Supplementary Materials:

DSPNet: Towards Slimmable Pretrained Networks based on Discriminative Self-supervised Learning

This is a supplementary material for our submission "DSPNet: Towards Slimmable Pretrained Networks based on Discriminative Self-supervised Learning". Due to spatial constraints in the main paper, some more detailed explanations are moved to this supporting document. In specific, we provide more **implementation details** for our method to facilitate the reproduction of our results, and additionally show that our obtained full-size online encoder based on DSPNet can also sever as an **ideal initialization for the supervised training of slimmable networks** (Yu et al., 2018; Yu & Huang, 2019), which is better than the random initialization or BYOL (Grill et al., 2020) pretraining.

A Implementation Details

A.1 Architecture

We use MobileNet v3 (Howard et al., 2019) at width $1\times$ as the architecture of the whole online encoder. A *multi-layer perceptron* (MLP) network, which consists of a linear layer with output size 4096 followed by batch normalization (Ioffe & Szegedy, 2015), rectified linear units (ReLU) (Nair & Hinton, 2010), and a final linear layer with output dimension 256, is adopted as the projectors of both the online and target branches, *i.e.*, g_θ and g_ξ^M respectively. The predictor q_χ uses the same architecture as g_θ. The above architecture settings are also applied to ResNet-based experiments without modifications.

A.2 Linear and kNN Evaluation

We provide more details of our linear evaluation in Fig. 4 and Tab. 1a. We largely follow the setting in BYOL (Grill et al., 2020), which includes training a linear classifier on top of the frozen representation, *i.e.*, without updating the backbone feature extraction network parameters or the batch statistics. We add an additional BN (Ioffe & Szegedy, 2015) layer on top of the linear classifier. In our setting, it affects the performance little, but makes the training less sensitive to the learning rate. The learning rate for training the linear classifiers with respect to MobileNet v3 (Howard et al., 2019) and ResNet (He et al., 2016) is 0.2 with a batch size 256. SGD with a momentum of 0.9 is adopted. We train the classifiers for 80 epochs in total with a cosine decay learning rate schedule (Loshchilov & Hutter, 2016). For kNN evaluation in Tab. 1a, a sample is classified by taking the most frequent label of its K-nearest neighbors (K = 20) based on the same frozen representation as the above linear evaluation.

A.3 Semi-supervised Evaluation

For the semi-supervised evaluation, the training set is altered to 1% or 10% subset. By default, we adopt fine-tuning evaluation (Fig. 5, and the entry marked with *fine-tuning* in Tab. 1b), where the parameters in both the backbone and the classifier are all trainable. And distinct learning rates are applied to the backbone and the classifier following (Caron et al., 2020), *i.e.*, 0.001 and 0.02 respectively. The same training settings are adopted for MobileNet v3 (Howard et al., 2019) and ResNet (He et al., 2016).

Furthermore, to make fair comparisons to SSL methods with distillation (Fang et al., 2020; Gao et al., 2021) as shown in Tab. 1b, we adopt the same semi-supervised linear evaluation protocol as them by training a linear classifier on top of the frozen representation of ResNet18 slimmed from our DSPNet. Specifically, the configurations largely follow the linear evaluation in Appendix A.2, except that the learning rate for the classifier is set to 0.02 and the number of training epochs is reduced to 50. The performance of semi-supervised linear evaluation for SEED (Fang et al., 2020) and Disco (Gao et al., 2021) in Tab. 1b is directly borrowed from their papers, both of which employ a large pretrained R152 (He et al., 2016) as a teacher.

A.4 DETECTION AND SEGMENTATION

For detection and segmentation tasks, we fine-tune Mask R-CNN (He et al., 2017) on COCO (Lin et al., 2014). The typical FPN (Lin et al., 2017) backbone is adopted. "SyncBN" (Peng et al., 2018) is adopted in both backbone and FPN. We fine-tune all layers (including BN) end-to-end. The schedule is the default 2× in (Wu et al., 2019). Specifically, all the models are trained for 180k iterations with a batch size 16. The learning rate starts from 0.02, and is divided by 10 at [120k, 160k] iterations. The short-edge of the training image is in [640, 800], while fixed at 800 during inference. We adopt the same training settings for both MobileNet v3 (Howard et al., 2019) and ResNet (He et al., 2016).

In Tab. 1c of the main paper, we adopt the pretrained ResNet18 (He et al., 2016) provided by torchvision[2] for the *Supervised-IN.* entry, and utilize the open-source pretrained checkpoints provided by the authors for SEED (Fang et al., 2020) and CompRess (Abbasi Koohpayegani et al., 2020). The teacher models for them are a 2× wider ResNet50 (He et al., 2016) network pretrained for 400 epochs with SwAV (Caron et al., 2020) method and a ResNet50 network pretrained for 800 epochs with SwAV (Caron et al., 2020) method respectively.

