MambaOut: Do We Really Need Mamba for Vision?*

Supplementary Material

1. More details of MambaOut models

In this section, we show more details of MambaOut models. The overall framework of MambaOut is illustrated in Figure 1 and our PyTorch implementation of Gated CNN block [2] is shown in Algorithm 1. Moreover, the MambaOut model configurations are shown in Table 1 and the hyper-parameters to train MambaOut on ImageNet are shown in Table 2.

2. Ablation study

The pivotal hyper-parameter in the MambaOut architecture is the kernel size of the depthwise convolution within the Gated CNN blocks [2]. To assess its impact, we conduct ablation study of the kernel size and the results are shown in Table 3. We observe that increasing the kernel size from 3×3 to 7×7 results in performance gains on both ImageNet and ADE20K datasets. However, further increasing the kernel size to 9×9 leads to a decline in performance on ADE20K. We conjecture that the performance drop may be caused by the optimization difficulty of large-kernel convolutions [3]. Training large-kernel convolutions [3, 13] is another line of research orthogonal to this paper. In this work, we aim to verify the Mamba concept for vision tasks instead of building state-of-the-art convolutional neural networks.

3. Throughput comparison

Besides MACs (FLOPs), throughput is another important metric, particularly in practical scenarios. The results in Table 4 indicate that MambaOut achieves at least $5\times$ higher throughput than VMamba [14].

Model	Params (M)	MACs (G)	Top-1 (%)	Throughput (img/s) Train	Infer
VMamba-T	22	5.6	82.2	194	582
MambaOut-Tiny	27	4.5	82.7	1055 (**5.4**×)	3370 (**5.8**×)
VMamba-S	44	11.2	83.5	117	326
MambaOut-Small	48	9.0	84.1	665 (**5.7**×)	2150 (**6.7**×)
VMamba-B	75	18.0	83.7	88	240
MambaOut-Base	85	15.8	84.2	441 (**5.0**×)	1375 (**5.7**×)

Table 4. **Throughput comparision between Vmamba [14] and MambaOut.** The throughputs are measured on an RTX 4090 GPU using PyTorch 2.3, CUDA 12.1, and float16 precision. Top-1 denotes the accuracy on ImageNet.

4. More MambaOut variants

In this section, we construct various MambaOut variants to facilitate a more comprehensive comparison with visual Mamba models.

Algorithm 1 PyTorch code of Gated CNN block

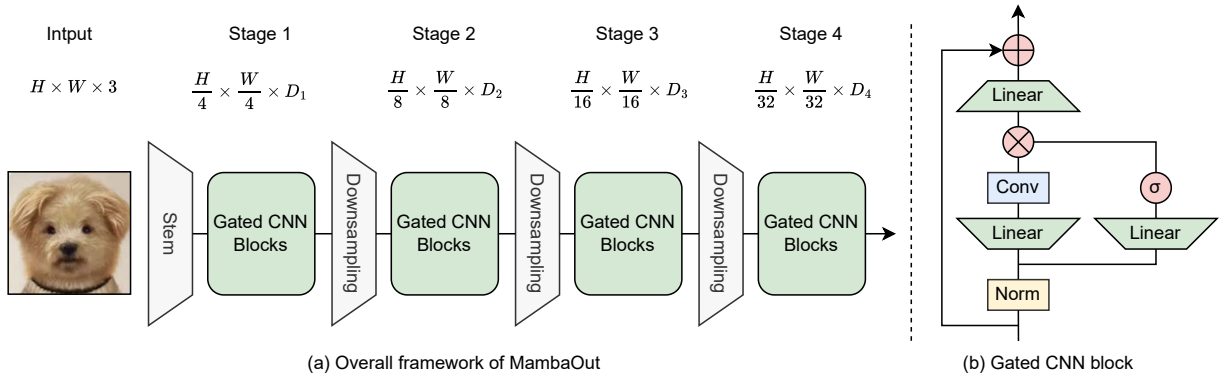
```
import torch
import torch.nn as nn

class GatedCNNBlock(nn.Module):
    """ Our implementation of Gated CNN Block: https
        ://arxiv.org/pdf/1612.08083
    Args:
        dim (int): Number of input and output channels
            (embedding dimensions).
        expansion_ratio (float): Gated MLP expansion
            ratio. Default: 8/3.
        kernel_size (int): Kernel size of the token
            mixer of depthwise convolution. Default:
            7.
        conv_ratio (float): Ratio of convolution
            channels to embedding dim. Default: 1.0.
            Conduct convolution on partial channels can
            improve paraitcal efficiency.
            The idea of partial channels is from
            ShuffleNet V2 (https://arxiv.org/abs
            /1807.11164) and
            also used by InceptionNeXt (https://arxiv.
            org/abs/2303.16900) and FasterNet (
            https://arxiv.org/abs/2303.03667).
        norm_layer: Normalization layer. Default: nn.
            LayerNorm.
        act_layer: Activation layer. Default: nn.GELU.

    """
    def __init__(self, dim, expansion_ratio=8/3,
        kernel_size=7, conv_ratio=1.0,
            norm_layer=partial(nn.LayerNorm,eps=1e
                -6),
            act_layer=nn.GELU):
        super().__init__()
        self.norm = norm_layer(dim)
        hidden = int(expansion_ratio * dim)
        self.fc1 = nn.Linear(dim, hidden * 2)
        self.act = act_layer()
        conv_channels = int(conv_ratio * dim)
        self.split_indices = (hidden, hidden -
            conv_channels, conv_channels)
        self.conv = nn.Conv2d(conv_channels,
            conv_channels, kernel_size=kernel_size,
            padding=kernel_size//2, groups=
            conv_channels)
        self.fc2 = nn.Linear(hidden, dim)

    def forward(self, x):
        shortcut = x # [B, H, W, C] = x.shape
        x = self.norm(x)
        g, i, c = torch.split(self.fc1(x), self.
            split_indices, dim=-1)
        c = c.permute(0, 3, 1, 2) # [B, H, W, C] -> [B
            , C, H, W]
        c = self.conv(c)
        c = c.permute(0, 2, 3, 1) # [B, C, H, W] -> [B
            , H, W, C]
        x = self.fc2(self.act(g) * torch.cat((i, c),
            dim=-1))
        return x + shortcut
```

