented as:

$$A = (a)_{ij} = Softmax(QK^T) \tag{1}$$

$$Y = AV \tag{2}$$

with Query map Q, Key map K, and Value map V. $A \in R^{N \times N}$ is the attention matrix and α_{ij} the relationship between the i^{th} and j^{th} elements, and Y is the attention feature map.

In Dosovitskiy et al. (2021), ViT is proposed for image classification with convolution layers replaced by self-attention layers. The input image is split into patches, then embedded as tokens containing features of patches. Tokens with manually designed position information are processed by the transformer encoder that is composed of self-attention layers. In Zhang et al. (2019), the Self-Attention Generative Adversarial Network (SAGAN) is proposed for image generation. Since image generation highly depends on the quality of each pixel, instead of splitting the feature map into patches, SAGAN directly utilizes the global attention feature map as a replacement of the convolution feature map. In Perreault et al. (2020), SpotNet is proposed to output the bounding boxes for the object detection task. Inspired by the spatial relationship inside the attention feature map, this map acts as an aide for the convolution feature map instead of fully replacing it. Through multiplication in the last stage, the convolution feature map and attention feature maps are integrated to suppress non-relevant areas. This has been shown to significantly improve the performance.

Self-Attention Extensions: Inspired by advantages of self-attention, researchers have further proposed full attention networks which replace the convolution layers inside CNN models with attention layers while otherwise retaining the network architecture (Ramachandran et al., 2019). Full attention networks have achieved many state-of-the-art models for various tasks, e.g. Hu et al. (2019); Li et al. (2022). Despite their popularity, global self-attention layers in such networks incur high computation and memory demands. To address this issue, researchers have introduced local attention layers to constrain the attention pattern fixed local windows (Hu et al., 2019; Ramachandran et al., 2019; Zhao et al., 2020).

Although the local attention layer reduces the computation cost, the receptive field is also constrained. To enlarge the receptive field while reducing the computation cost as much as possible, different methods have been proposed. Swin Transformer (Liu et al., 2021) uses a hierarchical representation computed with shifted windows. This allows for the flexible modeling at various scales, but the method leads to a slower increase of the receptive field. Based on Swin Transformer, other vision transformers introduce data-dependent sparse attention to flexibly model relevant features (Xia et al., 2022). Although a sparse convolution is realized in deformable convolution network (DCN, Dai et al. (2017)), applying DCN to transformer models is a non-trivial problem. Compared to the convolution layer, the space complexity of the self-attention layer is generally bi-quadratic.

Deformable DETR (Zhu et al., 2020) implements the idea of deformable attention with a lower number of keys at each scale to reduce the computation cost. This works well as a detection head, but also causes a loss of information due to the strongly reduced number of keys in the backbone network. Based on the assumption that different queries may have similar attention maps, DAT (Xia et al., 2022) proposes to generate a few groups of position offsets for the local attention layers, so that it can use shared shifted keys and values for each query to achieve an efficient trade-off. However, since different pixels in the same group use a shared query offset, the network concentrates on specific regions which limits the receptive field.

Another issue is that the self-attention layer is the content-based summarization of features. This requires the introduction of manually designed position information into self-attention layers, such as in Vision Transformers (Dosovitskiy et al., 2021) and Swin Transformers (Liu et al., 2021). However, optimal position information might differ between vision tasks, image domain and network depth. Therefore, a fixed position encoding might not be optimal. Convolution layers implicitly encode position information along with the feature maps they extract (Islam et al., 2020). Inspired by

this observation, researchers have introduced local convolution-style self-attention layers, such that the position information is implicitly encoded in a convolution way. For example, CoT blocks (Li et al., 2022) are proposed as a local attention layer to replace the convolution layer. To reduce the computation cost of the self-attention layer, CoT uses two successive 1×1 convolution layers to produce the attention matrix. A 3×3 convolution layer is then applied in the query region to obtain the local context information. The convolution output is concatenated with the query as the input for the next stage. The CoT block requires fewer FLOPs and parameters compared to the same network with solely convolution layers. CoT can implicitly learn the position information because the attention matrix is generated from convolution layers. Still, the performance is constrained due to the lack of data specificity of the self-attention, which plays a more importance role than long-range dependency, especially in CNNs (Park & Kim, 2022).

