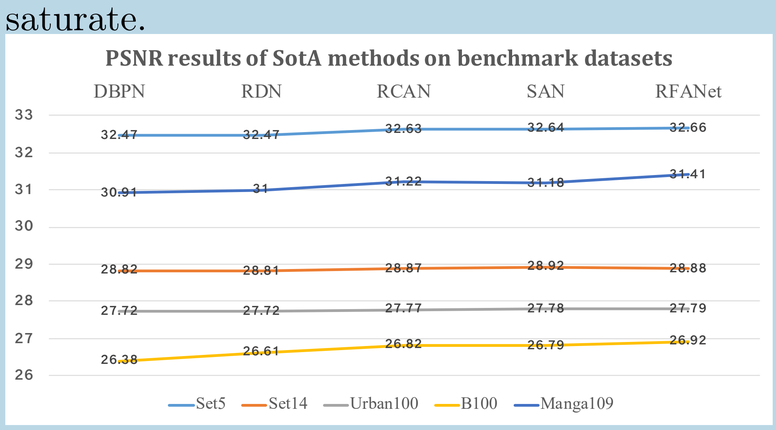
Efficient Super Resolution by Recursive Aggregation

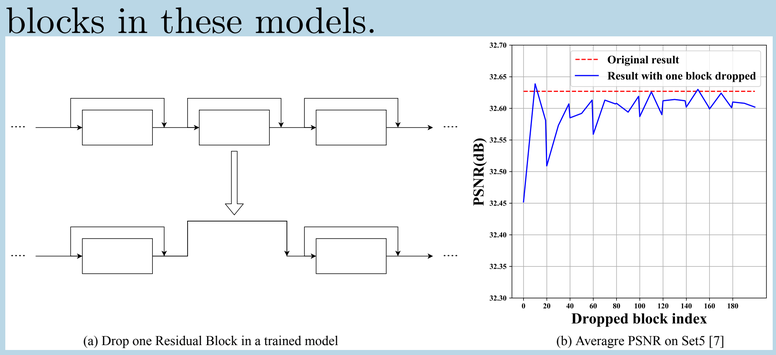
Zhengxiong Luo, Huang Yan, Shang Li, Liang Wang and Tieniu Tan
zhengxiong.luo@cripac.ia.ac.cn, yhuang@nlpr.ia,ac.cn, lishang2018@ia.ac.cn {wangliang, tnt}@nlpr.ia.ac.cn

Problems

Nowadays super-resolutions are becoming dramatically deep and large. While their performance on benchmark datasets are beginning to saturate.



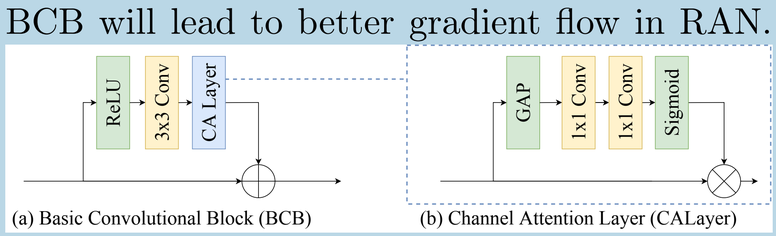
Gradients cannot flow to all layers when the model is too deep. Therefore, there might be a large proportion of sub-optimized layers or blocks in these models.



(a) Drop one Residual Block in a trained model (b) Averagre PSNR on Set5 [7]

Basic Convolutional Block

We simplify the residual block to only one convolutional layer. Thus it allows more skip connections with the same number of layers. The multiple skip connections introduced by BCB will lead to better gradient flow in RAN.



(a) Basic Convolutional Block (BCB) (b) Channel Attention Layer (CALayer)

References

[1] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012.

[2] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[3] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010.

