Rethinking Mobile Block for Efficient Attention-based Models

Jiangning Zhang[1,2] Xiangtai Li[3] Jian Li[1] Liang Liu[1] Zhucun Xue[4]
Boshen Zhang[1] Zhengkai Jiang[1] Tianxin Huang[2] Yabiao Wang[1†] Chengjie Wang[1†]
[1]Youtu Lab, Tencent [2]Zhejiang University [3]Peking University [4]Wuhan University

Abstract

*This paper focuses on developing modern, efficient, lightweight models for dense predictions while trading off parameters, FLOPs, and performance. Inverted Residual Block (IRB) serves as the infrastructure for lightweight CNNs, but no counterpart has been recognized by attention-based studies. This work rethinks lightweight infrastructure from efficient IRB and effective components of Transformer from a unified perspective, extending CNN-based IRB to attention-based models and abstracting a one-residual Meta Mobile Block (MMB) for lightweight model design. Following simple but effective design criterion, we deduce a modern **I**nverted **R**esidual **M**obile **B**lock (iRMB) and build a ResNet-like Efficient MOdel (EMO) with only iRMB for down-stream tasks. Extensive experiments on ImageNet-1K, COCO2017, and ADE20K benchmarks demonstrate the superiority of our EMO over state-of-the-art methods,* e.g.*, EMO-1M/2M/5M achieve 71.5, 75.1, and 78.4 Top-1 that surpass equal-order CNN-/Attention-based models, while trading-off the parameter, efficiency, and accuracy well: running 2.8-4.0× ↑ faster than EdgeNeXt on iPhone14. Code is available.*

1. Introduction

With a recent increasing demand for storage/computing restricted applications, mobile models with *fewer parameters* and *low FLOPs* have attracted significant attention from developers and researchers. The earliest attempt to design an efficient model dates back to the Inceptionv3 [55] era, which uses asymmetric convolutions to replace standard convolution. Then, MobileNet [20] proposes *depth-wise separable convolution* to significantly decrease the amount of computation and parameters, which is viewed as a fundamental CNN-based component for subsequent works [81, 43, 48, 15]. Remarkably, MobileNetv2 [51] proposes an efficient *Inverted Residual Block* (IRB) based on *Depth-Wise Convolution* (DW-Conv) that is recognized as the infrastructure of efficient models [58] until now. Inevitably, limited by the natural induction bias of static CNN, the *accuracy* of CNN-pure models still maintains a low level of accuracy that needs further improvements. In summary, one extreme core is to *advance a stronger fundamental block going beyond IRB*.

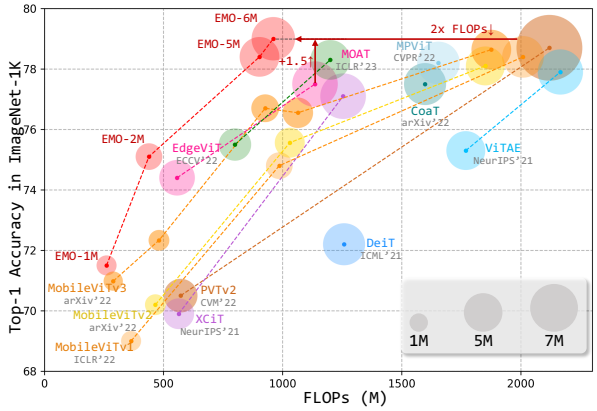


Figure 1: *Performance vs. FLOPs* with concurrent methods.

On the other hand, stared from vision transformer (ViTs) [13], many follow-ups [61, 65, 66, 38, 37, 79, 78] have achieved significant improvements over CNN. This is due to its ability to model dynamically and learn from the extensive dataset, and how to migrate this capability to lightweight CNN is worth our explorations. However, limited by the quadratic amount of computations for Multi-Head Self-Attention (MHSA), the attention-based model requires massive resource consumption, especially when the channel and resolution of the feature map are large. Some works attempt to tackle the above problems by designing variants with linear complexity [29, 7], decreasing the spatial resolution of features [69, 65, 31], rearranging channel [45], using local window attention [38] *etc*. However, these methods still cannot be deployed on devices.

Recently, researchers aim to design efficient hybrid models with lightweight CNNs, and they obtain better performances than CNN-based models with trading off accuracy, parameters, and FLOPs. However, current methods introduce complex structures [46, 47, 64, 6, 44] or multiple hybrid modules [44, 49], which is very detrimental to optimize for applications. So far, little work has been done to explore attention-based counterparts as IRB, and this inspires us to think: *Can we build a lightweight IRB-like infrastructure for attention-based models based on the basic operators?*

Based on the above motivation, we rethink efficient *Inverted Residual Block* in MobileNetv2 [51] and effective MHSA/FFN modules in Transformer [63] from a unified perspective, expecting to integrate both advantages at the infrastructure design level. As shown in Fig. 2-Left, while working to bring one-residual IRB with inductive bias into the attention model, we observe two underlying sub-modules (*i.e.*, FFN and MHSA) in two-residual Transformer share the similar structure to IRB. Thus, we inductively abstract a one-residual Meta Mobile Block (MMB, *c.f.*, Sec. 2.2) that takes parametric arguments *expansion ratio* λ and *efficient operator* $\mathcal{F}$ to instantiate different modules, *i.e.*, IRB, MHSA, and FFN. We argue that *MMB can reveal the consistent essence expression of the above three modules, and it can be regarded as an improved lightweight concentrated aggregate of Transformer*. Furthermore, a simple yet effective *Inverted Residual Mobile Block* (iRMB) is deduced that only contains fundamental Depth-Wise Convolution and our improved EW-MHSA (*c.f.*, Sec. 2.3) and we build a ResNet-like 4-phase Efficient MOdel (EMO) with only iRMBs (*c.f.*, Sec. 2.4). Surprisingly, our method performs better over the SoTA lightweight attention-based models even without complex structures, as shown in Fig. 1. In summary, this work follows simple design criteria while gradually producing an efficient attention-based lightweight model.

Our contributions are four folds: **1)** We extend CNN-based IRB to the two-residual transformer and abstract a one-residual *Meta Mobile Block* (MMB) for lightweight model design. This meta paradigm could describe the current efficient modules and is expected to have the guiding significance in concreting novel efficient modules. **2)** Based on inductive MMB, we deduce a simple yet effective modern *Inverted Residual Mobile Block* (iRMB) and build a ResNet-like Efficient MOdel (EMO) with only iRMB for downstream applications. In detail, iRMB only consists of naive DW-Conv and the improved EW-MHSA to model short-/long-distance dependency, respectively. **3)** We provide detailed studies of our method and give some experimental findings on building attention-based lightweight models, hoping our study will inspire the research community to design powerful and efficient models. **4)** Even without introducing complex structures, our method still achieves very competitive results than concurrent attention-based methods on several benchmarks, *e.g.*, our EMO-1M/2M/5M reach 71.5, 75.1, and 78.4 Top-1 over current SoTA CNN-/Transformer-based models. Besides, EMO-1M/2M/5M armed SSDLite obtain 22.0/25.2/27.9 mAP with only 2.3M/3.3M/6.0M parameters and 0.6G/0.9G/1.8G FLOPs, which exceeds recent MobileViTv2 [47] by +0.8↑/+0.6↑/+0.1↑ with decreased FLOPs by -33%↓/-50%↓/-62%↓; EMO-1M/2M/5M armed DeepLabv3 obtain 33.5/35.3/37.98 mIoU with only 5.6M/6.9M/10.3M parameters and 2.4G/3.5G/5.8G FLOPs, surpassing MobileViTv2 by +1.6↑/+0.6↑/+0.8↑ with much lower FLOPs.