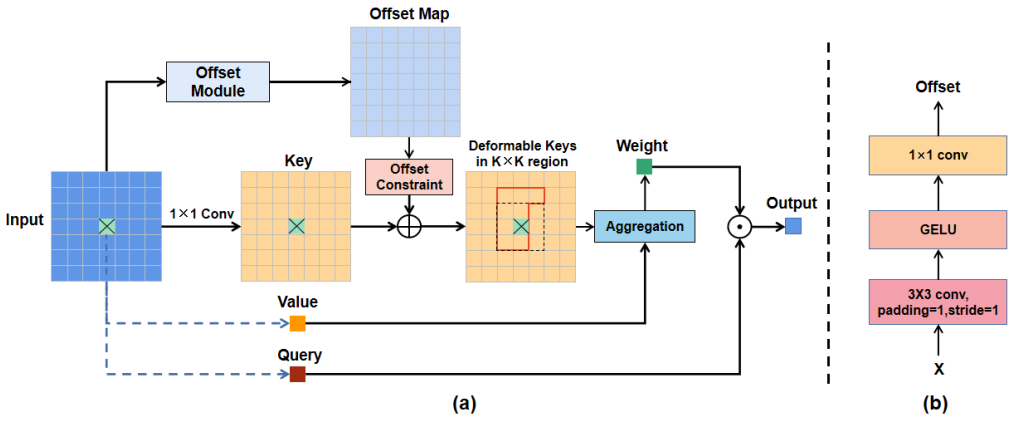

Figure 2: Schematic overview of the LA-layer. (a) Computation process, which is repeated for each element of the input feature map. The output is the sum of deformable features with assigned locality constraint. (b) The offset module in the LA-layer. The input feature map passes through a 3×3 convolution, GELU, and 1×1 convolution. The output is a feature map with size $H \times W \times 2$. Values are the deformed (x, y) position indices.

3 Local Attention Layer (LA-layer)

We introduce a local attention layer (LA-layer) as a replacement for convolution layers in CNNs, to produce full attention networks while preserving the original model structure. Compared to the traditional local attention module, our LA-layer implicitly learns the position information and inherits the ability to extract deformable regions of interest with significantly reduced computation and memory cost. More importantly, the LA-layer integrates the improvements from locality constraints of self-attention layers, which brings better performance. The framework of the LA-layer is shown in Figure 2. We will first introduce the LA-layer with its components, and then detail the implementation of full attention networks using the LA-layer.

3.1 Overview of the LA-Layer

For the LA-layer, similar to the kernel in the convolution layer, we assign a local region, i.e., a $k \times k$ neighborhood, to aggregate the input features. We first revisit the local attention layer in recent vision transformers (Ramachandran et al., 2019). Taking a flattened feature map $x \in R^{W \times H \times C}$ as the input, the Query map Q, Key map K, and Value map V are generated through the self-attention mechanism, which is formulated as:

$$Q = xW_q \tag{3}$$
$$K = xW_k \tag{4}$$
$$V = xW_v \tag{5}$$

where $W_q, W_k, W_v \in R^{d_{out} \times d_{in}}$ are learned transformation matrices.

The local Query map Q_k, Key map K_k, and Value map V_k are extracted from the feature maps K, Q, and V. At each anchor position q_{ij}, the attention matrix will be produced after the multiplication

