ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformers have shown great potential in various computer vision tasks owing to their strong capability in modeling long-range dependency using the self-attention mechanism. Nevertheless, vision transformers treat an image as 1D sequence of visual tokens, lacking an intrinsic inductive bias (IB) in modeling local visual structures and dealing with scale variance. Alternatively, they require large-scale training data and longer training schedules to learn the IB implicitly. In this paper, we propose a novel **Vi**sion **T**ransformer **A**dvanced by **E**xploring intrinsic IB from convolutions, *i.e.*, ***ViTAE***. Technically, ViTAE has several spatial pyramid reduction modules to downsample and embed the input image into tokens with rich multi-scale context by using multiple convolutions with different dilation rates. In this way, it acquires an intrinsic scale invariance IB and is able to learn robust feature representation for objects at various scales. Moreover, in each transformer layer, ViTAE has a convolution block in parallel to the multi-head self-attention module, whose features are fused and fed into the feed-forward network. Consequently, it has the intrinsic locality IB and is able to learn local features and global dependencies collaboratively. Experiments on ImageNet as well as downstream tasks prove the superiority of ViTAE over the baseline transformer and concurrent works. Source code and pretrained models will be made public.

1 Introduction

Transformers [65, 13, 29, 10, 35, 47] have shown a domination trend in NLP studies owing to their strong ability in modeling long-range dependencies by the self-attention mechanism [53, 67, 40]. Such success and good properties of transformers has inspired following many works that apply them in various computer vision tasks [15, 81, 80, 66, 6]. Among them, ViT [15] is the pioneering pure transformer model that embeds images into a sequence of visual tokens and models the global dependencies among them with stacked transformer blocks. Although it achieves promising performance on image classification, it requires large-scale training data and a longer training schedule. One important reason is that ViT lacks intrinsic inductive bias (IB) in modeling local visual structures (*e.g.*, edges and corners) and dealing with objects at various scales like convolutions. Alternatively, ViT has to learn such IB implicitly from large-scale data.

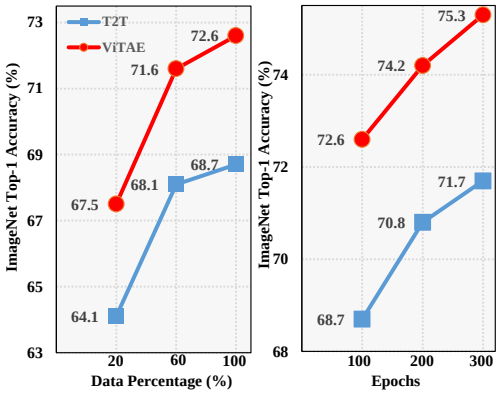

Figure 1: Comparison of data and training efficiency of T2T-ViT and ViTAE on ImageNet.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Unlike vision transformers, Convolution Neural Networks (CNNs) naturally equip with the intrinsic IBs of scale-invariance and locality and still serve as prevalent backbones in vision tasks [20, 56, 48, 7, 79]. The success of CNNs inspires us to explore intrinsic IBs in vision transformers. We start by analyzing the above two IBs of CNNs, *i.e.*, locality and scale-invariance. Convolution that computes local correlation among neighbor pixels is good at extracting local features such as edges and corners. Consequently, CNNs can provide plentiful low-level features at the shallow layers [77], which are then aggregated into high-level features progressively by a bulk of sequential convolutions [23, 54, 57]. Moreover, CNNs have a hierarchy structure to extract multi-scale features at different layers [54, 27, 20]. Besides, intra-layer convolutions can also learn features at different scales by varying their kernel sizes and dilation rates [19, 56, 7, 34, 79]. Consequently, scale-invariant feature representation can be obtained via intra- or inter-layer feature fusion. Nevertheless, CNNs are not well suited to model long-range dependencies[1], which is the key advantage of transformers. An interesting question comes up: Can we improve vision transformers by leveraging the good properties of CNNs? Recently, DeiT [62] explores the idea of distilling knowledge from CNNs to transformers to facilitate training and improve the performance. However, it requires an off-the-shelf CNN model as the teacher and consumes extra training cost.

Different from DeiT, we explicitly introduce intrinsic IBs into vision transformers by re-designing the network structures in this paper. Current vision transformers always obtain tokens with single-scale context [15, 76, 66, 70, 36, 55, 63] and learn to adapt to objects at different scales from data. For example, T2T-ViT [76] improves ViT by delicately generating tokens in a soft split manner. Specifically, it uses a series of Tokens-to-Token transformation layers to aggregate single-scale neighboring contextual information and progressively structurizes the image to tokens. Motivated by the success of CNNs in dealing with scale variance, we explore a similar design in transformers, *i.e.*, intra-layer convolutions with different receptive fields [56, 74], to embed multi-scale context into tokens. Such a design allows tokens to carry useful features of objects at various scales, thereby naturally having the intrinsic scale-invariance IB and explicitly facilitating transformers to learn scale-invariant features more efficiently from data. On the other hand, low-level local features are fundamental elements to generate high-level discriminative features. Although transformers can also learn such features at shallow layers from data, they are not skilled as convolutions by design. Recently, [72, 32, 16] stack convolutions and attention layers sequentially and demonstrate that locality is a reasonable compensation of global dependency. However, this serial structure ignores the global context during locality modeling (and vice versa). To avoid such a dilemma, we follow the "divide-and-conquer" idea and propose to model locality and long-range dependencies in parallel and then fuse the features to account for both. In this way, we empower transformers to learn local and long-range features within each block more effectively.

