Research Article

Partial Convolution Meets Visual Attention

Haiduo Huang[1,2], Fuwei Yang[2], Dong Li[2], Ji Liu[2], Lu Tian[2], Jinzhang Peng[2], Pengju Ren[1], Emad Barsoum[2]

1. Xi'an Jiaotong University, Xi'an, China; 2. Advanced Micro Devices, Inc., China

Designing an efficient and effective neural network has remained a prominent topic in computer vision research. Depthwise onvolution (DWConv) is widely used in efficient CNNs or ViTs, but it needs frequent memory access during inference, which leads to low throughput. FasterNet attempts to introduce partial convolution (PConv) as an alternative to DWConv but compromises the accuracy due to underutilized channels. To remedy this shortcoming and consider the redundancy between feature map channels, we introduce a novel Partial visual ATtention mechanism (PAT) that can efficiently combine PConv with visual attention. Our exploration indicates that the partial attention mechanism can completely replace the full attention mechanism and reduce model parameters and FLOPs. Our PAT can derive three types of blocks: Partial Channel-Attention block (PAT_ch), Partial Spatial-Attention block (PAT_sp) and Partial Self-Attention block (PAT_sf). First, PAT_ch integrates the enhanced Gaussian channel attention mechanism to infuse global distribution information into the untouched channels of PConv. Second, we introduce the spatial-wise attention to the MLP layer to further improve model accuracy. Finally, we replace PAT_ch in the last stage with the self-attention mechanism to extend the global receptive field. Building upon PAT, we propose a novel hybrid network family, named PATNet, which achieves superior top-1 accuracy and inference speed compared to FasterNet on ImageNet-1K classification and excel in both detection and segmentation on the COCO dataset. Particularly, our PATNet-T2 achieves 1.3% higher accuracy than FasterNet-T2, while exhibiting 25% higher GPU throughput and 24% lower CPU latency.

1. Introduction

To design an efficient network, many prior works adopt depthwise separable convolution (DWConv)[1] as a substitute for regular dense convolution. For instance, some CNN-based models[2][3] leverage DWConv to reduce the model's FLOPs and parameters, while Hybrid-based models[4][5][6] employ DWConv to simulate self-attention operations to decrease computation complexity. Nevertheless, some studies[7][8] have revealed that DWConv may suffer from frequent memory access and low parallelism. Recent works have attempted to optimize the network's inference speed on specific hardware[9][10][11][12][13]. From the perspective of versatility, regular convolution (Conv) still has certain advantages.

Notably, FasterNet[9] proposes to use partial convolution (PConv) as an alternative to DWConv. Based on PConv, the FasterNet family achieves exceptional speed across various devices. PConv leverages redundancy within feature maps to selectively apply Conv to a subset of input channels, leaving the remaining channels untouched. That leads to lower FLOPs compared to regular Conv and higher FLOPS[1] than DWConv[9]. However, we analyze that PConv underutilizes the untouched part and is constrained by the local dependencies inherent to CNNs, which may compromise accuracy. The primary reason for the decrease in accuracy is that PConv employs sparse (partial) parameters. So, how to maintain the inference speed of PConv while further enhancing its accuracy? Our motivation is integrating visual attention into partial convolution to enhance the feature representation ability of the untouched channels. We introduce a novel partial visual attention mechanism that can completely replace the conventional

 doi.org/10.32388/1L3TE6

full attention mechanism without compromising accuracy and can reduce the model's parameter count and FLOPs compared with the full attention mechanism. The approach mainly involves substituting partial convolution with partial attention convolution, which is illustrated in Figure 1 (a).

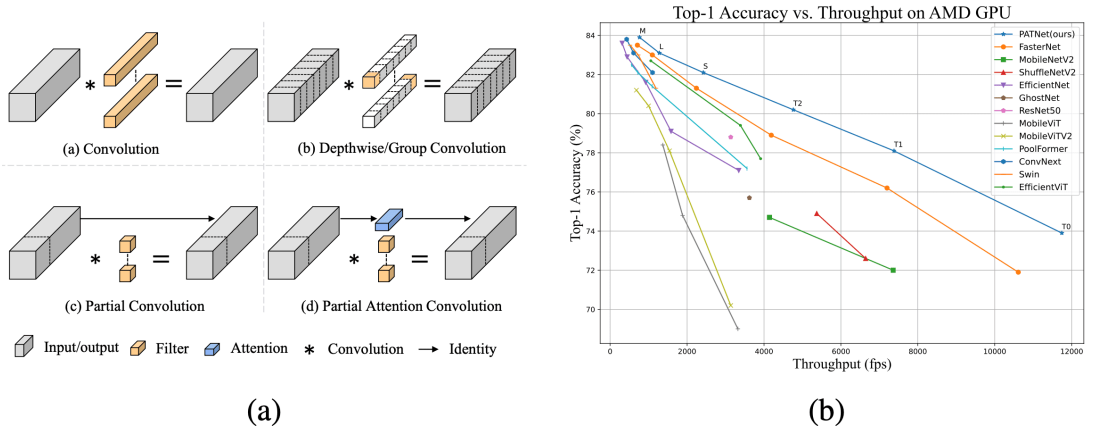

Figure 1. Comparison of different convolution types and efficient networks. Our PATNet incorporates the visual attention mechanism in Partial Convolution named Partial Attention Convolution, which surpasses the performance of FasterNet[9] on various model variants.

How to choose a proper visual attention mechanism to achieve the optimal trade-off between model inference speed and accuracy? To address this problem, we propose three novel efficient partial visual attention blocks, *i.e.*, Partial Channel-Attention block (PAT_ch), Partial Spatial-Attention block (PAT_sp) and Partial Self-Attention block (PAT_sf). Firstly, we construct PAT_ch by integrating an enhanced Gaussian channel attention mechanism[14], facilitating richer inter-channel information interaction. Secondly, we extend the concept of partial convolution to MLP layer to further improve model performance. The convolution part of PA_sp can be fused with the Conv1×1 in the MLP during inference, resulting in efficient computation. Unlike previous spatial-wise attention[15], our approach is simple and effective, involving only a Conv1×1 operation and Hard-Sigmoid[16] activation. Lastly, we refer to the MetaFormer-based[17] paradigm and integrate global self-attention into the last stage of the CNN architecture to expand its global receptive field. The proposed PAT_sf substantially boosts model accuracy in the ImageNet1k classification task.

