CMT: Convolutional Neural Networks Meet Vision Transformers

Jianyuan Guo[1,2], Kai Han[2], Han Wu[1], Yehui Tang[2], Xinghao Chen[2], Yunhe Wang[2]*, Chang Xu[1]*

[1] School of Computer Science, Faculty of Engineering, University of Sydney. [2] Huawei Noah's Ark Lab.

{jianyuan.guo, kai.han, yunhe.wang}@huawei.com; c.xu@sydney.edu.au

Abstract

Vision transformers have been successfully applied to image recognition tasks due to their ability to capture long-range dependencies within an image. However, there are still gaps in both performance and computational cost between transformers and existing convolutional neural networks (CNNs). In this paper, we aim to address this issue and develop a network that can outperform not only the canonical transformers, but also the high-performance convolutional models. We propose a new transformer based hybrid network by taking advantage of transformers to capture long-range dependencies, and of CNNs to extract local information. Furthermore, we scale it to obtain a family of models, called CMTs, obtaining much better trade-off for accuracy and efficiency than previous CNN-based and transformer-based models. In particular, our CMT-S achieves 83.5% top-1 accuracy on ImageNet, while being 14x and 2x smaller on FLOPs than the existing DeiT and EfficientNet, respectively. The proposed CMT-S also generalizes well on CIFAR10 (99.2%), CIFAR100 (91.7%), Flowers (98.7%), and other challenging vision datasets such as COCO (44.3% mAP), with considerably less computational cost.

1. Introduction

The past decades have witnessed the extraordinary contribution of CNNs [16, 46, 47, 52, 53] in the field of computer vision due to its ability of extracting deep discriminative features. Meanwhile, self-attention based transformers [9, 58] has become the de facto most popular models for natural language processing (NLP) tasks, and shown excellent capability of capturing long-distance relationships. Recently, many researchers attempt to apply the transformer-based architectures to vision domains, and achieve promising results in various tasks such as image classification [10, 57], object detection [2, 70], and semantic segmentation [69]. Vision transformer (ViT) [10] is the first work to replace the

conventional CNN backbone with a pure transformer. Input images (224×224×3) are first split into 196 non-overlapping patches (with a fixed size of 16×16×3 per patch), which are analogous to the word tokens in NLP. The patches are then fed into stacked standard transformer blocks to model global relations and extract feature for classification. The design paradigm of ViT has heavily inspired the following transformer based models for computer vision, such as IPT [3] for low-level vision and SETR [69] for semantic segmentation.

Despite that transformers have demonstrated excellent capabilities when migrated to vision tasks, their performances are still far inferior to similar-sized convolutional neural network counterparts, *e.g.*, EfficientNets [53]. We believe the reason of such weakness is threefold. Firstly, images are split into patches in ViT [10] and other transformer-based models such as IPT [3] and SETR [69]. Doing so can greatly simplify the process of applying transformer to image-based tasks. And the sequence of patches can be directly fed into a standard transformer where long-range dependencies between patches can be well captured. However, it ignores the fundamental difference between sequence-based NLP tasks and image-based vision tasks, *e.g.*, the 2D structure and spatial local information within each patch. Secondly, transformer is difficult to explicitly extract low-resolution and multi-scale features due to the fixed patch size, which poses a big challenge to dense prediction tasks such as detection and segmentation. Thirdly, the computational and memory cost of self-attention modules in transformers are quadratic ($\mathcal{O}(N^2C)$) to the resolution of inputs, compared to $\mathcal{O}(NC^2)$ of convolution-based CNNs. High resolution images are prevalent and common, *e.g.*, 1333×800 in COCO [35] and 2048×1024 in Cityscapes [7]. Using transformers to process such images would inevitably cause the problem of insufficient GPU memory and low computation efficiency.

In this paper, we stand upon the intersection of CNNs and transformers, and propose a novel CMT (CNNs meet transformers) architecture for visual recognition. The proposed CMT takes the advantages of CNNs to compensate for the aforementioned limitations when utilizing pure transformers. As shown in Figure 2(c), input images first go through the convolution stem for fine-grained feature extraction, and are

---

Model	Top1 Acc.	# Params	# FLOPs
DeiT-Ti [57]	72.2%	**5M**	1.3B
CPVT-Ti-GAP [6]	74.9%	6M	1.3B
CeiT-T [67]	76.4%	5M	**1.2B**
EfficientNet-B3 [53]	81.6%	12M	1.8B
CMT-XS	**81.7%**	14M	1.5B
ResNet-50 [16]	76.2%	26M	4.1B
DeiT-S [57]	79.8%	22M	4.6B
RegNetY-4GF [44]	80.0%	21M	4.0B
T2T-ViT-14 [68]	80.6%	21M	4.8B
ResNeXt-101 [64]	80.9%	84M	32B
PVT-M [60]	81.2%	44M	6.7B
Swin-T [36]	81.3%	29M	4.5B
CPVT-S-GAP [6]	81.5%	23M	4.6B
CeiT-S [67]	82.0%	24M	4.5B
CvT-13-NAS [63]	82.2%	**18M**	4.1B
EfficientNet-B4 [53]	82.9%	19M	4.2B
CMT-S	**83.5%**	25M	**4.0B**

Model	mAP	# FLOPs
ConT-M [66]	37.9%	217B
ResNet-101 [16]	38.5%	315B
RelationNet++ [4]	39.2%	266B
ResNeXt-101-64x4d [64]	41.0%	473B
Swin-T [36]	41.5%	245B
PVT-M [60]	42.0%	283B
Twins-SVT-S [5]	42.3%	209B
CMT-S	**44.3%**	230B