Technically, we propose a novel **Vi**sion **T**ransformers **A**dvanced by **E**xploring Intrinsic Inductive Bias (***ViTAE***), which is a combination of two types of basic cells, *i.e.*, reduction cell (RC) and normal cell (NC). RCs are used to downsample and embed the input images into tokens with rich multi-scale context while NCs aim to jointly model locality and global dependencies in the token sequence. Moreover, these two types of cells share a simple basic structure, *i.e.*, paralleled attention module and convolutional layers followed by a feed-forward network (FFN). It is noteworthy that RC has an extra pyramid reduction module with atrous convolutions of different dilation rates to embed multi-scale context into tokens. Following the setting in [76], we stack three reduction cells to reduce the spatial resolution by $1/16$ and a series of NCs to learn discriminative features from data. ViTAE outperforms representative vision transformers in terms of data efficiency and training efficiency (see Figure 1), as well as classification accuracy and generalization on downstream tasks.

Our contributions are threefold. **First**, we explore two types of intrinsic IB in transformers, *i.e.*, scale invariance and locality, and demonstrate the effectiveness of this idea in improving the feature learning ability of transformers. **Second**, we design a novel transformer architecture named ViTAE based on two new reduction and normal cells to intrinsically incorporate the above two IBs. The proposed ViTAE embeds multi-scale context into tokens and learns both local and long-range features effectively. **Third**, ViTAE outperforms representative vision transformers regarding classification accuracy, data efficiency, training efficiency, and generalization on downstream tasks. ViTAE achieves 75.3% and 82.0% top-1 accuracy on ImageNet with 4.8M and 23.6M parameters, respectively.

[1] Despite projection in transformer can be viewed as 1×1 convolution [8], the term of convolution here refers to those with larger kernels, *e.g.*, 3×3, which are widely used in typical CNNs to extract spatial features.

2 Related Work

2.1 CNNs with intrinsic IB

CNNs have led to a series of breakthroughs in image classification [27, 77, 20, 78, 71] and downstream computer vision tasks. The convolution operations in CNNs extract local features from the neighbor pixels within the receptive field determined by the kernel size [31]. Following the intuition that local pixels are more likely to be correlated in images [30], CNNs have the intrinsic IB in modeling locality. In addition to the locality, another critical topic in visual tasks is scale-invariance, where multi-scale features are needed to represent the objects at different scales effectively [38, 73]. For example, to effectively learn features of large objects, a large receptive field is needed by either using large convolution kernels [73, 74] or a series of convolution layers in deeper architectures [20, 23, 54, 57]. To construct multi-scale feature representation, the classical idea is using image pyramid [7, 1, 43, 4, 28, 12], where features are hand-crafted or learned from a pyramid of images at different resolutions respectively [33, 7, 41, 49, 24, 3]. Accordingly, features from the small scale image mainly encode the large objects while features from the large scale image respond more to small objects. In addition to the above inter-layer fusion way, another way is to aggregate multi-scale context by using multiple convolutions with different receptive fields within a single layer, *i.e.*, intra-layer fusion [79, 57, 56, 56, 58]. Either inter-layer fusion or intra-layer fusion empower CNNs an intrinsic IB in modeling scale-invariance. This paper introduces such an IB to vision transformers by following the intra-layer fusion idea and utilizing multiple convolutions with different dilation rates in the reduction cells to encode multi-scale context into each visual token.

2.2 Vision transformers with learned IB

ViT [15] is the pioneering work that applies a pure transformer to vision tasks and achieves promising results. However, since ViT lacks intrinsic inductive bias in modeling local visual structures, it indeed learns the IB from amounts of data implicitly. Following works along this direction are to simplify the model structures with fewer intrinsic IBs and directly learn them from large scale data [39, 60, 61, 17, 14], which have achieved promising results and been studied actively. Another direction is to leverage the intrinsic IB from CNNs to facilitate the training of vision transformers, *e.g.*, using less training data or shorter training schedules. For example, DeiT [62] proposes to distill knowledge from CNNs to transformers during training. However, it requires an off-the-shelf CNN model as a teacher, introducing extra computation cost during training. Recently, some works try to introduce the intrinsic IB of CNNs into vision transformers explicitly [18, 46, 16, 32, 11, 72, 69, 75, 5, 36, 9]. For example, [32, 16, 69] stack convolutions and attention layers sequentially, resulting in a serial structure and modeling the locality and global dependency accordingly. However, this serial structure may ignore the global context during locality modeling (and vice versa). Instead, we follow the "divide-and-conquer" idea and propose to model locality and global dependencies simultaneously via a parallel structure within each transformer layer. Conformer [46], the most relevant concurrent work to us, employs a unit to explore inter-block interactions between parallel convolution and transformer blocks. In contrast, in ViTAE, the convolution and attention modules are designed to be complementary to each other within the transformer block. In addition, Conformer is not designed to have inherent scale invariance IB.

