

SELF-ORGANIZED RESIDUAL BLOCKS FOR IMAGE SUPER-RESOLUTION

Onur Keleş* A. Murat Tekalp* Junaid Malik† Serkan Kiranyaz‡

*Dept. of Electrical & Electronics Eng., Koç University, 34450 İstanbul, Turkey

†Tampere University, Tampere, Finland

‡Qatar University, Doha, Qatar

ABSTRACT

It has become a standard practice to use the convolutional networks (ConvNet) with RELU non-linearity in image restoration and super-resolution (SR). Although the universal approximation theorem states that a multi-layer neural network can approximate any non-linear function with the desired precision, it does not reveal the best network architecture to do so. Recently, operational neural networks (ONNs) that choose the best non-linearity from a set of alternatives, and their “self-organized” variants (Self-ONN) that approximate any non-linearity via Taylor series have been proposed to address the well-known limitations and drawbacks of conventional ConvNets such as network homogeneity using only the McCulloch-Pitts neuron model. In this paper, we propose the concept of self-organized operational residual (SOR) blocks, and present hybrid network architectures combining regular residual and SOR blocks to strike a balance between the benefits of stronger non-linearity and the overall number of parameters. The experimental results demonstrate that the proposed architectures yield performance improvements in both PSNR and perceptual metrics.

Index Terms— Convolutional networks, self-organized networks, operational neural networks, generative neurons, Taylor/Maclaurin series, hybrid networks, super-resolution

1. INTRODUCTION

The widely popular convolutional networks (ConvNet) [1] are built using the McCulloch-Pitts neuron model with the commonly used rectified linear unit (RELU) activation. The universal approximation theorem (UAT) [2, 3] states that any non-linear function can be approximated by a multi-layer network with the desired accuracy. However, it is only an existence theorem and does not provide network architectures with performance guarantees. Since the common neuron model performs only linear transformations with a simple non-linear activation, many such neurons may be needed for a sufficiently good approximation, and an architecture with

a given number of layers and channels may not satisfy the performance requirements of a particular problem at hand.

To boost the learning capability with more powerful network models, operational neural networks (ONN) [4], based on generalized operational perceptron (GOP) [5–9], and self-organized ONNs (Self-ONNs) [10, 11] based on generative neuron model have recently been proposed. The main idea in both models is to make the network learn the best non-linear kernel transformation. While conventional ONNs have to search for the best transformation from a pre-determined operator set library, Self-ONNs can approximate the best kernel transformation by a Taylor series expansion of the desired order. It has been noted that exhaustive search makes the use of ONNs expensive [4, 10]; hence, Self-ONNs have become a more promising and computationally efficient network architecture.

In this paper, we explore the use of self-organized layers for image super-resolution (SR) and compare its performance with the popular EDSR residual ConvNet architecture [12]. The rest of the paper is organized as follows: We review EDSR and Self-ONN architectures in Section 2. We introduce self-organized residual (SOR) blocks and network architectures with SOR blocks in Section 3. In Section 4, we present the experimental results over the benchmark dataset. Section 5 concludes the paper.

2. RELATED WORK

Self-ONN is a recent network architecture [10, 11]. In the following, we briefly review the popular convolutional residual network architectures for image SR and the foundations of Self-ONN, which constitute the background for paper.

2.1. Convolutional Super-Resolution Networks

The most popular convolutional single image SR architectures are the enhanced deep super-resolution (EDSR) [12] and residual channel attention networks (RCAN) [13]. The baseline EDSR model uses 16 residual blocks [14] with 64 channels each, and a pixelshuffler layer [15] for upsampling. RCAN uses a residual in residual (RIR) structure, which

This work was supported by TUBITAK projects 217E033 and 120C156, and a grant from Turkish Is Bank to KUIS AI Center. A. M. Tekalp also acknowledges support from Turkish Academy of Sciences (TUBA).