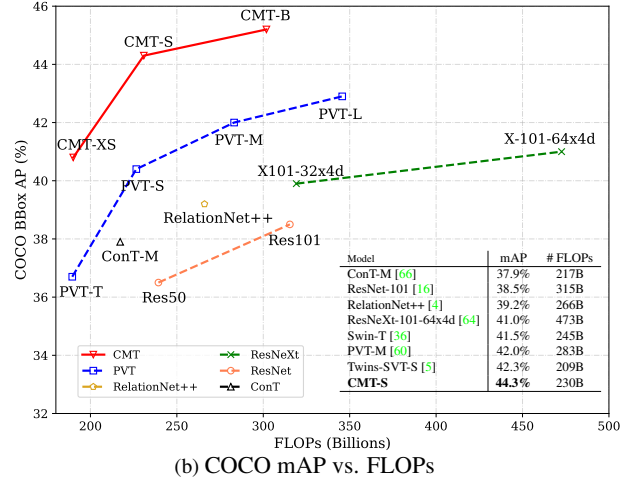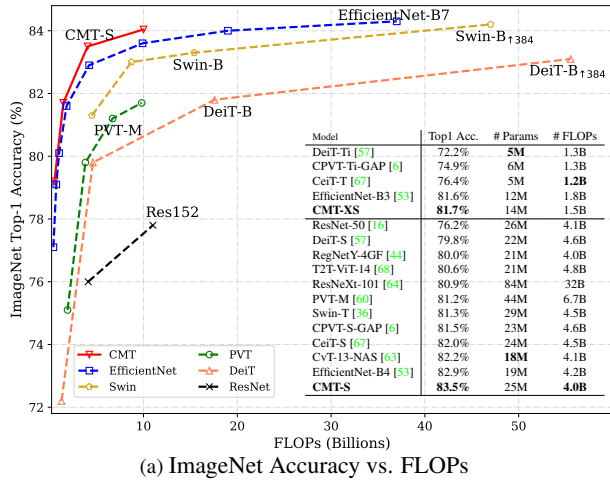(a) ImageNet Accuracy vs. FLOPs   (b) COCO mAP vs. FLOPs

Figure 1. **Performance comparison between CMT and other models.** (a) Top-1 accuracy on ImageNet [8]. (b) Object detection results on COCO val2017 [35] of different backbones using RetinaNet framework, all numbers are for single-scale, "1x" training schedule.

then fed into a stack of CMT blocks for representation learning. Specifically, the introduced CMT block is an improved variant of transformer block whose local information is enhanced by depth-wise convolution. Compared to ViT [10], the features generated from the first stage of CMT can maintain higher resolution, *i.e.*, $H/4 \times W/4$ against $H/16 \times W/16$ in ViT, which are essential for other dense prediction tasks. Furthermore, we adopt the stage-wise architecture design similar to CNNs [16,46,53] by using four convolutional layer with stride 2, to gradually reduce the resolution (sequence length) and increase the dimension flexibly. The stage-wise design helps to extract multi-scale features and alleviate the computation burden caused by high resolution. The local perception unit (LPU) and inverted residual feed-forward network (IRFFN) in CMT block can help capture both local and global structure information within the intermediate features and promote the representation ability of the network. Finally, the average pooling is used to replace the class token in ViT for better classification results. In addition, we propose a simple scaling strategy to obtain a family of CMT variants. Extensive experiments on ImageNet and other downstream tasks demonstrate the superiority of our CMT in terms of accuracy and FLOPs. For example, our CMT-S achieves 83.5% ImageNet top-1 with only 4.0B FLOPs, while being 14x and 2x less than the best existing DeiT [57] and Efficient-Net [53], respectively. In addition to image classification, CMT can also be easily transferred to other vision tasks and serve as a versatile backbone. Using CMT-S as the backbone, RetinaNet [34] can achieve 44.3% mAP on COCO val2017, outperforming the PVT-based RetinaNet [60] by 3.9% with less computational cost.

2. Related Work

The computer vision community prospered in past decades riding the wave of deep learning, and the most popular deep neural networks are often built upon basic blocks, in which a series of convolutional layers are stacked sequentially to capture local information within intermediate features. However, the limited receptive field of small convolutional kernels makes it difficult to obtain global information, withholding the networks of high performance on challenging tasks such as classification, object detection, and semantic segmentation. Therefore, many researchers start to dig deeper into self-attention based transformers which have the ability to capture long-range information. Here we briefly review the conventional CNNs and recently proposed vision transformers.

Convolutional neural networks. The first standard CNN was proposed by LeCun *et al*. [32] for handwritten character recognition, and the past decades have witnessed that many powerful networks [16, 22, 31, 47, 51] achieved unprecedented success on large scale image classification task [8]. AlexNet [31] and VGGNet [47] showed that a deep neural network composed of convolutional layers and pooling layers can obtain adequate results in recognition. GoogleNet [51] and InceptionNet [52] demonstrated the effectiveness of multiple paths within a basic block. ResNet [16] showed better generalization by adding shortcut connections every two layers to the base network. To alleviate the limited receptive fields in prior research, some researches [20,21,41,45,59,62] incorporated attention mechanisms as an operator for adaptation between modalities. Wang *et al*. [59] proposed to stack attention modules sequentially between the intermediate stages of deep residual networks. SENet [21] and GENet [20] adaptively recalibrated channel-wise feature responses by modeling interdependencies between channels. NLNet [61] incorporated the self-attention mechanism into neural networks, providing pairwise interactions across all spatial positions to augment the long-range dependencies. In addition to above archi-

tectural advances, there has also been works [13, 24, 39] focusing on improving over-parameterized deep neural networks by trading accuracy for efficiency. For example, MobileNets [19, 46] and EfficientNets [53] both leveraged neural architecture search (NAS) to design efficient mobile-size network and achieved new state-of-the-art results.

Vision transformers. Since transformers achieved remarkable success in natural language processing (NLP) [9, 58], many attempts [6, 10, 12, 14, 33, 36, 55–57, 60, 63, 67] have been made to introduce transformer-like architectures to vision tasks. The pioneering work ViT [10] directly applied the transformer architecture inherited from NLP to classification with image patches as input. While ViT required a large private dataset JFT-300M [49] to achieve promising result, DeiT [57] introduced a new training paradigm to extend ViT to a data-efficient transformer directly trained on ImageNet-1K. T2T-ViT [68] proposed to embed visual tokens by recursively aggregating neighboring tokens into one token. TNT [68] proposed to model both patch-level and pixel-level representation by the inner and outer transformer block, respectively. PVT [60] introduced the pyramid structure into ViT, which can generate multi-scale feature maps for various pixel-level dense prediction tasks. CPVT [6] and CvT [63] are the most related to our work which leverage a convolutional projection into conventional transformer block, but we carefully investigate how to maximize the advantage of utilizing both CNNs and transformers by studying the different components including shortcut and normalization functions and successfully obtain a more superior result. Besides, transformers are also used to solve other vision tasks such as object detection [2, 70], semantic segmentation [69], image retrieval [11], and low-level vision task [3].