Stage	Output	ResNet-50	LA-ResNet-50	ResNeXt-50	LA-ResNeXt-50
res1	112×112	7×7 conv,64,stride2	7×7 conv,64,stride2	7×7 conv,64,stride2	7×7 conv,64,stride2
res2	56×56	3×3 max pool, stride 2 {1×1,64 3×3 conv,64 1×1,256}×3	3×3 max pool, stride 2 {1×1,64 3×3 LA-layer,64 1×1,256}×3	3×3 max pool, stride 2 {1×1,64 3×3 conv,64 1×1,256}×3	3×3 max pool, stride 2 {1×1,64 3×3 LA-layer,64 1×1,256}×3
res3	28×28	{1×1,64 3×3 conv,64 1×1,256}×4	{1×1,64 3×3 LA-layer,64 1×1,256}×4	{1×1,64 3×3 LA-layer,64 1×1,256}×4	{1×1,64 3×3 LA-layer,64 1×1,256}×4
res4	14×14	{1×1,64 3×3 conv,64 1×1,256}×6	{1×1,64 3×3 LA-layer,64 1×1,256}×6	{1×1,64 3×3 conv,64 1×1,256}×6	{1×1,64 3×3 LA-layer,64 1×1,256}×6
res5	7×7	{1×1,64 3×3 conv,64 1×1,256}×3	{1×1,64 3×3 LA-layer,64 1×1,256}×3	{1×1,64 3×3 conv,64 1×1,256}×3	{1×1,64 3×3 LA-layer,64 1×1,256}×3
	1×1	global average pool 1000-d fc,softmax	global average pool 1000-d fc,softmax	global average pool 1000-d fc,softmax	global average pool 1000-d fc,softmax
Params (M)		25.5	21.0	25.0	24.2
FLOPs (G)		4.1	3.4	4.2	4.2

Table 1: Comparison of ResNet-50 and ResNext-50 with and without LA-layer with 3×3 kernel and 8 channels.

of q_{ij} and K_k. The content-based integrated feature y_{ij} will be generated after V_k and the attention matrix pass through the feature multiplication. Finally, we obtain the attention feature map by sliding the kernel through the entire input. Mathematically, for input x_{ij} with corresponding position p_{ij}, the output y_{ij} is computed as:

$$y_{ij} = \sum_{a,b \in N_k(i,j)} softmax_{ab}(q_{ij}^T k_{ab}) v_{ab} \tag{6}$$

In practice, multiple attention heads are used to learn multiple distinct representations of the input (Vaswani et al., 2017). This works by partitioning the pixel features x_{ij} depth-wise into m groups $x_{ij}^n \in R^{d_{in}/m}$ and to calculate each group independently using distinct transformations. Finally, we concatenate the resulting representations into the final output.

For the traditional local attention mechanism, the kernel size limits the receptive field of the local attention layer. Inspired by the convolution operation, we apply the deformable mechanism in our local attention kernel. Following Dai et al. (2017), a sub-network is adopted for offset generation. It processes the local features and outputs the offset values for reference points in the kernel region. The input features are first passed through a 3×3 convolution to capture local features. Then, GELU activation and a 1×1 convolution is applied to produce the 2D offsets. We drop the bias in the 1×1 convolution to avoid the compulsive shift for all locations. Inspired by the improvement from locality constraints in self-attention (Park & Kim, 2022), we introduce a simple locality constraint operation into the traditional deformable mechanism to improve the performance of LA-layer. We first round up the 2D offsets Δp, then we assign the size of constraint region l, which is used to constrain the position offsets as:

$$\Delta p' = \frac{\Delta p}{max(\Delta p) + |min(\Delta p)|} l \tag{7}$$

l is typically in the order of 2–4 times the kernel size k. We perform ablations in Section 4.4.

After producing the offsets, for a target position t in local attention region $k \times k$, we perform a weighted average of values to aggregate the Key map and Query map, the result of which is combined with the Value map through multiplication. For the local Query map Q_k, Key map K_k, and Value map V_k extracted from feature maps K, Q, and V, we perform the following operation:

$$y_t = v_t \bigodot \frac{\sum (k_{t'} + \Delta p_{t'}^{'}) \bigodot q_t}{\sum (k_{t'} + \Delta p_{t'}^{'})} \tag{8}$$

where t' is the reference point index in the kernel region, y_t is the output of the LA-layer at position t, and $k_{t'}$ is the key value inside Key map K_k (see Figure 2 for an illustration).

The aggregation function is simply composed of keys and a set of learned position biases. Consequently, we effectively avoid the expensive computation and storage of the attention matrix. This not only reduces the computation and memory cost, but also maintains the spatial interaction between query and values.

