ARM: Any-Time Super-Resolution Method (Supplementary Material)

Bohong Chen[1], Mingbao Lin[3], Kekai Sheng[3], Mengdan Zhang[3], Peixian Chen[3], Ke Li[3], Liujuan Cao[1]⋆, Rongrong Ji[1,2,4]

[1]School of Informatics, Xiamen University.
[2]Institute of Artificial Intelligence, Xiamen University.
[3]Tencent Youtu Lab. [4]Institute of Energy Research, Jiangxi Academy of Sciences.
bhchen@stu.xmu.edu.cn, linmb001@outlook.com
{saulsheng,davinazhang,peixianchen,tristanli}@tencent.com,
{caoliujuan,rrji}@xmu.edu.cn,

1 More Ablation Studies

1.1 Effect of η in Eq. (5)

Recall that in Eq. (5), η is a manually set hyper-parameter used to balance the numerical difference between the calculated cost and the PSNR estimate in the inference stage. A larger η leads to a preference to the subnet with a larger estimated PSNR $\bar{p}^j_{\hat{k}}$. To better illustrate the process of subnet selection in Eq. (5), we plot the Edge-to-PSNR lookup tables of different subnets as well as the interpolation branch (denoted as $\mathcal{N}_{W[0:0]}$) using different η in Fig. 1.

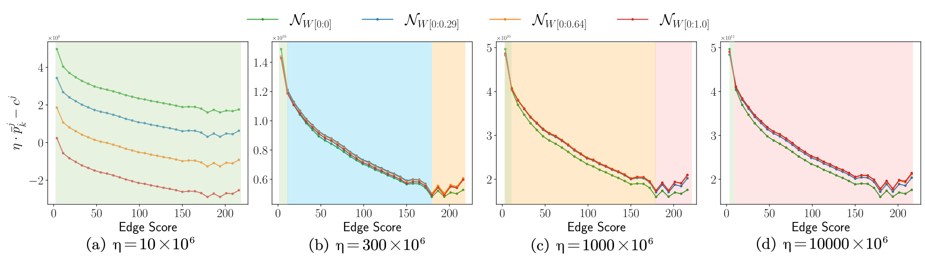

Fig. 1. The impact of η in Eq. (5). The background color indicates the selected subnet or interpolation branch according to the edge score.

As shown in Fig. 1 (a), when the available computation is close to zero, we take a small value for η and then ARM automatically selects the branch with the highest PSNR, *i.e.*, interpolation branch (*green* background), for all edge score patches to satisfy the computational constraint. When there are more calculations available, we can set η to a larger value (such as Fig. 1 (b)(c)). In

⋆ Corresponding Author

Fig. 1 (b), for example, when the edge score of the patches is small (*e.g.*, edge score < 10), ARM will select the interpolation branch (*green* background) for these patches; when the edge score is moderate (*e.g.*, edge score $\in (10, 180)$), ARM will select $subnet_1$ (*blue* background) for these patches; For larger edge scores, $subnet_2$ (*yellow* background) will be selected. Finally, we set η to a large value when there are many computation resources available. In this case, ARM selects the largest $subnet_3$ to super-resolution almost all patches. It is worth noting that for very "easy" patches, ARM still chooses interpolation rather than always selecting the subnets, since for these patches, interpolation outperforms all sizes of subnets.

We use some images from Test2K as examples to illustrate how ARM automatically adjusts the calculation based on the same pre-trained ARM supernet with different η settings at inference time. The results are shown in Fig. 3. As the η increases, more and more patches are selected to use a larger subnet for super-resolution, thus gradually improving the PSNR. It can also be seen that in Fig 3, the ARM achieves better performance with fewer FLOPs than the backbone network and the previous SOTA dynamic SISR method [1].

1.2 Effect of K subintervals

Our Edge-to-PSNR Lookup Tables are constructed by splitting the edge score interval into a total of K subintervals and then averaging over all PSNR values within each interval as the estimated PSNR. Fig. 2 analyzes the impact of K. For a small value of K, the estimated PSNR is loosely scattered, which causes an inaccurate PSNR estimation. Increasing the value of K leads to a more well-fitted edge-psnr mapping. Though a larger K may result in a better estimation, more parameters from the lookup tables are introduced. In our experiment, we set $K = 30$ across all the experiments for a better tradeoff and observe high-performing results as well.

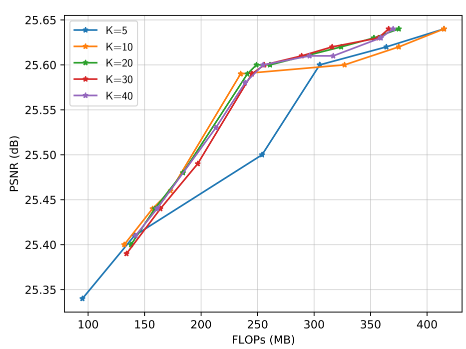

Fig. 2. Effect of different values of K.