3 Methodology

3.1 Revisit vision transformer

We first give a brief review of vision transformer in this part. To adapt transformers to vision tasks, ViT [15] first splits an image $x \in R^{H \times W \times C}$ into tokens with a reduction ratio of p (*i.e.*, $x_t \in R^{((H \times W)/p^2) \times D}$), where H, W and C denote the height, width, and channel dimensions of the input image, $D = Cp^2$ denotes the token dimension. Then, an extra class token is concatenated to the visual tokens before adding position embeddings in an element-wise manner. The resulting tokens are fed into the following transformer layers. Each transformer layer is composed of two parts, *i.e.*, a multi-head self-attention module (MHSA) and a feed forward network (FFN).

MHSA Multi-head self-attention extends single-head self-attention (SHSA) by using different projection matrices for each head. Specifically, the input tokens x_t are first projected to queries (Q),

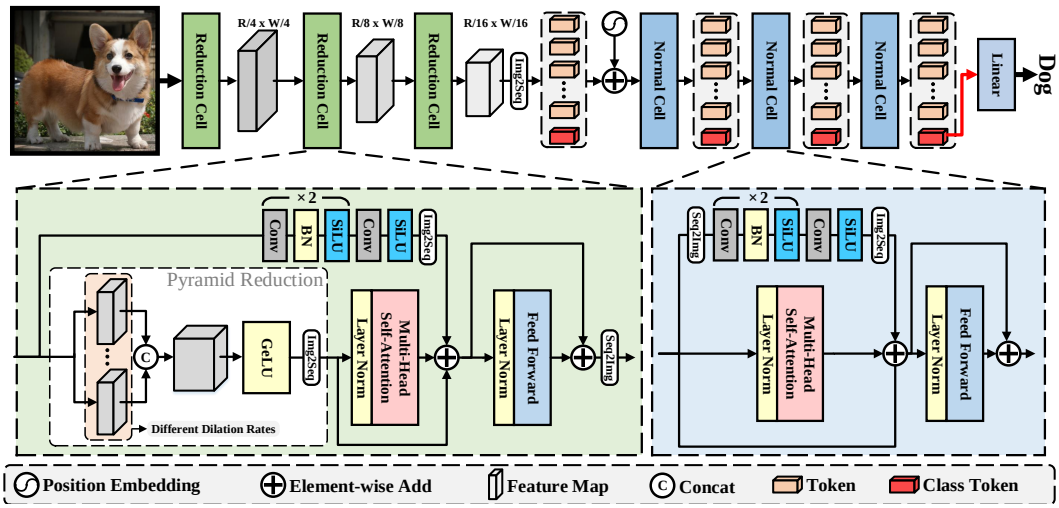

Figure 2: The structure of the proposed ViTAE. It is constructed by stacking three RCs and several NCs. Both types of cells share a simple basic structure, *i.e.*, an MHSA module and a parallel convolutional module followed by an FFN. In particular, RC has an extra pyramid reduction module using atrous convolutions with different dilation rates to embed multi-scale context into tokens.

keys (K) and values (V) using projection matrices, *i.e.*, $Q, K, V = x_t W_Q, x_t Q_K, x_t Q_V$, where $W_{Q/K/V} \in R^{D \times D}$ denotes the projection matrix for query, key, and value, respectively. Then, the self-attention operation is calculated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D}})V. \tag{1}$$

This SHSA module is repeated for h times to formulate the MHSA module, where h is the number of heads. The output features of the h heads are concatenated along the channel dimension and formulate the output of the MHSA module.

FFN FFN is placed on top of the MHSA module and applied to each token identically and separately. It consists of two linear transformations with an activation function in between. Besides, a layer normalization [2] and a shortcut are added before and aside from the MHSA and FFN, respectively.

3.2 Overview architecture of ViTAE

ViTAE aims to introduce the intrinsic IB in CNNs to vision transformers. As shown in Figure 2, ViTAE is composed of two types of cells, *i.e.*, RCs and NCs. RCs are responsible for embedding multi-scale context and local information into tokens, and NCs are used to further model the locality and long-range dependencies in the tokens. Taken an image $x \in R^{H \times W \times C}$ as input, three RCs are used to gradually downsample x by $4\times$, $2\times$, and $2\times$, respectively. Thereby, the output tokens of the RCs are of size $[H/16, W/16, D]$ where D is the token dimension (64 in our experiments). The output tokens of RCs are then flattened as $R^{HW/256 \times D}$, concatenated with the class token, and added by the sinusoid position encoding. Next, the tokens are fed into the following NCs, which keep the length of the tokens. Finally, the prediction probability is obtained using a linear classification layer on the class token from the last NC.

3.3 Reduction cell

Instead of directly splitting and flatten images into visual tokens based on a linear image patch embedding layer, we devise the reduction cell to embed multi-scale context and local information into visual tokens, which introduces the intrinsic scale-invariance and locality IBs from convolutions. Technically, RC has two parallel branches responsible for modeling locality and long-range dependency, respectively, followed by an FFN for feature transformation. We denote the input feature of

the i_{th} RC as $f_i \in R^{H_i \times W_i \times D_i}$. The input of the first RC is the image x. In the global dependencies branch, f_i is firstly fed into a Pyramid Reduction Module (PRM) to extract multi-scale context, *i.e.*,

$$f_i^{ms} \triangleq PRM_i(f_i) = Cat([Conv_{ij}(f_i; s_{ij}, r_i) | s_{ij} \in \mathcal{S}_i, r_i \in \mathcal{R}]), \tag{2}$$

