Co-Scale Conv-Attentional Image Transformers

Weijian Xu* Yifan Xu* Tyler Chang Zhuowen Tu
University of California San Diego
{wex041, yix081, tachang, ztu}@ucsd.edu

Abstract

In this paper, we present Co-scale conv-attentional image Transformers (CoaT), a Transformer-based image classifier equipped with co-scale and conv-attentional mechanisms. First, the co-scale mechanism maintains the integrity of Transformers' encoder branches at individual scales, while allowing representations learned at different scales to effectively communicate with each other; we design a series of serial and parallel blocks to realize the co-scale mechanism. Second, we devise a conv-attentional mechanism by realizing a relative position embedding formulation in the factorized attention module with an efficient convolution-like implementation. CoaT empowers image Transformers with enriched multi-scale and contextual modeling capabilities. On ImageNet, relatively small CoaT models attain superior classification results compared with similar-sized convolutional neural networks and image/vision Transformers. The effectiveness of CoaT's backbone is also illustrated on object detection and instance segmentation, demonstrating its applicability to downstream computer vision tasks.

1. Introduction

A notable recent development in artificial intelligence is the creation of attention mechanisms [38] and Transformers [31], which have made a profound impact in a range of fields including natural language processing [7, 20], document analysis [39], speech recognition [8], and computer vision [9, 3]. In the past, state-of-the-art image classifiers have been built primarily on convolutional neural networks (CNNs) [15, 14, 27, 26, 11, 36] that operate on layers of filtering processes. Recent developments [30, 9] however begin to show encouraging results for Transformer-based image classifiers.

In essence, both the convolution [15] and attention [38] operations address the fundamental representation problem for structured data (e.g. images and text) by modeling the local contents, as well as the contexts. The receptive fields in CNNs are gradually expanded through a series of convolution operations. The attention mechanism [38, 31] is, however, different from the convolution operations: (1) the receptive field at each location or token in self-attention [31] readily covers the entire input space since each token is "matched" with all tokens including itself; (2) the self-attention operation for each pair of tokens computes a dot product between the "query" (the token in consideration) and the "key" (the token being matched with) to weight the "value" (of the token being matched with).

Moreover, although the convolution and the self-attention operations both perform a weighted sum, their weights are computed differently: in CNNs, the weights are learned during training but fixed during testing; in the self-attention mechanism, the weights are dynamically computed based on the similarity or affinity between every pair of tokens. As a consequence, the self-similarity operation in the self-attention mechanism provides modeling means that are potentially more adaptive and general than convolution operations. In addition, the introduction of position encodings and embeddings [31] provides Transformers with additional flexibility to model spatial configurations beyond fixed input structures.

Of course, the advantages of the attention mechanism are

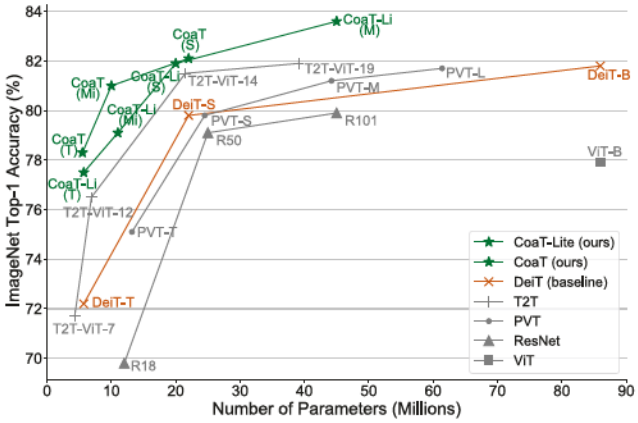

Figure 1. **Model Size vs. ImageNet Accuracy.** Our CoaT model significantly outperforms other image Transformers. Details are in Table 2.

* indicates equal contribution.
Code at https://github.com/mlpc-ucsd/CoaT.

not given for free, since the self-attention operation computes an affinity/similarity that is more computationally demanding than linear filtering in convolution. The early development of Transformers has mainly focused on natural language processing tasks [31, 7, 20] since text is "shorter" than an image, and text is easier to tokenize. In computer vision, self-attention has been adopted to provide added modeling capability for various applications [34, 37, 44]. With the underlying framework increasingly developed [9, 30], Transformers start to bear fruit in computer vision [3, 9] by demonstrating their enriched modeling capabilities.

In the seminal DEtection TRansformer (DETR) [3] algorithm, Transformers are adopted to perform object detection and panoptic segmentation, but DETR still uses CNN backbones to extract the basic image features. Efforts have recently been made to build image classifiers from scratch, all based on Transformers [9, 30, 33]. While Transformer-based image classifiers have reported encouraging results, performance and design gaps to the well-developed CNN models still exist. For example, in [9, 30], an input image is divided into a single grid of fixed patch size. In this paper, we develop Co-scale conv-attentional image Transformers (CoaT) by introducing two mechanisms of practical significance to Transformer-based image classifiers. The contributions of our work are summarized as follows:

- We introduce a co-scale mechanism to image Transformers by maintaining encoder branches at separate scales while engaging attention across scales. Two types of building blocks are developed, namely a serial and a parallel block, realizing **fine-to-coarse**, **coarse-to-fine**, and **cross-scale** image modeling.
- We design a conv-attention module to realize **relative position embeddings** with convolutions in the **factorized attention** module that achieves significantly enhanced computation efficiency when compared with vanilla self-attention layers in Transformers.

Our resulting Co-scale conv-attentional image Transformers (CoaT) learn effective representations under a modularized architecture. On the ImageNet benchmark, CoaT achieves state-of-the-art classification results when compared with the competitive convolutional neural networks (e.g. EfficientNet [29]), while outperforming the competing Transformer-based image classifiers [9, 30, 33], as shown in Figure 1.

