Supplemental material of CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows

Xiaoyi Dong[1], Jianmin Bao[2], Dongdong Chen[3], Weiming Zhang[1],
Nenghai Yu[1], Lu Yuan[3], Dong Chen[2], Baining Guo[2]
[1]University of Science and Technology of China
[2]Microsoft Research Asia [3]Microsoft Cloud + AI
{dlight@mail., zhangwm@, ynh@}.ustc.edu.cn cddlyf@gmail.com
{jianbao, luyuan, doch, bainguo }@microsoft.com

Experiment Details

In this section, we provide more detailed experimental settings about ImageNet and downstream tasks.

ImageNet-1K Classification. For a fair comparison, we follow the training strategy in DeiT [22]. Specifically, all our models are trained for 300 epochs with the input size of 224×224. We use the AdamW optimizer with weight decay of 0.05 for CSWin-T/S and 0.1 for CSWin-B. The default batch size and initial learning rate are set to 2048 and $2e-3$ respectively, and the cosine learning rate scheduler with 20 epochs linear warm-up is used. We adopt most of the augmentation in [22], including RandAugment [8] (rand-m9-mstd0.5-inc1) , Mixup [29] ($prob = 0.8$), CutMix [28] ($prob = 1.0$), Random Erasing [31] ($prob = 0.25$) and Exponential Moving Average [19] ($ema\text{-}decay = 0.99984$), increasing stochastic depth [14] ($prob = 0.2, 0.4, 0.5$ for CSWin-T, CSWin-S, and CSWin-B respectively).

When fine-tuning with 384×384 input, we follow the setting in [17] that fine-tune the models for 30 epochs with the weight decay of $1e$-8, learning rate of $5e$-6, batch size of 256. We notice that a large ratio of stochastic depth is beneficial for fine-tuning and keeping it the same as the training stage.

COCO Object Detection and Instance Segmentation. We use two classical object detection frameworks: Mask R-CNN [12] and Cascade Mask R-CNN [1] based on the implementation from mmdetection [3]. For Mask R-CNN, we train it with ImageNet-1K pretrained model with two settings: $1\times$ schedule and $3\times$+MS schedule. For $1\times$ schedule, we train the model with single-scale input (image is resized to the shorter side of 800 pixels, while the longer side does not exceed 1333 pixels) for 12 epochs. We use AdamW [18] optimizer with a learning rate of 0.0001, weight decay of 0.05 and batch size of 16. The learning rate declines at the 8 and 11 epoch with decay rate 0.1. The stochastic depth is also same as the ImageNet-1K setting that 0.1, 0.3, 0.5 for CSWin-T, CSWin-S, and CSWin-B respectively. For $3\times$+MS schedule, we train the model with multi-scale input (image is resized to the shorter side between 480 and 800 while the longer side is no longer than 1333) for 36 epochs. The other settings are same as the $1\times$ except we decay the learning rate at epoch 27 and 33. When it comes to Cascade Mask R-CNN, we use the same $3\times$+MS schedule as Mask R-CNN.

ADE20K Semantic segmentation. Here we consider two semantic segmentation frameworks: UperNet [25] and Semantic FPN [15] based on the implementation from mmsegmentaion [7]. For UperNet, we follow the setting in [17] and use AdamW [18] optimizer with initial learning rate $6e^{-5}$, weight decay of 0.01 and batch size of 16 (8 GPUs with 2 images per GPU) for 160K iterations. The learning rate warmups with 1500 iterations at the beginning and decays with a linear decay strategy. We use the default augmentation setting in mmsegmentation including random horizontal flipping, random re-scaling (ratio range [0.5, 2.0]) and random photo-metric distortion. All the models are trained with input size 512×512. The stochastic depth is set to 0.2, 0.4, 0.6 for CSWin-T, CSWin-S, and CSWin-B respectively. When it comes to testing, we report both single-scale test result and multi-scale test ([0.5, 0.75, 1.0, 1.25, 1.5, 1.75]$\times$ of that in training).

For Semantic FPN, we follow the setting in [23]. We use AdamW [18] optimizer with initial learning rate $1e^{-4}$, weight decay of $1e^{-4}$ and batch size of 16 (4 GPUs with 4 images per GPU) for 80K iterations.

Ablation study details. In the ablation study part, we evaluate each component with previous methods. For the "Parallel Multi-Head Grouping" part, we use CSWin-Tiny as backbone, including LePE, conv position embedding. To reduce the influence of the last stage, we use full attention in the last stage for all settings, *i.e.* we only apply different attention mechanisms in the first three stages (1+2+21 blocks) and

full attention in the last stage(1 block).

