A comparative study on wavelets and residuals in deep super resolution

Ruofan Zhou*, Fayez Lahoud*, Majed El Helou and Sabine Süsstrunk
School of Computer and Communication Sciences, EPFL, Switzerland

Abstract

Despite the advances in single-image super resolution using deep convolutional networks, the main problem remains unsolved: recovering fine texture details. Recent works in super resolution aim at modifying the training of neural networks to enable the recovery of these details. Among the different method proposed, wavelet decomposition are used as inputs to super resolution networks to provide structural information about the image. Residual connections may also link different network layers to help propagate high frequencies. We review and compare the usage of wavelets and residuals in training super resolution neural networks. We show that residual connections are key in improving the performance of deep super resolution networks. We also show that there is no statistically significant performance difference between spatial and wavelet inputs. Finally, we propose a new super resolution architecture that saves memory costs while still using residual connections, and performing comparably to the current state of the art.

Keywords: *super resolution, deep learning, wavelet decomposition, residual learning*

Introduction

Single-image super resolution is the process of obtaining a high-resolution (HR) image from a low-resolution (LR) sample. Super resolution has been used in many machine vision and image processing applications such as medical imaging [1], remote sensing [2], satellite imaging [3], etc. Super resolution is an inverse problem involving the recovery of missing high frequency content from LR images. It is an ill-posed problem since there are multiple reconstructions that could lead to the same low-resolution observation.

Some conventional super resolution methods utilize aliased low-resolution images to reconstruct high-resolution results [4]. The wavelet representation of an image allows the exploitation of both aliasing information and the self-similarities between local neighboring regions. Multiple algorithms leveraged this property to construct super resolved images based on wavelet decompositions [5, 6]. More recently, convolutional neural networks (CNN) have shown superior performance in many image restoration tasks. CNN-based super resolution methods were introduced in [7, 8] to model a mapping from LR patches to HR patches in the spatial domain.

Among deep learning methods, residual learning techniques detain the state-of-the-art performance in multiple image restoration problems [9, 10]. The residual mapping corresponding to the degradation model is easier to learn than the natural image manifold. Residual networks thus increase the accuracy and reduce the training time of image restoration neural networks, particularly super resolution networks. However, residual connections force all feature maps in the network to be as large as the input, thus inducing large memory and computational costs on the network.

Other deep learning techniques [11,12], inspired by the original use of wavelets for super resolution, build networks to predict HR wavelet decompositions from their corresponding LR signals. They state that using wavelets as inputs to neural networks promotes sparsity in the intermediate representations and provides more structural information about the image. Therefore, the claim is that wavelet inputs help deep network architectures reconstruct better HR results with less artifacts.

We investigate the impact of both residual learning and wavelet decompositions on the performance of super resolution networks in order to understand how they can improve the results. We hypothesize that the residual learning has a much larger effect on the results than the wavelet-domain learning. We build a setup to compare the same architecture trained with variations of residual connections and spatial and wavelet inputs. We show that wavelet representations do not lead to any significant performance improvement in super resolution networks. However, wavelet input is half the size of spatial input, so they reduce the memory requirements of the network at inference, as they require smaller intermediate feature maps.

Finally, we show that the residual connections are the main factor in improving the performance of super resolution networks. However, they require intermediate feature maps to have the same size as the input, thus demanding large memory at inference. We propose an architecture that benefits from the residual connections' ability to propagate high frequencies without forcing all the feature maps in the network to remain as large as the input. We base our model on PixelShuffle [13] ($\mathcal{PS}$), an upsampling method that takes N LR channels and outputs 1 HR channel. We build residual blocks that downsample their inputs using convolutions before upsampling them with $\mathcal{PS}$, achieving residual connections with smaller intermediate feature maps. Using this new module, we propose a new architecture that takes advantage of residual connections without suffering from the memory costs typically incurred by the conventional connection model.

Related Work

Typically, a network takes as input a bicubic-interpolated, upsampled image, and learns to restore the high frequencies found in the real image [7, 8].