In conclusion, the enhanced model is dubbed **PATNet**, which achieves overall performance exceeding FasterNet in the ImageNet1K classification task while maintaining similar throughput, as is presented in Figure 1 (b). Our main contributions can be described as:

- We are the first to propose a novel partial visual attention mechanism that integrates visual attention into PConv, which can significantly improve model performance while minimizing the impact on inference speed.
- We develop three types of partial visual attention blocks including of PAT_ch, PAT_sp, and PAT_sf. The PAT_ch exhibits high potential as a replacement for regular convolution and DWConv. PAT_sp can effectively reinforce MLP layers at minimal cost, while PAT_sf integrates local and global features, achieving higher accuracy.
- Building upon PAT, we design a new hybrid-based model family named PATNet that shows improved performance on standard vision benchmarks over FasterNet with higher throughput and lower latency.

doi.org/10.32388/1L3TE6

2. Related Work

Efficient CNNs and ViTs. DWConv is widely adopted in the design of efficient neural networks, such as MobileNets[2][18], EfficientNets[3][19], MobileViT[20], and EdgeViT[21]. Despite its efficiency limitations on modern parallel devices, DWConv still holds unparalleled advantages on mobile devices. Given the drawbacks of DWConv, numerous works have aimed to improve it. For example, RepLKNet[8] uses larger-kernel DWConv to alleviate the issue of underutilized calculations. PoolFormer[17], following the MetaFormer principles, achieves strong performance through spatial interaction with pooling operations alone. Recently, FasterNet[9] reduces FLOPs and memory accesses simultaneously by introducing partial convolution. Nevertheless, FasterNet does not outperform other vision models in accuracy. In contrast, our proposed PATNet addresses this limitation by integrating the visual attention mechanism into partial convolution, effectively enhancing the performance of FasterNet.

Attention Mechanism. Why are Vision Transformers (ViTs) so effective? Some studies attribute their success to the role of attention mechanisms[22][23]. In visual tasks, attention mechanisms are commonly categorized into three types: Channel Attention, Spatial Attention, and Self-Attention. Some works[24][6][25][26] employ various techniques to implement the Self-Attention mechanism efficiently, *e.g.,* Linear Attention[27][26]. Furthermore, the effectiveness of Channel Attention and Spatial Attention has already been validated in SRM[28], SE-Net[14] and CBAM[15]. Similarly, we have incorporated attention mechanisms, but with a partial attention mechanism to mitigate the impact of element-wise multiplication on overall inference speed.

Additional Enhanced Technology Some State-Of-The-Art networks employ additional technologies. For instance, MobileNetV3[18] utilizes NAS[29] techniques to attain an optimal network structure. Networks like MobileOne[12] and RIFormer[11] rely on structured re-parameterization[30] techniques, involving the addition of branches during training to expand its width and the merging of branches during inference to compress it. Furthermore, RIFormer[11], LeViT[31], SwiftFormer[25], and RepViT[32] leverage knowledge distillation[33] technology to transfer prior knowledge from large models to student models, thereby improving accuracy. Self-supervised pre-training[34] technology is employed in models like ConvNeXtV2[35] to achieve better model initialization. However, our PATNet follows regular training as the same FasterNet[9] without using bells and whistles.

3. Methodology

In this section, we first elaborate on our motivation for integrating the visual attention mechanism into partial convolution and introduce **P**artial visual **At**tention mechanism (**PAT**). Subsequently, we delve into our innovative Partial Channel-Attention block (PAT_ch), Partial Spatial-Attention block (PAT_sp), and Partial Self-Attention block (PAT_sf). Finally, we design **PATNet** architecture and explain its details.

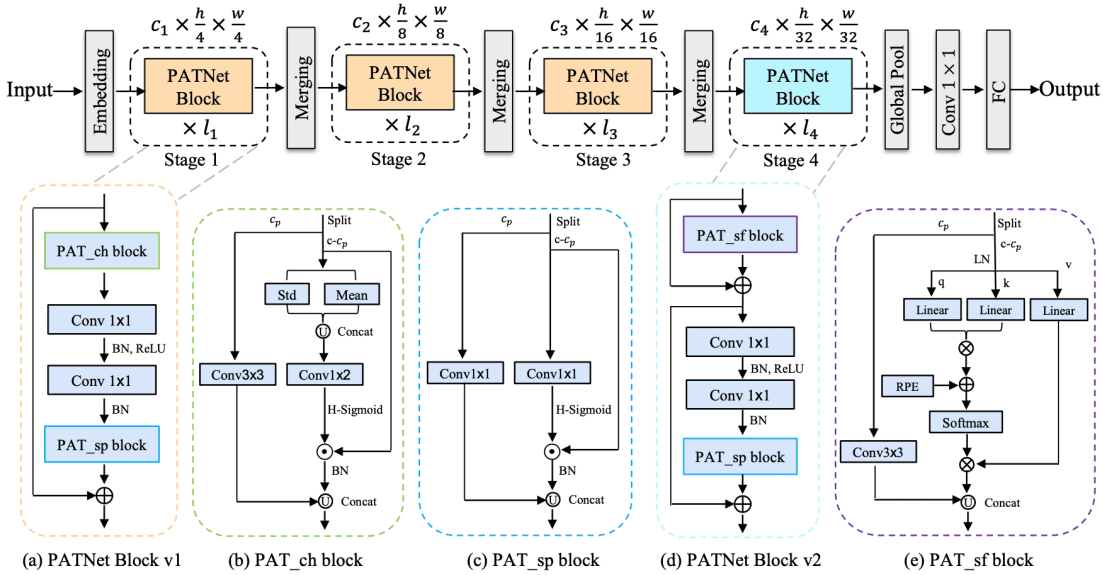

Figure 2. The overall architecture of our PATNet, consisting of four hierarchical stages, each incorporating a series of PATNet blocks followed by an embedding or merging layer. The last three layers are dedicated to feature classification. Where ⊙ and ⊗ denote element-wise multiplication and matrix multiplication respectively.

3.1. Partial Visual Attention Mechanism

Generally, designing an efficient and effective neural network necessitates comprehensive consideration and optimization from various perspectives, including fewer FLOPs, smaller model sizes, lower memory access, and comparable accuracy. Recently, the emerging FasterNet[9] may have met the aforementioned requirements to some degree and demonstrated its effectiveness across various vision tasks and terminal devices without additional technology enhancements. However, it does not exhibit a noticeable accuracy advantage when compared to models with similar parameters or FLOPs.

We empirically analyze that FasterNet mainly conducts Conv3×3 operations on a portion of input channels of PConv, leaving the rest as direct identity mappings. These identity mappings are then concatenated with the processed Conv3×3 portion. While this approach significantly reduces FLOPs and latency, it results in limited feature interaction and fusion, lacking global information interaction. Natural, we explore the integration of the visual attention mechanism into the identity mapping part (untouched part). Previous research[36][9] has demonstrated that redundancy exists among feature map channels, making attention operations applied to the untouched parts a form of global information interaction.

Unlike regularly dense visual attention methods, our PAT is more efficient due to using only a subset of channels for the computationally expensive element-wise multiplication. Indeed, running two operations in parallel on separate branches allows for simultaneous computation, optimizing resource utilization on the GPU[37]. We also find that PAT is not only capable of applying channel-wise and spatial-wise mixing to enhance global information but also combines self-attention mechanisms to expand the model receptive field, proving to be highly effective. Below, We describe our PAT mechanism in formula.

Suppose the input and output of our PAT is $X, Y \in \mathbb{R}^{H \times W \times C}$, where C, H, W represent the number of channels, height and width of a channel, respectively. We keep the number of channels unchanged after PAT. Then, the output can be formulated as

doi.org/10.32388/1L3TE6

$$Y = Y^{C_p} \cup Y^{C-C_p} = Conv(X^{C_p}) \cup Atten(X^{C-C_p}) \tag{1}$$

where the symbol $\cup$ denotes the concatenation operation. $Conv$ denotes for regular convolution function and $Atten$ denotes for attention function, which can be one of channel attention, spatial attention and self attention. And $C_p = r_p \times C$ is defined as the number of front or last consecutive partial channels of the feature map. r_p is a hyperparameter representing the $ratio$ used to select a portion of the channels. Detailed hyperparameter setup refer to the appendix.