2. Related Works

Our work is inspired by the recent efforts [9, 30] to realize Transformer-based image classifiers. ViT [9] demonstrates the feasibility of building Transformer-based image classifiers from scratch, but its performance on ImageNet [23] is not achieved without including additional training data; DeiT [30] attains results comparable to convolution-based classifiers by using an effective training strategy together with model distillation, removing the data requirement in [9]. Both ViT [9] and DeiT [30] are however based on a single image grid of fixed patch size.

The development of our co-scale conv-attentional transformers (CoaT) is motivated by two observations: (1) multi-scale modeling typically brings enhanced capability to representation learning [11, 22, 32]; (2) the intrinsic connection between relative position encoding and convolution makes it possible to carry out efficient self-attention using conv-like operations. As a consequence, the superior performance of the CoaT models shown in the experiments comes from two of our new designs in Transformers: (1) a co-scale mechanism that allows cross-scale interaction; (2) a conv-attention module to realize an efficient self-attention operation. Next, we highlight the differences of the two proposed modules with the standard operations and concepts.

- **Co-Scale vs. Multi-Scale**. Multi-scale approaches have a long history in computer vision [35, 19]. Convolutional neural networks [15, 14, 11] naturally implement a fine-to-coarse strategy. U-Net [22] enforces an extra coarse-to-fine route in addition to the standard fine-to-coarse path; HRNet [32] provides a further enhanced modeling capability by keeping simultaneous fine and coarse scales throughout the convolution layers. In a parallel development [33] to ours, layers of different scales are in tandem for the image Transformers but [33] merely carries out a fine-to-coarse strategy. The co-scale mechanism proposed here differs from the existing methods in how the responses are computed and interact with each other: CoaT consists of a series of highly modularized serial and parallel blocks to enable attention with fine-to-coarse, coarse-to-fine, and cross-scale information on tokenized representations. The joint attention mechanism across different scales in our co-scale module provides enhanced modeling power beyond existing vision Transformers [9, 30, 33].
- **Conv-Attention vs. Attention.** Pure attention-based models [21, 13, 44, 9, 30] have been introduced to the vision domain. [21, 13, 44] replace convolutions in ResNet-like architectures with self-attention modules for better local and non-local relation modeling. In contrast, [9, 30] directly adapt the Transformer [31] for image recognition. Recently, there have been works [1, 6] enhancing the attention mechanism by introducing convolution. LambdaNets [1] introduce an efficient self-attention alternative for global context modeling and employ convolutions to realize the relative position embeddings in local context modeling. CPVT [6] designs 2-D depthwise convolutions as the conditional positional encoding after self-attention. In our conv-attention, we: (1) adopt an efficient factorized attention following [1]; (2) extend it to be a combination of

depthwise convolutional relative position encoding and convolutional position encoding, related to CPVT [6]. Detailed discussion of our network design and its relation with LambdaNets [1] and CPVT [6] can be found in Section 4.1 and 4.2.

3. Revisit Scaled Dot-Product Attention

Transformers take as input a sequence of vector representations (i.e. tokens) $\mathbf{x}_1, ..., \mathbf{x}_N$, or equivalently $X \in \mathbb{R}^{N \times C}$. The self-attention mechanism in Transformers projects each $\mathbf{x}_i$ into corresponding query, key, and value vectors, using learned linear transformations W^Q, W^K, and $W^V \in \mathbb{R}^{C \times C}$. Thus, the projection of the whole sequence generates representations $Q, K, V \in \mathbb{R}^{N \times C}$. The *scaled dot-product attention* from original Transformers [31] is formulated as :

$$\text{Att}(X) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{C}}\Big)V \tag{1}$$

In vision Transformers [9, 30], the input sequence of vectors is formulated by the concatenation of a class token `CLS` and the flattened feature vectors $\mathbf{x}_1, ..., \mathbf{x}_{HW}$ as image tokens from the feature maps $F \in \mathbb{R}^{H \times W \times C}$, for a total length of $N = HW + 1$. The softmax logits in Equation 1 become not affordable for high-resolution images (i.e. $N \gg C$) due to its $O(N^2)$ space complexity and $O(N^2C)$ time complexity. To reduce the length of the sequence, ViT [9, 30] tokenizes the image by patches instead of pixels. However, the coarse splitting (e.g. 16×16 patches) limits the ability to model details within each patch. To address this dilemma, we propose a *co-scale* mechanism that provides enhanced multi-scale image representation with the help of an efficient *conv-attentional* module that lowers the computation complexity for high-resolution images.

4. Conv-Attention Module

4.1. Factorized Attention Mechanism

In Equation 1, the materialization of the softmax logits and attention maps leads to the $O(N^2)$ space complexity and $O(N^2C)$ time complexity. Inspired by recent works [5, 25, 1] on linearization of self-attention, we approximate the softmax attention map by factorizing it using two functions $\phi(\cdot), \psi(\cdot) : \mathbb{R}^{N \times C} \rightarrow \mathbb{R}^{N \times C'}$ and compute the second matrix multiplication (keys and values) together:

$$\text{FactorAtt}(X) = \phi(Q)\Big(\psi(K)^\top V\Big) \tag{2}$$

The factorization leads to a $O(NC' + NC + CC')$ space complexity (including output of $\phi(\cdot), \psi(\cdot)$ and intermediate steps in the matrix product) and $O(NCC')$ time complexity, where both are linear functions of the sequence length N. Performer [5] uses random projections in ϕ and ψ for a provable approximation, but with the cost of relatively large C'. Efficient-Attention [25] applies the softmax function for both ϕ and ψ, which is efficient but causes a significant performance drop on the vision tasks in our experiments. Here, we develop our factorized attention mechanism following LambdaNets [1] with ϕ as the identity function and ψ as the softmax:

$$\text{FactorAtt}(X) = \frac{Q}{\sqrt{C}}\Big(\text{softmax}(K)^\top V\Big) \tag{3}$$

where $\text{softmax}(\cdot)$ is applied across the tokens in the sequence in an element-wise manner and the projected channels $C' = C$. In LambdaNets [1], the scaling factor $1/\sqrt{C}$ is implicitly included in the weight initialization, while our factorized attention applies the scaling factor explicitly. This factorized attention takes $O(NC + C^2)$ space complexity and $O(NC^2)$ time complexity. It is noteworthy that the proposed factorized attention following [1] is not a direct approximation of the scaled dot-product attention, but it can still be regarded as a generalized attention mechanism modeling the feature interactions using query, key and value vectors.

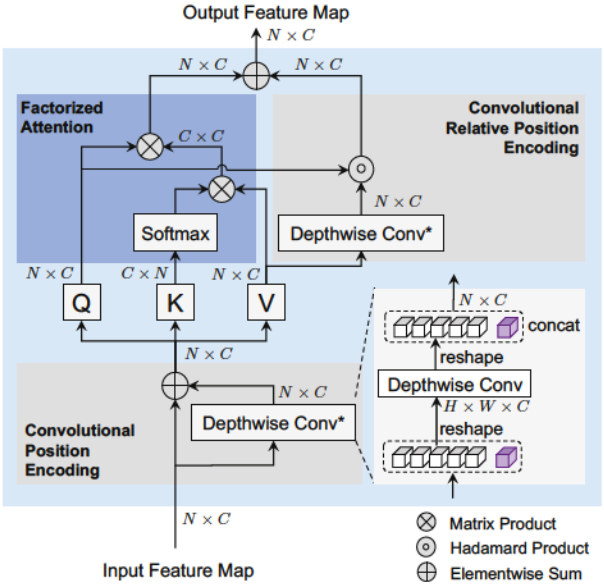

Figure 2. **Illustration of the conv-attentional module.** We apply a convolutional position encoding to the image tokens from the input. The resulting features are fed into a factorized attention with a convolutional relative position encoding.

4.2. Convolution as Position Encoding

Our factorized attention module mitigates the computational burden from the original scaled dot-product attention. However, because we compute $L = \text{softmax}(K)^\top V \in \mathbb{R}^{C \times C}$ first, L can be seen as a global data-dependent linear transformation for every feature vector in the query map Q.

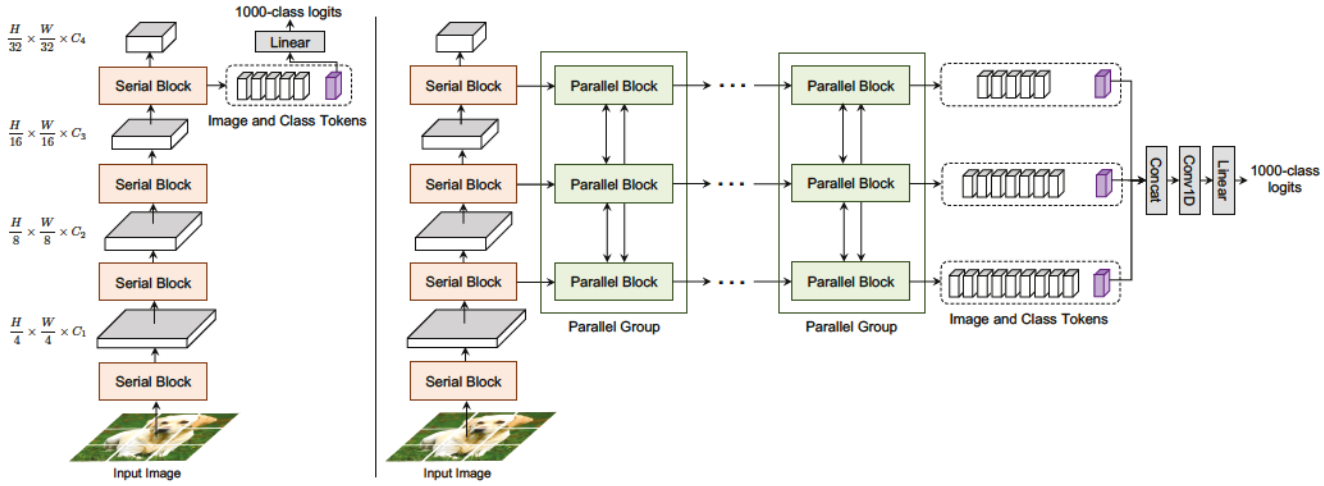

Figure 3. **CoaT model architecture.** (Left) The overall network architecture of **CoaT-Lite**. CoaT-Lite consists of serial blocks only, where image features are down-sampled and processed in a sequential order. (Right) The overall network architecture of **CoaT**. CoaT consists of serial blocks and parallel blocks. Both blocks enable the co-scale mechanism.

This indicates that if we have two query vectors $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^C$ from Q and $\mathbf{q}_1 = \mathbf{q}_2$, then their corresponding self-attention outputs will be the same:

$$\text{FactorAtt}(X)_1 = \frac{\mathbf{q}_1}{\sqrt{C}} L = \frac{\mathbf{q}_2}{\sqrt{C}} L = \text{FactorAtt}(X)_2 \quad (4)$$

Without the position encoding, the Transformer is only composed of linear layers and self-attention modules. Thus, the output of a token is dependent on the corresponding input without awareness of any difference in its locally nearby features. This property is unfavorable for vision tasks such as semantic segmentation (e.g. the same blue patches in the sky and the sea are segmented as the same category).