3.2 Full Attention Network Implementation

We use the LA-layer to replace the convolution layers in the ResNet architecture for our full attention networks. We also compare our LA-layer performance with other state-of-the-art local attention approaches. More recent ResNet extensions such as ResNext (Xie et al., 2017) and ResNeSt (Zhang et al., 2022) also benefit from the introduction of LA-layers.

The core building block of a ResNet is a bottleneck block with a structure of a 1×1 down-projection convolution, a 3×3 spatial convolution, and a 1×1 up-projection convolution, as well as a residual connection between the input of the block and the output of the last convolution in the block. The bottleneck block is repeated multiple times across layers to form the ResNet, with the output of the current block being the input of the next.

The proposed LA-layer only replaces the 3×3 spatial convolution operation. All other settings, including the number of layers and when spatial downsampling is applied, are preserved. A comparison of networks with and without the LA-layer is shown in Table 1.

	Model	**FLOPs** (G)	**Params (M)** (M)	**Top-1** Acc.(%)	**Top-5** Acc.(%)
Backbones	ResNet-50 (He et al., 2016)	4.1	25.5	77.3	93.6
	ResNet-101	7.9	44.6	78.5	94.2
	ResNeXt-50 (Xie et al., 2017)	4.2	25.0	78.2	93.9
	ResNeXt-101	8.0	44.2	79.1	94.4
Local attention	Stand-Alone-50Ramachandran et al. (2019)	3.6	18.0	77.6	-
	Swin-ResNet-50 (Park & Kim, 2022)	4.3	32.7	78.4	94.0
	DAT-ResNet-50 (Xia et al., 2022)	4.3	32.7	78.9	94.5
	LR-Net-50 (Hu et al., 2019)	4.3	23.3	77.3	93.6
	LR-Net-101	8.0	42.0	78.5	94.3
	AA-ResNet-50 (Bello et al., 2019)	4.2	25.8	77.7	93.8
	AA-ResNet-101	8.1	45.4	78.7	94.4
	CoTNet-50 (Li et al., 2022)	3.3	22.2	79.2	94.5
	CoTNet-101	6.1	38.3	80.0	94.9
	CoTNeXt-50	4.3	30.1	79.5	94.5
	CoTNeXt-101	8.2	53.4	80.3	95.0
LA-layer	LA-ResNet-50 (**ours**)	3.4	21.0	79.7 (+2.4)	94.9 (+1.3)
	LA-ResNet-101 (**ours**)	5.7	35.9	81.0 (+2.5)	95.7 (+1.5)
	LA-ResNeXt-50 (**ours**)	4.2	24.2	79.6 (+1.4)	94.4 (+0.5)
	LA-ResNeXt-101 (**ours**)	7.6	39.5	**81.5** (+2.4)	**95.8** (+1.4)

Table 2: ImageNet-1K image classification on ResNet and ResNext backbones. Comparison with the state-of-the-art local attention approaches. For LA models, the difference with the original ResNet/ResNext models is shown in parentheses. Best results in **bold**.

4 Experiments

We empirically validate our LA-layer on several common computer vision tasks: image classification, object detection, and semantic segmentation (Sections 4.1-4.3). We conduct an ablation study (Section 4.4), and present qualitative results of our LA-layer to better understand how our LA-layer lead to the improved performance of ResNets (Section 4.5).

4.1 ImageNet Classification

Setup: We perform experiments on ImageNet-1K image classification (Russakovsky et al., 2015), which contains 1.28M training images and 50k test images. For the LA-layer, we use a kernel size of $k = 3$, constraint region size of $L = 7$ and $m = 8$ attention heads. During training, we adopt the

training setup as in Li et al. (2022). Specifically, we perform SGD optimization with a batch size of 512 on 8 GPUs for all experiments. We train our models for 100 epochs, with a weight decay of 0.0001 and a momentum of 0.9. For the first five epochs, the initial learning rate is 0.4 with linear warm-up in the first 5 epochs, and the learning rate is further decayed via a cosine schedule (Loshchilov & Hutter, 2016).

To demonstrate the generalization of the LA-layer, we replace the convolution layers by our local attention layer in ResNet-50, ResNet-101, ResNeXt-50, and ResNeXt-101. The corresponding networks are referred to as LA-ResNet or LA-ResNeXt.