3.2. Efficient Integrated Visual Attention Information

In this section, we explain our three types of partial visual attention in detail.

PAT_ch: We integrate channel attention and Conv3×3 because both involve spatial information interaction: Conv3x3 convolves and sums pixels within a local window, while our enhanced Gaussian-SE module computes channels' mean and variance to squeeze global spatial information. Unlike SENet[14], it only considers the mean information of the channel and ignores the statistical information of std. Considering that the feature maps obey an approximately normal distribution[38][39] during training, we fully utilize the Gaussian statistical to express the channel-wise representation information, as shown in Figure 3 (a).

PAT_sp: We integrate spatial attention with Conv1×1 because both operations mix channel wise information. Our spatial attention employs a point-wise convolution to squeeze global channel information into tensor with only 1 single channel. After passing through a Hard-Sigmoid activation, this tensor serves as the spatial attention map to weight features. We position PAT_sp after the MLP layer, enabling the Conv1×1 component of PAT_sp to merge with the second Conv1×1 in the MLP layer during inference, as shown in Figure 3 (b) and Figure 3 (d). This setup further minimizes the impact of attention on inference speed.

PAT_sf: Since PAT_sf also engages with spatial information interaction, it can replace PAT_ch and extend the model's effective receptive field. However, because the computational complexity of self-attention operations increases quadratically with the size of the feature map, we restrict the use of PAT_sf to the last stage to achieve a superior speed-accuracy trade-off. Beside, we employee relative position encoding (RPE) [40] into the attention map, which can further enhances model accuracy, as shown in Figure 3 (c).

Notable, unlike conventional CNNs combined with attention, which process steps one after the other, we process steps simultaneously on the same input, improving the balance between speed and accuracy. In addition, our PAT is not limited to the above three combinations, it can be efficiently combined with more visual attention modules. Hence, the combination of the above three types of PAT blocks into a efficient PATNet.

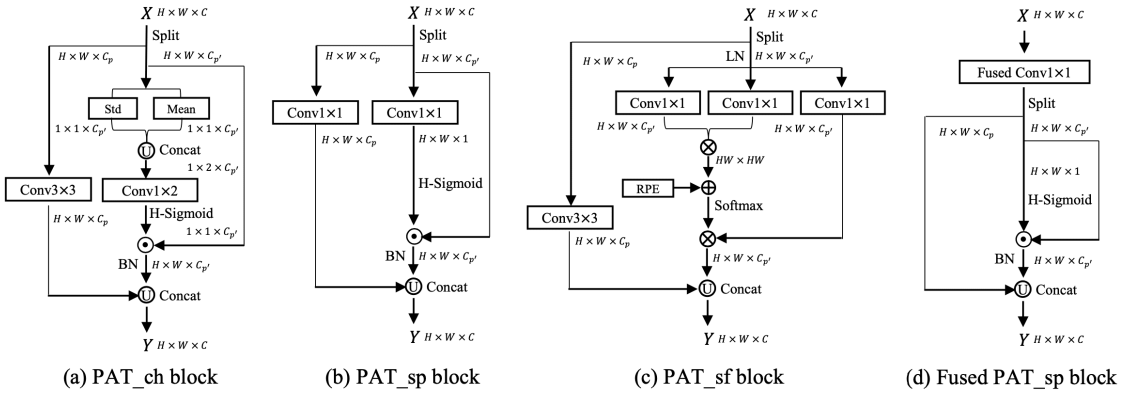

Figure 3. Combination of different partial visual attention blocks. Where $\odot$ and $\otimes$ denote element-wise multiplication and matrix multiplication respectively, and $C = C_p + C_{p'}$.

3.3. PATNet Architecture

Our proposed PATNet refer to the recently introduced FasterNet[9]. The overall architecture, as depicted in Figure 2, consists of four hierarchical stages, each of which precedes an embedding layer (a regular Conv4×4 with stride 4) or a merging layer (a regular Conv2×2 with stride 2). These layers serve for spatial downsampling and channel number expansion. Each stage comprises a set of PATNet blocks. In the first three stages of the PATNet, we employ "PATNet Block v1" including PAT_ch block and PAT_sp block, as shown in Figure 2 (a). However, we employ "PATNet Block v2" by replacing PAT_ch with PAT_sf in the last stage and modifying the shortcut connection way to achieve stable training, as shown in Figure 2 (d). Furthermore, we adjust the depth ratios across the four stages. In previous designs[17][41][9], the depth of the last stage equals that of the first or second stage. We experimental find the critical importance of the last stage for network accuracy. Consequently, we adjusted the depth of the last stage to twice that of the first two stages. This adjustment substantially enhances model accuracy while minimally affecting throughput and latency.

Following the FasterNet design principles, we maintain normalization or activation layers only after each intermediate Conv1×1 to preserve feature diversity and achieve higher throughput. We also incorporate batch normalization into adjacent Conv layers to expedite inference without sacrificing performance. For the activation layer, the smaller PATNet variant uses GELU[42], while the larger PATNet variant employs ReLU. Similarly, the last three layers consist of global average pooling, Conv1×1, and a fully connected layer[18]. These layers collectively serve for feature transformation and classification. We offer tiny, small, medium, and large variants of PATNet, which are denoted as PATNet-T0/1/2, PATNet-S, PATNet-M, and PATNet-L. These variants share a similar architecture but differ in depth and width. The width of PATNet has been reduced compared to FasterNet to achieve faster inference speed. Detailed architectural specifications refer to the appendix.