2. Methodology: Induction and Deduction

2.1. Criteria for General Efficient Model

When designing efficient visual models for mobile applications, we advocate the following criteria subjectively and empirically that an efficient model should satisfy as much as possible: ① **Usability.** Simple implementation that does not use complex operators and is easy to optimize for applications. ② **Uniformity.** As few core modules as possible to reduce model complexity and accelerated deployment. ③ **Effectiveness.** Good performance for classification and dense prediction. ④ **Efficiency.** Fewer parameters and calculations with accuracy trade-off. We make a summary of current efficient models in Tab. 1: *1)* Performance of MobileNet series [20, 51, 64] is now seen to be slightly lower, and its parameters are slightly higher than counterparts. *2)* Recent MobileViT series [46, 47, 64] achieve notable performances, but they suffer from higher FLOPs and slightly complex modules. *3)* EdgeNeXt [44] and EdgeViT [49] obtain pretty results, but their basic blocks also consist of elaborate modules. Comparably, the design principle of our EMO follows the above criteria without introducing complicated operations (*c.f.*, Sec. 2.4), but it still obtains impressive results on multiple vision tasks (*c.f.*, Sec. 3).

Table 1: **Criterion comparison for current efficient models**. ①: Usability; ②: Uniformity; ③: Effectiveness; ④: Efficiency. ✔: Satisfied. ✚: Partially satisfied. ✘: Unsatisfied.

Method *vs.* Criterion	①	②	③	④
MobileNet Series [20, 51, 64]	✔	✔	✚	✚
MobileViT Series [46, 47, 64]	✚	✚	✔	✚
EdgeNeXt [44]	✚	✘	✔	✔
EdgeViT [49]	✔	✚	✔	✚
EMO (Ours)	✔	✔	✔	✔

2.2. Meta Mobile Block

Motivation. 1) Recent Transformer-based works [73, 38, 12, 53, 35, 59, 60] are dedicated to improving spatial token mixing under the MetaFormer [74] for high-performance network. CNN-based *Inverted Residual Block* [51] (IRB) is recognized as the infrastructure of efficient models [51, 58], but little work has been done to explore attention-based counterpart. This inspires us to build a lightweight IRB-like infrastructure for attention-based models. 2) While working to bring one-residual IRB with inductive bias into the attention model, we stumble upon two underlying sub-modules (*i.e.*, FFN and MHSA) in two-residual Transformer that happen to share a similar structure to IRB.

Induction. We rethink Inverted Residual Block in MobileNetv2 [51] with core MHSA and FFN modules in Transformer [63], and inductively abstract a general Meta Mobile Block (MMB) in Fig. 2, which takes parametric arguments *expansion ratio* λ and *efficient operator* $\mathcal{F}$ to instantiate

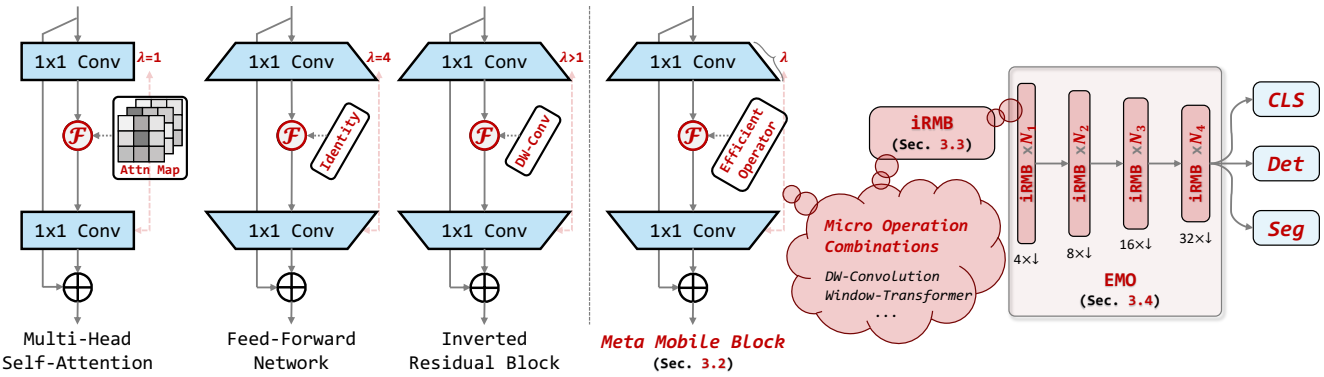


Figure 2: **Left**: Abstracted unified ***Meta-Mobile Block*** from *Multi-Head Self-Attention* / *Feed-Forward Network* [63] and *Inverted Residual Block* [51] (*c.f*. Sec 2.2). The inductive block can be deduced into specific modules using different *expansion ratio* λ and *efficient operator* $\mathcal{F}$. **Right**: ResNet-like *EMO* composed of only deduced *iRMB* (*c.f*. Sec 2.3).

different modules. We argue that *the MMB can reveal the consistent essence expression of the above three modules, and MMB can be regarded as an improved lightweight concentrated aggregate of Transformer*. Also, this is the basic motivation for our elegant and easy-to-use EMO, which only contains one deduced iRMB absorbing advantages of lightweight CNN and Transformer. Take image input $\boldsymbol{X}(\in \mathbb{R}^{C\times H\times W})$ as an example, MMB firstly use a expansion MLP_e with output/input ratio equaling λ to expand channel dimension:

$$\boldsymbol{X}_e = \text{MLP}_e(\boldsymbol{X})(\in \mathbb{R}^{\lambda C\times H\times W}). \tag{1}$$

Then, intermediate operator $\mathcal{F}$ enhance image features further, *e.g*., identity operator, static convolution, dynamic MHSA, *etc*. Considering that MMB is suitable for efficient network design, we present $\mathcal{F}$ as the concept of *efficient operator*, formulated as:

$$\boldsymbol{X}_f = \mathcal{F}(\boldsymbol{X}_e)(\in \mathbb{R}^{\lambda C\times H\times W}). \tag{2}$$

Finally, a shrinkage MLP_s with inverted input/output ratio equaling λ to shrink channel dimension:

$$\boldsymbol{X}_s = \text{MLP}_s(\boldsymbol{X}_f)(\in \mathbb{R}^{C\times H\times W}), \tag{3}$$

where a residual connection is used to get the final output $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{X}_s(\in \mathbb{R}^{C\times H\times W})$. Notice that normalization and activation functions are omitted for clarity.

Relation to MetaFormer. We discuss the differences between our *Meta Mobile Block* and *MetaFormer* [74] in Fig. 3. ***1)*** From the structure, two-residual MetaFormer contains two sub-modules with two skip connections, while our Meta Mobile Block contains only one sub-module that covers one-residual IRB in the field of lightweight CNN. Also, shallower depths require less memory access and save costs [43] that is more general and hardware friendly. ***2)*** From the motivation, MetaFormer is the induction of high-performance Transformer/MLP-like models, while our Meta Mobile Block is the induction of efficient IRB in MobileNetv2 [51] and effective MHSA/FFN in Transformer [63, 13] for designing efficient infrastructure. ***3)*** To a certain extent, the inductive one-residual Meta Mobile Block can be regarded as a conceptual extension of two-residual MetaFormer in the lightweight field. We hope our work inspires more future research dedicated to lightweight model design domain based on attention.