Results: Table 2 shows the results of the full attention ResNet (LA-ResNet) compared to the convolution baseline and other state-of-the-art local attention approaches. Compared to the ResNet-50 baseline, the full attention LA-ResNet-50 achieves 2.4% higher classification accuracy (from 77.3% to 79.7%), while having 17.1% fewer floating point operations (FLOPs) and 17.6% fewer parameters to train. This performance gain is consistent for ResNet-101 (+2.5%), ResNeXt-50 (+1.4%) and ResNet-101 (+2.4%). Compared with other local attention approaches, our LA-layer outperforms all local attention methods with comparable FLOPs or parameters. For example, LA-ResNet-50 achieves higher top-1 accuracy (+1.3%) than Swin-ResNet-50 (Park & Kim, 2022) with 20% fewer FLOPs. With similar FLOPs, our LA-ResNets outperform LR-Nets (Hu et al., 2019) by 2.4-2.5%.

4.2 Object Detection

Setup: To understand how our local attention layer performs on a more fine-grained task, we experiment on object detection on MS COCO (Lin et al., 2014). We use LA-ResNets pretrained on ImageNet-1K as backbone with Faster R-CNN (He et al., 2017) and Cascade R-CNN (Cai & Vasconcelos, 2018) as the detection heads, and the standard AP metric of single scale is adopted for evaluation. For a fair comparison, we follow the method in Zhang et al. (2022) and train our models on the COCO-2017 training set (118K images) and evaluate them on COCO-2017 validation set (5K images). During the training process, the 1x learning rate schedule is utilized, and the size of the shorter side is sampled from the range [640, 800] for each input image during the data augmentation process. All the other hyper-parameters remain the same for fair comparison with backbones.

Method	Model	FLOPs	AP	AP_{50}	AP_{75}
Faster R-CNN	ResNet-50 (He et al., 2016)	180G	39.34	59.47	42.76
	ResNet-101	246G	41.46	61.99	45.38
	ResNeXt-50 (Xie et al., 2017)	279G	41.31	62.23	44.91
	ResNeXt-101	406G	42.91	63.77	46.89
	ResNeSt-50 (Zhang et al., 2022)	291G	42.39	63.73	46.02
	ResNeSt-101	422G	44.13	61.91	47.67
	LA-ResNet-50 (**ours**)	164G	44.28 (+4.94)	64.81 (+5.34)	47.98 (+5.22)
	LA-ResNet-101 (**ours**)	215G	46.63 (+5.19)	67.25(+5.26)	49.34(+3.96)
	LA-ResNeXt-50 (**ours**)	274G	44.60 (+3.29)	65.76(+3.53)	48.10(+3.19)
	LA-ResNeXt-101 (**ours**)	384G	**46.71** (+3.80)	**67.43**(+3.66)	**50.75**(+3.86)
Cascade R-CNN	ResNet-50 (He et al., 2016)	201G	42.45	59.76	46.09
	ResNet-101	274G	44.13	61.91	47.67
	ResNeXt-50 (Xie et al., 2017)	313G	44.53	62.45	48.38
	ResNeXt-101	422G	45.83	63.61	49.89
	ResNeSt-50 (Zhang et al., 2022)	336G	45.41	63.92	48.70
	ResNeSt-101	451G	47.51	66.06	51.35
	LA-ResNet-50 (**ours**)	173G	46.81 (+4.36)	64.80 (+5.04)	50.16 (+4.07)
	LA-ResNet-101 (**ours**)	226G	48.83 (+4.70)	67.06 (+5.15)	52.61 (+4.94)
	LA-ResNeXt-50 (**ours**)	301G	47.56 (+3.03)	65.43 (+2.98)	50.96 (+2.58)
	LA-ResNeXt-101 (**ours**)	399G	**49.64** (+3.81)	**69.25** (+5.64)	**53.39**(+3.50)

Table 3: Object detection results on MS COCO validation set, with Fast R-CNN and Cascade R-CNN detection heads. For LA models, the difference with the original ResNet/ResNext models is shown in parentheses. Best results for each detection head in bold.