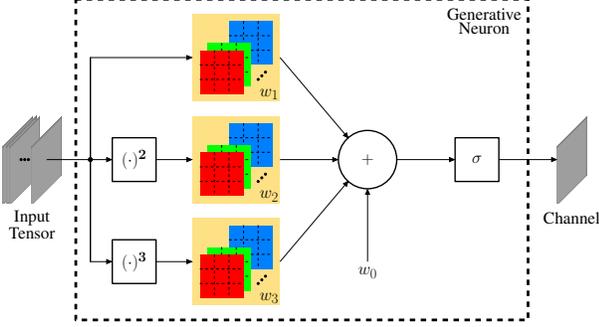


Fig. 1: Illustration of a generative neuron for $q = 3$.

consists of several residual groups with long skip connections. RIR allows low-frequency information to be bypassed through skip connections, making the network focus on learning high-frequency information. Although RCAN yields slightly better results compared to EDSR, in our hybrid network we employ EDSR residual blocks due to its simplicity.

2.2. Generative Neurons and Self Organized Networks

A Self-ONN layer is formed by generative neurons. A generative neuron approximates a non-linear function $f(x)$ by a Taylor series expansion

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n \quad (1)$$

around the point a . If we truncate the series to q terms, we have the approximation $g(\mathbf{w}, x, a)$ given by

$$g(\mathbf{w}, x, a) = w_0 + w_1(x - a) + \dots + w_q(x - a)^q \quad (2)$$

where

$$w_n = \frac{f^{(n)}(a)}{n!}. \quad (3)$$

For a c -channel input tensor, the parameters $w_n, n = 1, \dots, q$ denote q banks of c -channel convolution kernels and w_0 denotes a bias. These parameters can be learned by the classical back-propagation algorithm.

A generative neuron with 3×3 kernels, $a = 0$, $q = 3$, and activation function $\sigma(\cdot)$ is illustrated in Figure 1. Each neuron takes c -channels as input and outputs a single channel. The activation function limits outputs within a range about the value a before they are input to the next neuron, since the Taylor series is expanded around a . So, for $a = 0.5$, $\sigma(\cdot)$ can be taken as sigmoid bounding the output in the range $[0 \ 1]$, or if $a = 0$, $\sigma(\cdot)$ can be $\tanh(x)$ to bound the outputs in the range $[-1 \ 1]$. We note that if we choose $q = 1$ and $a = 0$, the generative neuron model reduces to the classic convolutional neuron.

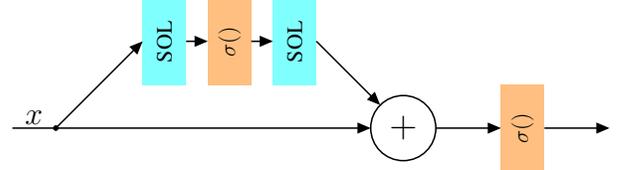


Fig. 2: Illustration of SOR block.

3. NETWORKS WITH SELF-ORGANIZED RESIDUAL BLOCKS

We first introduce self-organized residual blocks in Section 3.1. We then present network architectures for image SR with self-organized residual blocks in Section 3.2. Details of training procedures are discussed in Section 3.3.

3.1. Self-Organized Residual Blocks

A self-organized residual (SOR) block can be obtained by replacing all regular convolutional layers in a residual block by Self-ONN layers. In analogy with the EDSR residual blocks, we define a Self-ONN layer (SOL) without the activation function $\sigma(\cdot)$ in the generative neuron to replace standard convolutional layers and use the activation function $\sigma(\cdot)$ as a separate layer. Figure 2 depicts a SOR block consisting of SOL, activation function and SOL, and another activation function after the summation.

Likewise, using SOR blocks in place of regular convolutional residual blocks, any ConvNet architecture with residual blocks, e.g., RCAN, can be transformed into a self-organized residual network architecture. The main advantage of SOR blocks over standard residual blocks is that we can obtain better performance with a fewer number of blocks, which will be demonstrated in Section 4.

3.2. Self-Organized Residual Network Architectures

We choose the EDSR baseline model, which has 16 residual blocks with 64 channels (McCulloch-Pitts neurons) each, depicted in Fig 3, as our starting architecture. In the following, we propose two alternative network architectures for super-resolution (SR), one using only SOR blocks and another using a combination of regular residual blocks and SOR blocks in order to evaluate their performance compared with that of the EDSR network.

3.2.1. Self-ONN with the EDSR Architecture

We propose to replace all EDSR residual blocks with SOR blocks, and the input and output convolution layers, including those in the upsampling layer, with self-organized layers. The main advantage of this Self-ONN is using less number of blocks than the original EDSR. The performance of the Self-ONN for different number of SOR blocks and neurons per SOL is investigated in Table 1.