doi.org/10.32388/1L3TE6

4. Experiments

Network	Type	Params (M)	FLOPs (G)	Throughput V100 (FPS) ↑	Throughput MI250 (FPS) ↑	Latency CPU (ms) ↓	Top-1 (%) ↑
ShuffleNetV2 x1.5[7]	cnn	3.5	0.30	5315	6642	13.7	72.6
MobileNetV2[2]	cnn	3.5	0.31	3924	7359	13.7	72.0
FasterNet-T0[9]	cnn	3.9	0.34	**8546**	10612	**10.5**	71.9
MobileViTv2-0.5[24]	hybrid	1.4	0.46	3094	3135	15.8	70.2
PATNet-T0(ours)	hybrid	4.3	0.25	7777	**11744**	12.2	**73.9**
EfficientNet-B0[3]	cnn	5.3	0.39	2934	3344	22.7	77.1
ShuffleNetV2 x2[7]	cnn	7.4	0.59	4290	5371	22.6	74.9
MobileNetV2 x1.4[2]	cnn	6.1	0.60	2615	4142	21.7	74.7
FasterNet-T1[9]	cnn	7.6	0.85	**4648**	7198	22.2	76.2
PATNet-T1(ours)	hybrid	7.8	0.55	4403	**7379**	**21.5**	**78.1**
EfficientNet-B1[3]	cnn	7.8	0.70	1730	1583	35.5	79.1
ResNet50[43]	cnn	25.6	4.11	1258	3135	94.8	78.8
FasterNet-T2[9]	cnn	15.0	1.91	2455	4189	43.7	78.9
PoolFormer-S12[17]	hybrid	11.9	1.82	1927	3558	56.1	77.2
MobileViTv2-1.0[24]	hybrid	4.9	1.85	1391	1543	41.5	78.1
EfficientViT-B1[26]	hybrid	9.1	0.52	3072	3387	**25.7**	79.4
PATNet-T2(ours)	hybrid	12.6	1.03	**3074**	**4761**	35.2	**80.2**
EfficientNet-B3[3]	cnn	12.0	1.80	768	926	73.5	81.6
ConvNeXt-T[41]	cnn	28.6	4.47	902	1103	99.4	**82.1**
FasterNet-S[9]	cnn	31.1	4.56	1261	2243	96.0	81.3
PoolFormer-S36[17]	hybrid	30.9	5.00	675	1092	152.4	81.4
MobileViTv2-2.0[24]	hybrid	18.5	7.50	551	684	103.7	81.2
Swin-T[44]	hybrid	28.3	4.51	808	1192	107.1	81.3
PATNet-S(ours)	hybrid	29.0	2.71	**1559**	**2422**	**72.5**	**82.1**
EfficientNet-B4[3]	cnn	19.0	4.20	356	442	156.9	82.9
ConvNeXt-S[41]	cnn	50.2	8.71	510	610	185.5	**83.1**
FasterNet-M[9]	cnn	53.5	8.74	621	1098	181.6	83.0

doi.org/10.32388/1L3TE6

Network	Type	Params (M)	FLOPs (G)	Throughput V100 (FPS) ↑	Throughput MI250 (FPS) ↑	Latency CPU (ms) ↓	Top-1 (%) ↑
PoolFormer-M36[17]	hybrid	56.2	8.80	444	721	244.3	82.1
Swin-S[44]	hybrid	49.6	8.77	477	732	199.1	83.0
PATNet-M(ours)	hybrid	61.3	6.69	**799**	**1280**	**155.3**	**83.1**
EfficientNet-B5[3]	cnn	30.0	9.90	246	313	333.3	83.6
ConvNeXt-B[41]	cnn	88.6	15.38	322	430	317.1	83.8
FasterNet-L[9]	cnn	93.5	15.52	384	709	312.5	83.5
PoolFormer-M48[17]	hybrid	73.5	11.59	335	556	322.3	82.5
Swin-B[44]	hybrid	87.8	15.47	315	520	333.8	83.5
PATNet-L(ours)	hybrid	104.3	11.91	**426**	**765**	**272.5**	**83.9**

Table 1. Comparison on ImageNet-1k Benchmark: models with similar top-1 accuracy are grouped together. The best results are in bold. Full comparison please refer to appendix.

4.1. PATNet on ImageNet-1k Classification

Setup. ImageNet-1K[45] is one of the most extensively used datasets in computer vision. It encompasses 1K common classes, consisting of approximately 1.3M training images and 50K validation images. We train our model on the ImageNet-1k dataset for 300 epochs using AdamW optimizer with 20 epochs linear warm-up. And we use the same regularization and augmentation techniques and multi-scale training as FasterNet[9]. For detailed experimental settings, please refer to the appendix. In inference speed, we test the model's throughput in Nvidia V100 and AMD Instinct MI250 GPUs with batch size of 256, we test latency in AMD EPYC^{TM} 73F3 CPU with one core.

Results. Table 1 provides a comparison of our proposed PATNet models (T0, T1, T2, S, M, and L) with previous state-of-the-art cnn-based and hybrid-based models. The experimental results demonstrate that PATNet consistently surpasses recent models like FasterNet[9] across all model variants. For example, PATNet-T2 achieves 1.3% higher accuracy than FasterNet-T2 while exhibiting around 25.2%(or 13.7%) increase in V100(or MI250) throughput and 24.1% lower CPU latency. This comprehensive evaluation underscores the advantages of PATNet regarding accuracy and throughput (or latency) across various model sizes. So, it also demonstrates that the combination of visual attention and partial convolution significantly improves model performance without impacting throughput.

4.2. PATNet on Downstream Tasks

Setup. We utilize the ImageNet1K pre-trained PATNet as the backbone within the Mask-RCNN[46] detector for object detection and instance segmentation on the MS-COCO 2017 dataset[47], comprising 118K training images and 5K validation images. To

highlight the effectiveness of the backbone itself, we follow the FasterNet[9] approach and employ the AdamW[48] optimizer, conduct training of 12 epochs, use a batch size of 16, image size of 1333×800, and maintain other training settings without further hyperparameter tuning.

Backbone	Params (M)	FLOPs (G)	Throughput MI250 (FPS)↑	AP^b ↑	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50[43]	44.2	253	121	38.0	58.6	41.4	34.4	55.1	36.7
PoolFormer-S24[17]	41.0	233	68	40.1	62.2	43.4	37.0	59.1	39.6
PVT-Small x1.5[49]	44.1	238	98	40.4	62.9	43.8	37.8	60.1	40.3
FasterNet-S[9]	49.0	258	121	39.9	61.2	43.6	36.9	58.1	39.7
PATNet-S(ours)	46.9	216	**122**	**42.7**	**64.9**	**46.5**	**39.3**	**61.8**	**42.2**
ResNet101[20]	63.2	329	62	40.4	61.1	44.2	36.4	57.7	38.8
ResNeXt101-32×4d[50]	62.8	333	51	41.9	62.5	45.9	37.5	59.4	40.2
PoolFormer-S36[17]	50.5	266	44	41.0	63.1	44.8	37.7	60.1	40.0
PVT-Medium[49]	63.9	295	52	42.0	64.4	45.6	39.0	61.6	42.1
FasterNet-M[9]	71.2	344	62	43.0	64.4	47.4	39.1	61.5	42.3
PATNet-M(ours)	78.2	295	**65**	**44.3**	**65.8**	**48.5**	**40.6**	**63.3**	**43.7**
ResNeXt101-64×4d[50]	101.9	487	29	42.8	63.8	47.3	38.4	60.6	41.3
PVT-Large×4d[49]	81.0	358	26	42.9	65.0	46.6	39.5	61.9	42.5
FasterNet-L[9]	110.9	484	35	44.0	65.6	48.2	39.9	62.3	43.0
PATNet-L(ours)	122.0	397	**39**	**44.7**	**66.3**	**49.0**	**41.0**	**63.7**	**44.2**

Table 2. Results using PATNet as a backbone on dense prediction tasks: Object detection and instance segmentation benchmark on the COCO dataset.

Results. Table 2 presents a comparison of PATNet with representative models, reporting performance in terms of average precision (mAP) for both detection and instance segmentation. As shown in Table 2, PATNet consistently outperforms FasterNet, achieving higher average precision (AP) while maintaining similar latency. The results further confirm the generalization capabilities of our proposed PATNet across various tasks.