Multiple architectures have been proposed to super resolve images using deep neural networks. Most commonly, the use of residuals has been proposed to super resolve natural images. In

*Both authors contribute equally to this work.

some methods [14, 15], residual learning is used to link the input image to the output layer, enabling the network to learn the degradation model, which is simpler than the natural images model. In [15, 16], residual blocks were proposed to propagate the high frequency details through deep neural network architectures.

Recently, neural networks have been proposed that learn to super resolve the wavelet decompositions of an image instead of its spatial form. Works such as [11] apply it for general image super resolution while [12] specifically applies for super resolving faces. Wavelet methods allow the networks to have sparser intermediate representations and help recovering details at different levels, thus claiming to achieve better super resolved images.

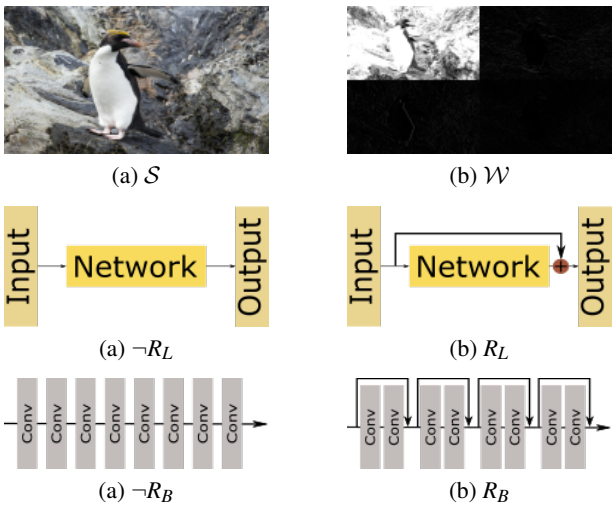
Residuals and Wavelets

Our goal is to compare multiple techniques used in super resolution neural networks. In order to obtain a fair comparison, we set up a single network architecture with eight different variations. The versions are based on combinations of three parameters. The first parameter controls whether the network obtains a spatial $\mathcal{S}$ or wavelet $\mathcal{W}$ representation of an image as input. The second parameter controls whether the network is trained using a residual connection R_L from the input image to the output or not (denoted by $\neg R_L$). Finally, the third parameter controls whether the network contains residual blocks R_B or regular convolutional blocks (denoted by $\neg R_B$). The notation of a network configuration is a triplet indicating which techniques were used to train it. For example, the triplet $(\mathcal{W}, R_L, \neg R_B)$ shows that the network has been trained with wavelet inputs, a residual connection from the input to the output and regular convolutional blocks. Fig. 1 illustrates, on each row, the two variations for each parameter.

All networks consists of 12 convolutional layers, having 64 convolutional kernel of size 3×3 except the last channel which has the same number of 3×3 kernels as the number of output channels in the input. We train all the networks using a single dataset: DIV2K [17]. It contains 800 high-resolution images and their corresponding low-resolution images with 2x, 3x, and 4x downsampling factors. We validate on another set of 100 images provided by DIV2K. Similar to [11], we use the Haar wavelet transform to generate wavelet inputs. The decompositions are generated in each color channel separately. We use the same training strategy for each network. We use L2 loss for all networks and Adam optimizer for training. The initial learning rate is set to 0.001 and is decayed by a factor of 10 after every 30 epochs. The training uses batches of size 64×64. All the networks are trained for 100 epochs and tested on the epoch that records the smallest loss on the validation set. Finally, the networks are initialized using Xavier [18], we set the same seed for all the networks to reduce variations due to random starts.