Convolutional Relative Position Encoding. To enable vision tasks, ViT and DeiT [9, 30] insert absolute position embeddings into the input, which may have limitations in modeling relations between local tokens. Instead, following [24], we can integrate a relative position encoding $P = \{\mathbf{p}_i \in \mathbb{R}^C, i = -\frac{M-1}{2}, ..., \frac{M-1}{2}\}$ with window size M to obtain the relative attention map $EV \in \mathbb{R}^{N \times C}$; in attention formulation, if tokens are regarded as a 1-D sequence:

$$\text{RelFactorAtt}(X) = \frac{Q}{\sqrt{C}} \Big(\text{softmax}(K)^\top V \Big) + EV \quad (5)$$

where the encoding matrix $E \in \mathbb{R}^{N \times N}$ has elements:

$$E_{ij} = \mathbb{1}(i,j) \mathbf{q}_i \cdot \mathbf{p}_{j-i}, \;\; 1 \leq i, j \leq N \quad (6)$$

in which $\mathbb{1}(i,j) = \mathbb{1}_{\{|j-i| \leq (M-1)/2\}}(i,j)$ is an indicator function. Each element E_{ij} represents the relation from query $\mathbf{q}_i$ to the value $\mathbf{v}_j$ within window M, and $(EV)_i$ aggregates all related value vectors with respect to query $\mathbf{q}_i$. Unfortunately, the EV term still requires $O(N^2)$ space complexity and $O(N^2C)$ time complexity. In CoaT, we propose to simplify the EV term to $\hat{EV}$ by considering each channel in the query, position encoding and value vectors as *internal heads*. Thus, for each internal head l, we have:

$$E_{ij}^{(l)} = \mathbb{1}(i,j) q_i^{(l)} p_{j-i}^{(l)}, \;\; \hat{EV}_i^{(l)} = \textstyle\sum_j E_{ij}^{(l)} v_j^{(l)} \quad (7)$$

In practice, we can use a 1-D depthwise convolution to compute $\hat{EV}$:

$$\hat{EV}^{(l)} = Q^{(l)} \circ \text{Conv1D}(P^{(l)}, V^{(l)}), \quad (8)$$

$$\hat{EV} = Q \circ \text{DepthwiseConv1D}(P, V) \quad (9)$$

where $\circ$ is the Hadamard product. It is noteworthy that in vision Transformers, we have two types of tokens, the class (CLS) token and image tokens. Thus, we use a 2-D depthwise convolution (with window size $M \times M$ and kernel weights P) and apply it only to the reshaped image tokens (i.e. $Q^{\text{img}}, V^{\text{img}} \in \mathbb{R}^{H \times W \times C}$ from Q, V respectively):

$$\hat{EV}^{\text{img}} = Q^{\text{img}} \circ \text{DepthwiseConv2D}(P, V^{\text{img}}) \quad (10)$$

$$\hat{EV} = \text{concat}(\hat{EV}^{\text{img}}, \mathbf{0}) \quad (11)$$

$$\text{ConvAtt}(X) = \frac{Q}{\sqrt{C}} \Big(\text{softmax}(K)^\top V \Big) + \hat{EV} \quad (12)$$

Based on our derivation, the depthwise convolution can be seen as a special case of relative position encoding.

Convolutional Relative Position Encoding vs Other Relative Position Encodings. The commonly referenced relative position encoding [24] works in standard scaled dot-product attention settings since the encoding matrix E is combined with the softmax logits in the attention maps, which are not materialized in our factorized attention. Related to our

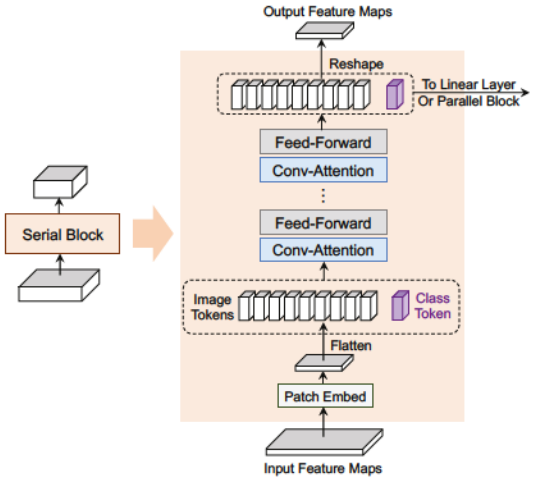

Figure 4. **Schematic illustration of the serial block in CoaT.** Input feature maps are first down-sampled by a patch embedding layer, and then tokenized features (along with a class token) are processed by multiple conv-attention and feed-forward layers.

work, the main results of the original LambdaNets [1] use a 3-D convolution to compute EV directly and reduce the channels of queries and keys to C_K where $C_K < C$, but it costs $O(NCC_K)$ space complexity and $O(NCC_KM^2)$ time complexity, which leads to relatively heavy computation when channel sizes C_K, C are large. A recent update in LambdaNets [1] provides an efficient variant with depth-wise convolution under resource constrained scenarios. Our factorized attention computes $\hat{EV}$ with only $O(NC)$ space complexity and $O(NCM^2)$ time complexity, aiming to achieve better efficiency.

Convolutional Position Encoding. We then extend the idea of convolutional relative position encoding to a general convolutional position encoding case. Convolutional relative position encoding models local position-based relationships between queries and values. Similar to the absolute position encoding used in most image Transformers [9, 30], we would like to insert the position relationship into the input image features directly to enrich the effects of relative position encoding. In each conv-attentional module, we insert a depthwise convolution into the input features X and concatenate the resulting position-aware features back to the input features following the standard absolute position encoding scheme (see Figure 2 lower part), which resembles the realization of conditional position encoding in CPVT [6].