Isotropic MambaOut. We conduct experiments of isotropic architecture for MambaOut, following ViT [4] and Vision Mamba (Vim) [23]. The results are presented in Table 5. LocalViM-S [9] is not included in the table because it additionally incorporates a spatial and channel attention



(a) Overall framework of MambaOut (b) Gated CNN block

Figure 1. **(a) The overall framework of MambaOut for visual recognition.** Similar to ResNet [8], MambaOut adopts hierarchical architecture with four stages. D_i represents the channel dimensions at the i-th stage. **(b) The architecture of Gated CNN block.** The difference between the Gated CNN block [2] and the Mamba block [6] lies in the absence of the SSM (state space model) in the Gated CNN block.

Size	Femto	Tiny	Small	Base
Stem	3×3 conv with stride 2; Norm; GELU; 3×3 conv with stride 2, Norm			
Downsampling layers	3×3 conv with stride 2			
Token mixer	7×7 depthwise conv			
Expansion ratio	8/3			
Classifier head	Global average pooling, Norm, MLP			
# Blocks	(3, 3, 9, 3)	(3, 3, 9, 3)	(3, 4, 27, 3)	(3, 4, 27, 3)
# Channel	(48, 96, 192, 288)	(96, 192, 384, 576)	(96, 192, 384, 576)	(128, 256, 512, 768)
Parameters (M)	7.3	26.5	48.5	84.8
MACs (G)	1.2	4.5	9.0	15.8

Table 1. Configurations of MambaOut models. The contents in the tuples represent the configurations in the four stages of the models.

	MambaOut			
	Femto	Tiny	Small	Base
Input resolution	224^2			
Epochs	300			
Batch size	4096			
Optimizer	AdamW			
Adam ϵ	1e-8			
Adam (β_1, β_2)	(0.9, 0.999)			
Learning rate	4e-3			
Learning rate decay	Cosine			
Gradient clipping	None			
Warmup epochs	20			
Weight decay	0.05			
Rand Augment	9/0.5			
Repeated Augmentation	off			
Cutmix	1.0			
Mixup	0.8			
Cutmix-Mixup switch prob	0.5			
Random erasing prob	0.25			
Label smoothing	0.1			
Peak stochastic depth rate	0.025	0.2	0.4	0.6
Random erasing prob	0.25			
EMA decay rate	None			

Table 2. Hyper-parameters of MambaOut on ImageNet image classification.

module. Note that a Transformer block has two residual sub-blocks. We can see that MambaOut-S-*iso.* matches the performance of Vim-S, which supports Hypothesis 1.