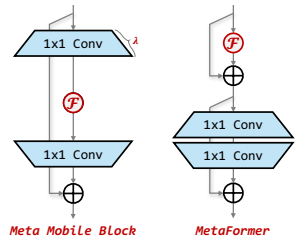


Figure 3: Paradigm illustration with MetaFormer.

2.3. Micro Design: Inverted Residual Mobile Block

Based on the inductive Meta Mobile Block, we instantiate an effective yet efficient modern *Inverted Residual Mobile Block* (iRMB) from a microscopic view in Fig. 4.

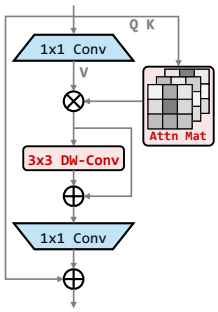


Figure 4: Paradigm of iRMB.

Design Principle. Following criteria in Sec. 2.1, $\mathcal{F}$ in iRMB is modeled as cascaded *MHSA* and *Convolution* operations, formulated as $\mathcal{F}(\cdot) = \text{Conv}(\text{MHSA}(\cdot))$. This design absorbs CNN-like efficiency to model local features and Transformer-like dynamic modelling capability to learn long-distance interactions. However, naive implementation can lead to unaffordable expenses for two main reasons:

1) λ is generally greater than one that the intermediate dimension would be multiple to input dimension, causing quadratic λ increasing of parameters and computations. Therefore, components of $\mathcal{F}$ should be independent or linearly dependent on the number of channels.

2) FLOPs of MHSA is proportional to the quadratic of total

Table 2: Complexity and Maximum Path Length analysis of modules. Input/output feature maps are in $\mathbb{R}^{C\times W\times W}$, $L=W^2$, $l=w^2$, W and w are feature map size and window size, while k and G are kernel size and group number.

Module	#Params	FLOPs	MPL
MHSA	$4(C+1)C$	$8C^2L+4CL^2+3L^2$	$O(1)$
W-MHSA	$4(C+1)C$	$8C^2L+4CLl+3Ll$	$O(Inf)$
Conv	$(Ck^2/G+1)C$	$(2Ck^2/G)LC$	$O(2W/(k-1))$
DW-Conv	$(k^2+1)C$	$(2k^2)LC$	$O(2W/(k-1))$

image pixels, so the cost of a naive Transformer is unaffordable. The specific influences can be seen in Tab. 2.

Deduction. We employ efficient Window-MHSA (W-MHSA) and Depth-Wise Convolution (DW-Conv) with a skip connection to trade-off model cost and accuracy.

Improved EW-MHSA. Parameters and FLOPs for obtaining Q,K in W-MHSA is quadratic of the channel, so we employ unexpanded $\boldsymbol{X}$ to calculate the attention matrix more efficiently, *i.e.*, Q=K=$\boldsymbol{X}$ ($\in \mathbb{R}^{C\times H\times W}$), while the expanded value $\boldsymbol{X}_e$ as V ($\in \mathbb{R}^{\boldsymbol{\lambda}C\times H\times W}$). This improvement is termed as *Expanded Window MHSA* (EW-MHSA) that is more applicative, formulated as:

$$\mathcal{F}(\cdot) = (\text{DW-Conv, Skip})(\text{EW-MHSA}(\cdot)). \tag{4}$$

Also, this cascading manner can increase the expansion speed of the receptive field and reduce the maximum path length of the model to $O(2W/(k-1+2w))$, which has been experimentally verified with consistency in Sec. 3.3.

Flexibility. Empirically, current transformer-based methods [44, 32, 72] reach a consensus that inductive CNN in shallow layers while global Transformer in deep layers composition could benefit the performance. Unlike recent EdgeNeXt that employs different blocks for different depths, our iRMB satisfies the above design principle using only two switches to control whether two modules are used (Code level is also concise in #Supp).

Efficient Equivalent Implementation. MHSA is usually used in channel-consistent projection ($\boldsymbol{\lambda}$=1), meaning that the FLOPs of multiplying attention matrix times expended $\boldsymbol{X}_e$ ($\boldsymbol{\lambda}$>1) will increase by $\boldsymbol{\lambda}$ - 1. Fortunately, the information flow from $\boldsymbol{X}$ to expended V ($\boldsymbol{X}_e$) involves only linear operations, *i.e.*, $\text{MLP}_e(\cdot)$, so we can derive an equivalent proposition:"*When the groups of MLP_e equals to the head number of W-MHSA, the multiplication result of exchanging order remains unchanged.*" To reduce FLOPs, matrix multiplication before MLP_e is used by default.

Choice of Efficient Operators. We also replace the component of $\mathcal{F}$ with group convolution, asymmetric [55] convolution, and performer [7], but they make no further improvements with much higher parameters and FLOPs at the same magnitude for our approach.

Boosting Naive Transformer. To assess iRMB performance, we set $\boldsymbol{\lambda}$ to 4 and replace standard Transformer structure in columnar DeiT [61] and pyramid-like PVT [65]. As shown in Tab. 3, we surprisingly found that iRMB can improve performance with fewer parameters and computations in the same training setting, especially for the columnar ViT. This proves that the one-residual iRMB has obvious advantages over the two-residual Transformer in the lightweight model.

Table 3: A toy experiment for assessing iRMB.

Model	#Params ↓	FLOPs ↓	Top-1 ↑
DeiT-Tiny [61]	5.7M	1258	72.2
DeiT-Tiny w/iRMB	4.9M -14% ↓	1102 -156M ↓	74.3 +2.1% ↑
PVT-Tiny [65]	13.2M	1943	75.1
PVT-Tiny w/iRMB	11.7M -11% ↓	1845 -98M ↓	75.4 +0.3% ↑

Parallel Design of $\mathcal{F}$. We also implement the parallel structure of DW-Conv and EW-MHSA with half the number of channels in each component, and some configuration details are adaptively modified to ensure the same magnitude. Comparably, this parallel model gets 78.1 (-0.3↓) Top-1 in ImageNet-1k dataset with 5.1M parameters and 964M FLOPs (+63M↑ than EMO-5M), but its throughput will slow down by about -7%↓. This phenomenon is also discussed in the work [43] that: "Network fragmentation reduces the degree of parallelism".

2.4. Macro Design of EMO for Dense Prediction

Based on the above criteria, we design a ResNet-like 4-phase Efficient MOdel (EMO) based on a series of iRMBs for dense applications, as shown in Fig. 2-**Right**.

1) For the overall framework, EMO consists of only iRMBs without diversified modules②, which is a departure from recent efficient methods [46, 44] in terms of designing idea.

2) For the specific module, iRMB consists of only standard convolution and multi-head self-attention without other complex operators①. Also, benefitted by DW-Conv, iRMB can adapt to down-sampling operation through the stride and does not require any position embeddings for introducing inductive bias to MHSA②.

3) For variant settings, we employ gradually increasing expansion rates and channel numbers, and detailed configurations are shown in Tab. 4. Results for basic classification and multiple downstream tasks in Sec. 3 demonstrate the superiority of our EMO over SoTA lightweight methods on magnitudes of 1M, 2M, and 5M③④.