CoaT and CoaT-Lite share the convolutional position encoding weights and convolutional relative position encoding weights for the serial and parallel modules within the same scale. We set convolution kernel size to 3 for the convolutional position encoding. We set convolution kernel size to 3, 5 and 7 for image features from different attention heads for convolutional relative position encoding.

The work of CPVT [6] explores the use of convolution as conditional position encodings by inserting it after the feed-forward network under a single scale ($\frac{H}{16} \times \frac{W}{16}$). Our work focuses on applying convolution as relative position encoding and a general position encoding with the factorized attention.

Conv-Attentional Mechanism The final conv-attentional module is shown in Figure 2: We apply the first convolutional position encoding on the image tokens from the input. Then, we feed it into ConvAtt$(\cdot)$ including factorized attention and the convolutional relative position encoding. The resulting map is used for the subsequent feed-forward networks.

5. Co-Scale Conv-Attentional Transformers

5.1. Co-Scale Mechanism

The proposed co-scale mechanism is designed to introduce fine-to-coarse, coarse-to-fine and cross-scale information into image transformers. Here, we describe two types of building blocks in the CoaT architecture, namely serial and parallel blocks, in order to model multiple scales and enable the co-scale mechanism.

CoaT Serial Block. A serial block (shown in Figure 4) models image representations in a reduced resolution. In a typical serial block, we first down-sample input feature maps by a certain ratio using a patch embedding layer, and flatten the reduced feature maps into a sequence of image tokens. We then concatenate image tokens with an additional `CLS` token, a specialized vector to perform image classification, and apply multiple conv-attentional modules as described in Section 4 to learn internal relationships among image tokens and the `CLS` token. Finally, we separate the `CLS` token from the image tokens and reshape the image tokens to 2-D feature maps for the next serial block.

CoaT Parallel Block. We realize a co-scale mechanism between parallel blocks in each parallel group (shown in Figure 5). In a typical parallel group, we have sequences of input features (image tokens and `CLS` token) from serial blocks with different scales. To enable fine-to-coarse, coarse-to-fine, and cross-scale interaction in the parallel group, we develop two strategies: (1) direct cross-layer attention; (2) attention with feature interpolation. In this paper, we adopt attention with feature interpolation for better empirical performance. The effectiveness of both strategies is shown in Section 6.4.

Direct cross-layer attention. In direct cross-layer attention, we form query, key, and value vectors from input features for each scale. For attention within the same layer, we use the conv-attention (Figure 2) with the query, key and value vectors from current scale. For attention across different layers, we down-sample or up-sample the key and value

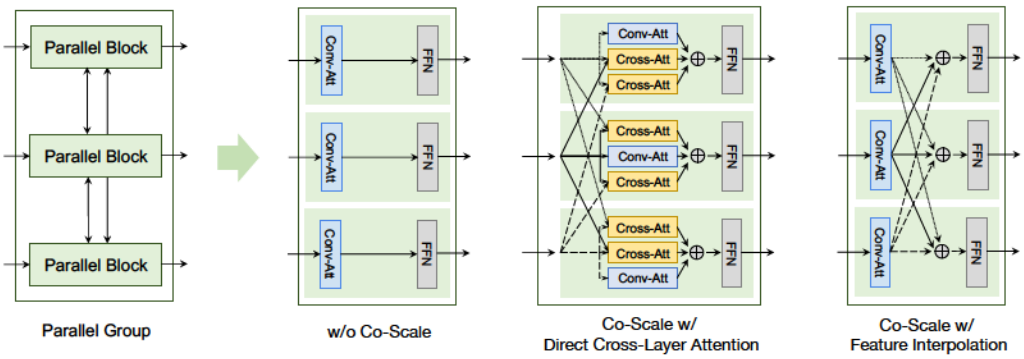

Figure 5. **Schematic illustration of the parallel group in CoaT.** For "w/o Co-Scale", tokens learned at the individual scales are combined to perform the classification but absent intermediate co-scale interaction for the individual paths of the parallel blocks. We propose two co-scale variants, namely direct cross-layer attention and attention with feature interpolation. Co-scale with feature interpolation is adopted in the final CoaT-Lite and CoaT models reported on the ImageNet benchmark.

Table 1. **Architecture details of CoaT-Lite and CoaT models.** C_i represents the hidden dimension of the attention layers in block i; H_i represents the number of attention heads in the attention layers in block i; R_i represents the expansion ratio for the feed-forward hidden layer dimension between attention layers in block i. Multipliers indicate the number of conv-attentional modules in block i.