Results. We summarize our results in Table 3. Our LA-layer consistently shows better performance than the baselines with either a Fast R-CNN or Cascase R-CNN detection head. For example, compared to ResNeXt-101, our LA-ResNext-101 achieves 3.29% higher AP with Fast R-CNN, and 3.03% higher AP with Cascade R-CNN. The number of FLOPs for all methods is comparable.

Again, these results demonstrate that the improvements are not achieved by using a more complex model, but instead from the ability to encode more informative features.

4.3 INSTANCE SEGMENTATION

Setup: We also evaluate the effectiveness of the LA-layer on a challenging scene parsing dataset: ADE20K (Zhou et al., 2017). We use DeepLabV3 (Chen et al., 2017) as the instance segmentation approach with ResNet and ResNext backbones pretrained on ImageNet-1K. A resolution of 512×2048 is used, and we report the pixel accuracy (pixAcc) and mean intersection-of-union (mIoU) for comparison with backbones. For a fair comparison, we follow the method in Zhang et al. (2022) and train the models for 120 epochs on the ADE20K training set (20K images). The trained networks are evaluated on the ADE20K validation set.

Results: Results are shown in Table 4. We observe that networks with LA-layers consistently outperform their ResNet and ResNeSt baselines with solely convolution layers for both pixel accuracy (Pix Acc.) and mean intersection-of-union (mIoU). The improvements for pixel accuracy are modest but consistent. For the mIoU, the improvements are a bit higher. This further validates the effectiveness of our LA-layer when applied to downstream tasks.

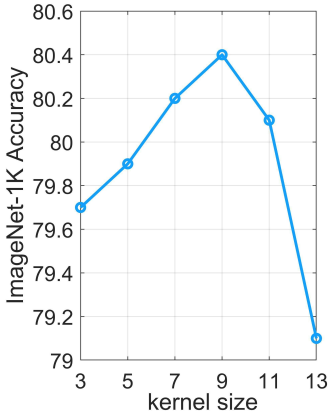

Figure 3: Ablation study of kernel size

Backbone	Pix Acc.	mIoU
ResNet-50 (He et al., 2016)	80.39	42.10
ResNet-101	81.11	44.14
ResNeSt-50 (Liu et al., 2021)	81.17	45.12
ResNeSt-101	82.07	46.91
LA-ResNet-50 (**ours**)	80.91 (+0.42)	43.31 (+1.21)
LA-ResNet-101 (**ours**)	81.73 (+0.62)	45.46 (+1.32)
LA-ResNeSt-50 (**ours**)	81.70 (+0.53)	46.22 (+1.10)
LA-ResNeSt-101 (**ours**)	**82.59** (+0.52)	**48.02** (+1.11)

Table 4: Results for semantic segmentation on ADE20K validation split. For LA models, the difference with the original ResNet/ResNeSt models is shown in parentheses. Best results in **bold**.

4.4 ABLATION STUDY

Effect of kernel size. We investigate how much the kernel size $k \times k$ of the LA-layer contributes to the overall accuracy through an ablation study on the ImageNet-1K dataset. We use the LA-layer to build a full attention network with the same hyper-parameters as ResNet-50, and compare with the same constraint region ($L = 7$). As shown in Figure 3, the best performance is obtained when kernel size is 9×9, while the performance decreases sharply for $k > 11$. This suggests that the larger receptive field due to the larger kernel allows to extract more informative features. However, larger kernels may also introduce additional features as noise to lower the performance. For example, on the segmentation task for dogs, if there are two dogs in the same image, similar features in different dogs may confuse the network and thus decrease the performance.

The effect of locality constraint We also investigate the effect of the value of the locality constraint l on the overall accuracy through an ablation study on the ImageNet-1K dataset. We use the same hyper-parameters as before. We vary l from 1 to 11, while the kernel size is fixed at 3×3. Figure 4 shows that the accuracy gradually increases until a constraint region size of 7, after which it sharply decreases. A possible assumption for this situation could be that the model weight for some specific features is much higher than for others. When increasing the offset, kernels in different areas tend to concentrate on these specific features and produce higher weights, thus lower the receptive field of the network to some extent. In this case, informative features with lower weights are neglected, which makes the model difficult to collect enough information, so that the performance starts to

decrease. We therefore conclude that a good balance between focusing on more globally relevant features and including more local features is optimal. Our LA-layer achieves just this balance.