Details. Since MHSA is better suited for modelling semantic features for deeper layers, we only turn it on at stage-3/4 following previous works [44, 32, 72]. Note that this never violates the uniformity criterion, as the shutdown of MHSA was a special case of iRMB structure. To further increase the stability of EMO, BN [26]+SiLU [18] are bound to DW-Conv while LN [2]+GeLU [18] are bound to EW-MHSA. Also, iRMB is competent for down-sampling operations.

Relation to MetaFormer. ***1)*** From the structure,

Table 4: Core configurations of EMO variants.

Items	EMO-1M	EMO-2M	EMO-5M
Depth	[2, 2, 8, 3]	[3, 3, 9, 3]	[3, 3, 9, 3]
Emb. Dim.	[32, 48, 80, 168]	[32, 48, 120, 200]	[48, 72, 160, 288]
Exp. Ratio	[2.0, 2.5, 3.0, 3.5]	[2.0, 2.5, 3.0, 3.5]	[2.0, 3.0, 4.0, 4.0]

MetaFormer extended dense prediction model employs an extra patch embedding layer for down-sampling, while our EMO only consists of iRMB. ***2)*** From the result, our instantiated EMO-5M (w/ 5.1M #Params and 903M FLOPs) exceeds instantiated PoolFormer-S12 (w/ 11.9M #Params and 1,823M FLOPs) by +1.2↑, illustrating that a stronger efficient operator makes a advantage. ***3)*** We further replace Token Mixer in MetaFormer with $\mathcal{F}$ in iRMB and build a 5.3M model *vs*. our EMO-5M. It only achieves 77.5 Top-1 on ImageNet-1k, *i.e*., -0.9↓ than our model, meaning that our proposed Meta Mobile Block has a better advantage for constructing lightweight models than two-residual MetaFormer.

Importance of Instantiated Efficient Operator. Our defined *efficient operator* $\mathcal{F}$ contains two core modules, *i.e*., EW-MHSA and DW-Conv. In Tab. 5, we conduct an ablation experiment to study the effect of both modules. The first row means that neither EW-MHSA nor DW-Conv is used, *i.e*., the model is almost composed of MLP layers with several DW-Conv for down-sampling, and $\mathcal{F}$ degenerates to Identity operation. Surprisingly, this model still produces a respectable result, *i.e*., 73.5 Top-1. Comparatively, results of the second and third rows demonstrate that each component contributes to the performance, *e.g*., +3.1↑ and +4.1↑ when adding DW-Conv and EW-MHSA, respectively. Our model achieves the best result, *i.e*., 78.4 Top-1 when both components are used. Besides, this experiment illustrates that the specific instantiation of iRMB is very important to model performance.

Table 5: Ablation study on components in iRMB.

EW-MHSA	DW-Conv	Top-1
✘	✘	73.5
✔	✘	76.6 +3.1 ↑
✘	✔	77.6 +4.1 ↑
✔	✔	78.4 +4.9 ↑

Order of Operators. Based on EMO-5M, we switch the order of DW-Conv/EW-MHSA and find a slight drop in performance (-0.6↓), so EW-MHSA performs firstly by default.

3. Experiments

3.1. Image Classification

Setting. Due to various training recipes of SoTA methods [19, 13, 61, 46, 47, 42, 44] that could lead to potentially unfair comparisons (summarized in Tab. 6), we employ a weaker training recipe to increase model persuasion and open the source code for subsequent fair comparisons in #Supp. All experiments are conducted on ImageNet-1K dataset [11] without extra datasets and pre-trained models. Each model is trained for standard 300 epochs from scratch at 224×224, and AdamW [41] optimizer is employed with betas (0.9, 0.999), weight decay $5e^{-2}$, learning rate $6e^{-3}$, and batch size 2,048. We use Cosine scheduler [40] with 20 warmup epochs, Label Smoothing 0.1 [56], stochastic depth [22], and RandAugment [10] during training, while LayerScale [62], Dropout [54], MixUp [77], CutMix [76], Random Erasing [83], Position Embeddings [13], Token Labeling [28], and Multi-Scale training [46] are *disabled*. EMO is implemented by PyTorch [50], based on TIMM [67], and trained with 8×V100 GPUs.

Table 6: Comparison of **training recipes among contemporary methods** and we employ the same setting in all experiments. Please zoom in for clearer comparisons. Abbreviated MNet and MViT: MobileNet and MobileViT.

Super-Params.	MNetv3 [19] ICCV'19	ViT [13] ICLR'21	DeiT [61] ICML'21	MViTv1 [46] ICLR'22	MViTv2 [47] arXiv'22	EdgeNeXt [44] arXiv'22	EMO Ours
Epochs	300	300	300	300	300	300	300
Batch size	512	4096	1024	1024	1024	4096	2048
Optimizer	RMSprop	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Learning rate	$6.4e^{-2}$	$3e^{-3}$	$1e^{-3}$	$2e^{-3}$	$2e^{-3}$	$6e^{-3}$	$6e^{-3}$
Learning rate decay	$1e^{-5}$	$3e^{-1}$	$5e^{-2}$	$1e^{-2}$	$5e^{-2}$	$5e^{-2}$	$5e^{-2}$
Warmup epochs	3	3.4	5	2.4	16	20	20
Label smoothing	0.1	✘	0.1	0.1	0.1	0.1	0.1
Drop out rate	✘	0.1	✘	0.1	✘	✘	✘
Drop path rate	0.2	✘	0.1	✘	✘	0.1	0.1
RandAugment	9/0.5	✘	9/0.5	✘	9/0.5	9/0.5	9/0.5
Mixup alpha	✘	✘	0.8	✘	0.8	✘	✘
Cutmix alpha	✘	✘	1.0	✘	1.0	✘	✘
Erasing probability	0.2	✘	0.25	✘	0.25	✘	✘
Position embedding	✘	✔	✔	✘	✘	✔	✘
Multi-scale sampler	✘	✘	✘	✔	✘	✔	✘

Results. EMO is evaluated with SoTAs on three small scales, and quantitative results are shown in Tab. 7. Surprisingly, our method obtains the current best results without using complex modules and MobileViTv2-like strong training recipe. For example, the smallest EMO-1M obtains SoTA 71.5 Top-1 that surpasses CNN-based MobileNetv3-L-0.50 [19] by +2.7↑ with nearly half parameters and Transformer-based MobileViTv2-0.5 [47] by +1.3↑ with only 56% FLOPs. Larger EMO-2M achieves SoTA 75.1 Top-1 with only 439M FLOPs, nearly half of MobileVit-XS [46]. Comparatively, the latest EdgeViT-XXX [49] obtains a worse 74.4 Top-1 while requiring +78%↑ parameters and +27%↑ FLOPs. Consistently, EMO-5M obtains a superior trade-off between #Params (5.1M) / #FLOPs (903M) and accuracy (78.4), which is more efficient than contemporary counterparts. Surprisingly, after increasing the channel of the fourth stage of EMO-5M from 288 to 320, the new EMO-6M reaches 79.0 Top-1 with only 961M FLOPs.