4.3. Ablation Studies

Partial Attention vs. Full Attention. To prove the superiority of our PAT over full attention mechanisms, we conduct comparative experiments on the PATNet_T2, as shown in Table 3. Specifically, We replace PAT blocks with corresponding full visual attention for comparison respectively. Full visual attention involves conducting visual attention calculations on all channels of the input

doi.org/10.32388/1L3TE6

feature map, without considering the split operation and convolution operation of another branch, which is the common way of conventional visual attention mechanism. The results demonstrate the feasibility of performing attention operations on part channels and also confirm the effectiveness of our improved visual attention mechanism. The results indicate that our PAT achieves a superior balance between inference speed and performance compared to the Full visual attention counterpart.

ch	sp	sf	Params (M)	FLOPs (G)	Throughput (FPS)↑	Latency (ms)↓	Top-1 (%)↑
P	P	P	12.6	1.03	**4761**	**35.2**	**80.2**
F	P	P	13.0	1.04	4662	36.5	80.1
P	F	P	12.6	1.04	4688	35.6	79.9
P	P	F	14.5	1.12	4600	38.6	**80.2**

Table 3. Comparison on PATNet-T2 of partial attention (P), and full attention (F) on ImageNet1K dataset. Where the "ch", "sp", and "sf" denote channel-wise attention, spatial-wise attention, and self-attention respectively.

Effect of PAT blocks. To demonstrate the individual effects of our three PAT blocks, we conducted ablation studies by progressively adding each PAT block one by one, as indicated in Table 4. Experiment results indicate that the three proposed PAT blocks consistently enhance model performance.

Stages	PAT_ch	PAT_sp	PAT_sf	Params (M)	FLOPs (G)	Throughput (FPS)↑	Latency (ms)↓	Top-1 (%)↑
2-2-6-4				11.1	0.92	**6405**	**25.7**	76.0
2-2-6-4	✓			11.1	0.92	5440	30.9	77.4
2-2-6-4	✓	✓		11.5	0.92	5157	31.7	78.9
2-2-6-4	✓	✓	✓	12.6	1.03	4761	35.2	**80.2**
2-2-8-2	✓	✓	✓	9.7	0.98	4976	32.7	78.8

Table 4. Ablation experiments of PATNet-T2 with different configurations of PAT blocks across different model stages on the ImageNet1K dataset.

Different Stage Settings. We adhere to the model design convention of utilizing four stages. However, previous works overlook the importance of the last stage, *e.g.*, FasterNet[9] and MetaFormer[17]. We conduct the comparative experiments between different stage settings (2-2-6-4 vs. 2-2-8-2). The last two rows of Table 4 show that our adjusted stage depths (i.e., 2-2-6-4) can bring more accuracy gain (78.8%→80.2%) with a slight performance drop.

Partial Visual Convolution vs. Regular (or DepthWise) Convolution. To further verify the advantages of our proposed partial visual convolution (PAT_ch) over regular convolution (Conv), we conducted ablation experiments on PATNet-T2 in Table 5. To make a fair comparison, we widen DWConv to keep the throughput of the three convolution types in the same range. Experimental results show that our proposed PAT_ch surpasses regular (or DepthWise) convolution in all metrics including Params, Flops, throughput, latency and Top-1 accuracy, which validates the efficiency and effectiveness of PAT.

Conv3×3	Params (M)	FLOPs (G)	Throughput (FPS)↑	Latency (ms)↓	Top-1 (%)↑
PAT_ch	12.6	1.03	4761	35.2	**80.2**
Conv	15.8	2.12	4190	49.9	79.9
DWConv	15.8	1.28	4017	35.4	79.6

Table 5. Ablation on PATNet-T2 with different convolution types on ImageNet.

5. Conclusion

This paper introduces the concept of partial visual attention mechanism which strategically integrates visual attention mechanisms into partial convolution. We propose three novel partial visual attention blocks including of Partial Channel-Attention block, Partial Spatial-Attention block, and Partial Self-Attention block, which enable models to achieve higher performance while maintaining efficiency. Building upon these innovations, we introduce the PATNet network which outperforms the recent FasterNet network in ImageNet1K classification, as well as COCO detection and segmentation tasks. This underscores the effectiveness of the Partial visual Attention mechanism and signifies a novel convolution approach that strikes an optimal balance between high accuracy and efficiency for various vision tasks. The idea of partial attention still has great potential in the natural language processing (NLP) or large language model (LLM) domains.

Appendix

A.1. Overview

In this supplementary material, we present more explanations and experimental results.

- We first make detailed explanations of our experimental setting and different PATNet variants.
- We then present a full comparison on ImageNet-1k Benchmark.
- We also provide further ablation studies for our proposed Partial Visual Attention mechanism (**PAT**).

A.2. Clarifications on Experimental Setting

Firstly, the configurations of different PATNet variants are presented in Table 6. We also provide ImageNet-1k training and evaluation settings in Table 7. They can be used for reproducing our main results in Figure 1 of the main paper. Different PATNet variants vary in the magnitude of regularization and augmentation techniques. The magnitude increases as the model becomes larger to alleviate overfitting and improve accuracy. Note that most of the compared works in Figure 1 of the main paper, *e.g.*,

MobileViT, FastNet, ConvNeXt, Swin, etc., also adopt such advanced training techniques (ADT). Some even heavily rely on the hyper-parameter search. For others w/o ADT, *e.g.*, ShuffleNetV2, MobileNetV2, and GhostNet, though the comparison is not totally fair, we include them for reference.

Name	Output size	Layer specification		T0	T1	T2	S	M	L			
Embedding	$\frac{h}{4} \times \frac{w}{4}$	Conv_4_c_4,	BN	# Channels c	32	48	64	96	128	160		
Stage 1	$\frac{h}{4} \times \frac{w}{4}$	$\begin{bmatrix} \text{PAT\_ch\_3\_}c\text{\_1\_1/4}, \\ \text{Conv\_1\_2}c\text{\_1}, \\ \text{BN, Acti}, \\ \text{Conv\_1\_}c\text{\_1}, \\ \text{PAT\_sp\_1\_}c\text{\_1\_1/4} \end{bmatrix} \times b_1$	# Blocks b_1	1	2	2	2	2	2			
Merging	$\frac{h}{8} \times \frac{w}{8}$	Conv_2_2c_2,	BN	# Channels $2c$	64	96	128	192	256	320		
Stage 2	$\frac{h}{8} \times \frac{w}{8}$	$\begin{bmatrix} \text{PAT\_ch\_3\_2}c\text{\_1\_1/4}, \\ \text{Conv\_1\_4}c\text{\_1}, \\ \text{BN, Acti}, \\ \text{Conv\_1\_2}c\text{\_1}, \\ \text{PAT\_sp\_1\_2}c\text{\_1\_1/4} \end{bmatrix} \times b_2$	# Blocks b_2	2	2	2	2	3	3			
Merging	$\frac{h}{16} \times \frac{w}{16}$	Conv_2_4c_2,	BN	# Channels $4c$	128	192	256	384	512	640		
Stage 3	$\frac{h}{16} \times \frac{w}{16}$	$\begin{bmatrix} \text{PAT\_ch\_3\_4}c\text{\_1\_1/4}, \\ \text{Conv\_1\_8}c\text{\_1}, \\ \text{BN, Acti}, \\ \text{Conv\_1\_4}c\text{\_1}, \\ \text{PAT\_sp\_1\_4}c\text{\_1\_1/4} \end{bmatrix} \times b_3$	# Blocks b_3	6	6	6	9	16	20			
Merging	$\frac{h}{32} \times \frac{w}{32}$	Conv_2_8c_2,	BN	# Channels $8c$	256	384	512	768	1024	1280		
Stage 4	$\frac{h}{32} \times \frac{w}{32}$	$\begin{bmatrix} \text{PAT\_ch\_3\_8}c\text{\_1\_1/4}, \\ \text{Conv\_1\_16}c\text{\_1}, \\ \text{BN, Acti}, \\ \text{Conv\_1\_8}c\text{\_1}, \\ \text{PAT\_sf\_1\_8}c\text{\_1\_1/4} \end{bmatrix} \times b_4$	# Blocks b_4	4	4	4	4	4	4			
Classifier	1×1	Global average pool,	Conv_1_1280_1,	Acti,	FC_1000	Acti	GELU	GELU	ReLU	ReLU	ReLU	ReLU
Params (M)				4.3	7.8	12.6	29.0	61.3	104.4			
FLOPs (G)				0.25	0.55	1.03	2.71	6.69	11.91			