4.5 Qualitative Analysis of LA-layer Attention Map

To understand qualitatively how the LA-layer facilitates the extraction of spatially distributed image patterns, we use Grad-CAM (Selvaraju et al., 2017) to visualize which parts of the input a trained model attends to. We compare ResNet-50, DAT-ResNet-50, and LA-ResNet-50 models trained on the image classification task on ImageNet-1K. Figure 5 shows heatmaps for these models on three random samples. Both DAT and LA-layer models attend to more of the object regions than ResNet, which explains the higher performance of these models (see Table 2). For DAT, since different pixels in the same group use shared query offsets, the network tends to concentrate on specific regions, which limits the receptive field. LA-layer covers more of the object of interest, which allows for the integration of a multitude of informative features to contribute to the final classification.

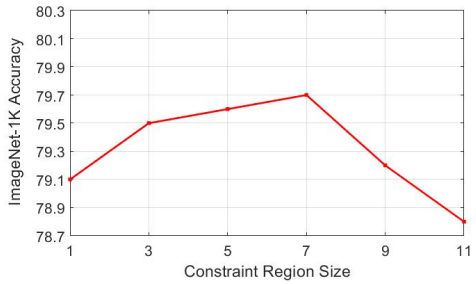

Figure 4: Ablation study of constraint region size

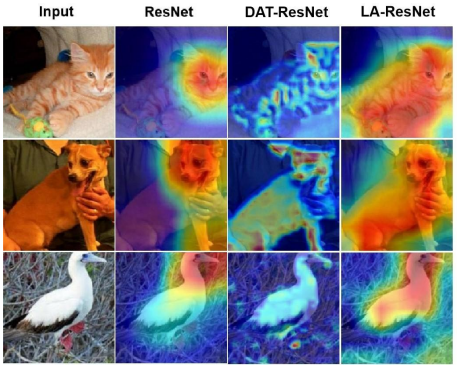

Figure 5: Grad-CAM visualizations for three images

5 Conclusion

This paper introduces a novel local attention layer (LA-layer), a basic image feature extractor that overcomes the short-range problem of convolution layers and addresses limitations of self-attention layers. LA-layers can straightforwardly replace convolution layers to obtain full attention models. Experimentation on ImageNet-1K image classification demonstrates improved performance over both original ResNet and ResNext backbones, as well as on current state-of-the-art models with self-attention. LA-layers require fewer parameters and FLOPs than related methods. This suggests that the improved performance is due to the extraction of more informative features, rather than being the result of a more complex model. On object detection (MS COCO) and instance segmentation (ADE20K) tasks, models with LA-layers also show significantly better performance compared to the original networks. We expect that these performance gains of the LA-layer also extend to more complex CNN architectures.

References

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3286–3295, 2019.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4013–4022, 2018.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

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*, pp. 11963–11975, 2022.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Changhee Han, Hideaki Hayashi, Leonardo Rundo, Ryosuke Araki, Wataru Shimoda, Shinichi Muramatsu, Yujiro Furukawa, Giancarlo Mauri, and Hideki Nakayama. GAN-based synthetic brain MR image generation. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 734–738. IEEE, 2018.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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.

Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3464–3473, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Vladimir Iglovikov, Selim Seferbekov, Alexander Buslaev, and Alexey Shvets. TernausNetV2: Fully convolutional network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 233–237, 2018.

Md Amirul Islam, Sen Jia, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations*, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual transformer networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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*, pp. 740–755. Springer, 2014.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3202–3211, 2022.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

Hughes Perreault, Guillaume-Alexandre Bilodeau, Nicolas Saunier, and Maguelonne Héritier. Spotnet: Self-attention multi-task network for object detection. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pp. 230–237. IEEE, 2020.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

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*, 115(3):211–252, 2015.

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 *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

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*, pp. 5998–6008, 2017.

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4794–4803, 2022.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

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

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pp. 7354–7363. PMLR, 2019.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. ResNeSt: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2736–2746, 2022.

Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10076–10085, 2020.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.