Training Recipes Matters. We evaluate EMO with different training recipes:

MNetv3	DeiT	EdgeNeXt	EMO
NaN	78.1	78.3	78.4

We find that the simple training recipe (Ours) is enough to get good results for our lightweight EMO, while existing stronger recipes (especially in EdgeNeXt [44]) will not improve the model further. NaN indicates the model did not

Table 7: Classification performance on ImageNet-1K dataset. White, yellow, and blue backgrounds indicate CNN-based, Transformer-based, and our EMO, respectively. This kind of display continues for all subsequent experiments. Unit: (M). Abbreviated MNet and MViT: MobileNet and MobileViT.

Model	#Params ↓	FLOPs ↓	Reso.	Top-1	#Pub
MNetv3-L-0.50 [19]	2.6	69	224^2	68.8	ICCV'19
MViTv1-XXS [46]	1.3	364	256^2	69.0	ICLR'22
MViTv2-0.5 [47]	1.4	466	256^2	70.2	arXiv'22
EdgeNeXt-XXS [44]	1.3	261	256^2	71.2	ECCVW'22
EMO-1M	1.3	261	224^2	71.5	
MNetv2-1.40 [51]	6.9	585	224^2	74.7	CVPR'18
MNetv3-L-0.75 [19]	4.0	155	224^2	73.3	ICCV'19
MoCoViT-1.0 [42]	5.3	147	224^2	74.5	arXiv'22
PVTv2-B0 [66]	3.7	572	224^2	70.5	CVM'22
MViTv1-XS [46]	2.3	986	256^2	74.8	ICLR'22
MFormer-96M [6]	4.6	96	224^2	72.8	CVPR'22
EdgeNeXt-XS [44]	2.3	538	256^2	75.0	ECCVW'22
EdgeViT-XXS [49]	4.1	557	256^2	74.4	ECCV'22
tiny-MOAT-0 [72]	3.4	800	224^2	75.5	ICLR'23
EMO-2M	2.3	439	224^2	75.1	
MNetv3-L-1.25 [19]	7.5	356	224^2	76.6	ICCV'19
EfficientNet-B0 [58]	5.3	399	224^2	77.1	ICML'19
DeiT-Ti [61]	5.7	1258	224^2	72.2	ICML'21
XCiT-T12 [1]	6.7	1254	224^2	77.1	NIPS'21
LightViT-T [23]	9.4	700	224^2	78.7	arXiv'22
MViTv1-S [46]	5.6	2009	256^2	78.4	ICLR'22
MViTv2-1.0 [47]	4.9	1851	256^2	78.1	arXiv'22
EdgeNeXt-S [44]	5.6	965	224^2	78.8	ECCVW'22
PoolFormer-S12 [74]	11.9	1823	224^2	77.2	CVPR'22
MFormer-294M [6]	11.4	294	224^2	77.9	CVPR'22
MPViT-T [30]	5.8	1654	224^2	78.2	CVPR'22
EdgeViT-XS [49]	6.7	1136	256^2	77.5	ECCV'22
tiny-MOAT-1 [72]	5.1	1200	224^2	78.3	ICLR'23
EMO-5M	5.1	903	224^2	78.4	
EMO-6M	6.1	961	224^2	79.0	

train well for the possibly unadapted hyper-parameters.

3.2. Downstream Tasks

Object detection. ImageNet-1K pre-trained EMO is integrated with light SSDLite [19] to evaluate its performance on MS-COCO 2017 [34] dataset at 320×320 resolution. Considering the fairness of the comparison and the friendliness of the community, we employ standard MMDetection library [4] for experiments and replace the optimizer with AdamW [41] without tuning other parameters.

Comparison results with SoTA methods are shown in Tab. 8, and our EMO surpasses corresponding counterparts by apparent advantages. For example, SSDLite equipped with EMO-1M achieves 22.0 mAP with only 0.6G FLOPs and 2.3M parameters, which boosts +2.1↑ compared with SoTA MobileViT [46] with only 66% FLOPs. Consistently, our EMO-5M obtains the highest 27.9 mAP so far with much fewer FLOPs, *e.g.*, 53% (1.8G) of MobileViT-S [46] (3.4G) and 0.3G less than EdgeNeXt-S (2.1G).

Semantic segmentation. ImageNet-1K pre-trained EMO is integrated with DeepLabv3 [5] and PSPNet [82] to adequately evaluate its performance on challenging ADE20K [84] dataset at 512×512 resolution. Also, we

Table 8: Object detection performance by SSDLite on MS-COCO. Abbreviated MNet/MViT: MobileNet/MobileViT.

Backbone	#Params ↓	FLOPs ↓	mAP
MNetv1 [20]	5.1	1.3G	22.2
MNetv2 [51]	4.3	0.8G	22.1
MNetv3 [19]	5.0	0.6G	22.0
MViTv1-XXS [46]	1.7	0.9G	19.9
MViTv2-0.5 [47]	2.0	0.9G	21.2
EMO-1M	2.3	0.6G	22.0
MViTv2-0.75 [47]	3.6	1.8G	24.6
EMO-2M	3.3	0.9G	25.2
ResNet50 [17]	26.6	8.8G	25.2
MViTv1-S [46]	5.7	3.4G	27.7
MViTv2-1.25 [47]	8.2	4.7G	27.8
EdgeNeXt-S [44]	6.2	2.1G	27.9
EMO-5M	6.0	1.8G	27.9

Table 9: Semantic segmentation performance on ADE20K dataset. Abbreviated MNet/MViT: MobileNet/MobileViT.

Backbone	DeepLabv3 [5]			PSPNet [82]		
	#Params	FLOPs	mIoU	#Params	FLOPs	mIoU
MViTv2-0.5 [47]	6.3	26.1	31.9	3.6	15.4	31.8
EMO-1M	5.6	2.4	33.5	4.3	2.1	33.2
MNetv2 [51]	18.7	75.4	34.1	13.7	53.1	29.7
MViTv2-0.75 [47]	9.6	40.0	34.7	6.2	26.6	35.2
EMO-2M	6.9	3.5	35.3	5.5	3.1	34.5
MViTv2-1.0 [47]	13.4	56.4	37.0	9.4	40.3	36.5
EMO-5M	10.3	5.8	37.8	8.5	5.3	38.2

employ standard MMSegmentation library [9] for experiments and replace the optimizer with AdamW [41] without tuning other parameters. Details can be viewed in the code.

Comparison results with SoTA methods are shown in Tab. 9, and our EMO is apparently superior over SoTA Transformer-based MobileViTv2 [47] at various scales when integrating into segmentation frameworks. For example, EMO-1M/2M/5M armed DeepLabv3 obtains 33.5/35.3/37.8 mIoU, surpassing MobileViTv2 counterparts by +1.6↑/+0.6↑/+0.6↑, while owning fewer parameters and FLOPs benefitted from efficient iRMB. Also, consistent conclusions can be reached when applying EMO as the backbone network of PSPNet. More qualitative results in #Supp.

3.3. Extra Ablation and Explanatory Analysis

Throughput Comparison. In Tab. 10, we present throughput evaluation results compared with SoTA EdgeNeXt [44]. The test platforms are AMD EPYC 7K62 CPU and V100 GPU with a resolution of 224×224 and a batch size of 256. Results indicate that our EMO has an faster speed on both platforms, even though both methods have similar FLOPs. For example, EMO-1M achieves speed boosts of +20%↑ for GPU and +116%↑ for CPU than EdgeNeXt-XXS over the same FLOPs. This gap is further widened on mobile devices (iPhone14), *i.e.*, *2.8*× ↑, *3.9*× ↑, and *4.80*× ↑ faster than SoAT EdgeNeXt [44]. This derives from our simple and

Table 10: Comparisons of throughput on CPU/GPU and running speed on mobile iPhone14 (ms).