Although there are many works successfully applying transformers for vision tasks, they have not shown satisfactory results compared to conventional CNNs, which are still the primary architectures for vision applications. Transformers are especially good at modeling long-range dependencies necessary for downstream vision tasks. However, locality should also be maintained for visual perception. In this paper, we demonstrate the potential of combining the transformer based network together with convolutional layer, the overall architecture follows the elaborated prior convolutional neural networks such as ResNet [16] and EfficientNet [53].

3. Approach

3.1. Overall Architecture

Our intention is to build a hybrid network taking the advantages of both CNNs and transformers. An overview of ResNet-50 [16], DeiT [57], and the proposed small version (CMT-S) of CMT architectures are presented in Figure 2. As shown in Figure 2(b), DeiT directly splits an input image into non-overlapping patches, however, the in-patch struc-

ture information can only be poorly modeled with linear projections. To overcome this limitation, we utilize the stem architecture [17] which has a 3×3 convolution with a stride of 2 and an output channel of 32 to reduce the size of input images, followed by another two 3×3 convolutions with stride 1 for better local information extraction. Following the design in modern CNNs (*e.g.*, ResNet [16]), our model has four stages to generate feature maps of different scales which are important for dense prediction tasks. To produce the hierarchical representation, a patch embedding layer consisting of a convolution and a layer normalization (LN) [1] is applied before each stage to reduce the size of intermediate feature (2x downsampling of resolution), and project it to a larger dimension (2x enlargement of dimension). In each stage, several CMT blocks are stacked sequentially for feature transformation while retaining the same resolution of the input. For example, the "Stage 3" of CMT-S contains 16 CMT blocks as illustrated in Figure 2(c). The CMT block is able to capture both local and long-range dependencies, and we will describe it in Sec. 3.2 in details. The model ends with a global average pooling layer, a projection layer, and a 1000-way classification layer with softmax.

Given an input image, we can obtain four hierarchical feature maps with different resolutions, similar to typical CNNs such as ResNet [16] and EfficientNet [53]. With the above feature maps whose strides are 4, 8, 16, and 32 with respect to the input, our CMT can obtain multi-scale representations of input images and can be easily applied to downstream tasks such as object detection and semantic segmentation.

3.2. CMT Block

The proposed CMT block consists of a local perception unit (LPU), a lightweight multi-head self-attention (LMHSA) module, and an inverted residual feed-forward network (IRFFN), as illustrated in Figure 2(c). We will describe these three parts in the following.

Local Perception Unit. Rotation and shift are two commonly used data augmentation manners in vision tasks, and these operations should not alter the final results of the model. In other words, we expect translation-invariance [27] in those tasks. However, the absolute positional encoding used in previous transformers, initially designed to leverage the order of tokens, damages such invariance because it adds unique positional encoding to each patch [6]. Besides, vision transformers ignore the local relation [38] and the structure information [26] inside the patch. To alleviate the limitations, we propose the local perception unit (LPU) to extract local information, which is defined as:

$$\text{LPU}(\mathbf{X}) = \text{DWConv}(\mathbf{X}) + \mathbf{X}. \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{H \times W \times d}$, $H \times W$ is the resolution of the input of current stage, d indicates the dimension of features.
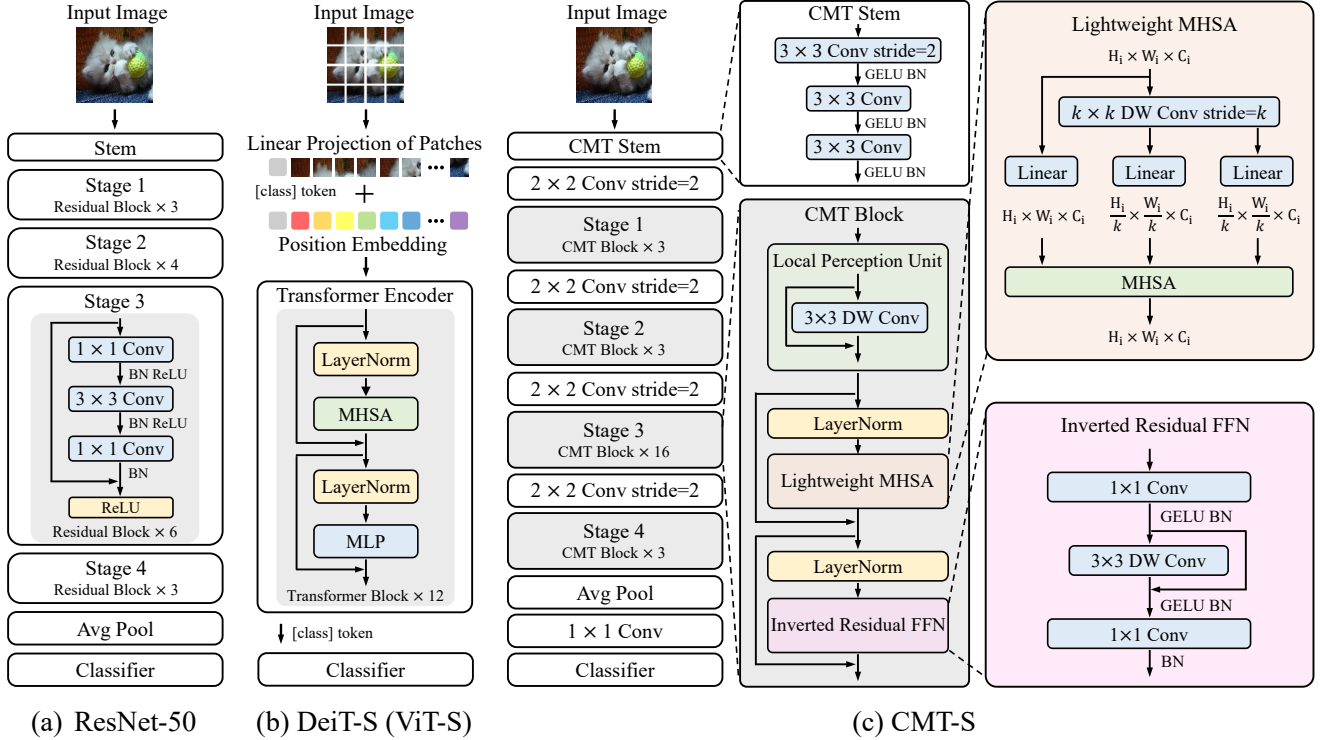
Figure 2. **Example of the CMT-S architecture.** (a) ResNet-50 [16]. (b) DeiT-S [57] (ViT-S [10]) architecture, where MHSA denotes the multi-head self-attention module. (c) The proposed CMT-S, described in Sec. 3. More details and other variants are shown in Table 1.