Set	Scale	Bicubic	$(\mathcal{S}, \neg R_L, \neg R_B)$	$(\mathcal{S}, \neg R_L, R_B)$	$(\mathcal{S}, R_L, \neg R_B)$	$(\mathcal{S}, R_L, R_B)$	$(\mathcal{W}, \neg R_L, \neg R_B)$	$(\mathcal{W}, \neg R_L, R_B)$	$(\mathcal{W}, R_L, \neg R_B)$	$(\mathcal{W}, R_L, R_B)$
Set5	×2	31.79	34.52	34.94	**34.99**	34.80	34.42	34.89	34.84	34.80
		0.917	0.942	0.945	**0.946**	0.945	0.940	0.945	0.945	0.944
	×3	26.95	27.77	27.99	**28.02**	27.99	27.80	27.95	27.96	28.00
		0.818	0.842	0.846	**0.847**	0.846	0.842	0.846	0.845	0.846
	×4	26.69	28.43	28.81	28.89	28.88	28.75	28.85	**28.94**	28.93
		0.789	0.838	0.850	0.851	0.851	0.845	0.851	**0.853**	**0.853**
Set14	×2	28.00	29.36	29.58	**29.62**	29.51	29.23	29.57	29.51	29.54
		0.849	0.878	0.882	**0.883**	0.881	0.875	0.882	0.880	0.880
	×3	24.44	24.58	**24.66**	24.64	24.64	24.58	24.61	24.62	24.64
		0.725	0.747	0.750	**0.751**	**0.751**	0.746	0.749	0.749	0.749
	×4	23.81	24.47	24.64	24.66	24.67	24.57	24.70	**24.74**	24.69
		0.673	0.706	**0.714**	**0.714**	**0.714**	0.709	**0.714**	**0.714**	**0.714**
BSDS100	×2	26.11	25.93	25.99	25.89	25.91	26.25	**26.46**	26.35	26.33
		0.785	0.801	0.800	0.800	0.800	0.811	**0.813**	0.809	0.810
	×3	24.66	24.72	**24.73**	**24.73**	24.70	24.71	24.70	24.70	24.70
		0.693	0.720	**0.722**	**0.722**	**0.722**	0.720	**0.722**	0.721	**0.722**
	×4	22.38	21.91	21.86	**23.17**	21.82	21.89	21.98	21.93	21.93
		0.566	0.568	0.567	**0.570**	0.568	0.567	**0.570**	0.568	**0.570**
Urban100	×2	25.43	28.25	28.65	**28.67**	28.51	27.96	28.51	28.42	28.43
		0.838	0.898	0.904	**0.905**	0.902	0.891	0.902	0.900	0.900
	×3	**21.30**	21.13	21.13	21.14	21.10	21.09	21.06	21.06	21.07
		0.673	0.698	0.701	**0.702**	0.701	0.697	0.700	0.699	0.700
	×4	21.70	22.93	23.24	23.22	23.24	23.13	23.28	23.28	**23.30**
		0.652	0.715	0.731	0.728	0.729	0.724	**0.732**	0.731	**0.732**
Manga109	×2	26.79	27.22	27.47	27.38	27.35	27.50	27.87	27.87	**27.90**
		0.899	0.915	0.920	0.920	0.918	0.911	**0.923**	0.922	0.922
	×3	24.61	25.99	26.21	**26.31**	26.23	26.04	26.18	26.19	26.20
		0.829	0.867	0.877	**0.878**	0.877	0.866	0.876	0.875	0.875
	×4	22.05	22.28	22.37	22.45	22.35	22.36	**22.62**	22.46	22.47
		0.742	0.765	0.778	**0.779**	0.778	0.768	0.778	0.772	0.776

Table 1. PSNR and SSIM results on public test sets. Bold red indicates the best performance. Brown indicates statistical significance between the model and the best performance, and black indicates no statistical significance.



(a) $\mathcal{S}$ (b) $\mathcal{W}$

(a) $\neg R_L$ (b) R_L

(a) $\neg R_B$ (b) R_B

Figure 1. *Training variants for wavelet and residual comparison*

Image Quality

For evaluating and comparing different networks, we use five datasets, namely: BSDS100 [19], Set5 [20], Set14 [21], Urban100 [22], and Manga109 [23]. Set5, Set14, and BSDS100 consist of natural scenes; Urban100 is more challenging as it contains a larger amount of details in urban areas (e.g., high-rise buildings with multiple small windows); and Manga109 is a dataset of Japanese manga consisting of synthetic non-natural images with fewer high frequency details than natural images. They all have as many images as their name indicates. All images are downsampled and upsampled using MATLAB's bicubic interpolation implementation. We evaluate the super resolved images with two commonly used image quality metrics: PSNR and SSIM [24]. Table 1 shows the results of all the networks evaluated on the previously mentioned datasets. The brightness of red show, per row, the statistical difference in means relative to the highest performing architecture, with darker colors signifying a larger difference.

