Direct Differentiable Augmentation Search Supplementary Materials

Aoming Liu[1], **Zehao Huang**[2], **Zhiwu Huang**[1], **Naiyan Wang**[2]
[1]ETH Zürich, Switzerland, [2]TuSimple, Beijing
{aoliu@student, zhiwu.huang@vision.ee}.ethz.ch, {zehaohuang18,winsty}@gmail.com

1. Addtional Experiments

1.1. Experiment on SVHN

Search Setting. We also search augmentation on SVHN [3]. Similar as the experiments on CIFAR-10/100, we first search augmentation on a proxy task with a small network, Wide-ResNet-40-2 on part of the dataset for 20 epochs. We split 3000 images for training dataset S_{train} and 3000 images for S_{val}. The training mini-batch size is 32 while the validation mini-batch size is 256. At each step, we sample $\hat{L} = 3$ augmentation policies randomly according to a uniform distribution. The number of operation N_o in an augmentation policy is 2. The search is carried out on a single RTX 2080Ti GPU. As for initialization, we initialize $\mathbf{p}_o$ equally and p_{tp} as 0.35.

α_{tp} and $\boldsymbol{\alpha_o}$ are updated with Adam optimizers. The learning rate for $\boldsymbol{\alpha_o}$ is 0.005, and that for α_{tp} is 0.001. For the 2 optimizers, we set $\beta_1 = 0.5$, $\beta_2 = 0.999$. The network parameters of Wide-ResNet-40-2 are updated with SGD optimizer with momentum as 0.9. The learning rate is 0.05, with cosine decay, and the weight decay is 0.0005.

Training Setting. After search, we apply the searched augmentation to Wide-ResNet-28-10. We train the network for 160 epochs on the full training set and report the performance evaluated on test set. We set initial learning as 0.005, batch size as 128, momentum as 0.9, weight decay as 0.001, and cosine learning rate decay.

Result. The search cost and test error rate are shown in Table.1. We run the experiment for three times and report the average test error rate. Compared with other efficient automatic augmentation methods, our DDAS can also achieve comparable performance and efficiency.

1.2. The effectiveness of policy number $\hat{L}$.

Our DDAS achieves efficient search because it can search for good augmentation policies with limited sampled augmentation policies at each step ($\hat{L} = 2,\ 3$). We further explore the effectiveness of sampled augmentation policy number $\hat{L}$ on the performance of searched policies. We search for augmentation policies with different $\hat{L}$ values and show the results in Table 2. We can see that simply increasing the sampled augmentation policy number $\hat{L}$ does not increase the performance.

2. Implemntation Details

2.1. Image Classsification

Operations. Here we list the augmentation operations used for image classification.

- `Identity`
- `Rotate`
- `Posterize`
- `Sharpness`
- `Translate-x/y`
- `FlipLR`
- `AutoContrast`
- `Solarize`
- `Contrast`
- `Shear-x/y`
- `Invert`
- `FlipUD`
- `Equalize`
- `Color`
- `Brightness`
- `Smooth`
- `Blur`

Similar as AA [1], the operations are from PIL, a popular Python image library.[1] In addition, we add `Cutout` to search space of ImageNet, and use it defaultly with region size as 16 pixels on CIFAR-10/100 and SVHN.

Magnitude. As for the magnitude, we follow RA [2] and use the same linear scale for indicating the magnitude (strength) of each transformation. As mentioned in Method, we discretely sampled magnitude value, and the selected magnitude values for CIFAR-10/100, SVHN and ImageNet are listed in Table 3.

2.2. Object Detection

For object detection, we use the operations in [4]. The magnitude setting keeps the same as [4]. The selected magnitude values for COCO is listed in Table 3.