DWConv(·) denotes the depth-wise convolution.

Lightweight Multi-head Self-attention. In original self-attention module, the input $\mathbf{X} \in \mathbb{R}^{n \times d}$ is linearly transformed into query $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$, key $\mathbf{K} \in \mathbb{R}^{n \times d_k}$, and value $\mathbf{V} \in \mathbb{R}^{n \times d_v}$, where $n = H \times W$ is the number of patches. And we omit the reshape operation from $H \times W \times d$ to $n \times d$ of tensors in Figure 2(c) for simplicity. The notation d, d_k and d_v are the dimensions of input, key (query) and value, respectively. Then the self-attention module is applied as:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}. \quad (2)$$

To mitigate the computation overhead, we use a $k \times k$ depth-wise convolution with stride k to reduce the spatial size of $\mathbf{K}$ and $\mathbf{V}$ before the attention operation, *i.e.*, $\mathbf{K}' = \text{DWConv}(\mathbf{K}) \in \mathbb{R}^{\frac{n}{k^2} \times d_k}$ and $\mathbf{V}' = \text{DWConv}(\mathbf{V}) \in \mathbb{R}^{\frac{n}{k^2} \times d_v}$ as shown in Figure 2(c). In addition, we add a relative position bias $\mathbf{B}$ to each self-attention module, and the corresponding lightweight attention is defined as:

$$\text{LightAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}'^T}{\sqrt{d_k}} + \mathbf{B})\mathbf{V}'. \quad (3)$$

where $\mathbf{B} \in \mathbb{R}^{n \times \frac{n}{k^2}}$ is randomly initialized and learnable. The learnt relative position bias can also be easily transferred to $\mathbf{B}' \in \mathbb{R}^{m_1 \times m_2}$ with a different size $m_1 \times m_2$ through bicubic interpolation, *i.e.*, $\mathbf{B}' = \text{Bicubic}(\mathbf{B})$. Thus

it is convenient to fine-tune the proposed CMT for other downstream vision tasks. Finally, the lightweight multi-head self-attention (LMHSA) module is defined by considering h "heads", *i.e.*, h LightweightAttention functions are applied to the input. Each head outputs a sequence of size $n \times \frac{d}{h}$. These h sequences are then concatenated into a $n \times d$ sequence.

Inverted Residual Feed-forward Network. The original FFN proposed in ViT [10] is composed of two linear layers separated by a GELU activation [18]. The first layer expands the dimension by a factor of 4, and the second layer reduces the dimension by the same ratio:

$$\text{FFN}(\mathbf{X}) = \text{GELU}(\mathbf{X}W_1 + b_1)W_2 + b_2. \quad (4)$$

where $W_1 \in \mathbb{R}^{d \times 4d}$ and $W_2 \in \mathbb{R}^{4d \times d}$ indicate weights of the two linear layers, respectively. The notation b_1 and b_2 are the bias terms. Figure 2(c) provides a schematic visualization of our design. The proposed inverted residual feed-forward network (IRFFN) appears similar to inverted residual block [46] consisting of an expansion layer followed by a depth-wise convolution and a projection layer. Specifically, we change the location of shortcut connection for better performance:

$$\text{IRFFN}(\mathbf{X}) = \text{Conv}(\mathcal{F}(\text{Conv}(\mathbf{X}))), \quad (5)$$
$$\mathcal{F}(\mathbf{X}) = \text{DWConv}(\mathbf{X}) + \mathbf{X}. \quad (6)$$

where the activation layer is omitted. We also include the batch normalization after the activation layer and the last

linear layer according to [46]. The depth-wise convolution is used to extract local information with negligible extra computational cost. The motivation for inserting shortcut is similar to that of classic residual networks, which can promote the propagation ability of gradient across layers. We show that such shortcut helps the network achieve better results in our experiments.

With the aforementioned three components, the CMT block can be formulated as:

$$\mathbf{Y}_i = \text{LPU}(\mathbf{X}_{i-1}), \tag{7}$$

$$\mathbf{Z}_i = \text{LMHSA}(\text{LN}(\mathbf{Y}_i)) + \mathbf{Y}_i, \tag{8}$$

$$\mathbf{X}_i = \text{IRFFN}(\text{LN}(\mathbf{Z}_i)) + \mathbf{Z}_i. \tag{9}$$

where $\mathbf{Y}_i$ and $\mathbf{Z}_i$ denote the output features of LPU and LMHSA module for the i-th block, respectively. LN denotes the layer normalization [1]. We stack several CMT blocks in each stage for feature transformation and aggregation.