First, a series of t-tests were conducted to compare the bicubic output against all other architectures over all the datasets. The adopted t-test is the paired-sample (or dependent sample) t-test, as the objective is to evaluate the statistical significance of the difference in means of PSNR and SSIM. All but one tests concluded a significant difference between the bicubic upsampled images and the network outputs. The smallest difference observed was between bicubic and $(\mathcal{S}, \neg R_L, \neg R_B)$ with $t_{psnr} = 3.92, p_{psnr} = 1 \times 10^{-5}$ and $t_{ssim} = 4.98, p_{ssim} = 7 \times 10^{-7}$. Therefore, the row (Urban x3) where bicubic upsampling results have better PSNR values than networks outputs is not statistically significant.

Over all the datasets, networks trained without residual learning $\neg R_L$ or blocks $\neg R_B$ did not achieve any top performance in comparison to other networks. This shows that any sort of residual connections can improve the performance of super resolution neural networks irrelevant of the modality of the input.

The network with residual learning and no residual blocks $(\mathcal{S}, R_L, \neg R_B)$ achieves the best performance on the majority of the test datasets, both in PSNR and SSIM scores. Comparing architecture pairs that only differ with respect to the residual connection (R_L vs. $\neg R_L$), the t-test shows that residual connections have a statistically significant impact on the performance of super resolution neural networks; $t_{psnr} = 4.45, p_{psnr} = 5 \times 10^{-4}$ and $t_{ssim} = 7.11, p_{ssim} = 5 \times 10^{-6}$.

Additionally, comparing networks where only the input changes between $\mathcal{S}$ and $\mathcal{W}$, we find that the average performances across all datasets are very similar with $\overline{PSNR}_{\mathcal{W}} = 26.15$ and $\overline{PSNR}_{\mathcal{S}} = 26.09$, and $\overline{SSIM}_{\mathcal{W}} = 0.798$ and $\overline{SSIM}_{\mathcal{S}} = 0.797$. T-tests conducted on pairs of networks with different inputs show that there is no statistically significant difference between their results; $t_{psnr} = 1.91, p_{psnr} = 0.07$ and $t_{ssim} = 1.02, p_{ssim} = 0.31$.

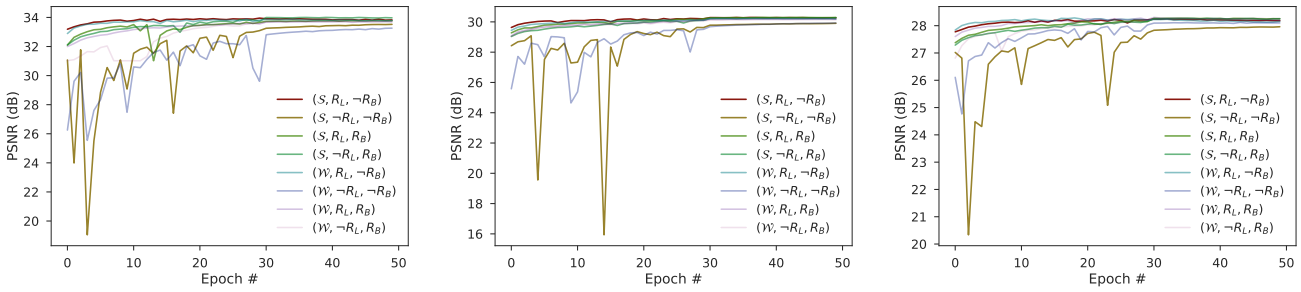
Furthermore, to illustrate the performance of each architecture, we present the average PSNR on the validation set in Fig. 2 per epoch. Networks with either a residual connection R_L or residual blocks R_B show superior performance in convergence and training stability. Inversely, there are instabilities in the training of networks without any residual learning due to the difficulty of learning the more complex manifold of natural high-resolution images. Comparing networks trained on spatial and wavelet inputs, we find that there is no significant difference in their evolution during training. Finally, in Fig. 3, we show visual results obtained with different interpolation techniques including all the different network architectures, for an upsampling scale factor of 4. The figure shows the reference image, the bicubic interpolation image and the output of the 8 different architectures, for two test images. Notice that all the networks trained with residual blocks, independently of other factors, show finer details in their reconstruction of the eyebrows in the first image.