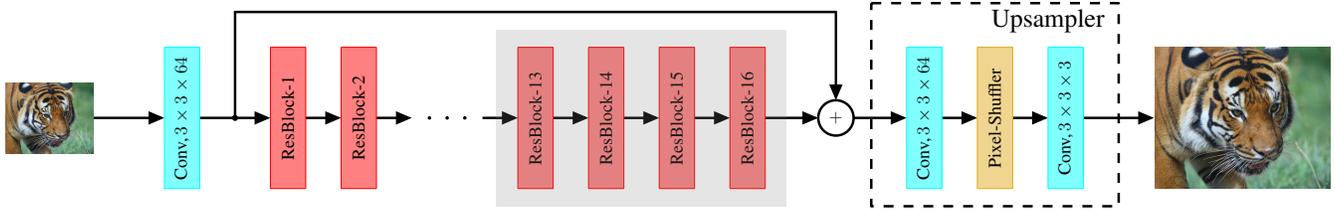


Fig. 3: The EDSR network architecture. We replace only the residual blocks in the gray box with SOR blocks in the hybrid architecture. We also replace the regular convolutional layers in the Upsampler with self-organizing layers.

3.2.2. Hybrid Network with EDSR and SOR Blocks

We next design a hybrid network with both EDSR and SOR blocks to evaluate whether combining some regular residual blocks and SOR blocks would improve performance. For this purpose, we replaced 4 of the EDSR residual blocks in Figure 3 with SOR blocks. The evaluation of the effect of which blocks to replace is shown in Table 2.

3.3. Training Details

We use the DIV2K dataset with 800 training images as our training set [16]. We normalize input images to the range $[0, 1]$ and then subtract the mean of R, G and B channels of all images in the training set from each R, G and B image, respectively, during training and tests [12].

The training is iteration based, and we use the same procedure for all architectures evaluated: We take 48×48 patches from low-resolution images and their corresponding regions in high-resolution images similar to [12]. The mini-batch size is 16. At each iteration, we select 16 images and locations of the patches within images randomly. In addition, each patch is subject to a random rotation of 0, 90, 180 and 270 degrees for data augmentation.

We minimize l_1 loss for 300K iterations using Adam optimizer [17] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate for EDSR (baseline) is 10^{-4} , and is halved after 200K iterations. For Self-ONNs and hybrid (EDSR + SOLs) architectures, the initial learning rate is 1.5×10^{-4} , and it is halved after each 100K iterations.

We train $\times 2$ SR networks with random initialization, while $\times 4$ SR networks are trained in two alternative ways: random initialization (Random), and fine-tuning a pre-trained (Pre-T) $\times 2$ SR network, except for the upsampling layer, which is randomly initialized.

4. EVALUATION

Evaluations are performed on 100 validation images from the DIV2K dataset for both $\times 2$ and $\times 4$ SR with bicubic downsampling.

4.1. Evaluation of Self-ONN Hyperparameters

We first evaluate the effect of the number of SOR blocks and number of neurons (channels) in each SOL on the performance of Self-ONN with a base configuration, where $q = 3$ for the SOR blocks and the upsampling/output layers.

Table 1: Self-ONN performance by hyperparameters.

Blocks (chan)	4 (32)	4 (64)	8 (32)	8 (64)
PSNR	34.148	34.424	34.377	34.616
SSIM	0.9608	0.9624	0.9621	0.9635
Parameters	365,059	1,448,963	586,499	2,334,311

The quantitative performance is measured in terms of PSNR and SSIM metrics. The test results with 4 or 8 SOR blocks and 32 or 64 neurons per layer are presented in Table 1. Clearly, the model with more SOR blocks yields better performance. The performance also increases with increasing number of neurons when the number of blocks are the same.

4.2. Evaluation of Hybrid Architecture

In the hybrid architecture, we consider replacing some residual blocks and all upsampling/output layers in the EDSR architecture with the proposed SOR blocks and SOL, respectively. To this effect, we conduct experiments to test whether to replace the first or last 4 residual blocks with SOR blocks ($q = 3$) and also whether to replace the upsampling/output layers with SOL ($q = 3$) or not in the case of $\times 2$ SR.