3.3. Complexity Analysis

We analyze the computational cost between standard ViT [10] and our CMT in this section. A standard transformer block consists of a MHSA module and a FFN. Given an input feature of size $n \times d$, the computational complexity (FLOPs) can be calculated as:

$$\mathcal{O}(\text{MHSA}) = 2nd(d_k + d_v) + n^2(d_k + d_v), \tag{10}$$

$$\mathcal{O}(\text{FFN}) = 2nd^2 r, \tag{11}$$

where r is the expansion ratio of FFN, d_k and d_v are dimensions of key and value, respectively. More specifically, ViT sets $d = d_k = d_v$ and $r = 4$, the cost can be simplified as:

$$\mathcal{O}(\text{Transformer block}) = \mathcal{O}(\text{MHSA}) + \mathcal{O}(\text{FFN})$$
$$= 12nd^2 + 2n^2d \tag{12}$$

Under above setting, the FLOPs of CMT block is as follows:

$$\mathcal{O}(\text{LPU}) = 9nd, \tag{13}$$

$$\mathcal{O}(\text{LMHSA}) = 2nd^2(1 + 1/k^2) + 2n^2d/k^2, \tag{14}$$

$$\mathcal{O}(\text{IRFFN}) = 8nd^2 + 36nd, \tag{15}$$

$$\mathcal{O}(\text{CMT block}) = \mathcal{O}(\text{LPU}) + \mathcal{O}(\text{LMHSA}) + \mathcal{O}(\text{IRFFN})$$
$$= 10nd^2(1 + 0.2/k^2) + 2n^2d/k^2 + 45nd, \tag{16}$$

where $k \geq 1$ is the reduction ratio in LMHSA. Compared to standard transformer block, the CMT block is more friendly to computational cost, and is easier to process the feature map under higher resolution (larger n).

3.4. Scaling Strategy

Inspired by [53], we propose a new compound scaling strategy suitable for transformer-based networks, which uses a compound coefficient ϕ to uniformly scale the number of layers (depth), dimensions, and input resolution in a principled way:

$$\text{depth}: \alpha^\phi, \quad \text{dimension}: \beta^\phi, \quad \text{resolution}: \gamma^\phi,$$
$$\text{s.t.} \quad \alpha \cdot \beta^{1.5} \cdot \gamma^2 \approx 2.5, \quad \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{17}$$

where α, β, and γ are constants determined by grid search to decide how to assign resources to network depth, dimension and input resolution, respectively. Intuitively, ϕ is the coefficient that controls how many more ($\phi \geq 1$) or less ($\phi \leq -1$) resources are available for model scaling. Notably, the FLOPs of the proposed CMT block is approximately proportional[1] to α, $\beta^{1.5}$, and γ^2 according to E.q. 16. And we constraint $\alpha \cdot \beta^{1.5} \cdot \gamma^2 \approx 2.5$ so that for a given new ϕ, the total FLOPS will approximately increase by 2.5^ϕ. This will strike a balance between the increase of computational cost and performance gain. In our experiments, we empirically set α=1.2, β=1.3, and γ=1.15.

We build our model CMT-S to have similar model size and computation complexity with DeiT-S (ViT-S) and EfficientNet-B4. We also introduce CMT-Ti, CMT-XS and CMT-B according to the proposed scaling strategy. The input resolutions are 160^2, 192^2, 224^2, and 256^2 for all four models, respectively. The detailed architecture hyper-parameters are shown in Table 1.

4. Experiments

In this section, we investigate the effectiveness of CMT architecture by conducting experiments on several tasks including image classification, object detection, and instance segmentation. We first compare the proposed CMT with previous state-of-the-art models on above tasks, and then ablate the important elements of CMT.

4.1. ImageNet Classification

Experimental Settings. ImageNet [8] is a image classification benchmark which contains 1.28M training images and 50K validation images of 1000 classes. For fair comparisons with recent works, we adopt the same training and augmentation strategy as that in DeiT [57], *i.e.*, models are trained for 300 epochs (800 for CMT-Ti that requires more epochs to converge) using the AdamW [37] optimizer. All models are trained on 8 NVIDIA Tesla V100 GPUs.

Results of CMT. Table 2 shows the performances of the proposed CMTs that are scaled from the CMT-S according to E.q. 17. Our models achieve better accuracy with fewer parameters and FLOPs compared to other convolution-based and transformer-based counterparts. In particular, our CMT-S achieves 83.5% top-1 accuracy with 4.0B FLOPs, which is

---

[1]The precious proportion is associated with n and d. For example, CMT-S has n=3136 $\gg$ d=64 in "stage 1" and n=49 $\ll$ d=512 in "stage 4". The above proportion can already generate good variants for CMT.

Table 1. **Architectures for ImageNet classification.** The output size corresponds to the input resolution of 224×224. Convolutions and CMT blocks are shown in brackets with the number of stacked blocks (see also Figure 2(c)). H_i and k_i are the number of heads and reduction rates in LMHSA of stage i, respectively. R_i denotes the expansion ratio in IRFFN of stage i.