where $Conv_{ij}(\cdot)$ indicates the jth convolutional layer in the PRM ($PRM_i(\cdot)$). It uses a dilation rate s_{ij} from the predefined dilation rate set $\mathcal{S}_i$ corresponding to the ith RC. Note that we use stride convolution to reduce the spatial dimension of features by a ratio r_i from the predefined reduction ratio set $\mathcal{R}$. The conv features are concatenated along the channel dimension, *i.e.*, $f_i^{ms} \in R^{(W_i/p) \times (H_i/p) \times (|\mathcal{S}_i| D)}$, where $|\mathcal{S}_i|$ denotes the number of dilation rates in $\mathcal{S}_i$. f_i^{ms} is then processed by an MHSA module to model long-range dependencies, *i.e.*,

$$f_i^g = MHSA_i(Img2Seq(f_i^{ms})), \tag{3}$$

where $Img2Seq(\cdot)$ is a simple reshape operation to flatten the feature map to a 1D sequence. In this way, f_i^g embeds the multi-scale context in each token. In addition, we use a Parallel Convolutional Module (PCM) to embed local context within the tokens, which are fused with f_i^g as follows:

$$f_i^{lg} = f_i^g + PCM_i(f_i). \tag{4}$$

Here, $PCM_i(\cdot)$ represents the PCM, which is composed of three stacked convolution layers and an $Img2Seq(\cdot)$ operation. It is noteworthy that the parallel convolution branch has the same spatial downsampling ratio as the PRM by using stride convolutions. In this way, the token features can carry both local and multi-scale context, implying that RC acquires the locality IB and scale-invariance IB by design. The fused tokens are then processed by the FFN, reshaped back to feature maps, and fed into the following RC or NC, *i.e.*,

$$f_{i+1} = Seq2Img(FFN_i(f_i^{lg}) + f_i^{lg}), \tag{5}$$

where the $Seq2Img(\cdot)$ is a simple reshape operation to reshape a token sequence back to feature maps. $FFN_i(\cdot)$ represents the FFN in the ith RC. In our ViTAE, three RCs are stacked sequentially to gradually reduce the input image's spatial dimension by 4×, 2×, and 2×, respectively. The feature maps generated by the last RC are of a size of $[H/16, W/16, D]$, which are then flattened into visual tokens and fed into the following NCs.

3.4 Normal cell

As shown in the bottom right part of Figure 2, NCs share a similar structure with the reduction cell except for the absence of the PRM. Due to the relatively small ($\frac{1}{16}\times$) spatial size of feature maps after RCs, it is unnecessary to use PRM in NCs. Given f_3 from the third RC, we first concatenate it with the class token t_{cls}, and then add it to the positional encodings to get the input tokens t for the following NCs. Here we ignore the subscript for clarity since all NCs have an identical architecture but different learnable weights. t_{cls} is randomly initialized at the start of training and fixed during the inference. Similar to the RC, the tokens are fed into the MHSA module, *i.e.*, $t_g = MHSA(t)$. Meanwhile, they are reshaped to 2D feature maps and fed into the PCM, *i.e.*, $t_l = Img2Seq(PCM(Seq2Img(t)))$. Note that the class token is discarded in PCM because it has no spatial connections with other visual tokens. To further reduce the parameters in NCs, we use group convolutions in PCM. The features from MHSA and PCM are then fused via element-wise sum, *i.e.*, $t_{lg} = t_g + t_l$. Finally, t_{lg} are fed into the FFN to get the output features of NC, *i.e.*, $t_{nc} = FFN(t_{lg}) + t_{lg}$. Similar to ViT [15], we apply layer normalization to the class token generated by the last NC and feed it to the classification head to get the final classification result.

3.5 Model details

We use two variants of ViTAE in our experiments for a fair comparison of other models with similar model sizes. The details of them are summarized in Table 1. In the first RC, the default convolution kernel size is 7×7 with a stride of 4 and dilation rates of $\mathcal{S}_1 = [1, 2, 3, 4]$. In the following two RCs, the convolution kernel size is 3×3 with a stride of 2 and dilation rates of $\mathcal{S}_2 = [1, 2, 3]$ and $\mathcal{S}_3 = [1, 2]$, respectively. Since the spatial dimension of tokens decreases, there is no need to use large kernels and dilation rates. PCM in both RCs and NCs comprises three convolutional layers with a kernel size of 3×3.

Table 1: Model details of two variants of ViTAE.

Model	Reduction Cell		Normal Cell			Params	Macs
	Dilation	Cells	Heads	Embed	Cells	(M)	(G)
ViTAE-T	$[1, 2, 3, 4] \downarrow$	3	4	256	7	4.8	1.5
ViTAE-S	$[1, 2, 3, 4] \downarrow$	3	6	384	14	23.6	5.6

4 Experiments

4.1 Implementation details

We train and test the proposed ViTAE model on the standard ImageNet [27] dataset, which contains about 1.3 million images and covers 1k classes. Unless explicitly stated, the image size during training is set to 224×224. We use the AdamW [37] optimizer with the cosine learning rate scheduler and uses the data augmentation strategy exactly the same as T2T [76] for a fair comparison. We use a batch size of 512 for training all our models and set the initial learning rate to be 5e-4. The results of our models can be found in Table 2, where all the models are trained for 300 epochs on 8 V100 GPUs. The models are built on PyTorch [45] and TIMM [68].

