Convolutional Embedding Makes Hierarchical Vision Transformer Stronger
–Supplemental Material–

Cong Wang[1,2], Hongmin Xu[1], Xiong Zhang[4], Li Wang[2], Zhitong Zheng[1], and Haifeng Liu[1,3]

[1] Data & AI Engineering System, OPPO, Beijing, China
{wangcong3575, xhmjimmy}@gmail.com, {liam,blade}@oppo.com
[2] Beijing Key Lab of Urban Intelligent Traffic Control Technology, North China University of Technology, Beijing, China
li.wang@ncut.edu.cn
[3] University of Science and Technology of China, Hefei, China
[4] Neolix Autonomous Vehicle, Beijing, China
zhangxiong@neolix.cn

1 Detailed Architectures of Fused-MBConv and MBConv

The structure of Fused-MBConv[6] and MBConv[5] are shown in Fig 1. The MBConv block introduces depth-wise convolution with fewer parameters and FLOPs than regular convolutions. The Fused-MBConv blocks replaced the 3×3 depth-wise convolution and the 1×1 convolution with a regular 3×3 convolution to better utilize modern accelerators. EfficientNetV2 [7] adopts a neural architecture search method to find the right combination of these two building blocks, MBConv and Fused-MBConv.

2 Detailed Structure of LEWin

The standard Multi-head Self-attention (MSA) module in a Transformer block's computation complexity is quadratic to the number of tokens. On the other hand, the linear projection for self-attention computation may lead the Transformer module lack of some certain desirable properties, for example, the local structure is significant for 2D image feature modeling because of spatially neighboring pixels are usually highly correlated. To calculate Multi-head Self-Attention (MSA) efficiently and introduce convolutional inductive bias at the same time, we present Locally Enhanced Window Self-Attention (LEWin). As shown in Fig 2, LEWin is achieved by performing self-attention in local windows with a convolutional projection.

3 Detailed Architectures of CETNet Variants

To compare with other baselines under similar model size and computation complexity, we consider three different variants of CETNet. The detailed architecture configurations are shown in Table 1. All model variants have four stages, and in all these variants, the head number of the four stages is set as 4, 8, 16, 32, respectively, and the expansion ratio of MLP is set to 4.

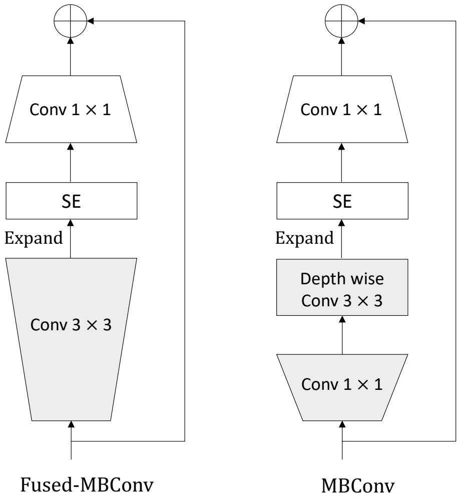

Fig. 1. Architectures of Fused-MBConv and MBConv

Table 1. Detailed configurations of different variants of our model. Channels is the channel number of the hidden layers in the first stage. The FLOPs are calculated with 224 × 224 input

Models	Channels	Blocks in each stages	heads in each stages	Params	FLOPs
CETNet-T	64	[2, 2, 18, 2]	[4, 8, 16, 32]	23M	4.3G
CETNet-S	64	[4, 4, 30, 2]	[4, 8, 16, 32]	34M	6.8G
CETNet-B	96	[4, 4, 30, 2]	[4, 8, 16, 32]	75M	15.1G

4 Detailed Experimental Settings

This section provides supplemental experiment details of image classification, object detection and instance segmentation, and semantic segmentation.

4.1 Image classification on ImageNet-1k

For training on 224×224 input size, we use the AdamW[3] optimizer with a weight decay of 0.05. The default batch size and initial learning rate are set to 1024 and 0.001 (Specially, 2048 and 0.002 for CETNet-T), respectively, and the cosine learning rate scheduler with 20 epochs linear warm-up is used. We use the same data augmentation and regularization strategies used in Swin[4] during training, including RandAugment [1], Mixup [9], Cutmix [8], random erasing [10] and stochastic depth [2]. An increasing degree of stochastic depth augmentation is employed for larger models, i.e., 0.2, 0.3, and 0.5 for CETNet-T, CETNet-S, and CETNet-B, respectively. All models are trained with 300 epochs. A center crop is used during evaluation on the validation set.

When training on a larger input size 384×384, we fine-tune the models trained on 224×224 instead of training from scratch to reduce GPU consumption. We fine-tune

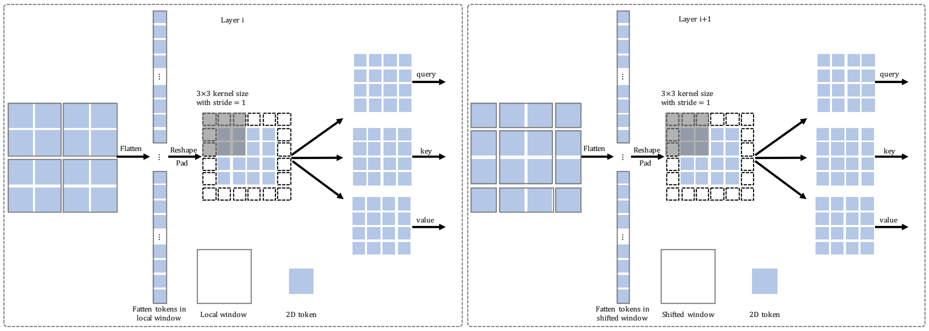

Fig. 2. Detail structure of LEWin. The convolutional projection is implemented by a depth-wise separable convolution with kernel size 3 × 3, stride 1, and padding 1. The window-based MSA is the same as in Swin[4]

the models for 30 epochs with the weight decay of 1e-8, learning rate of 1e-5, batch size of 512, and the same data augmentation and regularizations as the training on 224×224. Unlike the 224×224 input, in the evaluation stage, we removed the center crop during evaluation on the validation set.

