LEMORE: LEARN MORE DETAILS FOR LIGHTWEIGHT SEMANTIC SEGMENTATION

— Supplementary Material —

ImageNet Pre-training

To ensure a fair comparison, we initialize the LeMoRe model with pre-trained parameters from ImageNet. The classification head of LeMoRe includes an average pooling layer followed by a linear layer, leveraging global semantic representations to generate class scores. Given the low resolution of the input images (224 × 224), the target resolution of the input tokens for Nested Attention is set to $\frac{1}{64} \times \frac{1}{64}$ of the input dimensions. Quantitative results of the proposed LeMoRe model on the ImageNet-1K dataset are shown in Table 1.

Table 1. LeMoRe results for ImageNet classification.

Method	Input Size	Top-1 Accuracy(%)	Parameters
LeMoRe	224 × 224	64.1	1.67M

Datasets and Measures

For semantic segmentation, experiments are conducted on four benchmark datasets: ADE20K [1], CityScapes [2], PASCAL Context [3], and COCO-Stuff [4]. The **ADE20K** [1] dataset consists of 25,000 images spanning 150 class categories, with 20,000 images for training, 2,000 for validation, and 3,000 for testing. The **CityScapes** [2] dataset, containing 19 fine-class annotations, comprises 2,975 images for training and 500 images for validation and testing. The **PASCAL Context** [3] dataset includes 10,103 images, featuring 1 background and 59 semantic labels, divided into 4,998 training images and 5,105 testing images. **COCO-Stuff** [4], derived from pixel-level annotations on the COCO dataset, includes 10,000 images, with 9,000 for training and 1,000 for testing.

In line with recent literature [5, 6, 7], we report results using standard metrics: Mean Intersection over Union (mIoU) for segmentation accuracy, Giga Floating Point Operations per Second (GFLOPs), latency, and the number of parameters.

Implementation Details

Our implementation is based on the PyTorch framework and the MMSegmentation toolbox [8]. All models, including our proposed LeMoRe are initially pretrained on the ImageNet-1K dataset [9] before being fine-tuned on semantic segmentation datasets. Batch Normalization layers are applied after almost every convolution layer, except the final output layer. For the ADE20K dataset, we adopt data augmentations similar to those in [10] to ensure fair comparison. We use a batch size of 16 and follow a 160K scheduler as described in [10]. Applied augmentations include random scaling, cropping, horizontal flipping, and resizing. For the CityScapes dataset, we apply the same data augmentations as in [10], with images resized and rescaled to a crop size of 1024×512. Across all datasets, we set an initial learning rate of 1.2×10^{-4} with a weight decay value of 0.01. For the CityScapes dataset specifically, the initial learning rate is adjusted to 3×10^{-4}. Additionally, a Poly learning rate scheduler with a factor of 1.0 is utilized. For the PASCAL Context and COCO-Stuff datasets, we conduct 80K training iterations, incorporating the same data augmentations as in [8]. Training images are resized and cropped to 512×512 for COCO-Stuff and 480×480 for PASCAL Context. Single-scale results are reported for model comparisons.

Details of Feed-Forward Network

For the Feed-Forward Network, in the proposed LeMoRe model, we have integrated depth-wise convolution layer between 1 × 1 convolution layers and to further minimize the computational complexity, expansion factor of two is incorporated. The design of Feed-Forward Network is shown in Figure 1. Where, let $X_f^{'}$ and $X_f^{''}$ be the input and output of Feed-Forward Network, respectively.

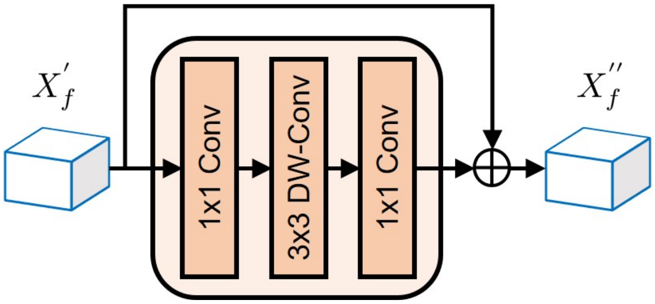

Fig. 1. Design of Feed-Forward Network.

More Visual Results

More visual results are shown in Figure 2 to demonstrate the efficacy of the proposed approach.

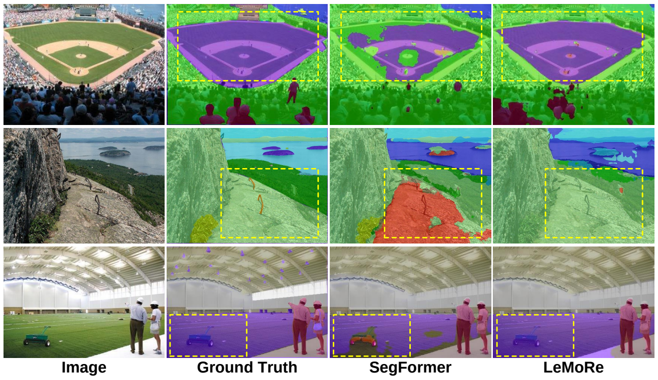

Fig. 2. Visualization of Image, Ground Truth, SegFormer, and LeMoRe results on the ADE20K validation set highlights the proposed model's effectiveness in producing high-quality segmentation maps with enhanced spatial consistency.

1. REFERENCES

[1] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[3] R. Mottaghi, X. Chen, X. Liu, N.G. Cho, S.W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *CVPR*, 2014.

[4] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *CVPR*, 2018.

[5] B. Kang, S. Moon, Y. Cho, H. Yu, and S.J. Kang, "Metaseg: Metaformer-based global contexts-aware network for efficient semantic segmentation," in *WACV*, 2024.

[6] N. Cavagnero, G. Rosi, C. Cuttano, F. Pistilli, M. Ciccone, G. Averta, and F. Cermelli, "Pem: Prototype-based efficient maskformer for image segmentation," in *CVPR*, 2024.

[7] W. Zhang, Z. Huang, G. Luo, T. Chen, X. Wang, W. Liu, G. Yu, and C. Shen, "Topformer: Token pyramid transformer for mobile semantic segmentation," in *CVPR*, 2022.

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

[9] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[10] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 12077–12090, 2021.