Journey Towards Tiny Perceptual Super-Resolution: Supplementary Material

Royson Lee[1*[0000−0002−6716−7994]], Łukasz Dudziak[1[0000−0003−4929−265X]], Mohamed Abdelfattah[1[0000−0002−4568−8932]], Stylianos I. Venieris[1[0000−0001−5181−6251]], Hyeji Kim[1[0000−0002−2256−5729]], Hongkai Wen[1,2[0000−0003−1159−090X]], and Nicholas D. Lane[1,3[0000−0002−2728−8273]]

[1] Samsung AI Center, Cambridge, UK
[2] University of Warwick
[3] University of Cambridge
*royson.lee@samsung.com

1 Experiment Details

Neural Architecture Search: Similar to previous approaches, we use a single LSTM layer with 100 hidden units as the trainable policy, π_θ. It takes an empty embedding as input and generates a sequence of l outputs, where l is the number of decisions to make in order to decide about a generator's structure. Each element of the sequence at position i is then passed through the following composition of functions, including a tanh constant of 2.5 and a sampling logit temperature of 5.0, in order to reduce its dimensionality and produce a probability distribution:

$$\text{softmax} \circ 2.5 \cdot \tanh \circ 0.2 \cdot \text{linear} \circ l_i \tag{1}$$

Each search is ran on five servers, totalling to 40 NVIDIA GeForce GTX 1080 Ti, and each GPU can fit up to three models which are trained in parallel depending on the sampled model's memory usage at each step. Our generator search took 2 days and our discriminator search took 10 days due to the memory requirement needed for the pre-trained generator, sampled discriminator, and two VGG networks to compute L_{vgg} and LPIPS respectively in each training pipeline. Due to the huge resource needed to run discriminator search, we limit the number of mult-add operations and run a constrained search for a generator, as mentioned in the paper. The performance of our generator for the discriminator search is listed in Table 1.

Each training pipeline is run by a separate process/worker and each worker asynchronously samples the probability distribution (Eq (1)), trains the resulting model, and returns the result (PSNR for generator search and LPIPS for discriminator search) from the validation set, $\hat{V}$, of the proxy task. The result would then be normalized using a minmax normalization, N, to scale it from 0 to 1 with a exponential moving average baseline, EMA, (decay of 0.95) to obtain the reward. We then add the sampled entropy, H, ($\zeta = 0.0001$) to the reward and use it to update π_θ via REINFORCE using an Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) with learning rate of $3.5e-4$. The training process is detailed in Eq. 2 where $\tilde{M}$ is the metric of choice (PSNR, LPIPS, etc), and $\tilde{L}$ is the training loss (L_G and L_D for generator and discriminator search respectively).

$$
\begin{aligned}
R(s) &= EMA(N(E(s))) + \zeta H(\pi) \\
E(s) &= \frac{1}{|\hat{V}|} \sum_{(I^{LR}, I^{HR}) \in \hat{V}} \tilde{M}(G_s(I^{LR}, W_s^*), I^{HR}) \\
W_s^* &= \arg\min_{W_s} \frac{1}{|\hat{T}|} \sum_{(I^{LR}, I^{HR}) \in \hat{T}} \tilde{L}(G_s(I^{LR}, W_s), I^{HR})
\end{aligned}
\tag{2}
$$

Table 1. Our TPSR-NOGAN model **serves as a good basis** for the $\times 4$ upscaling discriminator search as it is the most computationally efficient and has performance that is comparable with other distortion-driven models in the literature within the same computational regime. Given that our goal is to build a **perceptual-based** model, we do not optimize our base model further. Higher is better for distortion metrics. red/blue represents best/second best respectively

Scale	Model	Params (K)	Mult-Adds (G)	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM
	FSRCNN [3]	12	6.0	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020
$\times 2$	MOREMNAS-C [2]	25	5.5	37.06/0.9561	32.75/0.9094	31.50/0.8904	29.92/0.9023
	TPSR-NOGAN	60	14.0	37.38/0.9583	33.00/0.9123	31.75/0.8942	30.61/0.9119
	FSRCNN [3]	12	4.6	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280
$\times 4$	FEQE-P [7]	96	5.6	31.53/0.8824	28.21/0.7714	27.32/0.7273	25.32/0.7583
	TPSR-NOGAN	61	3.6	31.10/0.8779	27.95/0.7663	27.15/0.7214	24.97/0.7456

Super-resolution: The set of hyper-parameters chosen for both the proxy task and the full task is summarized in Table 2.