Table 2: Comparison of ViTAE and SOTA methods on the ImageNet validation set.

Type	Model	Params (M)	MACs (G)	Input Size	ImageNet Top-1	ImageNet Top-5	Real Top-1
CNN	ResNet-18 [20]	11.7	3.6	224	70.3	86.7	77.3
	ResNet-50 [20]	25.6	7.6	224	76.7	93.3	82.5
	ResNet-101 [20]	44.5	15.2	224	78.3	94.1	83.7
	ResNet-152 [20]	60.2	22.6	224	78.9	94.4	84.1
	EfficientNet-B0 [59]	5.3	0.8	224	77.1	93.3	83.5
	EfficientNet-B4 [59]	19.3	8.4	380	82.9	96.4	88.0
	MobileNetV1 [22]	4.3	0.6	224	72.3	-	-
	MobileNetV2(1.4) [51]	6.9	0.6	224	74.7	-	-
	RegNetY-600M [48]	20.6	1.2	224	75.5	-	
	RegNetY-4GF [48]	39.2	8.0	224	80.0	-	86.4
Transformer with learned IB	ViT-B/16 [15]	86.5	18.7	384	77.9	-	83.6
	ViT-L/16 [15]	304.3	65.8	384	76.5	-	82.2
	DeiT-T [62]	5.7	2.6	224	72.2	91.1	80.6
	DeiT-S [62]	22.1	9.8	224	79.9	95.0	85.7
	DeiT-B [62]	86.6	34.6	224	81.8	95.6	86.7
	DeiT-T⚗ [62]	5.7	2.6	224	74.5	91.9	82.1
	DeiT-S⚗ [62]	22.1	9.8	224	81.2	95.4	86.8
Transformer with intrinsic IB	PVT-T [66]	13.2	3.8	224	75.1	-	-
	PVT-S [66]	24.5	7.6	224	79.8	-	-
	PVT-M [66]	44.2	13.2	224	81.2	-	-
	PVT-L [66]	61.4	19.6	224	81.7	-	-
	LocalViT-T [32]	5.9	2.6	224	74.8	92.6	-
	LocalViT-T2T [32]	4.3	2.4	224	72.5	-	-
	LocalViT-PVT [32]	13.5	9.6	224	78.2	94.2	-
	Conformer-Ti [46]	23.5	5.2	224	81.3	-	-
	Swin-T [36]	29.0	9.0	224	81.3	-	-
	ConT-Ti [72]	5.8	1.6	224	74.9	-	-
	ConT-M [72]	19.2	6.2	224	80.2	-	-
	ConT-B [72]	39.6	12.8	224	81.8	-	-
	CrossViT-Ti [5]	6.9	3.2	224	73.4	-	-
	CrossViT-S [5]	26.7	11.2	224	81.0	-	-
	T2T-ViT-7 [76]	4.3	1.2	224	71.7	90.9	79.7
	T2T-ViT-14 [76]	21.5	5.2	224	81.5	95.7	86.8
	T2T-ViT-19 [76]	39.2	8.9	224	81.9	95.7	86.9
	ViTAE-T	4.8	1.5	224	75.3	92.7	82.9
	ViTAE-T $\uparrow$ **384**	4.8	5.7	384	77.2	93.8	84.4
	ViTAE-S	23.6	5.6	224	82.0	95.9	87.0
	ViTAE-S $\uparrow$ **384**	23.6	20.2	384	83.0	96.2	87.5

4.2 Comparison with the state-of-the-art

We compare our ViTAE with both CNN models and vision transformers with similar model sizes in Table 2. Both Top-1/5 accuracy and real Top-1 accuracy on the ImageNet validation set are reported. We categorize the methods into CNN models, vision transformers with learned IB, and vision transformers with introduced intrinsic IB. Compared with CNN models, our ViTAE-T achieves a 75.3% Top-1 accuracy, which is better than ResNet-18 with more parameters. The real Top-1 accuracy of the ViTAE model is 82.9%, which is comparable to ResNet-50 that has four more times

of parameters than ours. Similarly, our ViTAE-S achieves 82.0% Top-1 accuracy with half of the parameters of ResNet-101 and ResNet-152, showing the superiority of learning both local and long-range features from specific structures with corresponding intrinsic IBs by design. Similar phenomena can also be observed when comparing ViTAE-T with MobileNetV1 [22] and MobileNetV2 [51], where ViTAE obtains better performance with fewer parameters. When compared with larger models which are searched according to NAS [59], our ViTAE-S achieves a similar performance when using 384×384 images as input, which further shows the potential of vision transformers with intrinsic IB.

In addition, among the transformers with learned IB, ViT is the first pure transformer model for visual recognition. DeiT shares the same structure with ViT but uses different data augmentation and training strategies to facilitate the learning of transformers. DeiT⚗ denotes using an off-the-shelf CNN model as the teacher model to train DeiT, which introduces the intrinsic IB from CNN to transformer implicitly in a knowledge distillation manner, showing better performance than the vanilla ViT on the ImageNet dataset. It is exciting to see that our ViTAE-T with fewer parameters even outperforms the distilled model DeiT⚗, demonstrating the efficacy of introducing intrinsic IBs in transformers by design. Besides, compared with other transformers with explicit intrinsic IB, our ViTAE with fewer parameters also achieves comparable or better performance. For instance, ViTAE-T achieves comparable performance with LocalVit-T but has 1M fewer parameters, demonstrating the superiority of the proposed RCs and NCs in introducing intrinsic IBs.