MambaOut-T15 with 15 residual blocks. The number of residual blocks of MambaOut-Tiny is 18, following ConvNeXt-T [16]. The expansion ratio of the MambaOut block is set as 8/3 to match the parameters and MACs of the ConvNeXt block with MLP ratio of 4. However, VMamba-T [14] and LocalVMamba-T [9] use a different number of blocks, *i.e.*, 15. To ease more direct comparison, we construct MambaOut-T15 with the same number of blocks and set its expansion ratio to 4 to match the MACs. The results are shown in Table 6. We can see that MambaOut-T15 outperforms VMamba-T and LocalVmamba-T with the same number of blocks.

MambaOut-Attn-T. Mamba block consists of token mixers of convolution and SSM. To demonstrate the effectiveness of attention over Mamba for short sequences, we build two models, VMamba-Mixer-stage and MambaOut-Attn, as presented in Table 7. Both models share the same token mixing of convolution in stages 1 and 2 because these stages focus on local modeling. However, in stages 3 and 4,

Backbone	Token Mixing Type	ImageNet			UperNet on ADE20K			
		Param (M)	MAC (G)	Acc (%)	Param (M)	MAC (G)	mIoU (SS)	mIoU (MS)
VMamba-T [14]	Conv + SSM	22	5.6	82.2	55	964	47.3	48.3
LocalVMamba-T [9]	Conv + SSM	26	5.7	**82.7**	57	970	**47.9**	**49.1**
MambaOut-Tiny/k3	Conv	26	4.4	82.2	54	936	45.3	46.7
MambaOut-Tiny/k5	Conv	26	4.4	82.6	54	937	47.1	48.2
MambaOut-Tiny/k7 (default)	Conv	27	4.5	82.7	54	938	**47.4**	**48.6**
MambaOut-Tiny/k9	Conv	27	4.5	**82.9**	54	940	46.9	48.1
MambaOut-Tiny/k11	Conv	27	4.6	82.8	54	941	46.9	47.9

Table 3. Ablation study for Mambout of the kernel size of the depthwise convolution within the Gated CNN blocks [2]. The notation 'kn' denotes the kernel size of n $\times$ n.

Model	Residual blocks	Params (M)	MACs (G)	Top-1 (%)
ViT-S [4, 20]	24	22	4.6	79.8
Vim-S [23]	24	26	5.1	80.5
MambaOut-S (*iso.*)	18	24	4.3	80.5

Table 5. **Comparison among ViT [4], Vision Mamba (Vim) [23] and isotropic MambaOut.** MambaOut-S (*iso.*) is configured with embedding dimension of 384 and expansion ratio of 8/3. Top-1 denotes the accuracy on ImageNet.

Model	Residual blocks	Params (M)	MACs (G)	Top-1 (%)
Swin-T [15]	(4, 4, 12, 4)	29	4.5	81.3
ConvNeXt-T [16]	(3, 3, 9, 3)	29	4.5	82.1
NAT-T [7]	(6, 8, 36, 10)	28	4.3	83.2
VMamba-T [14]	(2, 2, 9, 2)	22	5.6	82.2
LocalVmamba-T [9]	(2, 2, 9, 2)	26	5.7	82.7
MambaOut-Tiny	(3, 3, 9, 3)	27	4.5	82.7
MambaOut-T15	(2, 2, 9, 2)	32	5.5	**82.9**

Table 6. **Comparison between MambaOut-T15 and other models.** Top-1 denotes the accuracy on ImageNet. MambaOut-T15 outperforms VMamba-T and LocalVmamba-T with the same number of residual blocks at each stage.

VMamba-Mixed-Stage utilizes Mamba blocks with token mixers of convolution and SSM, while MambaOut-Attn replaces SSM with attention in these blocks, resulting in token mixers of convolution and attention. MambaOut-Attn significantly outperforms VMamba-Mixer-stage, providing strong support for Hypothesis 1.

Pure SSM using neither gate nor convolution. To directly compare SSM, convolution and Attention, we remove the gate and convolution in Mamba block to obtain pure SSM block, and construct 12-layer isotropic model like ViT. The models are trained for 200 epochs on CIFAR-10 and CIFAR-100 [11] (image size 32^2 and patch size 4^2). As shown in Table 8, the pure SSM model fails to match the performance of pure Conv or Attention models on these image classification tasks.