Output Size	Layer Name	CMT-Ti	CMT-XS	CMT-S	CMT-B
112×112	Stem	$3 \times 3, 16$, stride 2 $[3 \times 3, 16] \times 2$	$3 \times 3, 16$, stride 2 $[3 \times 3, 16] \times 2$	$3 \times 3, 32$, stride 2 $[3 \times 3, 32] \times 2$	$3 \times 3, 38$, stride 2 $[3 \times 3, 38] \times 2$
56×56	Patch Embedding	$2 \times 2, 46$, stride 2	$2 \times 2, 52$, stride 2	$2 \times 2, 64$, stride 2	$2 \times 2, 76$, stride 2
Stage 1	LPU LMHSA IRFFN	$\begin{bmatrix} 3 \times 3, 46 \\ H_1=1, k_1=8 \\ R_1=3.6 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 52 \\ H_1=1, k_1=8 \\ R_1=3.8 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 64 \\ H_1=1, k_1=8 \\ R_1=4 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 76 \\ H_1=1, k_1=8 \\ R_1=4 \end{bmatrix} \times 4$
28×28	Patch Embedding	$2 \times 2, 92$, stride 2	$2 \times 2, 104$, stride 2	$2 \times 2, 128$, stride 2	$2 \times 2, 152$, stride 2
Stage 2	LPU LMHSA IRFFN	$\begin{bmatrix} 3 \times 3, 92 \\ H_2=2, k_2=4 \\ R_2=3.6 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 104 \\ H_2=2, k_2=4 \\ R_2=3.8 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 128 \\ H_2=2, k_2=4 \\ R_2=4 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 152 \\ H_2=2, k_2=4 \\ R_2=4 \end{bmatrix} \times 4$
14×14	Patch Embedding	$2 \times 2, 184$, stride 2	$2 \times 2, 208$, stride 2	$2 \times 2, 256$, stride 2	$2 \times 2, 304$, stride 2
Stage 3	LPU LMHSA IRFFN	$\begin{bmatrix} 3 \times 3, 184 \\ H_3=4, k_3=2 \\ R_3=3.6 \end{bmatrix} \times 10$	$\begin{bmatrix} 3 \times 3, 208 \\ H_3=4, k_3=2 \\ R_3=3.8 \end{bmatrix} \times 12$	$\begin{bmatrix} 3 \times 3, 256 \\ H_3=4, k_3=2 \\ R_3=4 \end{bmatrix} \times 16$	$\begin{bmatrix} 3 \times 3, 304 \\ H_3=4, k_3=2 \\ R_3=4 \end{bmatrix} \times 20$
7×7	Patch Embedding	$2 \times 2, 368$, stride 2	$2 \times 2, 416$, stride 2	$2 \times 2, 512$, stride 2	$2 \times 2, 608$, stride 2
Stage 4	LPU LMHSA IRFFN	$\begin{bmatrix} 3 \times 3, 368 \\ H_4=8, k_4=1 \\ R_4=3.6 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 416 \\ H_4=8, k_4=1 \\ R_4=3.8 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 512 \\ H_4=8, k_4=1 \\ R_4=4 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 608 \\ H_4=8, k_4=1 \\ R_4=4 \end{bmatrix} \times 4$
1×1	Projection	$1 \times 1, 1280$			
1×1	Classifier	Fully Connected Layer, 1000			
# Params		9.49 M	15.24 M	25.14 M	45.72 M
# FLOPs		0.64 B	1.54 B	4.04 B	9.33 B

3.7% higher than the baseline model DeiT-S [57] and 2.0% higher than CPVT [6], indicating the benefit of CMT block for capturing both local and global information. Note that all previous transformer-based models are still inferior to EfficientNet [53] which is obtained via a thorough architecture search, however, our CMT-S is 0.6% higher than EfficientNet-B4 with less computational cost, which demonstrates the efficacy of the proposed hybrid structure and show strong potential for further improvement. We also plot the accuracy-FLOPs curve in Figure 1(a) to have an intuitive comparison between these models. We can see that CMTs consistently outperform other models by a large margin.

4.2. Ablation Study

Stage-wise architecture. Transformer-based ViT/DeiT can only generate single-scale feature map, losing a lot of multi-scale information crucial for dense prediction tasks. We change the columnar DeiT-S to hierarchical DeiT-S-4Stage, which has 4 stages like CMT-S in Table 1, but maintains the original FFN. We also change MHSA to LMHSA to reduce computational cost. As shown in Table 3, DeiT-S-4Stage outperforms DeiT-S by 1.6% with less FLOPs, demonstrating that the widely-adopted stage-wise design in CNNs is a better choice for promoting transformer-based architecture.

CMT block. Ablations on different modules in CMT are shown in Table 5. DeiT-S-4Stage has 4 patch embedding layers (the first is a 4×4 convolution with stride 4). "+ Stem" indicates that we add the CMT stem into the network and replace the first patch embedding layer with a 2×2 convolution with stride 2. The improvement shows the benefit of the convolution-based stem. Besides, the proposed LPU and IRFFN can further boost the network by 0.8% and 0.6%, respectively. It is worth noticing that the shortcut connections in LPU and IRFFN are also crucial for the final performance.

Normalization function. Transformer-based models usually use LN [1] inherited from NLP. However, convolution-based models usually utilize batch normalization (BN) [25] to stabilize the training. CMT maintains the LN before LMHSA and

Table 3. **Ablation study of stage-wise architecture on ImageNet.**

Model	Params	FLOPs	Top-1
DeiT-S	22M	4.6B	79.8%
DeiT-S-4Stage	25M	3.7B	81.4%

Table 5. **Ablations of CMT block.**

Model	Params	FLOPs	Top-1
DeiT-S-4Stage	25M	3.7B	81.4%
+ Stem	25M	3.9B	81.9%
+ LPU	25M	3.9B	82.7%
w/o shortcut	25M	3.9B	82.0%
+ IRFFN	25M	3.9B	83.3%
w/o shortcut	25M	3.9B	82.5%
+ Projection	25M	4.0B	83.5%

Table 2. **ImageNet Results of CMT**. CNNs and transformers with similar accuracy are grouped together for comparison. The proposed CMTs consistently outperform other methods with less computational cost.

