Enriching Local Patterns with Multi-Token Attention for Broad-Sight Neural Networks (Supplementary Material)

Hankyul Kang
Ajou University
hankyulkang1997@gmail.com

Jongbin Ryu
Ajou University
jongbinryu@ajou.ac.kr

1. Architecture detail

We describe the model configuration of the proposed MAP with details in Tab. 1. There are several hyperparameters in MAP, such as the scale level of merged features *s*, the dimension of the merged feature channel *D*, the number of class tokens per group *N*, and the number of token groups *G*. For simplicity, we fix *D* as 384 and *s* as 3 and set the pair of token and group parameters (N, G) to (2, 4) for the small model and (4, 2) for the large model in Tab.3 of the manuscript. As explained in Sec.4 of the paper, this design choice maximizes the likelihood of capturing local patterns with little computational budget. Based on this design principle, we determine *D*, *s*, *N*, and *G* as the above optimal values.

Table 1. Details of MAP module variants. The input size is determined by the total sum of hidden dimensions of every stage.

Hyper-param	ResNet [6]	ConvNeXt [14]	MaxViT [23]	ResMLP [20]
input size	3904	1344	1344	1536
hidden size (D)	384	384	384	384
MLP size	1536	1536	1536	1536
heads	12	12	12	12
#group (G)	2	4	4	2
#token (N)	4	2	2	4
Params	13M	19M	19M	12M
FLOPs	0.4G	0.4G	0.4G	0.2G

2. Experimental setup

2.1. Image classification

Tab. 2 provides our training hyper-parameters used to train multiple backbone architectures on different datasets. Except for ResNet, we adhere to the original design used to train backbone networks for the ILSVRC-2012 pretraining task. We refer to the A2 configuration in [24] for our hyperparameter setting. We utilize the standard fine-tuning receipt from the [21] for our downstream tasks.

Table 2. Summary of ILSVRC-2012 training hyper-parameters. ViT includes DeiT, PiT, and PVT.

Hyper-param	ResNet [24]	ConvNeXt [14]	ViT [21]	ResMLP [20]
train res.	224	224	224	224
test res.	224	224	224	224
test crop ratio	0.95	0.875	0.95	0.95
epoch	300	300	300	350
batch size	2048	4096	1024	1024
criterion	BCE	CE	CE	CE
optimizer	LAMB	AdamW	AdamW	LAMB
lr	5e-3	4e-3	1e-3	5e-3
lr decay	cosine	cosine	cosine	cosine
weight decay	0.02	0.05	0.05	0.2
warmup epochs	5	20	5	5
h.flip	✓	✓	✓	✓
rand augmentation	7/0.5	9/0.5	9/0.5	9/0.5
cutmix alpha	1.0	1.0	1.0	1.0
mixup alpha	0.1	0.8	0.8	0.8
erasing prob.	0.0	0.25	0.25	0.25
ema	-	✓	-	-

2.2. Downstream task

Object detection and Instance segmentation. Mask R-CNN [17] and Cascade Mask R-CNN [1] are used as object detection models. We use the MMDetection [2] framework for training the detection models on the MS-COCO 2017 dataset [13]. We evaluate our model following ×1 schedule training receipt: AdamW [15] optimizer, 32 batch size, 12 epochs, 0.1 weight decay, 1e-4 learning rate, and a step-wise learning rate scheduler.

Semantic segmentation. Semantic FPN [10] and UperNet [25] are used in our implementation. We use the MMSegmentation [3] framework for training the segmentation model on the ADE20K dataset [28]. We evaluate our model

Table 3. Ablation study on the baseline networks used in the Sec.3 of the manuscript. We train each model with a distinct pooling layer using the same training receipt as shown in Tab. 2.

Model	GAP		CAP	
	Top-1 Acc.(%)	Param. (M)	Top-1 Acc.(%)	Param. (M)
ResNet50	79.8	25.6	80.6	59.1
DeiT-S	80.4	22.0	81.0	52.9

following ×1 schedule training receipt: AdamW [15] optimizer, 32 batch size, 40000 iterations, 1e-4 weight decay, 2e-4 learning rate, and polynomial decay learning rate scheduler.