4.3 Ablation study

We use T2T-ViT [76] as our baseline model in the following ablation study of our ViTAE. As shown in Table 3, we investigate the hyper-parameter settings in RCs and NCs by isolating them separately. All the models are trained for 100 epochs on ImageNet and follow the same training setting and data augmentation strategy as described in Section 4.1.

We use ✓ and $\times$ to denote whether or not the corresponding module is enabled during the experiments. If all columns under the RC and NC are marked $\times$ as shown in the first row, the model becomes the standard T2T-ViT model. "Pre" indicates the output features of PCM and MHSA are fused before FFN while "Post" indicates a late fusion strategy correspondingly. "BN" indicates whether PCM uses BN after the convolutional layer or not. "$\times 3$" in the first column denotes that the dilation rate set is the same in the three RCs. "$[1, 2, 3, 4] \downarrow$" denotes using lower dilation rates in deeper RCs, *i.e.*, $\mathcal{S}_1 = [1, 2, 3, 4]$, $\mathcal{S}_2 = [1, 2, 3]$, $\mathcal{S}_3 = [1, 2]$.

Table 3: Ablation Study of RCs and NCs in our ViTAE. "Pre" indicates the output features of PCM and MHSA are fused before FFN while "Post" indicates a late fusion strategy correspondingly. "BN" indicates whether PCM uses BN or not. "$[1, 2, 3, 4] \downarrow$" denotes using smaller dilation rates in deeper RCs, *i.e.*, $\mathcal{S}_1 = [1, 2, 3, 4]$, $\mathcal{S}_2 = [1, 2, 3]$, $\mathcal{S}_3 = [1, 2]$.

Reduction Cell		**Normal Cell**			**Top-1**
Dilation ($\mathcal{S}_1 \sim \mathcal{S}_3$)	**PCM**	**Pre**	**Post**	**BN**	
$\times$	$\times$	$\times$	$\times$	$\times$	68.7
$\times$	$\times$	✓	$\times$	$\times$	69.1
$\times$	$\times$	$\times$	✓	$\times$	69.0
$\times$	$\times$	$\times$	✓	✓	68.8
$\times$	$\times$	✓	$\times$	✓	69.9
$[1, 2] \times 3$	$\times$	$\times$	$\times$	$\times$	69.5
$[1, 2, 3] \times 3$	$\times$	$\times$	$\times$	$\times$	69.9
$[1, 2, 3, 4] \times 3$	$\times$	$\times$	$\times$	$\times$	69.2
$[1, 2, 3, 4, 5] \times 3$	$\times$	$\times$	$\times$	$\times$	68.9
$[1, 2, 3, 4] \downarrow$	$\times$	$\times$	$\times$	$\times$	69.8
$[1, 2, 3, 4] \downarrow$	✓	$\times$	$\times$	$\times$	71.7
$[1, 2, 3, 4] \downarrow$	✓	✓	$\times$	✓	72.6

As can be seen, using a pre-fusion strategy and BN achieves the best 69.9% Top-1 accuracy among other settings. It is noteworthy that all the variants of NC outperform the vanilla T2T-ViT, implying the effectiveness of PCM, which introduces the intrinsic locality IB in transformers. For the RC, we first investigate the impact of using different dilation rates in the PRM, as shown in the first column. As can be seen, using larger dilation rates (*e.g.*, 4 or 5) does not deliver better performance. We suspect that larger dilation rates may lead to plain features in the deeper RCs due to the smaller resolution of feature maps. To validate the hypothesis, we use smaller dilation rates in deeper RCs as denoted by $[1, 2, 3, 4] \downarrow$. As can be seen, it achieves comparable performance as $[1, 2, 3] \times$. However, compared with $[1, 2, 3, 4] \downarrow$, $[1, 2, 3] \times$ increases the amount of parameters from 4.35M to 4.6M. Therefore, we select $[1, 2, 3, 4] \downarrow$ as the default setting. In addition, after using PCM in the RC, it introduces the intrinsic locality IB, and the performance increases to 71.7% Top-1 accuracy. Finally, the combination of RCs and NCs achieves the best accuracy at 72.6%, demonstrating the complementarity between our RCs and RCs.

4.4 Data efficiency and training efficiency

To validate the effectiveness of the introduced intrinsic IBs in improving data efficiency and training efficiency, we compare our ViTAE with T2T-ViT at different training settings: (a) training them using 20%, 60%, and 100% ImageNet training set for equivalent 100 epochs on the full ImageNet training set; and (b) training them using the full ImageNet training set for 100, 200, and 300 epochs respectively. The results are shown in Figure 1. As can be seen, ViTAE consistently outperforms the T2T-ViT baseline by a large margin in terms of both data efficiency and training efficiency. For example, ViTAE using only 20% training data achieves comparable performance with T2T-ViT using all data. When 60% training data are used, ViTAE significantly outperforms T2T-ViT using all data by about an absolute 3% accuracy. It is also noteworthy that ViTAE trained for only 100 epochs has outperformed T2T-ViT trained for 300 epochs. After training ViTAE for 300 epochs, its performance is significantly boosted to 75.3% Top-1 accuracy. With the proposed RCs and NCs, the transformer layers in our ViTAE only need to focus on modeling long-range dependencies, leaving the locality and multi-scale context modeling to its convolution counterparts, *i.e.*, PCM and PRM. Such a "divide-and-conquer" strategy facilitates the training of vision transformers, making it possible to learn more efficiently with less training data and fewer training epochs.