Blocks	Output	CoaT-Lite				CoaT		
		Tiny	Mini	Small	Medium	Tiny	Mini	Small
Serial Block (S_1)	56×56	$\begin{bmatrix} C_1 = 64 \\ H_1 = 8 \\ R_1 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 64 \\ H_1 = 8 \\ R_1 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 64 \\ H_1 = 8 \\ R_1 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} C_1 = 128 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 3$	$\begin{bmatrix} C_1 = 152 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 152 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 152 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 2$
Serial Block (S_2)	28×28	$\begin{bmatrix} C_2 = 128 \\ H_2 = 8 \\ R_2 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} C_2 = 128 \\ H_2 = 8 \\ R_2 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} C_2 = 128 \\ H_2 = 8 \\ R_2 = 8 \end{bmatrix} \times 4$	$\begin{bmatrix} C_1 = 256 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 6$	$\begin{bmatrix} C_2 = 152 \\ H_2 = 8 \\ R_2 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_2 = 216 \\ H_2 = 8 \\ R_2 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 320 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 2$
Serial Block (S_3)	14×14	$\begin{bmatrix} C_3 = 256 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_3 = 320 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_3 = 320 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 6$	$\begin{bmatrix} C_1 = 320 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 10$	$\begin{bmatrix} C_3 = 152 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_3 = 216 \\ H_3 = 8 \\ R_3 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 320 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 2$
Serial Block (S_4)	7×7	$\begin{bmatrix} C_4 = 320 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_4 = 512 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_4 = 512 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 3$	$\begin{bmatrix} C_1 = 512 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 8$	$\begin{bmatrix} C_4 = 152 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_4 = 216 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 320 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 2$
Parallel Group	$\begin{bmatrix} 28 \times 28 \\ 14 \times 14 \\ 7 \times 7 \end{bmatrix}$					$\begin{bmatrix} C_4 = 152 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 6$	$\begin{bmatrix} C_4 = 216 \\ H_4 = 8 \\ R_4 = 4 \end{bmatrix} \times 6$	$\begin{bmatrix} C_1 = 320 \\ H_1 = 8 \\ R_1 = 4 \end{bmatrix} \times 6$
#Params		5.7M	11M	20M	45M	5.5M	10M	22M

vectors to match the resolution of other scales, which enables fine-to-coarse and coarse-to-fine interaction. We then perform cross-attention, which extends the conv-attention with queries from the current scale with keys and values from another scale. Finally, we sum the outputs of conv-attention and cross-attention together and apply a shared feed-forward layer. With direct cross-layer attention, the cross-scale information is fused in a cross-attention fashion.

Attention with feature interpolation. Instead of performing cross-layer attention directly, we also present attention with feature interpolation. First, the input image features from different scales are processed by independent conv-attention modules. Then, we down-sample or up-sample image features from each scale to match the dimensions of other scales using bilinear interpolation, or keep the same for its own scale. The features belonging to the same scale are summed in the parallel group, and they are further passed into a shared feed-forward layer. In this way, the conv-attentional module in the next step can learn cross-scale information based on the feature interpolation in the current step.

5.2. Model Architecture

CoaT-Lite. CoaT-Lite, Figure 3 (Left), processes input images with a series of serial blocks following a fine-to-coarse pyramid structure. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, each serial block down-samples the image features into lower resolution, resulting in a sequence of four resolutions: $F_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C_1}$, $F_2 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_2}$, $F_3 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_3}$, $F_4 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_4}$. In CoaT-Lite, we obtain the `CLS` token in the last serial block, and perform image classification via a linear projection layer based on the `CLS` token.

CoaT. Our CoaT model, shown in Figure 3 (Right), consists of both serial and parallel blocks. Once we obtain multi-scale feature maps $\{F_1, F_2, F_3, F_4\}$ from the serial blocks, we pass F_2, F_3, F_4 and corresponding `CLS` tokens into the parallel group with three separate parallel blocks. To perform classification with CoaT, we aggregate the `CLS`

Table 2. **CoaT performance on ImageNet-1K validation set.** Our CoaT models consistently outperform other methods while being parameter efficient. ConvNets and ViTNets with similar model size are grouped together for comparison. "#GFLOPs" and Top-1 Acc are measured at input image size. "*" results are adopted from [33].

Arch.	Model	#Params	Input	#GFLOPs	Top-1 Acc.
ConvNets	EfficientNet-B0 [29]	5.3M	224^2	0.4	77.1%
	ShuffleNet [43]	5.4M	224^2	0.5	73.7%
ViTNets	DeiT-Tiny [30]	5.7M	224^2	1.3	72.2%
	CPVT-Tiny [6]	5.7M	224^2	-	73.4%
	CoaT-Lite Tiny (Ours)	5.7M	224^2	1.6	77.5%
	CoaT Tiny (Ours)	5.5M	224^2	4.4	**78.3%**
ConvNets	EfficientNet-B2[29]	9M	260^2	1.0	80.1%
	ResNet-18* [11]	12M	224^2	1.8	69.8%
ViTNets	PVT-Tiny [33]	13M	224^2	1.9	75.1%
	CoaT-Lite Mini (Ours)	11M	224^2	2.0	79.1%
	CoaT Mini (Ours)	10M	224^2	6.8	**81.0%**
ConvNets	EfficientNet-B4 [29]	19M	380^2	4.2	**82.9%**
	ResNet-50* [11]	25M	224^2	4.1	78.5%
	ResNeXt50-32x4d* [36]	25M	224^2	4.3	79.5%
ViTNets	DeiT-Small [30]	22M	224^2	4.6	79.8%
	PVT-Small [33]	24M	224^2	3.8	79.8%
	CPVT-Small [6]	22M	224^2	-	80.5%
	T2T-ViT$_t$-14 [40]	22M	224^2	6.1	81.7%
	Swin-T [17]	29M	224^2	4.5	81.3%
	CoaT-Lite Small (Ours)	20M	224^2	4.0	81.9%
	CoaT Small (Ours)	22M	224^2	12.6	82.1%
ConvNets	EfficientNet-B6 [29]	43M	528^2	19	84.0%
	ResNet-101* [11]	45M	224^2	7.9	79.8%
	ResNeXt101-64x4d* [36]	84M	224^2	15.6	81.5%
ViTNets	PVT-Large [33]	61M	224^2	9.8	81.7%
	T2T-ViT$_t$-24 [40]	64M	224^2	15	82.6%
	DeiT-Base [30]	86M	224^2	17.6	81.8%
	CPVT-Base [6]	86M	224^2	-	82.3%
	Swin-B [17]	88M	224^2	15.4	83.5%
	Swin-B [17]	88M	384^2	47	**84.5%**
	CoaT-Lite Medium (Ours)	45M	224^2	9.8	83.6%
	CoaT-Lite Medium (Ours)	45M	384^2	28.7	**84.5%**

tokens from all three scales.

