SuperMix: Supplementary Material

Anonymous CVPR submission

Paper ID 3387

1. Distillation Setup

For the sake of consistency with the previous works, major parts of the experimental setup are taken from the benchmark implementations of CRD [10][1].

1.1. Implementation Details for CIFAR-100

We train all models using Stochastic Gradient Descent (SGD) with the initial learning rate of 0.1, momentum of 0.9, weight decay of $5e-4$, batch size of 128, and the learning rate decay factor 0.1 at epochs 200, 300, 400, and 500. The total number of epochs is set to 600. The parameters for SuperMix are set as follows: $\sigma = 1$, spatial size of the masks equal to 8×8, $\lambda_s = 25$, and $\alpha = 3$. The size of the Gaussian kernel in SuperMix is set to 5σ. All the results for the baseline distillation approaches are reported from CRD [10]. The parameters for $\mathcal{L}_{KD}$ are selected according to the best performance reported in [10] as $\lambda_{KD} = 0.9$, and $\tau = 4$.

1.2. Network Architectures for CIFAR-100

The network architectures for the distillation experiments on CIFAR-100 are exactly the same as the benchmark models implemented in CRD [10]. We briefly describe the network architectures and denote their total number of parameters (TNP) in the following. For more details, please refer to our code or the code provided by Tian *et al*. [10].

WRN-[a]-[b]: Wide Residual Network [11] with depth **a** and width factor **b**. Convolutional layers do not have bias weights. TNP: WRN-40-2 = 2255156, WRN-40-1 = 569780, WRN-16-2 = 703284.

ResNet[a]: Residual Network [4] adapted for CIFAR dataset with 3 basic blocks, each with 16, 32, 64 channels and total number of **a** layers. Only ResNet50 uses buttleneck blocks. TNP: ResNet110 = 1736564, ResNet56 = 861620, ResNet32 = 472756, ResNet20 = 278324, ResNet50 = 23705252.

ResNet[a]x4: Four times wider Residual Network [4] adopted for CIFAR dataset with 3 basic blocks, each with 64, 128, 256 channels and total number of **a** layers. TNP: ResNet32x4 = 7433860, ResNet8x4 = 1233540.

VGG[a]: VGG [9] adapted from the original ImageNet model with 5 convolutional blocks and **a-3** convolutional layers. TNP: VGG13 = 9462180, VGG8 = 3965028.

MobileNetV2: The original MobileNetV2 [8] with the width factor of 0.5. TNP: 812836.

ShuffleNetV1 & ShuffleNetV2: The original ShuffleNets [12, 7] adapted for the CIFAR-100 dataset. TNP: ShuffleNetV1 = 949258, ShuffleNetV2 = 1355528.

1.3. Implementation Details for ImageNet

Models are trained using the standard practice provided by PyTorch for training on ImageNet. Stochastic Gradient Descent (SGD) is used with the initial learning rate of 0.025, momentum of 0.9, weight decay of $1e-4$, batch size of 256, and the learning rate decay factor 0.1 every 30 epochs. The total number of epochs is set to 100. The parameters for SuperMix are set as follows: $\sigma = 2$, spatial size of the masks equal to 16×16, $\lambda_s = 25$, $\alpha = 3$. The size of the Gaussian kernel in SuperMix is set to 5σ. All the results for the baseline distillation approaches are reported from CRD [10]. The parameters for $\mathcal{L}_{KD}$ are selected according to the best performance reported in [10] as $\lambda_{KD} = 0.9$, and $\tau = 4$. ResNet34 and ResNet18 are the benchmark implementation of deep residual Networks [4] provided by PyTorch with the total number of parameters of 21797672, and 11689512, respectively.

2. Object Classification Setup

The augmentation performance of SuperMix is compared to AutoAugment (AA) [1], Fast AutoAugment (FAA) [6], Population based Augmentation (PBA) [5], and RandAugment [2]. Network architectures for both CIFAR-100 and ImageNet are the same models provided in the official repository of FAA[2]. The results for the baselines are also reported directly from FAA. The training setup for both datasets is exactly the same as the distillation setup in Sec-

[1] http://github.com/HobbitLong/RepDistiller

[2] The official code is available at: https://github.com/kakaobrain/fast-autoaugment

tion 1 of the supplementary. It may be noted that the implementation of Wide ResNet [11] for this part is slightly different than the implementation for the distillation task in [10]. Specifically, convolutional layers in this part have bias weights. Therefore, the total number of parameters is greater than the same architecture detailed in the previous section. We notified this in the paper by referring to WRN-40-2_a and WRN-40-2_b for the classification and distillation tasks, respectively. In the following, we briefly describe the network architectures. For more information, please refer to the code.

WRN-[a]-[b]: Wide Residual Network discussed in Section 1.2 with convolutional layers that have bias weights. TNP: WRN-40-2 = 2258084, WRN-28-10 = 36546980.

SS-([a] 2×[b]d): Shake-shake model [3] with the depth of **a** and base channels **b** . TNP: SS-(26 2× 96d)=26366404.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 113–123, 2019. 1

[2] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019. 1

[3] Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[5] Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules. *arXiv preprint arXiv:1905.05393*, 2019. 1

[6] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *Advances in Neural Information Processing Systems*, pages 6662–6672, 2019. 1

[7] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 1

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[10] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 1, 2

[11] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1, 2

[12] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 1