Supplementary Material for Searching for TrioNet: Combining Convolution with Local and Global Self-Attention

Huaijin Pi[1]
hjpi@zju.edu.cn
Huiyu Wang[2]
hwang157@jhu.edu
Yingwei Li[2]
yingwei.li@jhu.edu
Zizhang Li[1]
zzli@zju.edu.cn
Alan Yuille[2]
alan.l.yuille@gmail.com

[1] Zhejiang University
Zhejiang, China

[2] Johns Hopkins University
Baltimore, USA

In this supplementary material, we mainly provide more impementation details of our TrioNet search method.

1 Implementation Details

Search Space Our search space contains four stages with $2,3,6,3$ blocks in total. The selected number of blocks is $1,2$ for the first stage, $2,3$ for the second, $3,4,5,6$ for the third and $1,2,3$ for the last stage. The other choices are shown in Tab. 1 in the main paper.

ImageNet Following the typical weight-sharing strategy, the original ImageNet training set is split into two subsets: 10k images for evolutionary search validation and the rest for training the supernet [10]. We report our results on the original validation set.

In our first stage of training, we apply SGD optimizer with learning rate 0.1, Nesterov momentum 0.9 and the weight decay $8e^{-5}$, which is only added to the largest model. Label smoothing [11] is also adopted. We do not use dropout [9] or color jitter since this training procedure is already strongly regularized. We train our supernet for 540 epochs, where contains 10 warmup epochs, with batchsize 32 per gpu and 256 in total. We use $\gamma = 0$ [3] in the final BN [5] layer for each residual block to stabilize the training procedure.

After searching, we train the searched model from scratch. We change the training epochs to 130, and keep other hyper-parameters not changed.

For OFA [1] retraining, we apply the same training procedure on the models provided by their codebase. We employ the provided models with MACCs 0.6B, 0.9B, 1.2B and 1.8B (FLOPs are 1.2B, 1.8B, 2.4B and 3.6B). By using our setting that image input size is 224×224 and convert the stem as conv stem [4, 8], these models FLOPs become 2.4B, 3.4B, 4.6B and 4.8B.

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Small Datasets. We modify our ImageNet recipe by training 60k iterations with weight decay $1e^{-3}$, dropout 0.1, attention-dropout 0.1, and strong color jittering since the datasets are small and easy to overfit. For each dataset, 500 of the training images are selected for evolutionary search. After training the supernet and the evolutionary search, we directly sample the weights [1, 11] from the supernet and finetune the searched model for 1k iterations with a base learning rate $1e^{-3}$ with batchsize 128. We do not employ ImageNet [7] pretraining in order to test the adaptation of our searching algorithm directly on the target data, instead of testing the transfer performance of the found architecture.

Segmentation For semantic segmentation on PASCAL VOC datasets [2], we use SGD optimizer with learning rate 0.01, momentum 0.9 and weight decay $5e^{-4}$. Models are trained with batchsize 16 for $20K$ iterations and the input image size is 512×512. We use output stride 32 and do not apply dilated operators in the backbone. For panoptic segmentation on COCO datasets, we apply Adam optimizer [6] with learning rate $6.25e^{-5}$ and without weight decay. Models are trained with batchsize 8 for $200K$ iterations and the input images size is 640×640.

References

[1] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HylxE1HKwS.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.

[3] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017.

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

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[8] Niki Parmar, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[11] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, T. Huang, Xiaodan Song, and Quoc V. Le. Bignas: Scaling up neural architecture search with big single-stage models. In *ECCV*, 2020.