Method	FLOPs	CPU	GPU	iPhone14
EdgeNeXt-XXS	261M	73.1	2860.6	12.6
EMO-1M	261M	158.4	3414.6	4.5 2.8× ↑
EdgeNeXt-XS	538M	69.1	1855.2	20.2
EMO-2M	439M	126.6	2509.8	5.1 3.9× ↑
EdgeNeXt-S	965M	54.2	1622.5	27.7
EMO-5M	903M	106.5	1731.7	6.8 4.0× ↑

Table 11: Performance *vs*. depth configurations.

Depth	#Params	FLOPs	Top-1
[2, 2, 10, 3]	5.3M	901M	78.0
[2, 2, 12, 2]	5.0M	970M	77.8
[4, 4, 8, 3]	4.9M	905M	78.1
[3, 3, 9, 3]	5.1M	903M	78.4

device-friendly iRMB with no other complex structures, *e.g.*, Res2Net module [14], transposed channel attention [1], *etc*.

Depth Configuration. We assess another three models with different depths on the order of 5M in Tab. 11. The selected depth configuration produces relatively better performance.

Normalization Type in Different Stages. BN and LN of the same dimension have the same parameters and similar FLOPs, but LN has a tremendous negative impact on the speed of vision models limited by the underlying optimization of GPU structure. Fig. 5A shows the throughput of EMO-5M with the LN layer applying to different stages, and LN is used to stage-3/4 (S-34) by default. As more stages replace BN with LN, *i.e.*, S-1234, throughput decreases significantly (1,693 → 952) while the benefit is modest (+0.2↑). We found that the model is prone to unstable NaNs when LN is not used; thus, we argue that LN is necessary but used in a few stages is enough for Attention-based EMO.

MHSA in Different Stages. Fig. 5B illustrates the changes in model accuracy when applying MHSA to different stages based on EMO-5M. Results indicate that MHSA always positively affects model accuracy, no matter the stage inserted. Our efficient model obtains the best result when applying MHSA to every stage, but this would take an extra 10%↑ more FLOPs, *i.e.*, from 903M to 992M. Therefore, only using MHSA in the last two stages is used by default, which trades off the accuracy and efficiency of the model.

Effect of Drop Path Rate. Fig. 5C explores the effect of drop path rate for training EMO-5M. Results show that the proposed model is robust to this training parameter in the range [0, 0.1] that fluctuates accuracy within 0.2, and 0.05 can obtain a slightly better result.

Effect of Batch Size. Fig. 5D explores the effect of batch size for training EMO. Small batch size ($\leq$ 512) will bring performance degradation, while high batch size will suffer from performance saturation, and it will also put higher

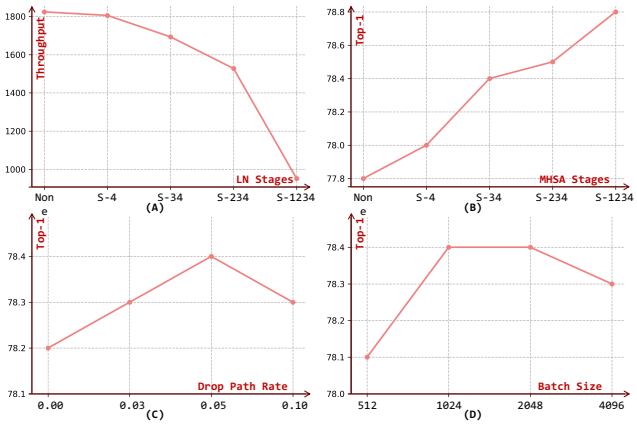


Figure 5: Ablation studies on ImageNet-1K with EMO-5M.

requirements on the hardware. Therefore, 1,024 or 2,048 is enough to meet the training requirement.

Distributions of #Params and FLOPs. iRMB mainly consists of DW-Conv and EW-MHSA modules, and Fig. 6 further displays distributions of #Params and FLOPs. In general, DW-Conv and MHSA account for a low proportion of #Params and FLOPs, *i.e.*, 4.6%/4.1% and 13.8%/14.6%, respectively. Also, we found that #Params is consistent with the proportion of FLOPs for our method, meaning that EMO is a relatively balanced model.

Figure 6: #Params and FLOPs distributions of EMO-5M in terms of core modules in iRMB. MLP represents the expansion/shrinkage operations outside DW-Conv and MHSA.

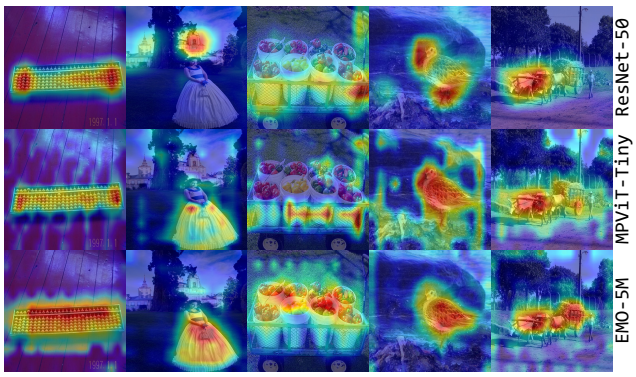


Figure 7: Visualizations by Grad-CAM among CNN-based ResNet, Transformer-based MPViT, and our EMO.

Attention Visualizations by Grad-CAM. To better illustrate the effectiveness of our approach, Grad-CAM [52] is used to highlight concerning regions of different models.

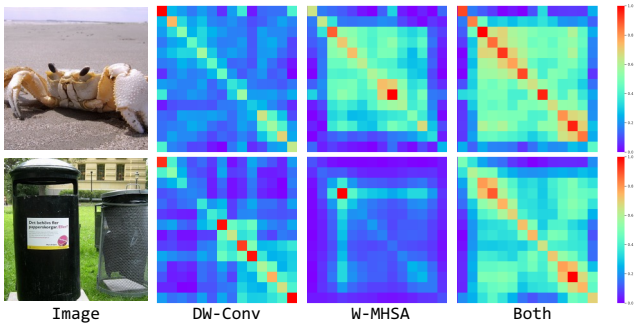


Figure 8: Diagonal similarity with different components.

As shown in Fig. 7, CNN-based ResNet tends to focus on specific objects, and Transformer-based MPViT pays more attention to global features. Comparatively, our EMO could focus more accurately on salient objects while keeping the capability of perceiving global regions. This potentially explains why EMO gets better results in various tasks.

Feature Similarity Visualizations. As mentioned in Sec. 2.3, cascaded *Convolution* and *MHSA* operations can increase the expansion speed of the receptive field. To verify the validation of this design, we visualize the similarity of diagonal pixels in Stage-3 with different compositions, *i.e.*, only DW-Conv, only EW-MHSA, and both modules. As shown in Fig. 8, results show that features tend to have short-distance correlations when only DW-Conv is used, while EW-MHSA brings more long-distance correlations. Comparatively, iRMB takes advantage of both modules with a larger receptive field, *i.e.*, distant locations have high similarities.