4.5 Generalization on downstream tasks

Table 4: Generalization of ViTAE and SOTA methods on different downstream tasks.

Model	Params	Cifar10	Cifar100	iNat19	Cars	Flowers	Pets
Grafit ResNet-50 [64]	25.6	-	-	75.9	92.5	98.2	-
EfficientNet-B5 [59]	30	98.1	91.1	-	-	98.5	-
ViT-B/16 [15]	86.5	98.1	87.1	-	-	89.5	93.8
ViT-L/16 [15]	304.3	97.9	86.4	-	-	89.7	93.6
DeiT-B [62]	86.6	99.1	90.8	77.7	92.1	98.4	-
T2T-ViT-14 [76]	21.5	98.3	88.4	-	-	-	-
ViTAE-T	4.8	97.3	86.0	73.3	89.5	97.5	92.6
ViTAE-S	23.6	98.8	90.8	76.0	91.4	97.8	94.2

We further investigate the generalization of the proposed ViTAE models on downstream tasks by fine-tuning them on the training sets of several fine-grained classification tasks, including Flowers [42], Cars [25], Pets [44], and iNaturalist19. We also fine-tune the proposed ViTAE models on Cifar10 [26] and Cifar100 [26]. The results are shown in Table 4. It can be seen that ViTAE achieves SOTA performance on most of the datasets using comparable or fewer parameters. These results demonstrate that the good generalization ability of our ViTAE.

4.6 Visual inspection of ViTAE

To further analyze the property of our ViTAE, we first calculate the average attention distance of each layer in ViTAE-T and the baseline T2T-ViT-7 on the ImageNet test set, respectively. The results are shown in Figure 3. It can be observed that with the usage of PCM, which focuses on modeling locality, the transformer layers in the proposed NCs can better focus on modeling long-range dependencies, especially in shallow layers. In the deep layers, the average attention distances of ViTAE-T and T2T-ViT-7 are almost the same since modeling long-range dependencies is much more important. These results confirm the effectiveness of the adopted "divide-and-conquer" idea in the proposed ViTAE, *i.e.*, introducing the intrinsic locality IB from convolutions into vision transformers makes it possible that transformer layers only need to be responsible to long-range dependencies, since locality can be well modeled by convolutions in PCM.

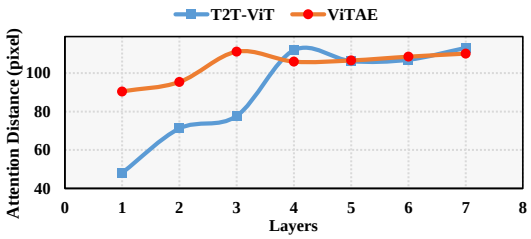

Figure 3: The average per-layer attention distance of T2T-ViT-7 and our ViTAE-T.

Besides, we apply Grad-CAM [52] on the MHSA's output in the last NC to qualitatively inspect ViTAE. The visualization results are provided in Figure 4. Compared with the baseline T2T-ViT, our ViTAE covers the single or multiple targets in the images more precisely and attends less to the

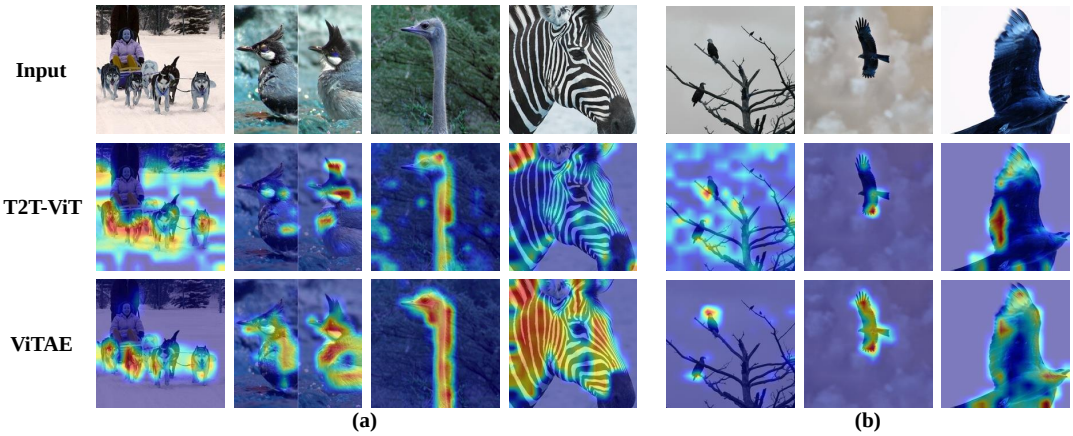

Figure 4: Visual inspection of T2T-ViT and ViTAE using Grad-CAM [52]. (a) Images containing multiple or single objects and the heatmaps. (b) Images containing the same class of objects at different scales and the heatmaps (Best viewed in color).

background. Moreover, ViTAE can better handle the scale variance issue as shown in Figure 4(b). Namely, it can precisely cover the birds no matter they are in small, middle, or large size. Such observations demonstrate that introducing the intrinsic IBs of locality and scale-invariance from convolutions to transformers helps ViTAE learn more discriminate features than the pure transformers.