Similarly, for the "Attention Mechanism Comparison" part, as some of the methods need even number of blocks in each stage, we use Swin-Tiny as backbone (non-overlapped token embedding, Relative Position Embedding) and change the attention mechanism for the first three stage (2+2+6 blocks) and full attention in the last stage(2 block).

More Experimetns

With the limitation of pages, we only compare with a few classical methods in our paper, here we make a comprehensive comparison with more current methods on ImageNet-1K. We find that our CSWin performs best in concurrent works.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1

[2] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification, 2021. 3

[3] 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. 1

[4] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer, 2021. 3

[5] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021. 3

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

[7] MMSegmentation Contributors. Mmsegmentation, an open source semantic segmentation toolbox. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[8] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. 1

[9] 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. 3

[10] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021. 3

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

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[13] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers, 2021. 3

[14] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 1

[15] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 1

[16] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers, 2021. 3

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1, 3

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 1

[19] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 1

[20] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces, 2020. 3

[21] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 3

[22] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1, 3

[23] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 1, 3

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

[25] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 1

ImageNet-1K 224^2 trained models			
Method	#Param.	FLOPs	Top-1
Reg-4G [20]	21M	4.0G	80.0
Eff-B4* [21]	19M	4.2G	**82.9**
DeiT-S [22]	22M	4.6G	79.8
PVT-S [23]	25M	3.8G	79.8
T2T-14 [27]	22M	5.2G	81.5
ViL-S [30]	25M	4.9G	82.0
TNT-S [11]	24M	5.2G	81.3
CViT-15 [2]	27M	5.6G	81.0
Visf-S [4]	40M	4.9G	82.3
LViT-S [16]	22M	4.6G	80.8
CoaTL-S [26]	20M	4.0G	81.9
CPVT-S [6]	23M	4.6G	81.5
Swin-T [17]	29M	4.5G	81.3
CvT-13 [24]	20M	4.5G	81.6
CSWin-T	23M	4.3G	82.7
ImageNet-1K 384^2 finetuned models			
CvT-13 [24]	20M	16.3G	83.0
T2T-14 [27]	22M	17.1G	83.3
CViT$_c$-15 [2]	28M	21.4G	83.5
CSWin-T	23M	14.0G	**84.3**

(a) Tiny Model

ImageNet-1K 224^2 trained models			
Method	#Param.	FLOPs	Top-1
Reg-8G [20]	39M	8.0G	81.7
Eff-B5* [21]	30M	9.9G	**83.6**
PVT-M [23]	44M	6.7G	81.2
PVT-L [23]	61M	9.8G	81.7
T2T-19 [27]	39M	8.9G	81.9
T2T$_t$-19 [27]	39M	9.8G	82.2
ViL-M [30]	40M	8.7G	83.3
MViT-B [10]	37M	7.8G	83.0
CViT-18 [2]	43M	9.0G	82.5
CViT$_c$-18 [2]	44M	9.5G	82.8
Twins-B [5]	56M	8.3G	83.2
Swin-S [17]	50M	8.7G	83.0
CvT-21 [24]	32M	7.1G	82.5
CSWin-S	35M	6.9G	**83.6**
ImageNet-1K 384^2 finetuned models			
CvT-21 [24]	32M	24.9G	83.3
CViT$_c$-18 [2]	45M	32.4G	83.9
CSWin-S	35M	22.0G	**85.0**

(b) Small Model

ImageNet-1K 224^2 trained models			
Method	#Param.	FLOPs	Top-1
Reg-16G [20]	84M	16.0G	82.9
Eff-B6* [21]	43M	19.0G	84.0
DeiT-B [22]	87M	17.5G	81.8
PiT-B [13]	74M	12.5G	82.0
T2T-24 [27]	64M	14.1G	82.3
T2T$_t$-24 [27]	64M	15.0G	82.6
CPVT-B [6]	88M	17.6G	82.3
TNT-B [11]	66M	14.1G	82.8
ViL-B [30]	56M	13.4G	83.2
Twins-L [5]	99M	14.8G	83.7
Swin-B [17]	88M	15.4G	83.3
CSWin-B	78M	15.0G	**84.2**
ImageNet-1K 384^2 finetuned models			
ViT-B/16 [9]	86M	49.3G	77.9
DeiT-B [22]	86M	55.4G	83.1
Swin-B [17]	88M	47.0G	84.2
CSWin-B	78M	47.0G	**85.4**

(c) Base Model

Table 1. Comparison of different models on ImageNet-1K classification. * means the EfficientNet are trained with other input sizes. Here the models are grouped based on the computation complexity.

[26] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021. 3

[27] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 3

[28] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. 1

[29] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. 1

[30] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. *arXiv preprint arXiv:2103.15358*, 2021. 3

[31] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation, 2017. 1