A.5 IMAGE SIZE

During self-supervised pretraining, we resize the images to 128×128 for ResNet (He et al., 2016), which is smaller than the common setting (224×224). Here we provide an alternative setup, experimentally analyze the effects of different image size settings by testing ResNet18 with BYOL-based pretraining for 300 epochs. When the images are resized to 224×224, the total mini-batch size is reduced to 1024 from 4096. The above settings along with training cost and top1 accuracy obtained by linear evaluation are shown in Tab. A1. Two image size settings achieve comparable performance with also comparable training cost. As larger batch size makes it easier to from multi-node distributed training with more machines, which contributes to shorter training time, we thus adopt 128×128 image size as the default setup.

Batch size	**Image size**	**Training cost** (GPU hour)	**Top1 accuracy**
1024	224	0.52k	61.0
4096	128	0.51k	60.8

Table A1: Comparison of different training setups.

B DSPNET AS THE INITIALIZATION OF SLIMMABLE NETWORKS

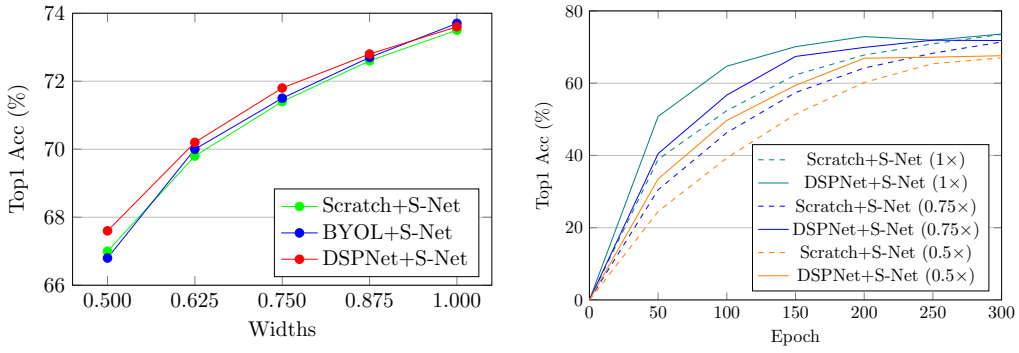

Figure A1: Top1 accuracy comparisons of slimmable networks (Yu et al., 2018) using different initialization methods *(Left)* and training convergence of several DNs *(Right)*. Our DSPNet can serve as an ideal pretraining.

In this section, we investigate the value of DSPNet on promoting the performance of slimmable networks (Yu et al., 2018). Slimmable networks (Yu et al., 2018; Yu & Huang, 2019; Yu et al.,

[2]https://pytorch.org/vision.

2020; Cai et al., 2019) were introduced that can switch among different widths and depths at runtime, permitting instant and adaptive accuracy-efficiency trade-offs. In this sense, our DSPNet can serve as an ideal pretraining method for slimmable networks (Yu et al., 2018). To validate this assumption, we conduct the DSPNet-based pretraining with MobileNet v3 at widths $[0.5, 0.625, 0.75, 0.875, 1.0]$, and then fine-tune the obtained full-size online encoder following the training algorithm of slimmable networks (Yu et al., 2018) on the ImageNet (Deng et al., 2009) classification task. The switches of the slimmable network during the supervised training are the same as during our DSPNet-based pretraining, *i.e.*, MobileNet v3 at widths $[0.5, 0.625, 0.75, 0.875, 1.0]$. The *sandwich rule* (Yu & Huang, 2019) and *inplace distillation* (Yu & Huang, 2019) training techniques are adopted. We adopt the training recipes provided by `timm` (Wightman, 2019), except that the number of training epochs is reduced to 300 from 600 for fast experiment. We show the effect of different initialization strategies for training slimmable networks and report the top1 accuracy of the five switches of them in Fig. A1.

In the comparison, we first adopt a pretrained MobileNet v3 at width $1\times$ with BYOL (Grill et al., 2020) as the initialization of the supervised training of slimmable network (BYOL+S-Net in Fig. A1). However, no improvement is observed compared with that trained from the scratch (Scratch+S-Net in Fig. A1). Then we adopt our DSPNet as the initialization (DSPNet+S-Net in Fig. A1), and improvements are observed especially for the smaller switches. The convergence is also sped up. We attribute it to that DSPNet provides a more suitable initialization for the subsequent supervised training of slimmable networks, in which all the desired sub-networks have relatively good representations at the very beginning of the training.

C Training Stability

Since randomness is involved during the training of our DSPNet, the training stability should be considered. We provide the training curves in Fig. A2, which show only a little more aggravated training perturbation for our DSPNet than BYOL baseline. Furthermore, our 3 tests with different random seeds achieve similar performance.

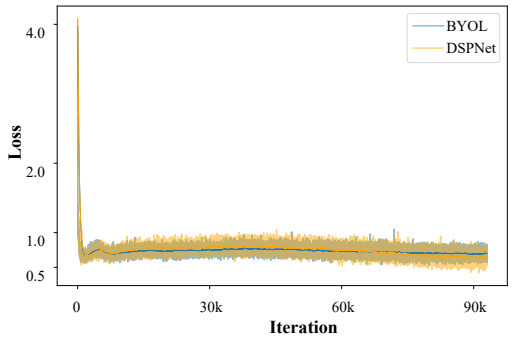

Figure A2: Training curves of BYOL and our DSPNet.