Table 6. Configurations of different PATNet variants. "Conv_k_c_s" means a convolutional layer with the kernel size of k, the output channels of c, and the stride of s. "PAT_ch_k_c_s_r" means a partial convolution with an extra parameter, the partial ratio of r. "FC_1000" means a fully connected layer with 1000 output channels. $h \times w$ is the input size while b_i is the number of PATNet blocks at stage i. The FLOPs are calculated given the input size of 224×224.

doi.org/10.32388/1L3TE6

Variants	T0	T1	T2	S	M	L
Train Res	Random select from {128,160,192,224,256,288}					
Test Res	224					
Epochs	300					
# of forward pass	188k					
Batch size	4096	4096	4096	4096	2048	2048
Optimizer	AdamW					
Momentum	0.9/0.999					
LR	0.004	0.004	0.004	0.004	0.002	0.002
LR decay	cosine					
Weight decay	0.005	0.01	0.02	0.03	0.05	0.05
Warmup epochs	20					
Warmup schedule	linear					
Label smoothing	0.1					
Dropout	✗					
Stoch. Depth	✗	0.02	0.05	0.1	0.2	0.3
Repeated Aug	✗					
Gradient Clip.	✗	✗	✗	✗	1	0.01
H. flip	✓					
RRC	✓					
Rand Augment	✗	3/0.5	5/0.5	7/0.5	7/0.5	7/0.5
Auto Augment	✗					
Mixup alpha	0.05	0.1	0.1	0.3	0.5	0.7
Cutmix alpha	1.0					
Erasing prob.	✗					
Color Jitter	✗					
PCA lighting	✗					
SWA	✗					
EMA	✗					
Layer scale	✗					
CE loss	✓					

doi.org/10.32388/1L3TE6

Variants	T0	T1	T2	S	M	L
BCE loss	✗					
Mixed precision	✓					
Test crop ratio	0.9					
Top-1 acc. (%)	73.9	78.1	80.2	82.1	83.1	83.9

Table 7. ImageNet-1k training and evaluation settings for different PATNet variants.

For object detection and instance segmentation on the COCO2017 dataset, we equip our PATNet backbone with the popular Mask R-CNN detector. We use ImageNet-1k pre-trained weights to initialize the backbone and Xavier to initialize the add-on layers. Detailed settings are summarized in Table 8.

Variants	S	M	L
Train and test Res	shorter side $=$ 800, longer side $\leq$ 1333		
Batch size	16 (2 on each GPU)		
Optimizer	AdamW		
Train schedule	1$\times$ schedule (12 epochs)		
Weight decay	0.0001		
Warmup schedule	linear		
Warmup iterations	500		
LR decay	StepLR at epoch 8 and 11 with decay rate 0.1		
LR	0.0002	0.0001	0.0001
Stoch. Depth	0.15	0.2	0.3

Table 8. Experimental settings of object detection and instance segmentation on the COCO2017 dataset.

A.3. Full Comparison on ImageNet-1k Benchmark

The full Comparison on ImageNet-1k Benchmark please refer to Table 9, which complements the results provided in Table 1 of the main paper.

Network	Type	Params (M)	FLOPs (G)	Throughput V100 (FPS) ↑	Throughput MI250 (FPS) ↑	Latency CPU (ms) ↓	Top-1 (%) ↑
ShuffleNetV2 x1.5[7]	cnn	3.5	0.30	5315	6642	13.7	72.6
MobileNetV2[2]	cnn	3.5	0.31	3924	7359	13.7	72.0
FasterNet-T0[9]	cnn	3.9	0.34	**8546**	10612	**10.5**	71.9
MobileViT-XXS[20]	hybrid	1.3	0.42	2900	3321	16.7	69.0
MobileViTv2-0.5[24]	hybrid	1.4	0.46	3094	3135	15.8	70.2
PATNet-T0(ours)	hybrid	4.3	0.25	7777	**11744**	12.2	**73.9**
EfficientNet-B0[3]	cnn	5.3	0.39	2934	3344	22.7	**77.1**
GhostNet x1.3[36]	cnn	7.4	0.24	3788	3620	16.7	75.7
ShuffleNetV2 x2[7]	cnn	7.4	0.59	4290	5371	22.6	74.9
MobileNetV2 x1.4[2]	cnn	6.1	0.60	2615	4142	21.7	74.7
FasterNet-T1[9]	cnn	7.6	0.85	**4648**	7198	22.2	76.2
EfficientViT-B1-192[26]	hybrid	9.1	0.38	4072	3912	**19.3**	77.7
MobileViT-XS[20]	hybrid	2.3	1.05	1663	1884	32.8	74.8
PATNet-T1(ours)	hybrid	7.8	0.55	4403	**7379**	21.5	**78.1**
EfficientNet-B1[3]	cnn	7.8	0.70	1730	1583	35.5	79.1
ResNet50[43]	cnn	25.6	4.11	1258	3135	94.8	78.8
FasterNet-T2[9]	cnn	15.0	1.91	2455	4189	43.7	78.9
PoolFormer-S12[17]	hybrid	11.9	1.82	1927	3558	56.1	77.2
MobileViT-S[20]	hybrid	5.6	2.03	1219	1370	52.4	78.4
MobileViTv2-1.0[24]	hybrid	4.9	1.85	1391	1543	41.5	78.1
EfficientViT-B1[26]	hybrid	9.1	0.52	3072	3387	**25.7**	79.4
PATNet-T2(ours)	hybrid	12.6	1.03	**3074**	**4761**	35.2	**80.2**
EfficientNet-B3[3]	cnn	12.0	1.80	768	926	73.5	81.6
ConvNeXt-T[41]	cnn	28.6	4.47	902	1103	99.4	**82.1**
FasterNet-S[9]	cnn	31.1	4.56	1261	2243	96.0	81.3
PoolFormer-S36[17]	hybrid	30.9	5.00	675	1092	152.4	81.4
MobileViTv2-1.5[24]	hybrid	10.6	4.00	812	1000	104.4	80.4
MobileViTv2-2.0[24]	hybrid	18.5	7.50	551	684	103.7	81.2