3. Experiments Details

3.1. CIFAR-10/100 & Sanity Check

Search Setting. As for search both α_{tp} and $\boldsymbol{\alpha_o}$ are updated with Adam optimizers. The learning rate for $\boldsymbol{\alpha_o}$ is 0.005, and that for α_{tp} is 0.001. For both of the 2 optimizers, we

[1]https://pillow.readthedocs.io/en/5.1.x/

Table 1. SVHN test error rates (%) and search cost (GPU hour). WRN is shorthand of Wide-ResNet.

	Baseline	Cutout	AA	PBA	Fast AA	RA	Faster AA	DADA	**DDAS**
Error									
WRN-28-10	1.5	1.3	1.1	1.2	1.1	1.0	1.2	1.2	1.2
Cost									
WRN-28-10	-	-	1000	1	1.5	-	0.06	0.1	0.1

Table 2. Sampled augmentation policy number $\hat{L}$ ablation.

$\hat{L}$	3	7	11
Error	16.6	16.8	17.1

Table 3. Manitudes sampled for different datasets

Dataset	mag
CIFAR-10/ CIFAR-100/ SVHN	$\{2, 6, 10, 14\}$
ImageNet	$\{7, 14\}$
COCO	$\{4, 6, 8\}$

set $\beta_1 = 0.5,\ \beta_2 = 0.999$. The network parameters of Wide-ResNet-40-2 are updated with SGD optimizer with momentum as 0.9. The learning rate is 0.05, with cosine decay, and the weight decay is 0.0005.

As for Sanity Check, the settings are basically similar as that of CIFAR-10/100 experiments. The difference is that we only use $N_o = 1$, magnitude as 2 and p_{tp} initialized as 0.75.

Training Setting. Both of the 2 training networks are trained with SGD optimizer whose momentum is 0.9. The batch size is 128 and the cosine LR schedule is adopted for all the models. For Wide-ResNet-28-10, we set initial learning as 0.1 and weight decay as 0.0005. For Shake-Shake(26 2x96d), we set initial learning as 0.01 and weight decay as 0.001. The $p_t(1), ..., p_t(T_{max})$ are smoothed with a mean filter whose size $F_s = 2$.

3.2. ImageNet

Search Setting. Both α_{tp} and $\boldsymbol{\alpha_o}$ are optimized with Adam optimizer. The learning rate for $\boldsymbol{\alpha_o}$ is 0.003, and that for α_{tp} is 0.002. For the 2 optimizers, we set $\beta_1 = 0.5,\ \beta_2 = 0.999$. The network parameters of ResNet-18 are updated with SGD optimizer. The learning rate is 0.01, with cosine decay, the momentum is 0.9 and the weight decay is 0.0001.

Training Setting. As for training, the 2 networks are trained with SGD optimizer whose momentum is 0.9. We set initial learning rate as 0.2, batch size as 512, momentum as 0.9, weight decay as 0.0001 and step learning decay by 0.1 at epoch 90, 180 and 240. For fair comparison we reproduce RA on ResNet-50 with our training setting and report the result. The p_ts are smoothed with a mean filter whose size $F_s = 6$.

3.3. COCO

Search Setting. Both $\boldsymbol{\alpha_o}$ and α_{tp} are updated with Adam optimizers. The learning rate for $\boldsymbol{\alpha_o}$ is 0.001, and that for α_{tp} is 0.001. And we set $\beta_1 = 0.5,\ \beta_2 = 0.999$ for the 2 optimizers. The parameters of RetinaNet are updated with SGD optimizer. The learning rate is 0.04, with step learning decay at epoch 24 and 28, the momentum is 0.9 and the weight decay is 0.0001.

Training Setting. Both of the 2 training networks are trained with SGD optimizer whose momentum is 0.9. We set the initial learning rate as 0.08, batch size as 64, momentum as 0.9, weight decay as 0.0001 and step learning rate decay at epoch 120 and 140.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation strategies from data. In *CVPR*, 2019.

[2] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

[3] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPSW*, 2011.

[4] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *ECCV*, 2020.