Table 2: Searching for the best hybrid network architecture.

SOR blocks	First 4		Last 4	
	Conv	SOL	Conv	SOL
UpS/Out layers				
PSNR	34.565	34.612	34.631	34.658
SSIM	0.9632	0.9635	0.9636	0.9638

The results in Table 2 indicate that replacing the last 4 blocks with SOR blocks and replacing the upsampling and output layers with SOL yield the best results.

4.3. Comparative Results

Based on the results in Tables 1 and 2, we compare the performances of i) Self-ONN with 8 SOR blocks and using SOL in

Table 3: Quantitative performance evaluation of the proposed models.

Method \ Metric	EDSR (baseline)			Self-ONN (8 SOR-64 + SOL UpS)			EDSR-12 + 4 SOR + SOL UpS		
	×2	×4		×2	×4		×2	×4	
		Random	Pre-T		Random	Pre-T		Random	Pre-T
PSNR ↑	34.557	28.924	28.970	34.616	28.983	29.016	34.658	29.018	29.064
SSIM ↑	0.9632	0.8857	0.8866	0.9635	0.8871	0.8875	0.9638	0.8877	0.8886
LPIPS ↓	0.0896	0.2775	0.2747	0.0873	0.2742	0.2714	0.0890	0.2726	0.2694

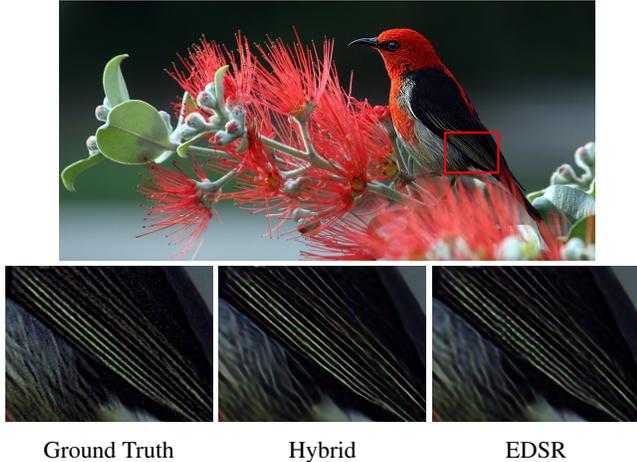


Fig. 4: Visual comparison of the EDSR baseline and hybrid network architecture with fine-tuned pre-trained model.

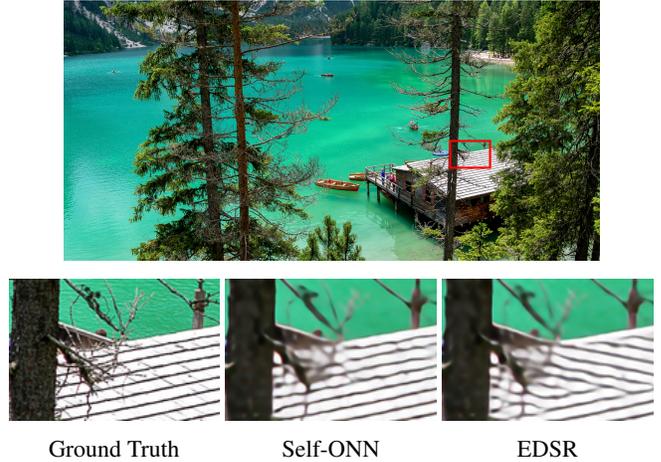


Fig. 5: Visual comparison of EDSR baseline and Self-ONN trained with random initialization for ×4 SR.

upsampling and output layers, and ii) the hybrid architecture, where the last four blocks are SOR blocks and upsampling and output layers are SOL with that of the EDSR baseline model in Table 3. All residual/SOR blocks in all networks have 64 channels. We set $q = 3$ for all SOL layers in the SOR blocks and upsampling/output layers in order to keep the number of parameters in all models approximately the same. In this study, we trained the EDSR network without using the geometric ensemble strategy adopted in [12].

We compare the performance of the proposed models with that of the EDSR baseline model using PSNR and SSIM measures. We also present LPIPS scores [18], which recently have been shown to correlate with human visual preferences. The arrows (↑) and (↓) next to the measures indicate whether a high or low value shows better performance, respectively.