Memory and Computational Complexity

Given that RGB and wavelet inputs show comparable performance, we evaluate memory and computational complexity of the different architectures with an NVIDIA Titan X GPU (12G Memory). Table 2 shows the memory usage and the average running time per image of the networks on the validation set, computed on 91 images with size of 1024×1024. Note that as the size of RGB inputs in width and height is twice larger than their wavelet representations, their corresponding feature maps occupy four times more memory than their spatial counterparts. Additionally, the smaller feature maps allow for faster convolutions and thus networks with wavelet inputs process images faster than those with spatial inputs. Finally, if we compare networks that differ only in R_B, we find that those with residual blocks are consistently slower at processing images due to the extra computational costs of adding features maps at the end of every residual block.

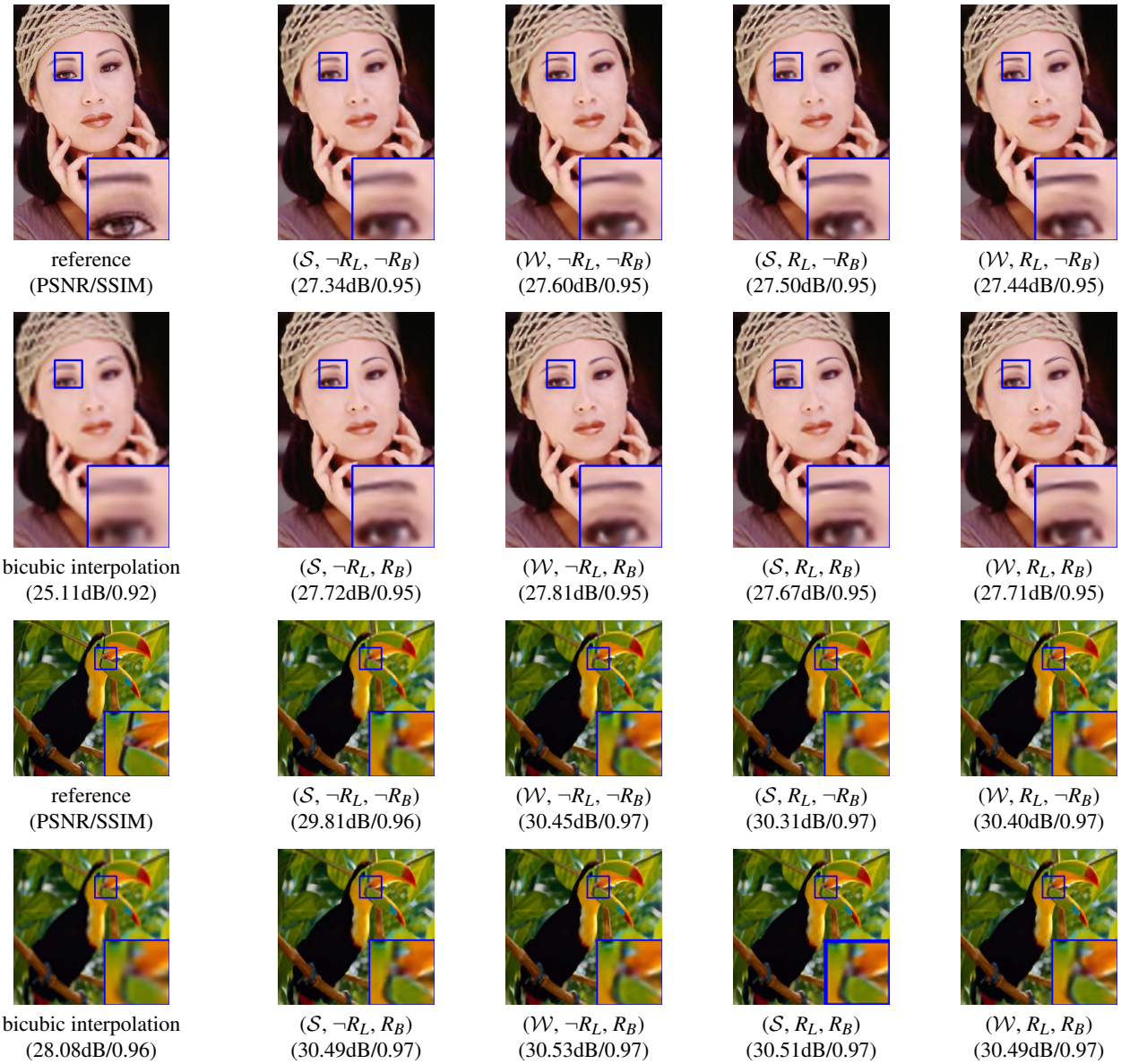
	Memory	**Running Time**
$(\mathcal{S}, \neg R_L, \neg R_B)$	5412MB	0.12s
$(\mathcal{S}, \neg R_L, R_B)$	5412MB	0.12s
$(\mathcal{S}, R_L, \neg R_B)$	5432MB	0.16s
$(\mathcal{S}, R_L, R_B)$	5432MB	0.16s
$(\mathcal{W}, \neg R_L, \neg R_B)$	1380MB	0.16s
$(\mathcal{W}, \neg R_L, R_B)$	1380MB	0.16s
$(\mathcal{W}, R_L, \neg R_B)$	1460MB	0.16s
$(\mathcal{W}, R_L, R_B)$	1460MB	0.16s