3. Analysis detail

In this section, we provide the measure of feature variance and dead neurons utilized in Sec. 3 of the manuscript in detail. We compute feature variance as:

$$\begin{aligned} \mathrm{var}_{\mathrm{ch}}(F) &= \frac{1}{HW}\sum_{h}^{H}\sum_{w}^{W}\mathrm{var}(F[:,h,w]), \\ \mathrm{var}_{\mathrm{sp}}(F) &= \frac{1}{C}\sum_{c}^{C}\mathrm{var}(F[c,:,:]), \end{aligned} \tag{1}$$

where $F \in \mathbb{R}^{C\times H\times W}$ denotes the feature map with height H, width W, and channel dimension C. We determine the average number of dead neurons by counting the number of zero activation units after the ReLU [16]. For our empirical analysis, we use two baseline networks (ResNet50, DeiT) with two different pooling layers (GAP and CAP). We train both networks on the ImageNet dataset by replacing the last pooling layer. The results of the two baseline networks are shown in Tab. 3.

4. Further experimental results

4.1. Image classification with various network scales

We apply our MAP to 10 different baseline networks to show its benefits in various network scales. For FasterViT [5], we train it without a sharpness-aware minimization loss for fair comparison with other networks. Tab. 4 indicates that ours works well with various network scales. In particular, our MAP demonstrates significantly improved performance compared to the scale-up version of baseline networks while using less resource overhead.

4.2. Scalability on input resolution

The suggested MAP performs effectively with scaling methods for high-resolution input images, which is crucial for recent visual recognition tasks. As shown in the 'accuracy vs. resolution' plots of Fig. 1, the proposed method delivers more performance gains as the input image's resolution increases. We assume that the reason for these results is that as the input resolution grows, there is more local information, but the current GAP is unable to learn it well.

Table 4. ImageNet Top-1 Acc on Rendition (R), V2, Real, and Val label. We report the network's throughput on an RTX 3090 GPU. We use public checkpoints to evaluate baseline networks. We compare the baseline, its up-scaled network, and our MAP method. †: denotes our reproduced results; otherwise, result of original paper.

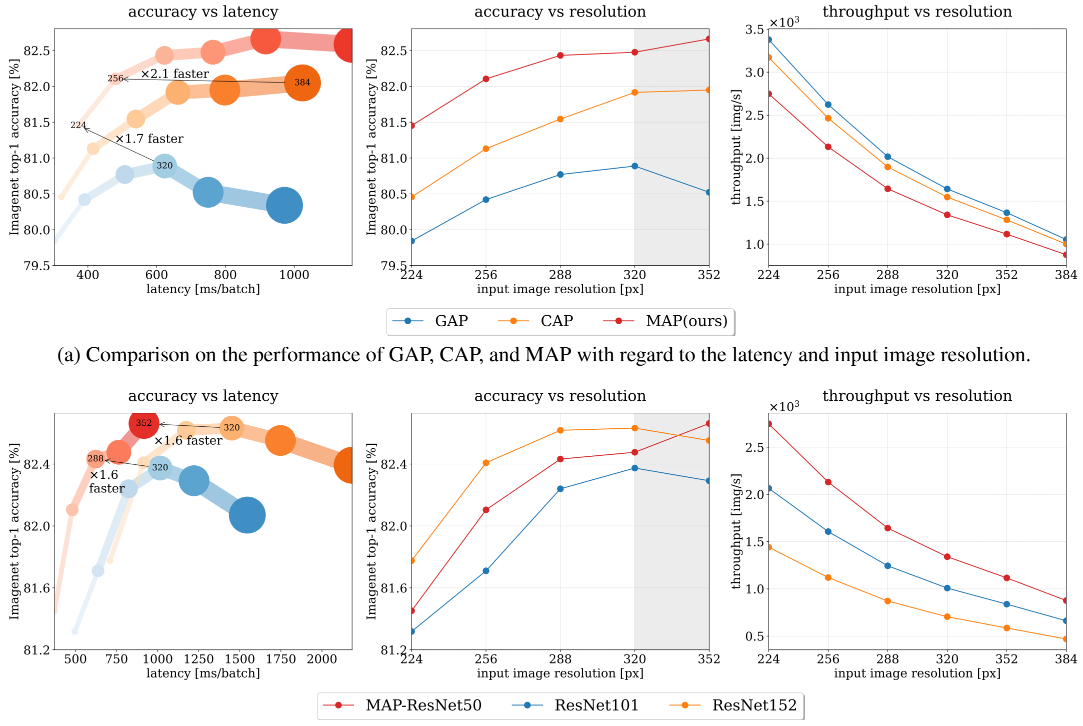
Network	Throughput (img/s)	Param. (M)	FLOPs (G)	ImageNet Top1 Acc. (%)			
				R	V2	Real	Val
MobileNetV1†	4066	4.2	0.6	30.3	58.8	78.8	71.3
MAP (Ours)	**3734**	4.9	0.6	31.9	60.9	80.9	73.4
ResNet50	3334	25.6	4.1	38.2	67.7	85.4	79.8
50 → 152	1446	60.2	11.5	41.5	71.0	86.5	81.8
MAP (Ours)	**2819**	38.0	4.5	45.9	70.2	86.7	81.8
DeiT-S	2611	22.1	4.3	42.3	68.5	85.7	79.8
S → B	1011	86.6	16.9	44.9	71.2	86.8	81.8
MAP (Ours)	**2287**	36.9	4.5	46.6	71.0	87.3	81.8
PiT-S	2580	23.5	2.4	43.9	69.9	86.3	80.9
S → B	935	73.8	10.5	44.1	71.7	86.7	82.0
MAP (Ours)	**2254**	36.2	2.6	47.5	70.8	87.3	81.9
ResMLP-S24	1926	30.0	6.0	40.7	67.9	85.3	79.4
S24 → S36	1310	44.7	8.9	43.0	68.4	85.6	79.7
MAP (Ours)	**1623**	43.3	6.2	44.9	69.7	86.8	81.0
ConvNeXt-T	2040	29.0	4.5	47.2	71.0	87.3	82.1
T → S	1257	50.0	8.7	49.6	72.4	88.1	83.1
MAP (Ours)	**1665**	47.8	4.9	48.7	72.5	88.0	83.3
ConvNeXt-S	1257	50.0	8.7	49.6	72.4	88.1	83.1
S → B	886	89.0	15.4	51.3	73.7	88.3	83.8
MAP (Ours)	**1111**	82.8	9.2	52.0	73.8	88.6	84.1
MaxViT-T	1009	30.9	5.4	48.8	72.9	88.0	83.6
T → S	654	69.0	11.7	50.9	73.9	88.5	84.5
MAP (Ours)	**907**	50.0	5.8	51.2	74.3	88.8	84.3
MaxViT-S	654	69.0	11.7	50.9	73.9	88.5	84.5
S → B	361	120.0	23.4	52.2	74.3	88.6	85.0
MAP (Ours)	**613**	100.9	11.8	54.1	74.8	88.9	85.0
FasterViT-3†	1087	159.5	18.5	45.3	72.4	87.2	83.1
MAP (Ours)	**970**	187.0	18.8	49.3	74.0	88.1	84.2