Model	Top-1 Acc.	Top-5 Acc.	Throughput	# Params	Resolution	# FLOPs	Ratio
CPVT-Ti-GAP [6]	74.9%	-	-	6M	224^2	1.3B	2.6×
DenseNet-169 [22]	76.2%	93.2%	-	14M	224^2	3.5B	7×
EfficientNet-B1 [53]	79.1%	94.4%	-	7.8M	240^2	0.7B	1.2×
CMT-Ti	**79.1%**	**94.5%**	1323.5	9.5M	160^2	**0.6B**	**1×**
ResNet-50 [16]	76.2%	92.9%	-	25.6M	224^2	4.1B	2.7×
CoaT-Lite Mini [65]	78.9%	-	-	11M	224^2	2.0B	1.3×
DeiT-S [57]	79.8%	-	940.4	22M	224^2	4.6B	3.1×
EfficientNet-B3 [53]	81.6%	95.7%	732.1	12M	300^2	1.8B	1.2×
CMT-XS	**81.8%**	**95.8%**	857.4	15.2M	192^2	**1.5B**	**1×**
ResNeXt-101-64x4d [64]	80.9%	95.6%	-	84M	224^2	32B	8×
T2T-ViT-19 [68]	81.2%	-	-	39.0M	224^2	8.0B	2×
PVT-M [60]	81.2%	-	528.1	44.2M	224^2	6.7B	1.7×
Swin-T [36]	81.3%	-	755.2	29M	224^2	4.5B	1.1×
CPVT-S-GAP [6]	81.5%	-	-	23M	224^2	4.6B	1.2×
RegNetY-8GF [44]	81.7%	-	591.6	39.2M	224^2	8.0B	2×
CeiT-S [67]	82.0%	95.9%	-	24.2M	224^2	4.5B	1.1×
EfficientNet-B4 [53]	82.9%	96.4%	349.4	19M	380^2	4.2B	1×
Twins-SVT-B [5]	83.1%	-	-	56.0M	224^2	8.3B	2.1×
CMT-S	**83.5%**	**96.6%**	562.5	25.1M	224^2	**4.0B**	**1×**
ViT-B/16$_{\uparrow 384}$ [10]	77.9%	-	85.9	55.5M	384^2	77.9B	8.4×
TNT-B [14]	82.8%	96.3%	-	65.6M	224^2	14.1B	1.5×
DeiT-B$_{\uparrow 384}$ [57]	83.1%	-	85.9	85.8M	384^2	55.6B	6.0×
CvT-21$_{\uparrow 384}$ [63]	83.3%	-	-	31.5M	384^2	24.9B	2.7×
Swin-B [36]	83.3%	-	88M	224^2	278.1	15.4B	1.5×
Twins-SVT-L [5]	83.3%	-	288.0	99.2M	224^2	14.8B	1.7×
CeiT-S$_{\uparrow 384}$ [67]	83.3%	96.5%	-	24.2M	384^2	12.9B	1.4×
BoTNet-S1-128 [48]	83.5%	96.5%	-	75.1M	256^2	19.3B	2.1×
EfficientNetV2-S [54]	83.9%	-	-	22M	224^2	8.8B	1×
EfficientNet-B6 [53]	84.0%	96.8%	96.9	43M	528^2	19.2B	2.0×
CMT-B	**84.5%**	**96.9%**	285.4	45.7M	256^2	**9.3B**	**1×**
EfficientNet-B7 [53]	84.3%	97.0%	55.1	66M	600^2	37B	1.9×
CMT-L	**84.8%**	**97.1%**	150.4	74.7M	288^2	**19.5B**	**1×**

Table 4. **Ablation study of the scaling strategy.**

Model (based on CMT-S)	FLOPs	Top-1	FLOPs	Top-1
Scale: α=2.2 (depth only)	1.7B (ϕ=-1)	80.8%	8.6B(ϕ=1)	83.4%
Scale: β=1.6 (dimension only)	1.7B (ϕ=-1)	81.3%	9.3B(ϕ=1)	83.8%
Scale: α=1.3, β=1.3, γ=1.15	1.5B (ϕ=-1)	81.8%	9.3B(ϕ=1)	84.5%

IRFFN, and inserts BN after the convolutional layer. If all LNs are replaced by BNs, the model cannot converge during training. If all BNs are replaced by LNs, the performance of CMT-S drops to 83.0%, indicating that proper application of normalization functions can improve the final performance.

Scaling strategy. Table 4 shows the ImageNet results of CMT architecture under different scaling strategies. Unidimensional scaling strategies are significantly inferior to the proposed compound scaling strategy, especially for depth-only scaling strategy which leads to a even worse result of 83.4% against 83.8% of the original CMT-S, when the network is scaled up.

4.3. Transfer Learning

4.3.1 Object Detection and Instance Segmentation

Experimental Settings. The experiments are conducted on COCO [35], which contains 118K training images and 5K validation images of 80 classes. We evaluate the proposed CMT-S using two typical framework: RetinaNet [34] and Mask R-CNN [15] for object detection and instance segmentation, respectively. Specifically, we replace the original backbone with our CMT-S to build new detectors. All models are trained under standard single-scale and "1x" schedule (12 epochs) following PVT [60].

Results of CMT. We report the performance comparison results of object detection task and instance segmentation task in Table 6 and Table 7 respectively. For object detection with RetinaNet as basic framework, CMT-S outperforms Twins-PCPVT-S [5] with 1.3% mAP and Twins-SVT-S [5]

Table 6. **Object detection results on COCO val2017.** All models use RetinaNet [34] as basic framework and are trained in "1x" schedule. FLOPs are calculated on 1280×800 input. † means the results are from [5].

Backbone	# Params	# FLOPs	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ConT-M [66]	27.0M	217B	37.9	58.1	40.2	23.0	40.6	50.4
ResNet-101 [16]	56.7M	315B	38.5	57.6	41.0	21.7	42.8	50.4
RelationNet++ [4]	39.0M	266B	39.4	58.2	42.5	-	-	-
ResNeXt-101-32x4d [64]	56.4M	319B	39.9	59.6	42.7	22.3	44.2	52.5
PVT-S [60]	34.2M	226B	40.4	61.3	43.0	25.0	42.9	55.7
Swin-T† [36]	38.5M	245B	41.5	62.1	44.2	25.1	44.9	55.5
Twins-SVT-S [5]	34.3M	209B	42.3	63.4	45.2	26.0	45.5	56.5
Twins-PCPVT-S [5]	34.4M	226B	43.0	64.1	46.0	27.5	46.3	57.3
CMT-S (ours)	44.3M	231B	**44.3**	**65.5**	**47.5**	**27.1**	**48.3**	**59.1**

Table 7. **Instance segmentation results on COCO val2017.** All models use Mask R-CNN [15] as basic framework and are trained in "1x" schedule. FLOPs are calculated on 1280×800 input. † means the results are from [5].