Model	Dim	Params	MACs	CIFAR-10	CIFAR-100
Conv	64	0.7M	43M	79.8 ± 0.4	52.1 ± 0.2
Attention	64	0.7M	45M	77.3 ± 0.5	50.8 ± 0.6
SSM	64	0.7M	56M	76.1 ± 0.6	47.0 ± 0.6
Conv	192	6M	357M	88.9 ± 0.1	68.5 ± 0.3
Attention	192	6M	361M	86.9 ± 0.3	66.8 ± 0.2
SSM	192	6M	399M	86.3 ± 0.1	63.1 ± 0.6

Table 8. **Performance comparison of pure SSM model (without convolution and gate) and pure Convolution/Attention model on CIFAR-10/100 [11].**

5. More other vision tasks

Fine-grained image classification. In the paper, we limit the scope to three commonly used visual tasks, ensuring the rigor of our discussion. In this section, we further evaluate MambaOut for fine-grained image classification. Using ImageNet pretrained checkpoints, we train and evaluate VMamba-T and MambaOut-Tiny on fine-grained datasets (resolution 224 $\times$ 224). From the results shown in Table 9, we see that MambaOut-Tiny achieves an average accuracy of 92.8, outperforming VMamba-T by 0.9 on these fine-grained datasets. Readers interested in further exploration can evaluate MambaOut on other specific visual tasks.

Video understanding. Video understanding is another important long-sequence visual task. As shown in Table 10, MambaOut-Femto does not match the performance of VideoMamba-Ti [12] on Something-Something V2 [5], which is consistent with Hypothesis 2.

Model	Pretrained	Params	MACs	SSV2
VideoMamba-Ti	ImageNet	7M	54G	65.1
MambaOut-Femto	ImageNet	7M	58G	63.6

Table 10. **Performance of video understanding.**

6. Fully-visible and causal modes on iLLaMA

Figure 3(b) in the paper illustrates that ViT [4] performs better in the fully-visible mode compared to the causal mode. We further conduct experiments on image LLaMA (iLLaMA) [22], a variant of the Vision Transformer that adopts

Model	Token mixers in stages 1 & 2	Token mixers in stages 3 & 4	Residual blocks	Params (M)	MACs (G)	Top-1 (%)
VMamba-Mixed-Stage-T	Conv	Conv + SSM	[3, 3, 9, 3]	28	5.2	82.1
MambaOut-Attn-T	Conv	Conv + Attention	[3, 3, 9, 3]	26	4.7	**83.3**

Table 7. **Comparison between VMamba-Mixed-Stage and MambaOut-Attn.** Top-1 denotes the accuracy on ImageNet.

Model	CUB-200-2011 [21]	Stanford Cars [10]	Stanford Dogs [1]	Oxford Pets [18]	FGVC Aircraft [17]	Average
VMamba-T [14]	87.9	93.2	95.5	94.8	88.0	91.9
MambaOut-Tiny	88.0	94.3	95.5	95.4	90.8	**92.8**

Table 9. **Accuracy of VMmaba-T [14] and MambaOut-Tiny on fine-grained image classification datasets.** VMamba-T and MambaOut-T are initialized with ImageNet pretrained checkpoints and then trained and evaluated on these datasets.

the LLaMA architecture and operates in causal mode by default. We utilize the official iLLaMA codebase but remove the causal mask to create a fully-visible mode version. The results in Table 11 demonstrate that iLLaMA with fully-visible mode significantly outperforms its causal mode counterpart, further supporting the conclusion that causal token mixing is not necessary for visual understanding tasks.

Mode	ViT-T	ViT-S	iLLaMA-T
Casual	70.6	78.9	75.0
Fully-visible	**72.2**	**79.8**	**76.2**

Table 11. **ImageNet accuracy of ViT [4] and iLLaMA [22] in casual or fully-visible modes.**

7. Visualization

We use Grad-CAM [19] to visualize the activation maps of MambaOut-T, as shown in Figure 2. The visualization shows that MambaOut-T accurately locates the key parts of the images, demonstrating the model's effectiveness. Additionally, we observe that the activation areas are relatively concentrated, which is characteristic of pure CNNs.

References

[1] E Dataset. Novel datasets for fine-grained image categorization. In *First Workshop on Fine Grained Visual Categorization, CVPR. Citeseer. Citeseer. Citeseer*, page 2. Citeseer, 2011. 4

[2] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017. 1, 2, 3

[3] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022. 1

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

[5] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 3

[6] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2

[7] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[9] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024. 1, 2, 3

[10] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 4

[11] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 3

[12] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pages 237–255. Springer, 2024. 3