4.2 Object Detection and Instance Segmentation on COCO

The dataset and the basic training configuration have been introduced in the main text detailedly. An additional point here is that the stochastic depth is set to 0.2, 0.2, and 0.3 for CETNet-T, CETNet-S, and CETNet-B, respectively.

4.3 Semantic segmentation on ADE20K

For this work, all the models are trained with input size 512×512. Stochastic depth with the ratio of 0.3 is applied for all CETNet models. In the testing stage, we report both single-scale test results and multi-scale test(using resolutions that are [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]× of that in training) results.

5 Comparison for model generalization of Hybrid CNNs/ViTs Design

As we discuss in Sec 5.4 (**Understanding the Role of CNNs in Hybrid CNNs/ViTs Design**), we explore how well CNNs in the deep layer of the hybrid CNNs/ViTs network improve ViTs. Here, we further present the models' (Table 8 in the formal paper) training loss and evaluation accuracy, summarized in Fig 3.

6 Feature Visualization

We show the feature maps of the corresponding stages of Swin-T[4] and our proposed CETNet-T trained on ImageNet-1k. From Fig 4 we can see that the low-level structure

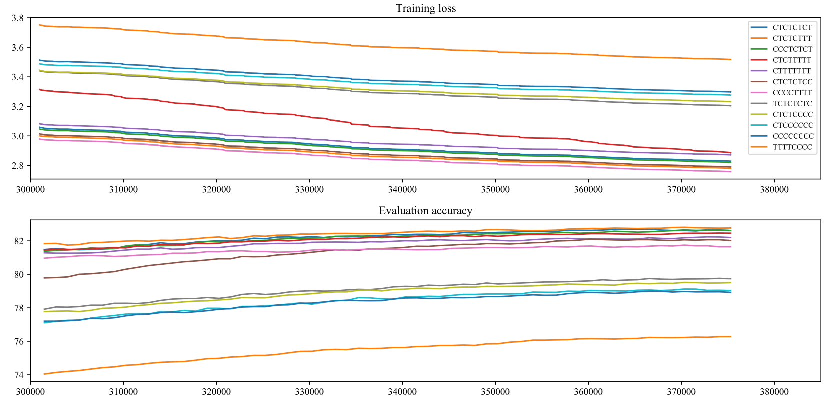

Fig. 3. Comparison for model generalization under different hybrid CNNs/ViTs structures. Top: comparison of training losses in different structures. Bottom: comparison of evaluation accuracy in different structures is top-1 accuracy on the ImageNet-1K validation set. The x-axis represents the number of iterations

features such as edges and lines in the early stage are better learned in our CETNet. In the later stages of the network (especially the 3^{rd} stage shown in the Fig 4), Swin-T learns many invalid feature maps with zero or too large values, which is not the case for our CETNet-T. The feature visualization results demonstrated that our design could improve the feature richness and reduce redundancy of networks.

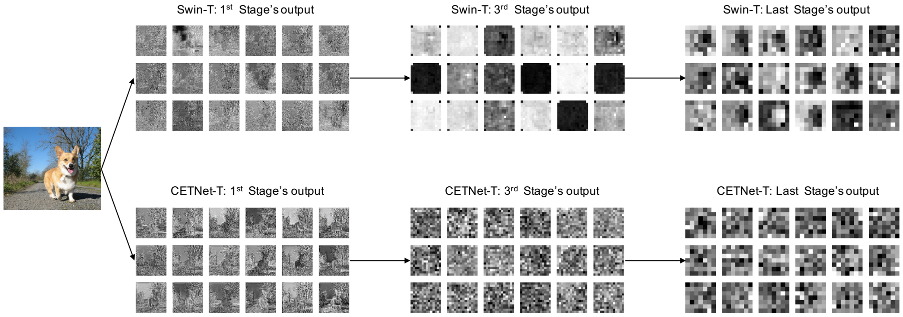
Swin-T: 1st Stage's output
Swin-T: 3rd Stage's output
Swin-T: Last Stage's output
CETNet-T: 1st Stage's output
CETNet-T: 3rd Stage's output
CETNet-T: Last Stage's output

Fig. 4. Feature visualization of Swin-T[4] and our proposed CETNet-T trained on ImageNet-1K, the feature maps visualized here are not attention maps, but image features reshaped from tokens

References

1. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 702–703 (2020)
2. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European conference on computer vision. pp. 646–661. Springer (2016)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
5. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
6. Suyog, G., Tan, M.: Efficientnet-edgetpu: Creating accelerator-optimized neural networks with automl (2019), https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html
7. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: International Conference on Machine Learning. pp. 10096–10106. PMLR (2021)
8. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
9. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
10. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 13001–13008 (2020)