[4] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

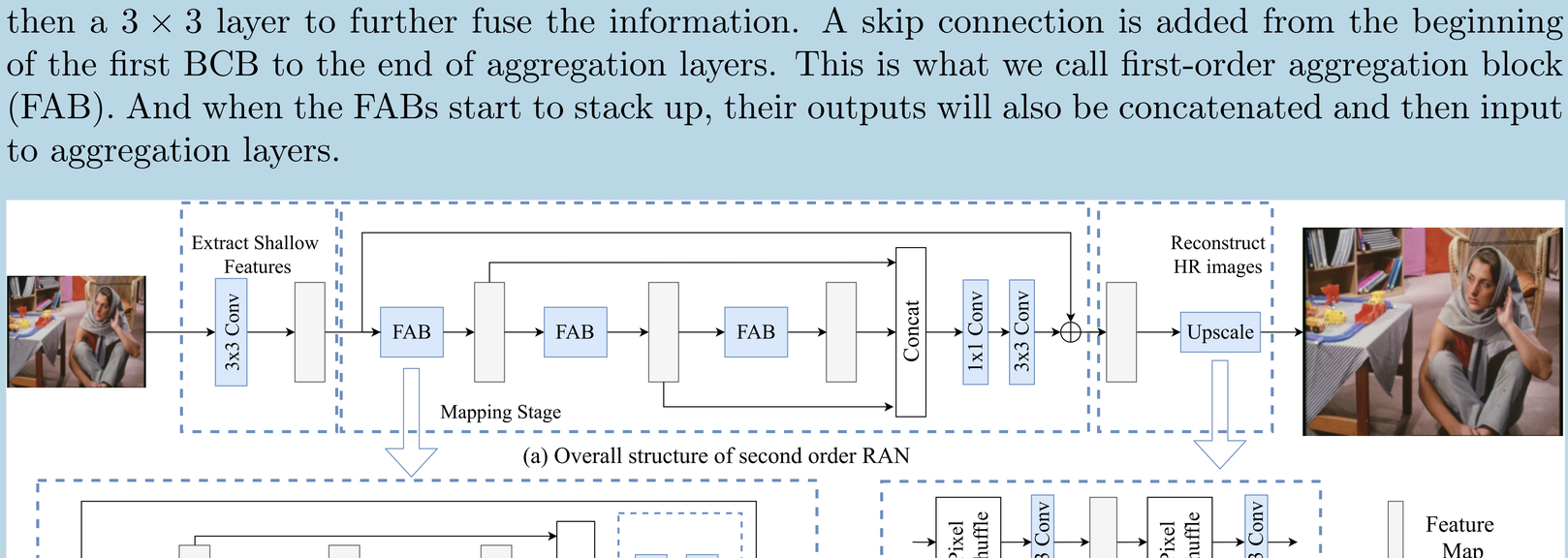
[5] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.

Acknowledgements

This work is jointly supported by National Key Research and Development Program of China (2016YFB1001000), Key Research Program of Frontier Sciences, CAS (ZDBS-LY-JSC032), Shandong Provincial Key Research and Development Program (2019JZZY010119), and CAS-AIR.

Recursive Aggregation

We construct the model starting from a basic convolutional block (BCB), i.e. the blue boxes in the figure. Every time a small number of BCBs are stacked, we concatenate the outputs as green boxes. The followed aggregation layers (yellow boxes) will further fuse their in- formation. And when the aggregated layers are stacked up, we also concatenate them by a second-order aggregation. In the same way, we can construct models with different orders by recursive aggregation. The gradients (pink solid arrows) are generated from the top the network, and the intensive shortcuts in the network can help gradients better conducted to all inner layers and make the networks more sufficiently optimized.



Network Instantiation

The details of a second-order RAN is shown below. We use a 3×3 convolutional layer to extract the shallow features, and then use PixelShuffle [4] layers to up-sample features to desired spatial sizes. For $\times 2$ and $\times 3$, a single PixelShuffle layer is used, and for $\times 4$, two $\times 2$ PixelShuffle layers are used. We mainly describe the mapping stage. As shown in (b), when several (3 in the figure) BCBs are sequentially stacked, their outputs will be concatenated and then input to aggregation layers. Inside the aggregation layer, there is first a 1×1 convolutional layer to reduce the number of channels, and then a 3×3 layer to further fuse the information. A skip connection is added from the beginning of the first BCB to the end of aggregation layers. This is what we call first-order aggregation block (FAB). And when the FABs start to stack up, their outputs will also be concatenated and then input to aggregation layers.