doi.org/10.32388/1L3TE6

Network	Type	Params (M)	FLOPs (G)	Throughput V100 (FPS) ↑	Throughput MI250 (FPS) ↑	Latency CPU (ms) ↓	Top-1 (%) ↑
Swin-T[44]	hybrid	28.3	4.51	808	1192	107.1	81.3
PATNet-S(ours)	hybrid	29.0	2.71	**1559**	**2422**	**72.5**	**82.1**
EfficientNet-B4[3]	cnn	19.0	4.20	356	442	156.9	82.9
ConvNeXt-S[41]	cnn	50.2	8.71	510	610	185.5	**83.1**
FasterNet-M[9]	cnn	53.5	8.74	621	1098	181.6	83.0
PoolFormer-M36[17]	hybrid	56.2	8.80	444	721	244.3	82.1
Swin-S[44]	hybrid	49.6	8.77	477	732	199.1	83.0
PATNet-M(ours)	hybrid	61.3	6.69	**799**	**1280**	**155.3**	**83.1**
EfficientNet-B5[3]	cnn	30.0	9.90	246	313	333.3	83.6
ConvNeXt-B[41]	cnn	88.6	15.38	322	430	317.1	83.8
FasterNet-L[9]	cnn	93.5	15.52	384	709	312.5	83.5
PoolFormer-M48[17]	hybrid	73.5	11.59	335	556	322.3	82.5
Swin-B[44]	hybrid	87.8	15.47	315	520	333.8	83.5
PATNet-L(ours)	hybrid	104.3	11.91	**426**	**765**	**272.5**	**83.9**

Table 9. Full comparison on ImageNet-1k Benchmark: models with similar top-1 accuracy are grouped together. The best results are in bold.

A.4. Ablation Studies

Partial Visual Attention vs. Conventional Visual Attention. To further prove the superiority of our PAT, we present experiment results for the combination of our partial attention and classic visual attention networks, and the results are shown in Table 10. The results demonstrate the effectiveness of our enhanced Gaussian-SE module.

Visual type	Params(M)	FLOPs(G)	Throughput(fps)↑	latency(ms)↓	Acc1(%)↑
SRM[28]	12.2	1.03	4751	35.2	79.6
SE-NET[14]	12.3	1.04	4910	32.3	79.8
PAT(ours)	12.6	1.03	4761	35.2	**80.2**

Table 10. Comparison on PATNet-T2 of partial visual attention and conventional visual attention on ImageNet1K dataset.

doi.org/10.32388/1L3TE6

Comparison On ImageNet-1k Under Same Training Settings. In order to further verify the effectiveness and fair comparison of our PATNet, we reproduce the results of FastNet on ImageNet-1k but based on our training experiment configuration, the results are shown in Table 11. It can be seen from the results that our PATNet still has great advantages.

Network	Type	Params (M)	FLOPs (G)	Throughput V100 (FPS)↑	Throughput MI250 (FPS)↑	Latency CPU (ms)↓	Top-1 (%)↑
FasterNet-T0[9]	cnn	3.9	0.34	**8546**	10612	**10.5**	71.9
FasterNet-T0*[9]	cnn	3.9	0.34	**8546**	10612	**10.5**	71.0
PATNet-T0(ours)	hybrid	4.3	0.25	7777	**11744**	12.2	**73.9**
FasterNet-T1[9]	cnn	7.6	0.85	**4648**	7198	22.2	76.2
FasterNet-T1*[9]	cnn	7.6	0.85	**4648**	7198	22.2	76.5
PATNet-T1(ours)	hybrid	7.8	0.55	4403	**7379**	21.5	**78.1**
FasterNet-T2[9]	cnn	15.0	1.91	2455	4189	43.7	78.9
FasterNet-T2*[9]	cnn	15.0	1.91	2455	4189	43.7	79.2
PATNet-T2(ours)	hybrid	12.6	1.03	**3074**	**4761**	35.2	**80.2**
FasterNet-S[9]	cnn	31.1	4.56	1261	2243	96.0	81.3
FasterNet-S[9]	cnn	31.1	4.56	1261	2243	96.0	81.5
PATNet-S(ours)	hybrid	29.0	2.71	**1559**	**2422**	**72.5**	**82.1**
FasterNet-M[9]	cnn	53.5	8.74	621	1098	181.6	83.0
FasterNet-M*[9]	cnn	53.5	8.74	621	1098	181.6	83.0
PATNet-M(ours)	hybrid	61.3	6.69	**799**	**1280**	**155.3**	**83.1**
FasterNet-L[9]	cnn	93.5	15.52	384	709	312.5	83.5
FasterNet-L*[9]	cnn	93.5	15.52	384	709	312.5	83.6
PATNet-L(ours)	hybrid	104.3	11.91	**426**	**765**	**272.5**	**83.9**

Table 11. Comparison on ImageNet-1k. The "*" denotes reproduction results based on our experimental setup.

Footnotes

[1] FLOPs stands for floating-point operations, representing the number of arithmetic operations performed. FLOPS stands for floating-point operations per second, indicating the rate or speed at which these operations are executed within a given timeframe.

doi.org/10.32388/1L3TE6

References

1. ^Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications". arXiv preprint arXiv:1704.04861. Available from: https://arxiv.org/abs/1704.04861.
2. a, b, c, d, e, fSandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks." In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 4510–4520.
3. a, b, c, d, e, f, g, h, i, j, k, lTan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR; 2019. p. 6105-6114.
4. ^Yang J, Li C, Dai X, Gao J (2022). "Focal modulation networks". Advances in Neural Information Processing Systems. **35**: 4203–4217.
5. ^Hou Q, Lu CZ, Cheng MM, Feng J (2022). "Conv2former: A simple transformer-style convnet for visual recognition". arXiv preprint arXiv:2211.11943. arXiv:2211.11943.
6. a, bRao Y, Zhao W, Tang Y, Zhou J, Lim SN, Lu J (2022). "Hornet: Efficient high-order spatial interactions with recursive gated convolutions". Advances in Neural Information Processing Systems. **35**: 10353–10366.
7. a, b, c, d, eMa N, Zhang X, Zheng HT, Sun J (2018). "Shufflenet v2: Practical guidelines for efficient cnn architecture design". In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131.
8. a, bDing X, Zhang X, Han J, Ding G (2022). "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns". Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11963–11975.
9. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^, ^Chen J, Kao S, He H, Zhuo W, Wen S, Lee C, Chan SG. "Run, don't walk: Chasing higher flops for faster neural networks." In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023:12021-12031.
10. ^Chen H, Wang Y, Guo J, Tao D (2023). "VanillaNet: the Power of Minimalism in Deep Learning". arXiv preprint arXiv:2305.12972.
11. a, b, cWang J, Zhang S, Liu Y, Wu T, Yang Y, Liu X, Chen K, Luo P, Lin D (2023). "RIFormer: Keep Your Vision Backbone Effective But Removing Token Mixer". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pages 14443–14452.
12. a, bVasu PK, Gabriel J, Zhu J, Tuzel O, Ranjan A. MobileOne: An Improved One Millisecond Mobile Backbone. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023:7907-7917.
13. ^Vasu PK, Gabriel J, Zhu J, Tuzel O, Ranjan A (2023). "FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization". arXiv preprint arXiv:2303.14189. Available from: https://arxiv.org/abs/2303.14189.
14. a, b, c, dHu J, Shen L, Sun G (2018). "Squeeze-and-excitation networks". In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141.
15. a, bWoo S, Park J, Lee JY, Kweon IS (2018). "Cbam: Convolutional block attention module". In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19.
16. ^Courbariaux M, Bengio Y, David JP (2015). "Binaryconnect: Training deep neural networks with binary weights during propagations". Advances in neural information processing systems. **28**.
17. a, b, c, d, e, f, g, h, i, j, k, l, m, nYu W, Luo M, Zhou P, Si C, Zhou Y, Wang X, Feng J, Yan S (2022). "Metaformer is actually what you need for vision". Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 10819–10829.
18. a, b, cHoward A, Sandler M, Chu G, Chen L, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, Le QV, Adam H (2019). "Searching for MobileNetV3". Proceedings of the IEEE/CVF International Conference on Computer Vision. **abs/1905.02244**. Available from: http://arx