4.3. Downstream tweak

We perform an ablation study on the proposed tweak for the downstream task in Tab. 5. Without the proposed tweaks, adding the final pooling layer results in a decrease

Table 5. Ablation study on the dense prediction task. The performances on downstream tasks are greatly increased when our architectural tweak for dense prediction is used.

Pooling	Tweak	CLS Acc.(%)	DET mAP(%)	SEG mIOU(%)
GAP	-	77.0	38.1	37.1
CAP	-	78.5	37.7	36.6
CAP	✓	78.5	39.2	37.7
MAP	✓	80.7	40.1	39.5



(a) Comparison on the performance of GAP, CAP, and MAP with regard to the latency and input image resolution.

(b) Comparison on the performance of MAP-ResNet50, ResNet101, and ResNet152 with regard to the latency and input image resolution.

Figure 1. Experimental study of extensive comparison of the proposed MAP with other pooling methods and networks. In the left figures, larger points signify that high input resolution is employed. (a) We confirm that, compared to GAP and CAP, MAP achieves much higher performance while using fewer resources. (b) MAP-ResNet50 shows better performance compared to vanilla ResNet with deeper layers, and in particular, as the resolution increases, MAP performs better.

Table 6. Model configurations used for scale-up networks of ResNet. These series of scale-up networks are used to compare the proposed MAP with SOTA networks in Tab. 7. MAP-R50 indicates ResNetD-50 with our MAP. #L and #C denote the number of layers and channels in each stage.

Stage	MAP-ResNet50		MAP-ResNet75		MAP-ResNet101	
	#L	#C	#L	#C	#L	#C
1	3	64	4	64	4	84
2	4	128	5	128	5	168
3	6	256	13	320	21	336
4	3	512	3	640	3	672

of 0.3% and 0.5% for segmentation and detection tasks, despite an increase in classification accuracy. However, with the proposed tweak, CAP improves the performance of baseline network detection and segmentation networks by 1.1% and 0.6%, respectively. Moreover, utilizing MAP boosts the accuracy of detection and segmentation networks by an additional 0.9% and 1.8%, respectively. As a result, the proposed tweak with MAP improves the performance of detection and segmentation as well as the classification task.