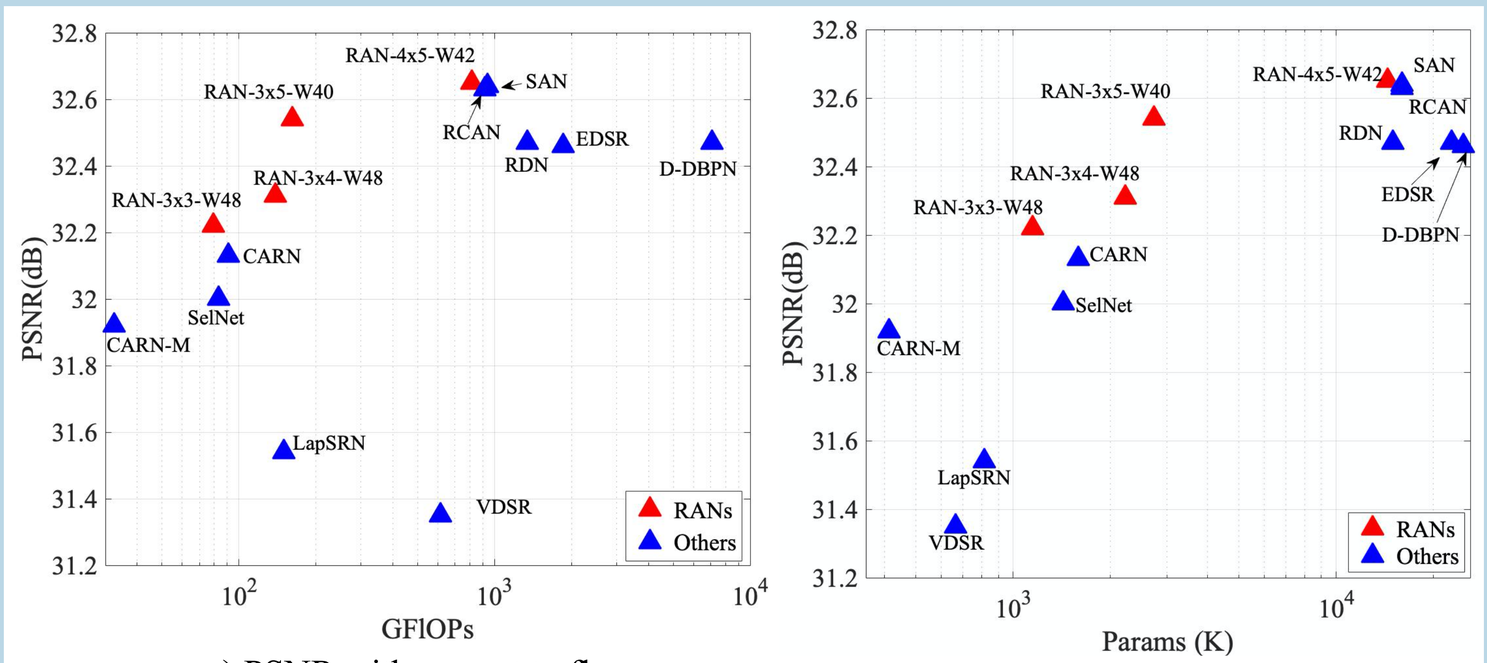
(a) Overall structure of second order RAN

(b) Structure of First-order Aggreagtion Block (FAB)

(c) Structure of Reconstruction Block

Efficiency Comparison

To intuitively compare the overall efficiency, we plot the PSNR results on Set5 [1] of different models with respect to their Flops and number of parameters. We compare with the state-of0-the-art models such as RCAN [2], RDN [5]. As one can see, RANs take up the left-up corner of the two diagrams. It indicates that RAN can achieve relatively better performance while with relative smaller model size and computational cost.



a) PSNR with respect to flops b) PSNR with respect to Params