Backbone	# Params	# FLOPs	AP^{box}	AP_{50}^{box}	AP_{75}^{box}	AP^{mask}	AP_{50}^{mask}	AP_{75}^{mask}
ResNet-101 [16]	63.2M	336B	40.0	60.5	44.0	36.1	57.5	38.6
PVT-S [60]	44.1M	245B	40.4	62.9	43.8	37.8	60.1	40.3
ResNeXt-101-32x4d [64]	62.8M	340B	41.9	62.5	45.9	37.5	59.4	40.2
Swin-T† [36]	47.8M	264B	42.2	64.6	46.2	39.1	61.6	42.0
Twins-SVT-S [5]	44.0M	228B	42.7	65.6	46.7	39.6	62.5	42.6
Twins-PCPVT-S [5]	44.3M	245B	42.9	65.8	47.1	40.0	62.7	42.9
CMT-S (ours)	44.5M	249B	**44.6**	**66.8**	**48.9**	**40.7**	**63.9**	**43.4**

Table 8. **Transfer Learning Results.** Models are fine-tuned with the ImageNet pretrained checkpoints. † means the results are from [28].

Model	# Params	# FLOPs	CIFAR10	CIFAR100	Cars	Flowers	Pets
ResNet-152† [16]	58.1M	11.3B	97.9%	87.6%	92.0%	97.4%	94.5%
Inception-v4† [50]	41.1M	16.1B	97.9%	87.5%	93.3%	98.5%	93.7%
EfficientNet-B7$_{\uparrow600}$ [53]	64.0M	37.2B	98.9%	**91.7%**	**94.7%**	98.8%	**95.4%**
ViT-B/16$_{\uparrow384}$ [10]	85.8M	17.6B	98.1%	87.1%	-	89.5%	93.8%
DeiT-B [57]	85.8M	17.6B	99.1%	90.8%	92.1%	98.4%	-
CeiT-S$_{\uparrow384}$ [67]	24.2M	12.9B	99.1%	90.8%	94.1%	98.6%	94.9%
TNT-S$_{\uparrow384}$ [14]	**23.8M**	17.3B	98.7%	90.1%	-	**98.8%**	94.7%
CMT-S (ours)	25.1M	**4.04B**	**99.2%**	91.7%	94.4%	98.7%	95.2%

with 2.0% mAP. For instance segmentation with Mask R-CNN as basic framework, CMT-S surpasses Twins-PCPVT-S [5] with 1.7% AP and Twins-SVT-S [5] with 1.9% AP. We also report the inference speed on COCO val2017 with 1280×800 input, CMT-S based RetinaNet and Mask R-CNN achieve 14.8 FPS and 11.2 FPS, respectively.

4.3.2 Other Vision Tasks

We also evaluate the proposed CMT on five commonly used transfer learning datasets, including CIFAR10 [30], CIFAR100 [30], Standford Cars [29], Flowers [40], and Oxford-IIIT Pets [42] (see Appendix for more details). We fine-tune the ImageNet pretrained models on new datasets following [14, 53]. Table 8 shows the corresponding results. CMT-S outperforms other transformer-based models in all datasets with less FLOPs, and achieves comparable performance against EfficientNet-B7 [53] with 9x less FLOPs, which demonstrates the superiority of CMT architecture.

5. Conclusion

This paper proposes a novel hybrid architecture named CMT for visual recognition and other downstream computer vision tasks such as object detection and instance segmentation, and addresses the limitations of utilizing transformers in a brutal force manner in the field of computer vision. The proposed CMT architectures take advantages of both CNNs and transformers to capture local and global information, promoting the representation ability of the network. In addition, a scaling strategy is proposed to generate a family of CMT variants for different resource constraints. Extensive experiments on ImageNet and other downstream vision tasks demonstrate the effectiveness and superiority of the proposed CMT architecture.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3, 5, 6

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020. 1, 3

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020. 1, 3

[4] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. *arXiv preprint arXiv:2010.15831*, 2020. 2, 8

[5] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021. 2, 7, 8

[6] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 2, 3, 6, 7

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009. 2, 5

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv preprint arXiv:1810.04805*, 2018. 1, 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 3, 4, 5, 7, 8

[11] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *arXiv preprint arXiv:2102.05644*, 2021. 3

[12] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. *arXiv preprint arXiv:2108.13341*, 2021. 3

[13] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[14] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. 3, 7, 8

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2017. 7, 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 1, 2, 3, 4, 7, 8

[17] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3

[18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[19] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 3

[20] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *arXiv preprint arXiv:1810.12348*, 2018. 2

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2, 7

[23] Mindspore. Huawei. https://www.mindspore.cn/. 2020. 1

[24] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 3

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015. 6

[26] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*, 2020. 3

[27] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[28] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 8

[29] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 2013. 8

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012. 2

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 2

[33] Wenshuo Li, Hanting Chen, Jianyuan Guo, Ziyang Zhang, and Yunhe Wang. Brain-inspired multilayer perceptron with spiking neurons. *arXiv preprint arXiv:2203.14679*, 2022. 3

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2017. 2, 7, 8

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer. 1, 2, 7

[36] 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. *arXiv preprint arXiv:2103.14030*, 2021. 2, 3, 7, 8

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[38] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, 1999. 3

[39] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. 3

[40] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 8

[41] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018. 2

[42] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 2012. 8

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 1

[44] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2, 7

[45] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 2

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 1, 2, 3, 4, 5

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[48] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021. 7

[49] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, 2017. 3

[50] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 8

[51] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. 2

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 1, 2

[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019. 1, 2, 3, 5, 6, 7, 8

[54] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, 2021. 7

[55] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. *arXiv preprint arXiv:2111.12294*, 2021. 3

[56] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. *arXiv preprint arXiv:2106.02852*, 2021. 3

[57] 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. 1, 2, 3, 4, 5, 6, 7, 8

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *arXiv preprint arXiv:1706.03762*, 2017. 1, 3

[59] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2

[60] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 2, 3, 7, 8

[61] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[62] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. 2

[63] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 2, 3, 7

[64] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2, 7, 8

[65] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021. 7

[66] Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv preprint arXiv:2104.13497*, 2021. 2, 8

[67] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021. 2, 3, 7, 8

[68] 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. *arXiv preprint arXiv:2101.11986*, 2021. 2, 3, 7

[69] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 1, 3

[70] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 3