Model Variants. In this paper, we explore CoaT and CoaT-Lite with several different model sizes, namely Tiny, Mini, Small and Medium. Architecture details are shown in Table 1. For example, tiny models represent those with a 5M parameter budget constraint. Specifically, these tiny models have four serial blocks, each with two conv-attentional modules. In CoaT-Lite Tiny architectures, the hidden dimensions of the attention layers increase in later blocks. CoaT Tiny sets the hidden dimensions of the attention layers in the parallel group to be equal, and performs the co-scale mechanism within the six parallel groups. Mini, small and medium models follow the same architecture design but with increased embedding dimensions and increased numbers of conv-attentional modules within blocks.

6. Experiments

6.1. Experiment Details

Image Classification. We perform image classification on the standard ILSVRC-2012 ImageNet dataset [23]. The standard ImageNet benchmark contains 1.3 million images in the training set and 50K images in the validation set, covering 1000 object classes. Image cropping sizes are set to 224×224. For fair comparison, we perform data augmentation such as MixUp [42], CutMix [41], random erasing [45], repeated augmentation [12], and label smoothing [28], following identical procedures in DeiT [30].

All experimental results for our models in Table 2 are reported at 300 epochs, consistent with previous methods [30]. All models are trained with the AdamW [18] optimizer under the NVIDIA Automatic Mixed Precision (AMP) framework. The learning rate is scaled as $5 \times 10^{-4} \times \frac{\text{global batch size}}{512}$.

Table 3. **Object detection and instance segmentation results based on Mask R-CNN on COCO val2017.** Experiments are performed under the MMDetection framework [4]. "*" results are adopted from Detectron2.

Backbone	#Params (M)	w/ FPN 1× AP^b	w/ FPN 1× AP^m	w/ FPN 3× AP^b	w/ FPN 3× AP^m
ResNet-18*	31.3	34.2	31.3	36.3	33.2
PVT-Tiny [33]	32.9	36.7	35.1	39.8	37.4
CoaT-Lite Mini (Ours)	30.7	41.4	38.0	42.9	38.9
CoaT Mini (Ours)	30.2	**45.1**	**40.6**	**46.5**	**41.8**
ResNet-50*	44.3	38.6	35.2	41.0	37.2
PVT-Small [33]	44.1	40.4	37.8	43.0	39.9
Swin-T [17]	47.8	43.7	39.8	46.0	41.6
CoaT-Lite Small (Ours)	39.5	45.2	40.7	45.7	41.1
CoaT Small (Ours)	41.6	**46.5**	**41.8**	**49.0**	**43.7**

Table 4. **Object detection and instance segmentation results based on Cascade Mask R-CNN on COCO val2017.** Experiments are performed using the MMDetection framework [4].

Backbone	#Params (M)	w/ FPN 1× AP^b	w/ FPN 1× AP^m	w/ FPN 3× AP^b	w/ FPN 3× AP^m
Swin-T [17]	85.6	48.1	41.7	50.4	43.7
CoaT-Lite Small (Ours)	77.3	49.1	42.5	48.9	42.6
CoaT Small (Ours)	79.4	**50.4**	**43.5**	**52.2**	**45.1**

Table 5. **Object detection results based on Deformable DETR on COCO val2017.** DD ResNet-50 represents the baseline result using the official checkpoint. ResNet-50 and our CoaT-Lite as DD backbones are directly comparable due to similar model size.

Backbone	Deformable DETR (Multi-Scale) AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DD ResNet-50 [46]	44.5	63.7	48.7	26.8	47.6	59.6
DD **CoaT-Lite Small** (Ours)	47.0	66.5	51.2	28.8	50.3	63.3
DD **CoaT Small** (Ours)	**48.4**	**68.5**	**52.4**	**30.2**	**51.8**	**63.8**

Object Detection and Instance Segmentation. We conduct object detection and instance segmentation experiments on the Common Objects in Context (COCO2017) dataset [16]. The COCO2017 benchmark contains 118K training images and 5K validation images. We evaluate the generalization ability of CoaT in object detection and instance segmentation with the Mask R-CNN [10] and Cascade Mask R-CNN [2]. We use the MMDetection [4] framework and follow the settings from Swin Transformers [17]. In addition, we perform object detection based on Deformable DETR [46] following its data processing settings.

For Mask R-CNN optimization, we train the model with the ImageNet-pretrained backbone on 8 GPUs via AdamW optimizer with learning rate 0.0001. The training period contains 12 epochs in 1× setting and 36 epochs in 3× setting.

For Cascade R-CNN experiments, we use three detection heads, with the same optimization and training period as Mask R-CNN. For Deformable DETR optimization, we train the model with the pretrained backbone for 50 epochs, using an AdamW optimizer with initial learning rate 2×10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We reduce the learning rate by a factor of 10 at epoch 40.

6.2. CoaT for ImageNet Classification

Table 2 shows top-1 accuracy results for our models on the ImageNet validation set comparing with state-of-the-art methods. We separate model architectures into two categories: convolutional networks (ConvNets), and Transformers (ViTNets). Under different parameter budget constraints, CoaT and CoaT-Lite show strong results compared to other ConvNet and ViTNet methods.

6.3. Object Detection and Instance Segmentation

Tables 3 and 4 demonstrate CoaT object detection and instance segmentation results under the Mask R-CNN and Cascade Mask R-CNN frameworks on the COCO val2017 dataset. Our CoaT and CoaT-Lite models show clear performance advantages over the ResNet, PVT [33] and Swin [17] backbones under both the $1\times$ setting and the $3\times$ setting. In particular, our CoaT models bring a large performance gain, demonstrating that our co-scale mechanism is essential to improve the performance of Transformer-based architectures for downstream tasks.