5 Limitation and discussion

In this paper, we explore two types of IBs and incorporate them into transformers through the proposed reduction and normal cells. With the collaboration of these two cells, our ViTAE model achieves impressive performance on the ImageNet with fast convergence and high data efficiency. Nevertheless, due to computational resource constraints, we have not scaled the ViTAE model and train it on large-size dataset, *e.g.*, ImageNet-21K [27] and JFT-300M [21]. Although it remains unclear by now, we are optimistic about its scale property from the following preliminary evidence. As illustrated in Figure 2, our ViTAE model can be viewed as an intra-cell ensemble of complementary transformer layers and convolution layers owing to the skip connection and parallel structure. According to the attention distance analysis shown in Figure 3, the ensemble nature enables the transformer layers and convolution layers to focus on what they are good at, *i.e.*, modeling long-range dependencies and locality. Therefore, ViTAE is very likely to learn better feature representation from large-scale data. Besides, we only study two typical IBs in this paper. More kinds of IBs such as constituting viewpoint invariance [50] can be explored in the future study.

6 Conclusion

In this paper, we re-design the transformer block by proposing two novel basic cells (reduction cells and normal cells) to incorporate two types of intrinsic inductive bias (IB) into transformers, *i.e.*, locality and scale-invariance, resulting in a simple yet effective vision transformer architecture named ViTAE. Extensive experiments show that ViTAE outperforms representative vision transformers in various respects including classification accuracy, data efficiency, training efficiency, and generalization ability on downstream tasks. We plan to scale ViTAE to the large or huge model size and train it on large-size datasets in the future study. In addition, other kinds of IBs will also be investigated. We hope that this study will provide valuable insights to the following studies of introducing intrinsic IB into vision transformers and understanding the impact of intrinsic and learned IBs.

Broad impacts This paper proposes a novel and effective vision transformer that can be used as backbone networks in many computer vision areas. However, like other basic network architectures, the proposed ViTAE may learn inappropriate capabilities from biased training data, for example, discriminatory outputs for specific demographic groups (*e.g.*, discrimination against gender, race, age) and compromise personal privacy. How to prevent models from exhibiting algorithmic discrimination and compromising personal privacy remains open and challenging.

References

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[4] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier, 1987.

[5] C.-F. Chen, Q. Fan, and R. Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*, 2021.

[6] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[8] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[9] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[10] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[11] S. d'Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[12] H. Demirel and G. Anbarjafari. Image resolution enhancement by using discrete and stationary wavelet decomposition. *IEEE transactions on image processing*, 20(5):1458–1460, 2010.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] X. Ding, X. Zhang, J. Han, and G. Ding. Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv preprint arXiv:2105.01883*, 2021.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

[17] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint arXiv:2105.02358*, 2021.

[18] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[24] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 2, pages II–II. IEEE, 2004.

[25] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[26] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[28] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[29] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[30] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[31] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[32] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.
[33] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.
[34] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
[37] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.
[38] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, volume 29, pages 4898–4906, 2016.
[39] L. Melas-Kyriazi. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. *arXiv: Computer Vision and Pattern Recognition*, 2021.
[40] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307, 2017.
[41] P. C. Ng and S. Henikoff. Sift: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13):3812–3814, 2003.
[42] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.
[43] H. Olkkonen and P. Pesola. Gaussian pyramid wavelet transform for multiresolution analysis of images. *Graphical Models and Image Processing*, 58(4):394–398, 1996.
[44] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019.
[46] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye. Conformer: Local features coupling global representations for visual recognition. *arXiv preprint arXiv:2105.03889*, 2021.
[47] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
[48] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár. Designing network design spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10428–10436, 2020.
[49] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE international conference on computer vision*, pages 2564–2571. Ieee, 2011.
[50] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.
[51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
[53] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.
[54] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[55] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021.
[56] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
[57] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
[58] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
[59] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[60] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[61] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

[62] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[63] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[64] H. Touvron, A. Sablayrolles, M. Douze, M. Cord, and H. Jégou. Grafit: Learning fine-grained image representations with coarse labels. *arXiv preprint arXiv:2011.12982*, 2020.

[65] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

[66] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[67] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[68] R. Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[69] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[70] J. Xie, R. Zeng, Q. Wang, Z. Zhou, and P. Li. So-vit: Mind visual tokens for vision transformer. *arXiv preprint arXiv:2104.10935*, 2021.

[71] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[72] H. Yan, Z. Li, W. Li, C. Wang, M. Wu, and C. Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv preprint arXiv:2104.13497*, 2021.

[73] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR 2016 : International Conference on Learning Representations 2016*, 2016.

[74] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[75] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.

[76] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[77] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[78] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[79] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[80] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[81] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please see Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please see Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code will be released publicly.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 4 and Table 1
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Section 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] We use ImageNet, Cifar, Pets, Cars, Flowers, iNaturalist dataset which are publicly available for academic research purpose.
   (c) Did you include any new assets either in the supplemental material or as a URL?[N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]