[13] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 1

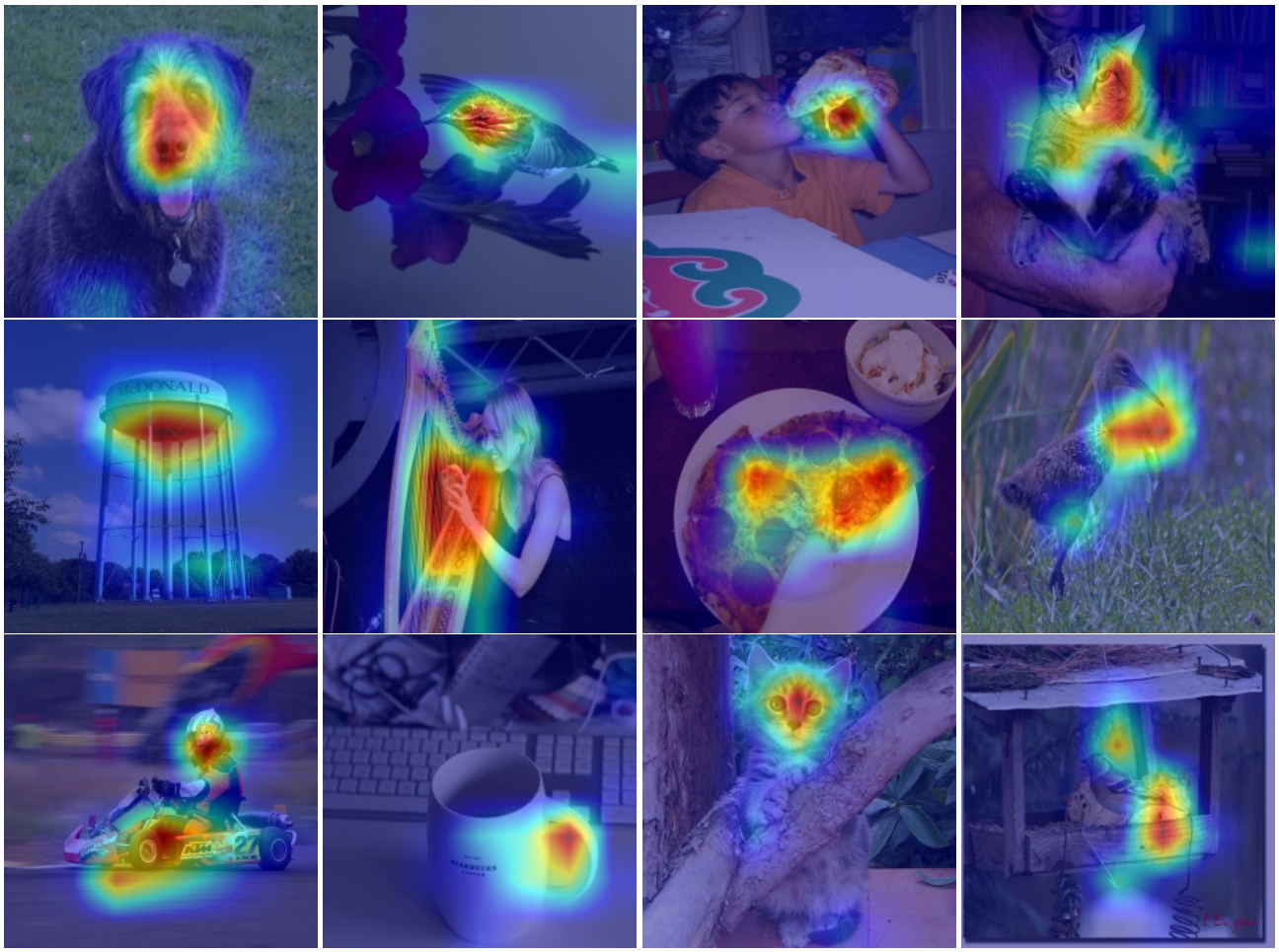


Figure 2. Grad-CAM [19] activation maps of MambaOut-T trained on ImageNet. The visualized images are from ImageNet val set. We can see that MambaOut-T can accurately locate key parts.

[14] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166v1*, 2024. 1, 2, 3, 4

[15] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[16] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 2, 3

[17] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 4

[18] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 4

[19] 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 *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4, 5

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 3

[21] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 4

[22] Jiahao Wang, Wenqi Shao, Mengzhao Chen, Chengyue Wu, Yong Liu, Taiqiang Wu, Kaipeng Zhang, Songyang Zhang, Kai Chen, and Ping Luo. Adapting llama decoder to vision transformer. *arXiv preprint arXiv:2404.06773*, 2024. 3, 4

[23] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62429–62442. PMLR, 2024. 1, 3