4.4. Distillation

Experiment setup. The configuration of our ResNet-based models in Tab. 7 is illustrated in Tab. 6. We augment our ResNet-based model by channel dimensions concurrently with the number of layers and tweaks from ResNet-D [7], a strategy that is widely used in current models [18, 27]. Other than these model augmentation, no additional architecture techniques (such as SE-module [9]) are used in our model.

CNN Distillation from ViT. Most previous studies [4, 21, 22] distill ViT from the CNN model using the distillation token. Initially, network learning utilizing tokens was developed in ViT; therefore, the distillation direction (CNN → ViT) has been a prevalent strategy. However, the proposed MAP uses class tokens on the last pooling layer, so we apply the distillation method in which ViT teaches CNN, as shown in Tab. 7. It demonstrates that the proposed MAP has the potential to distill the CNN from knowledge of the ViT. It is worth noting that the proposed distillation approach has the benefit that it can be applied to various network architectures. Tab. 7 shows the distillation with our MAP achieves

Table 7. Experimental study on knowledge distillation and scale-up architectures with the proposed MAP. Υ denotes a network trained by knowledge distillation from VOLO-D1 [26].

Model	ImageNet Top-1 acc. (%)				Throughput (img/s)		FLOPs (G)	
	224	320	224Υ	320Υ	224	320	224	320
MAP-ResNet50	81.8	82.9	82.5	83.7	2557.1	1233.4	5.6	11.3
MAP-ResNet75	82.5	83.5	83.4	84.3	1587.0	771.5	10.3	20.8
MAP-ResNet101	82.9	83.9	83.7	84.4	935.3	454.2	15.8	32.2

Table 8. Experimental result on the transfer learning. We fine-tune our MAP methods pre-trained on ImageNet to small datasets.

Method	IN1K (Acc. %)	C10 (Acc. %)	C100 (Acc. %)	Cars (Acc. %)	Throughput (img/s)
Convolution: ResNet50 [6]					
GAP [6]	79.8	98.2	88.7	87.8	3401
CAP [22]	80.6	98.6	89.6	91.3	3176
MAP	**81.8**	**98.7**	**90.3**	**91.5**	2819
Transformer: PiT-S [8]					
GAP [6]	79.8	98.8	90.1	90.4	2580
CAP [22]	81.2	99.0	91.0	90.2	2494
MAP	**81.9**	**99.0**	**91.3**	**90.5**	2254

competitive performance regarding accuracy and resource usage in most cases.

4.5. Transfer Learning

We finetune the pre-trained networks (*i.e*. ResNet50 [6], and PiT-S [8]) to small datasets such as CIFAR10/100 [12], and Stanford-Cars [11] to examine their ability to generalize on such datasets. In Tab. 8, the proposed MAP improves the accuracy for all small datasets. For instance, ResNet50 with MAP outperforms the baseline by about 0.5/1.5% in CIFAR10/100. Similarly, we observe a consistent performance improvement for PiT in all datasets. This finding verifies that replacing GAP with MAP improves generalization.

Table 9. Comparison of MAP vs GA. We report top-1 accuracy of 300 epochs on ImageNet-1K. †: GA are the original results [19].

Network	Pooling	Throughput (img/sec)	FLOPs (G)	Δ (↓)	Top-1 Acc. (%)	Δ (↑)
ResNet50	GA† [19]	2145	5.2	-	82.5	-
	MAP	2127	5.4	+0.2	82.9	**+0.4**
ViT-S	GA† [19]	2289	4.2	-	80.9	-
	MAP	2287	4.5	+0.3	81.8	**+0.9**

4.6. Experimental Comparison with GA

We further compare ours with the GA method [19], which is the baseline for our approach. Tab. 9 compares our MAP to the original results of the GA method [19]. It shows that our MAP works well compared to the GA method using manageable extra resources.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1

[2] 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:1906.07155*, 2019. 1

[3] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[4] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, 2021. 3

[5] Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. *ICLR*, 2024. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 4

[7] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, 2019. 3

[8] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, 2021. 4

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 3

[10] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1

[11] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 4

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009. 4

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[14] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 1

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 1, 2

[16] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 2

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 1

[18] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *WACV*, 2021. 3

[19] Jongbin Ryu, Dongyoon Han, and Jongwoo Lim. Gramian attention heads are strong yet efficient vision learners. In *ICCV*, 2023. 4

[20] 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. *PAMI*, 2022. 1

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 3

[22] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021. 3, 4

[23] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 1

[24] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv:2110.00476*, 2021. 1

[25] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 1

[26] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *PAMI*, 2023. 4

[27] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *CVPR*, 2022. 3

[28] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1