4. Related Work

Efficient CNN Models. With increasing demands of neural networks for mobile vision applications, efficient model designing has attracted extensive attention from researchers in recent years. SqueezeNet [24] replaces 3x3 filters with 1x1 filters and decreases channel numbers to reduce model parameters, while Inceptionv3 [55] factorizes the standard convolution into asymmetric convolutions. Later MobileNet [20] introduces depth-wise separable convolution to alleviate a large amount of computation and parameters, followed in subsequent lightweight models [21, 51, 81, 43, 48, 15]. Besides the above hand-craft methods, researchers exploit automatic architecture design in the pre-defined search space [19, 58, 57, 36, 3].

Hugging Vision Transformer with CNN. Since ViT [13] first introduces Transformer structure [63] into visual tasks, massive improvements have successfully been developed. DeiT [61] provides a benchmark for efficient transformer training, subsequent works [65, 66, 38] employ ResNet-like [17] pyramid structure to form pure Transformer-based models for dense prediction tasks. However, the absence of 2D convolution will potentially increase the optimization difficulty and damage the model accuracy for lacking local inductive bias, so researchers [16, 27] concentrate on how to better integrate convolution into Transformer for obtaining stronger hybrid models. *E.g.*, work [75] incorporate convolution design into FFN, works [8, 32, 71] regard convolution as the positional embedding for enhancing inductive bias of the model, and works [70, 69] for attention and QKV calculations, respectively. Unlike the above methods that improve naive Transformer to obtain high performance, we study how to build a simple but effective lightweight model based on an improved one-residual attention block.

Efficient Transformer Improvements. Recently, researchers have started to lighten Transformer-based models for low computational power. Tao *et al*. [23] introduce additional learnable tokens to capture global dependencies efficiently, and Chen *et al*. [23] design a parallel structure of MobileNet and Transformer with a two-way bridge in between. Works [80, 49] improve an efficient Transformer block by borrowing convolution operation, while EdgeNeXt [44] absorbs effective Res2Net [14] and transposed channel attention [1]. The recently popular MobileVit series [46, 47, 64] fuse improved MobileViT blocks with Mobile blocks [51] and achieve significant improvements over MobileNet [20, 51, 19] on several vision tasks. However, most current approaches build on transformer structure and require *elaborate complex modules*, which limits the mobility and usability of the model. In summary, how to balance parameters, computation, and accuracy while designing an easy-to-use mobile model still needs further research.

5. Conclusion and Future Works

This work rethinks lightweight infrastructure from efficient IRB and effective components of Transformer in a unified perspective, and we propose the concept of Meta Mobile Block for designing efficient models. In detail, we deduce a modern infrastructural iRMB and build a lightweight attention-based EMO with only iRMB for downstream tasks. Massive experiments on several datasets demonstrate the superiority of our approach. Also, we provide detailed studies of our method and give some experimental findings on building an attention-based lightweight model. Hope our study will inspire researchers to design more power efficient models and make interesting applications.

More complex operators may potentially improve the effectiveness of the model, *e.g.*, transposed channel attention [1], multi-scale Res2Net [14], and efficient Performer [7], *etc*., which should be thoroughly tried and experimented further to explore the upper limits of the efficient model structure. Also, higher resolution input, combined with Neural Architecture Search (NAS), distillation from heavy models, training on larger ImageNet-21K dataset, and stronger training augmentations/strategies [46, 44, 28] will further improve the model performance. Limited by

the current computational power, we will leave the above-mentioned attempts in our future works.

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, volume 34, 2021. 6, 7, 8, 12

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. 8

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6, 12

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 6

[6] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *CVPR*, pages 5270–5279, 2022. 1, 6

[7] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *ICLR*, 2021. 1, 4, 8

[8] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *The Eleventh International Conference on Learning Representations*, 2023. 8

[9] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6

[10] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 702–703, 2020. 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 5

[12] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, pages 12124–12134, 2022. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3, 5, 8

[14] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*, 43(2):652–662, 2019. 7, 8

[15] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *CVPR*, pages 1580–1589, 2020. 1, 8

[16] Mohammed Hassanin, Saeed Anwar, Ibrahim Radwan, Fahad S Khan, and Ajmal Mian. Visual attention methods in deep learning: An in-depth survey. *arXiv preprint arXiv:2204.07756*, 2022. 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 8, 12

[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 *ICCV*, pages 1314–1324, 2019. 5, 6, 8

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 2, 6, 8

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 8

[22] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016. 5

[23] Tao Huang, Lang Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Lightvit: Towards light-weight convolution-free vision transformers. *arXiv preprint arXiv:2207.05557*, 2022. 6, 8

[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. 8

[25] Apple Inc. Classifying images with vision and core ml. https://developer.apple.com/documentation/vision/classifying_images_with_vision_and_core_ml, 2023. 12

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. PMLR, 2015. 4

[27] Khawar Islam. Recent advances in vision transformer: A survey and outlook of recent work. *arXiv preprint arXiv:2203.01536*, 2022. 8

[28] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens

matter: Token labeling for training better vision transformers. In *NeurIPS*, volume 34, 2021. 5, 8

[29] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 1

[30] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, pages 7287–7296, 2022. 6, 12

[31] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022. 1

[32] Kunchang Li, Yali Wang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *ICLR*, 2022. 4, 8

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 12

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6, 12

[35] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *NeurIPS*, 34:9204–9215, 2021. 2

[36] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 8

[37] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, pages 12009–12019, 2022. 1

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 1, 2, 8, 12

[39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 12

[40] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5, 6, 12

[42] Hailong Ma, Xin Xia, Xing Wang, Xuefeng Xiao, Jiashi Li, and Min Zheng. Mocovit: Mobile convolutional vision transformer. *arXiv preprint arXiv:2205.12635*, 2022. 5, 6

[43] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, pages 116–131, 2018. 1, 3, 4, 8

[44] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *ECCVW*, 2022. 1, 2, 4, 5, 6, 8, 12

[45] Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Delight: Deep and light-weight transformer. In *ICLR*, 2021. 1

[46] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 1, 2, 4, 5, 6, 8

[47] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *arXiv preprint arXiv:2206.02680*, 2022. 1, 2, 5, 6, 8

[48] Sachin Mehta, Mohammad Rastegari, Linda Shapiro, and Hannaneh Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *CVPR*, pages 9190–9200, 2019. 1, 8

[49] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *ECCV*, 2022. 1, 2, 5, 6, 8, 12

[50] 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. In *NeurIPS*, volume 32, 2019. 5

[51] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 1, 2, 3, 6, 8

[52] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 7

[53] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng YAN. Inception transformer. In *NeurIPS*, 2022. 2

[54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. 15(1):1929–1958, 2014. 5

[55] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1, 4, 8

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 5

[57] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 8

[58] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. 1, 2, 6, 8

[59] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*, 34:24261–24272, 2021. 2

[60] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al.

Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 1, 4, 5, 6, 8, 12

[62] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, pages 32–42, 2021. 5

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 30, 2017. 2, 3, 8

[64] Shakti N Wadekar and Abhishek Chaurasia. Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features. *arXiv preprint arXiv:2209.15159*, 2022. 1, 2, 8

[65] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 1, 4, 8, 12

[66] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, pages 1–10, 2022. 1, 6, 8, 12

[67] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 5

[68] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. In *NeurIPSW*, 2021. 12

[69] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31, 2021. 1, 8

[70] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *ICCV*, pages 9981–9990, 2021. 8

[71] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. In *NeurIPS*, volume 34, 2021. 8, 12

[72] Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan Yuille, Hartwig Adam, and Liang-Chieh Chen. Moat: Alternating mobile convolution and attention brings strong vision models. 2023. 4, 6

[73] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. In *NeurIPS*, volume 34, pages 30008–30022, 2021. 2

[74] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, pages 10819–10829, 2022. 2, 3, 6, 12

[75] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *ICCV*, pages 579–588, 2021. 8

[76] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 5

[77] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 5

[78] Jiangning Zhang, Xiangtai Li, Yabiao Wang, Chengjie Wang, Yibo Yang, Yong Liu, and Dacheng Tao. Eatformer: improving vision transformer inspired by evolutionary algorithm. *arXiv preprint arXiv:2206.09325*, 2022. 1

[79] Jiangning Zhang, Chao Xu, Jian Li, Wenzhou Chen, Yabiao Wang, Ying Tai, Shuo Chen, Chengjie Wang, Feiyue Huang, and Yong Liu. Analogous to evolutionary algorithm: Designing a unified sequence model. *Advances in Neural Information Processing Systems*, 34:26674–26688, 2021. 1

[80] Qinglong Zhang and Yu-Bin Yang. Rest: An efficient transformer for visual recognition. In *NeurIPS*, volume 34, pages 15475–15485, 2021. 8

[81] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 1, 8

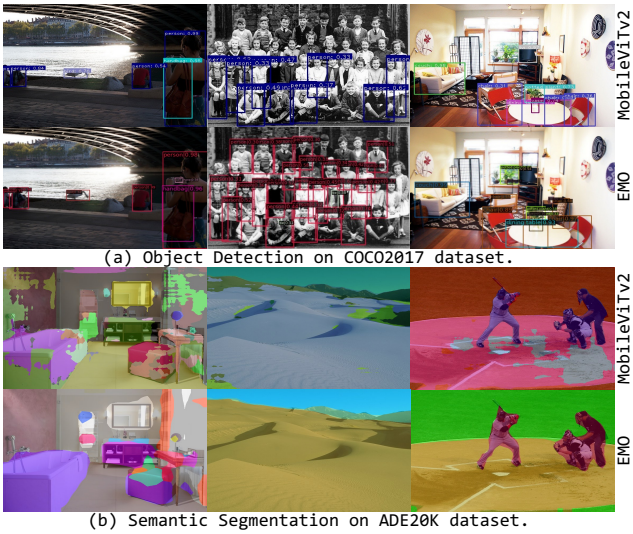
[82] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 6

[83] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, volume 34, pages 13001–13008, 2020. 5

[84] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127(3):302–321, 2019. 6

A. Qualitative Comparisons

Qualitative detection visualizations compared with MobileViTv2 by SSDLite are shown in Fig. 9-(a), and results indicate the superiority of our EMO for capturing adequate and accurate information on different scenes. Also, qualitative segmentation results compared with MobileViTv2 by DeepLabv3 are shown in Fig. 9-(b), and EMO-based model can obtain more accurate and stable results than the comparison approach, *e.g.*, more consistent bathtub, sand, and baseball field segmentation results.



(a) Object Detection on COCO2017 dataset.

(b) Semantic Segmentation on ADE20K dataset.

Figure 9: Qualitative comparisons with MobileNetv2 on two main downstream tasks. Zoom in for more details.

B. Object Detection by Heavy RetinaNet

We further explore EMO integrated with heavy RetinaNet [33] and report additional results at 5M parameters on MS-COCO 2017 [34] dataset at 320×320 resolution. Considering the fairness of the comparison and the friendliness of the community, we employ standard MMDetection library [4] for experiments and replace the optimizer with AdamW [41] without tuning other parameters. Data in Tab. 12 come from official EdgeViT [49], and our EMO consistently obtains better results over counterparts, *e.g.*, +2.6↑ AP than CNN-based ResNet-50 and +1.7↑ AP than Transformer-based PVTv2-B0. In addition, we report EMO-5M-based RetinaNet with 178.11 GFLOPs for the follow-up comparison.

C. Scale Up Assessment

We scale up instantiated EMO-10M and EMO-20M models to further evaluate the effect of our approach, and the specific structures are shown in Fig. 13. Tab. 14 shows comparison results with SoTA methods, and our EMO-10M consistently achieves superior accuracy with lower parameters and FLOPs over contemporary counterparts, *i.e.*, **81.0** Top-1. At the larger scale, the advantage of EMO-20M decreases slightly, but it still achieves a very competitive result, *i.e.*, **82.0** Top-1.

Table 12: Object detection results by RetinaNet on MS-COCO.

Backbone	#Params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 [17]	37.7	36.3	55.3	38.6	19.3	40.0	48.8
PVTv1-Tiny [65]	23.0	36.7	56.9	38.9	22.6	38.8	50.0
PVTv2-B0 [66]	13.0	37.2	57.2	39.5	23.1	40.4	49.7
EMO-5M	14.4	38.9	59.8	41.0	23.8	42.2	51.7

Table 13: Core configurations of scaled EMO variants.

Items	EMO-10M	EMO-20M
Depth	[3, 4, 9, 3]	[3, 4, 12, 3]
Emb. Dim.	[64, 96, 224, 384]	[64, 128, 320, 448]
Exp. Ratio	[2.0, 3.0, 4.0, 5.0]	[2.0, 3.0, 4.0, 5.0]

Table 14: Supplementary classification performance on ImageNet-1K dataset. White, orange, and blue backgrounds indicate CNN-based, Transformer-based, and our EMO, respectively. This kind of display continues for all subsequent experiments. Unit: (M).

Model	#Params ↓	FLOPs ↓	Reso.	Top-1	#Pub
PVTv2-B1 [66]	14.0	2120	224^2	78.7	ICCV'21
XCiT-T24 [1]	12.1	2354	224^2	79.4	NeurIPS'21
ViTAE-13M [71]	10.8	3054	224^2	81.0	NeurIPS'21
PoolFormer-S12 [74]	11.9	1823	224^2	77.2	CVPR'22
MPViT-XS [30]	10.5	2971	224^2	80.9	CVPR'22
EMO-10M	10.2	1874	224^2	81.0	
ResNet-50 [17, 68]	25.5	4112	224^2	80.4	CVPR'16
ConvNeXt-T [39]	28.5	4466	224^2	82.1	CVPR'22
DeiT-S [61]	22.0	4608	224^2	79.8	ICML'21
PVTv2-B2 [66]	25.3	4046	224^2	82.0	ICCV'21
Swin-T [38]	28.2	4509	224^2	81.3	ICCV'21
ViTAE-S [71]	24.0	6207	224^2	82.0	NeurIPS'21
MPViT-S [30]	22.8	4800	224^2	83.0	CVPR'22
PoolFormer-S24 [74]	21.3	3413	224^2	80.3	CVPR'22
PoolFormer-S36 [74]	30.8	5003	224^2	81.4	CVPR'22
EMO-20M	20.5	3808	224^2	82.0	

D. Details for Mobile Evaluation

Following official classification project [25], we test our EMO *vs.* SoTA EdgeNeXt [44] on iPhone14 mobile device. The source code to produce *mlmodel* is attached in the supplementary source code.