Table 2. Computational and memory complexity of different networks on the validation set. Image size 1024×1024



(a) Scale Factor 2 (b) Scale Factor 3 (c) Scale Factor 4

Figure 2. *Performance per epoch for all network architectures.*



reference (PSNR/SSIM)	$(\mathcal{S}, \neg R_L, \neg R_B)$ (27.34dB/0.95)	$(\mathcal{W}, \neg R_L, \neg R_B)$ (27.60dB/0.95)	$(\mathcal{S}, R_L, \neg R_B)$ (27.50dB/0.95)	$(\mathcal{W}, R_L, \neg R_B)$ (27.44dB/0.95)
bicubic interpolation (25.11dB/0.92)	$(\mathcal{S}, \neg R_L, R_B)$ (27.72dB/0.95)	$(\mathcal{W}, \neg R_L, R_B)$ (27.81dB/0.95)	$(\mathcal{S}, R_L, R_B)$ (27.67dB/0.95)	$(\mathcal{W}, R_L, R_B)$ (27.71dB/0.95)
reference (PSNR/SSIM)	$(\mathcal{S}, \neg R_L, \neg R_B)$ (29.81dB/0.96)	$(\mathcal{W}, \neg R_L, \neg R_B)$ (30.45dB/0.97)	$(\mathcal{S}, R_L, \neg R_B)$ (30.31dB/0.97)	$(\mathcal{W}, R_L, \neg R_B)$ (30.40dB/0.97)
bicubic interpolation (28.08dB/0.96)	$(\mathcal{S}, \neg R_L, R_B)$ (30.49dB/0.97)	$(\mathcal{W}, \neg R_L, R_B)$ (30.53dB/0.97)	$(\mathcal{S}, R_L, R_B)$ (30.51dB/0.97)	$(\mathcal{W}, R_L, R_B)$ (30.49dB/0.97)

Figure 3. *Qualitative comparison of different networks on a $\times 4$ super-resolution task. Better viewed on screen.*

PixelShuffle Residual Network

Typically, super resolution neural networks take images as input that have the same resolution than the output, therefore, intermediate convolution layers operate at HR scale. Given an upsampling scale r, the computational cost of applying a convolution on the HR image is r^2 times that of applying it on the LR image.