2 The inference speed

We measure the average inference time per image on Test2K dataset on a single core of *Intel Xeon Platinum 8255C* CPU. The results are listed in Table 1. Due to limited rebuttal time, no further speed optimization was conducted for hardware, so the actual speedup ratio was somewhat different from the theoretical speedup ratio. Despite, ARM still has some speed advantage over backbone and ClassSR.

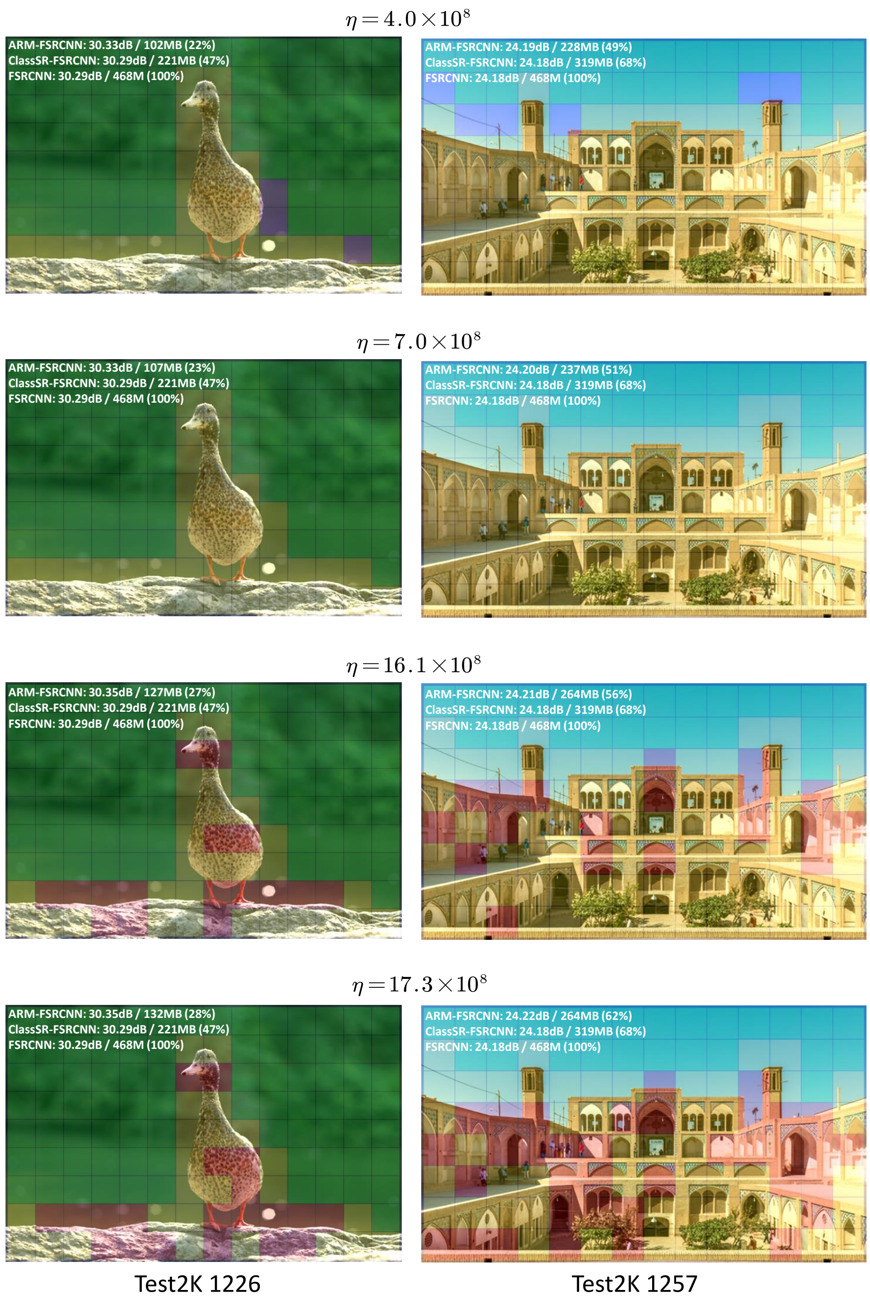

Fig. 3. Examples of super-resolution visualization of ARM-FSRCNN with different η. The green, blue, yellow and red masks on the patch indicate that ARM uses interpolation, $subnet_1$, $subnet_2$ and $subnet_3$ for super-resolution of the patch, respectively.

Table 1. The results of latency (Lat.) of Test2K with FSRCNN as the backbone. The latency is the average of five trials.

Model	FSRCNN	ClassSR	ARM-L	ARM-M	ARM-S
FLOPs/M	468 (100%)	311 (66%)	366 (78%)	289 (62%)	245 (52%)
Lat./ms	542.25	549.48	540.55	531.36	518.96

References

1. Kong, X., Zhao, H., Qiao, Y., Dong, C.: Classsr: A general framework to accelerate super-resolution networks by data characteristic. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12016–12025 (2021)