iv.org/abs/1905.02244.

19. ^Tan M, Le Q (2021). "Efficientnetv2: Smaller models and faster training." In: International conference on machine learning. PMLR. p. 10096–10106.
20. a, b, c, d, eMehta S, Rastegari M (2021). "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer". arXiv preprint arXiv:2110.02178. Available from: https://arxiv.org/abs/2110.02178.
21. ^Pan J, Bulat A, Tan F, Zhu X, Dudziak L, Li H, Tzimiropoulos G, Martinez B (2022). "Edgevits: Competing light-weight cnns on mobile devices with vision transformers". In: European Conference on Computer Vision. Springer; p. 294–311.
22. ^Raghu M, Unterthiner T, Kornblith S, Zhang C, Dosovitskiy A (2021). "Do vision transformers see like convolutional neural networks?" Advances in Neural Information Processing Systems. **34**: 12116–12128.
23. ^Paul S, Chen P-Y (2022). "Vision transformers are robust learners". Proceedings of the AAAI conference on Artificial Intelligence. **36** (2): 2071–2081.
24. a, b, c, d, e, f, g, hMehta S, Rastegari M (2022). "Separable self-attention for mobile vision transformers". arXiv preprint arXiv:2206.02680. Available from: https://arxiv.org/abs/2206.02680.
25. a, bShaker A, Maaz M, Rasheed H, Khan S, Yang MH, Khan FS (2023). "SwiftFormer: Efficient Additive Attention for Transformer-based Real-time Mobile Vision Applications". arXiv preprint arXiv:2303.15446. arXiv:2303.15446.
26. a, b, c, d, eCai H, Li J, Hu M, Gan C, Han S. EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023:17302-17313.
27. ^Wang S, Li BZ, Khabsa M, Fang H, Ma H (2020). "Linformer: Self-attention with linear complexity". arXiv preprint arXiv:2006.04768. Available from: https://arxiv.org/abs/2006.04768.
28. a, bLee HJ, Kim HE, Nam H (2019). "Srm: A style-based recalibration module for convolutional neural networks". Proceedings of the IEEE/CVF International Conference on Computer Vision. pages 1854–1862.
29. ^Tan M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, Le QV. "Mnasnet: Platform-aware neural architecture search for mobile." In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019. p. 2820-2828.
30. ^Ding X, Zhang X, Ma N, Han J, Ding G, Sun J (2021). "Repvgg: Making vgg-style convnets great again". Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pages 13733–13742.
31. ^Graham B, El-Nouby A, Touvron H, Stock P, Joulin A, Jégou H, Douze M (2021). "Levit: a vision transformer in convnet's clothing for faster inference". Proceedings of the IEEE/CVF international conference on computer vision. pages 12259–12269.
32. ^Wang A, Chen H, Lin Z, Pu H, Ding G (2023). "RepViT: Revisiting Mobile CNN From ViT Perspective". arXiv preprint arXiv:2307.09283. Available from: https://arxiv.org/abs/2307.09283.
33. ^Huang T, You S, Wang F, Qian C, Xu C (2022). "Knowledge distillation from a stronger teacher". Advances in Neural Information Processing Systems. **35**: 33716–33727.
34. ^He K, Chen X, Xie S, Li Y, Dollár P, Girshick R (2022). "Masked autoencoders are scalable vision learners". Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pages 16000–16009.
35. ^Woo S, Debnath S, Hu R, Chen X, Liu Z, Kweon IS, Xie S (2023). "Convnext v2: Co-designing and scaling convnets with masked autoencoders." In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16133–16142.
36. a, bHan K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020). "Ghostnet: More features from cheap operations". Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pages 1580–1589.

37. ^Kirk DB, Hwu WM. Programming massively parallel processors: a hands-on approach. Morgan Kaufmann; 2016.

38. ^Ioffe S, Szegedy C (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: International conference on machine learning. pmlr. pp. 448–456.

39. ^Glorot X, Bengio Y (2010). "Understanding the difficulty of training deep feedforward neural networks". In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings. pp. 249–256.

40. ^Wu K, Peng H, Chen M, Fu J, Chao H (2021). "Rethinking and improving relative position encoding for vision transformer". Proceedings of the IEEE/CVF International Conference on Computer Vision. pages 10033–10041.

41. a, b, c, d, e, f, gLiu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022). "A convnet for the 2020s." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pages 11976–11986.

42. ^Hendrycks D, Gimpel K (2016). "Gaussian error linear units (gelus)". arXiv preprint arXiv:1606.08415.

43. a, b, cHe K, Zhang X, Ren S, Sun J (2015). "Deep residual learning for image recognition". CoRR. **abs/1512.03385**. Available from: http://arxiv.org/abs/1512.03385.

44. a, b, c, d, e, fLiu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021). "Swin transformer: Hierarchical vision transformer using shifted windows". In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022.

45. ^Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. "Imagenet: A large-scale hierarchical image database." In: 2009 IEEE conference on computer vision and pattern recognition. Ieee; 2009. p. 248-255.

46. ^He K, Gkioxari G, Dollár P, Girshick R (2017). "Mask r-cnn". In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969.

47. ^Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: Common objects in context. In: Computer Vision--ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer; 2014. p. 740–755.

48. ^Loshchilov I, Hutter F (2017). "Decoupled weight decay regularization". arXiv preprint arXiv:1711.05101.

49. a, b, cWang W, Xie E, Li X, Fan D-P, Song K, Liang D, Lu T, Luo P, Shao L. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions." In: Proceedings of the IEEE/CVF international conference on computer vision. 2021:568-578.

50. a, bXie S, Girshick R, Dollár P, Tu Z, He K (2017). "Aggregated residual transformations for deep neural networks." In: Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.