Table 2. Hyper-parameters for the searching (proxy-task) and training (full-task) of the generator and discriminator. The generator search is done on $\times 2$ upscaling and the discriminator is done on $\times 4$ upscaling. We use the features before activations in the pre-trained VGG19 network provided by PyTorch to compute L_{vgg}. Each input patch is an RGB image. We used the same model in both proxy and full task to closely align the performance between both tasks. For speed ups during the search, we use lower fidelity estimates, lower patch size etc, that have been shown in previous works such as ESRGAN to preserve the ranking of the model.

Search	Hyper-parameter	Proxy-task	Full-task
Generator	Epochs	50	450
	Batch size	64	16
	Input patch size	12×12	96×96
	L_G (α)	1	1
Discriminator	Epochs	50	450
	Batch size	32	16
	Input patch size	24×24	48×48
	VGG19 Features	22	54
	L_G (α, λ, γ)	(0.01, 1, 0.005)	(0.01, 1, 0.005)

For all experiments, we use 800 images for training and 100 images for validation from the DIV2K dataset. We train for the stated number of epochs in Table 2 using an Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) with learning rate of $1e-4$ at the beginning and $5e-5$ after 200 epochs for the full task. All training patches are randomly flipped, both horizontally and vertically, rotated $90°$, and subtracted by the mean RGB values of the DIV2K dataset. All operations in the generator is followed by a PReLU and all operations in the discriminator is followed by batch normalization and PReLU.

Finally, all experiments are built on top of EDSR's [5] code base[4] using PyTorch 1.2. PSNR and SSIM [9] were evaluated on each image's Y-channel and NIQE [6] and PI [1] were evaluated using the official code for the PIRM 2018 SR Challenge[5] on Matlab R2018b. As mentioned in the main paper, all images are shaved by their scaling factor before evaluation apart from that of LPIPS (version 0.1), which we evaluate using the linear calibration of the features in the provided VGG network[6]. In order to compare to previous works, the number of Mult-Add operations are calculated by upscaling to an 1280 * 720 image. Images from state-of-the-art models are taken from Wang *et al.* [8][7] and Thang *et al.* [7][8] or generated using provided model by Dong *et al.* [4][9].

[4] https://github.com/thstkdgus35/EDSR-PyTorch
[5] https://github.com/roimehrez/PIRM2018
[6] https://github.com/richzhang/PerceptualSimilarity
[7] https://github.com/xinntao/ESRGAN
[8] https://github.com/thangvubk/FEQE
[9] https://github.com/tensorlayer/srgan

References

1. Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L.: The 2018 PIRM Challenge on Perceptual Image Super-Resolution. In: ECCV Workshops (2018)
2. Chu, X., Zhang, B., Xu, R., Ma, H.: Multi-objective reinforced evolution in mobile neural architecture search. ArXiv **abs/1901.01074** (2019)
3. Dong, C., Loy, C.C., Tang, X.: Accelerating the Super-Resolution Convolutional Neural Network. In: ECCV (2016)
4. Ledig, C., et al.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
5. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced Deep Residual Networks for Single Image Super-Resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2017)
6. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a “Completely Blind" Image Quality Analyzer. IEEE Signal Processing Letters **20**, 209–212 (2013)
7. Vu, T., Van Nguyen, C., Pham, T.X., Luu, T.M., Yoo, C.D.: Fast and Efficient Image Quality Enhancement via Desubpixel Convolutional Neural Networks. In: The European Conference on Computer Vision (ECCV) Workshops (September 2018)
8. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: ECCV Workshops (2018)
9. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**, 600–612 (2004)