We additionally perform object detection with the Deformable DETR (DD) framework in Table 5. We compare our models with the standard ResNet-50 backbone on the COCO dataset [16]. Our CoaT backbone achieves 3.9% improvement on average precision (AP) over the results of Deformable DETR with ResNet-50 [46].

6.4. Ablation Study

Effectiveness of Position Encodings. We study the effectiveness of the combination of the convolutional relative position encoding (CRPE) and convolutional position encoding (CPE) in our conv-attention module in Table 6. Our CoaT-Lite without any position encoding results in poor performance, indicating that position encoding is essential for vision Transformers. We observe great improvement for CoaT-Lite variants with either CRPE or CPE, and the combination of CRPE and CPE leads to the best performance (77.5% top-1 accuracy), making both position encoding schemes complementary rather than conflicting.

Effectiveness of Co-Scale. In Table 7, we present performance results for two co-scale variants in CoaT, direct cross-layer attention and attention with feature interpolation. We also report CoaT without co-scale as a baseline. Comparing to CoaT without a co-scale mechanism, CoaT with feature interpolation shows performance improvements on both image classification and object detection (Mask R-CNN w/ FPN $1\times$). Attention with feature interpolation offers a clear advantage over direct cross-layer attention due to less computational complexity and higher accuracy.

Table 6. **Effectiveness of position encodings.** All experiments are performed with the CoaT-Lite Tiny architecture. Performance is evaluated on the ImageNet-1K validation set.

Model	CPE	CRPE	Top-1 Acc.
CoaT-Lite Tiny	✗	✗	68.8%
	✗	✓	75.0%
	✓	✗	75.9%
	✓	✓	77.5%

Table 7. **Effectiveness of co-scale.** All experiments are performed with the CoaT Tiny architecture. Performance is evaluated on the ImageNet-1K validation set and the COCO val2017 dataset.

Model	#Params	Input	#GFLOPs	Top-1 Acc. @input	AP^b	AP^m
CoaT w/o Co-Scale	5.5M	224^2	4.4	77.8%	41.6	37.9
CoaT w/ Co-Scale						
- Direct Cross-Layer Attention	5.5M	224^2	4.8	77.0%	42.1	38.3
- Attention w/ Feature Interp.	5.5M	224^2	4.4	78.3%	42.5	38.6

Computational Cost. We report FLOPs, FPS, latency, and GPU memory usage in Table 8. In summary, CoaT models attain higher accuracy than similar-sized Swin Transformers, but CoaT models in general do have larger latency/FLOPs. The current parallel groups in CoaT are more computationally demanding, which can be mitigated by reducing high-resolution parallel blocks and re-using their feature maps in the co-scale mechanism in future work. The latency overhead in CoaT is possibly because operations (e.g. layers, position encodings, upsamples/downsamples) are not running in parallel.

Table 8. **ImageNet-1K validation set results** compared with the concurrent work Swin Transformer[17]. Computational metrics are measured on a single V100 GPU.

Model	#Params	Input	GFLOPs	FPS	Latency	Mem	Top-1 Acc.	Top-5 Acc.
Swin-T [17]	28M	224^2	4.5	755	16ms	222M	81.2%	95.5%
CoaT-Lite Small (Ours)	20M	224^2	4.0	634	32ms	224M	81.9%	95.6%
CoaT Small (Ours)	22M	224^2	12.6	111	60ms	371M	**82.1%**	**96.1%**
Swin-S [17]	50M	224^2	8.7	437	29ms	372M	83.2%	96.2%
Swin-B [17]	88M	224^2	15.4	278	30ms	579M	83.5%	96.5%
CoaT-Lite Medium (Ours)	45M	224^2	9.8	319	52ms	429M	**83.6%**	**96.7%**
Swin-B [17]	88M	384^2	47.1	85	33ms	1250M	**84.5%**	97.0%
CoaT-Lite Medium (Ours)	45M	384^2	28.7	97	56ms	937M	**84.5%**	**97.1%**

7. Conclusion

In this paper, we present a Transformer based image classifier, Co-scale conv-attentional image Transformer (CoaT), in which cross-scale attention and efficient conv-attention operations have been developed. CoaT models attain strong classification results on ImageNet, and their applicability to downstream computer vision tasks has been demonstrated for object detection and instance segmentation.

Acknowledgments. This work is supported by NSF Award IIS-1717431. Tyler Chang is partially supported by the UCSD HDSI graduate fellowship.

References

[1] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. In *ICLR*, 2021.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *TPAMI*, 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *ICLR*, 2021.

[6] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[8] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *Proc. IEEE Int. Conf. Acoust. Sph. Sig Process.*, 2018.

[9] 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 *ICLR*, 2021.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, 2020.

[13] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *ICCV*, 2019.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

[17] 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 *ICCV*, 2021.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[19] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[20] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[21] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *NIPS*, 2019.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[24] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL-HLT*, 2018.

[25] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *WACV*, pages 3531–3539, 2021.

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

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

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[29] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[30] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[32] J Wang, K Sun, T Cheng, B Jiang, C Deng, Y Zhao, D Liu, Y Mu, M Tan, X Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020.

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

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[35] Andrew Witkin. Scale-space filtering: A new approach to multi-scale description. In *Proc. IEEE Int. Conf. Acoust. Sph. Sig Process.*, volume 9, pages 150–153, 1984.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[37] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *CVPR*, 2018.

[38] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[39] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *SIGKDD*, 2020.

[40] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021.

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[43] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018.

[44] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.

[45] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proc. AAAI Conf. Artificial Intell.*, pages 13001–13008, 2020.

[46] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.