Inspection of Table 3 indicates that both pure and hybrid Self-ONN models outperform the EDSR model in all quantitative metrics and in visual quality. The PSNR difference can be as much as 0.1 dB. Example visual comparison results in ×4 SR with randomly initialized and pre-trained models are shown in Figure 5 and Figure 4, respectively. We observe that the hybrid model, where we employ 12 regular residual and 4 SOR blocks and SOL for the upsampler layers, outperforms the pure Self-ONN with 8 SOR blocks although both model have approximately the same number (2.3 million for x2 SR and 2.7 million for x4 SR) of parameters.

5. CONCLUSION

Self-ONNs achieve the ultimate level of network heterogeneity and expressive power whilst maximizing the network diversity along with computational efficiency, thanks to the generative neurons that have the ability to adapt the desired transformation function for each connection during training. In this paper, we propose pure and hybrid Self-ONN architectures with self-organized residual (SOR) blocks, which are composed of self-organized layers (SOL), and show that they outperform conventional convolutional networks in the single image super-resolution task.

While a SOL of order q has approximately q times more learnable parameters compared to a conventional convolutional block, it is important to note that a Self-ONN with 8 layers already exceeds the performance of EDSR with 16 layers. Since computations for each layer can be fully parallelized, the execution time of a network is proportional to the number of layers. Hence, Self-ONNs can achieve better performance faster than ConvNets.

Our experiments demonstrate that both pure and hybrid Self-ONNs exceed the state-of-the-art SR performance of the popular EDSR network. The results also show that the hybrid model with 12 regular residual blocks and 4 SOR blocks and SOL for upsampler layers outperforms the pure Self-ONN model with 8 SOR blocks, where both models have approximately the same number of parameters.

6. REFERENCES

- [1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [2] Ken-Ichi Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural networks*, vol. 2, no. 3, pp. 183–192, 1989.
- [3] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al., “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [4] Serkan Kiranyaz, Turker Ince, Alexandros Iosifidis, and Moncef Gabbouj, “Operational neural networks,” *Neural Computing and Applications*, pp. 1–24, 2020.
- [5] Serkan Kiranyaz, Turker Ince, Alexandros Iosifidis, and Moncef Gabbouj, “Progressive operational perceptrons,” *Neurocomputing*, vol. 224, pp. 142–154, 2017.
- [6] Serkan Kiranyaz, Turker Ince, Alexandros Iosifidis, and Moncef Gabbouj, “Generalized model of biological neural networks: progressive operational perceptrons,” in *Int. Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2477–2485.
- [7] Dat Thanh Tran, Serkan Kiranyaz, Moncef Gabbouj, and Alexandros Iosifidis, “Progressive operational perceptron with memory,” *arXiv*, pp. arXiv–1808, 2018.
- [8] Dat Thanh Tran, Serkan Kiranyaz, Moncef Gabbouj, and Alexandros Iosifidis, “Heterogeneous multilayer generalized operational perceptron,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 710–724, 2019.
- [9] Dat Thanh Tran, Serkan Kiranyaz, Moncef Gabbouj, and Alexandros Iosifidis, “Knowledge transfer for face verification using heterogeneous generalized operational perceptrons,” in *IEEE Int. Conf. on Image Processing (ICIP)*, 2019, pp. 1168–1172.
- [10] Serkan Kiranyaz, Junaid Malik, Habib Ben Abdallah, Turker Ince, Alexandros Iosifidis, and Moncef Gabbouj, “Self-organized operational neural networks with generative neurons,” *arXiv preprint arXiv:2004.11778*, 2020.
- [11] Junaid Malik, Serkan Kiranyaz, and Moncef Gabbouj, “Self-organized operational neural networks for severe image restoration problems,” *Neural Networks*, vol. 135, pp. 201 – 211, 2021.
- [12] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.
- [13] Y. Zhang, K. Li, Kai Li, L. Wang, B. Zhong, and Yun Fu, “Image super-resolution using very deep residual channel attention networks,” in *IEEE/CVF European Conf. Comp. Vision (ECCV)*, 2018.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [15] 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 sub-pixel convolutional neural network,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883.
- [16] Eirikur Agustsson and Radu Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [17] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in *Int. Conf. on Learning Representations (ICLR)*, 2014.
- [18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.