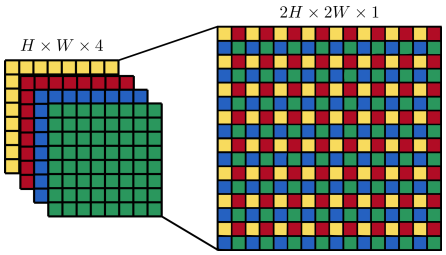


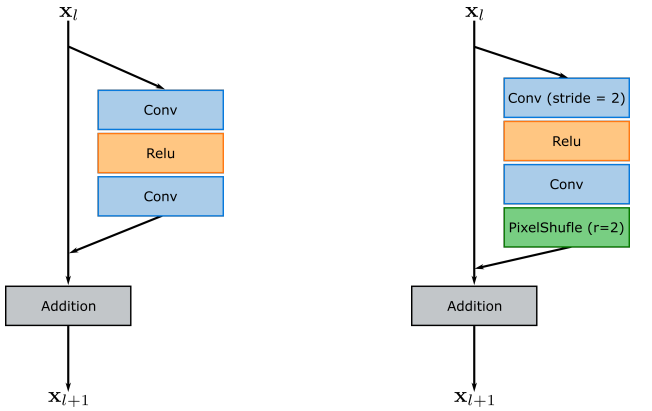
Figure 4. *$\mathcal{PS}$ upsampling for $r = 2$*

PixelShuffle [13] ($\mathcal{PS}$) is an upsampling method introduced to allow for efficient convolutions in LR followed by a pixel reshuffling that transfers information from the features dimension to the spatial dimensions. Prior to a $\mathcal{PS}$ layer, convolutions output LR feature maps of dimension $H \times W \times C \cdot r^2$. $\mathcal{PS}$ then reshuffles the pixels to obtain HR feature maps of shape $rH \times rW \times C$. An illustration of the upsampling done with $\mathcal{PS}$ is shown in Fig. 4. The $\mathcal{PS}$ layer can be modelled as

$$\mathcal{PS}(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C \cdot r \cdot \bmod (y,r) + C \cdot \bmod (x,r) + c} \qquad (1)$$

where T is the input feature map and x, y, and c are the pixel indices in T.

We propose to use a combination of convolutions with stride $r > 1$ and $\mathcal{PS}$ in the residual blocks to alleviate the computational and memory complexity. In Fig. 5 we compare the original residual blocks and the building blocks of our proposed ShuffleNet. In both illustrations, $\mathbf{x}_l$ and $\mathbf{x}_{l+1}$ have similar width w, height h and number of channels c. In the original residual blocks, the intermediate feature maps have the same dimensions as $\mathbf{x}_l$. In our proposed block, the first convolution has stride $r > 1$. The intermediate layers will have a width of $\frac{w}{r}$ and a height of $\frac{h}{r}$. The second convolution will output r^2 channels with these reduced dimensions, which will be expanded by $\mathcal{PS}$ to match the size $\mathbf{x}_l$ before addition. In this block, the intermediate representations occupy less memory. We train ShuffleNet using the same training strategy as the previous networks. We compare the performance on the datasets with the best network architecture (R_L, $\neg R_B$) in both spatial and wavelet domain. The results are shown in Table 3. From the average performances access all datasets: $\overline{PSNR}_{\mathcal{S}}$(ShuffleNet)= 26.17, $\overline{PSNR}_{\mathcal{S}}$(($R_L$, $\neg R_B$))= 26.25, $\overline{PSNR}_{\mathcal{W}}$(ShuffleNet)= 26.09, $\overline{PSNR}_{\mathcal{W}}$(($R_L$, $\neg R_B$))= 26.19, we find that ShuffleNet achieves comparable results. Moreover, ShuffleNet is able to reduce the GPU memory usage. For super-resolving a 1024×1024 image, it requires only 3492MB for spatial image input or 900MB for wavelets representation input, while (R_L, $\neg R_B$) requires 5432MB and 1460MB, respectively. Thus we can effectively reduce the network size while still retaining the same performance by using ShuffleNet.



Original Residual Block Proposed ShuffleNet block

Figure 5. *Integrating $\mathcal{PS}$ in a residual block*

Dataset	Scale	Spatial		Wavelets	
		ShuffleNet	(R_L, $\neg R_B$)	ShuffleNet	(R_L, $\neg R_B$)
Set5	×2	34.92	**34.99**	34.92	34.84
	×3	27.94	**28.02**	27.89	27.96
	×4	28.74	28.89	28.69	**28.94**
Set14	×2	**29.79**	29.62	29.51	29.51
	×3	24.61	**24.64**	24.56	24.62
	×4	24.54	24.66	24.57	**24.74**
BSDS100	×2	25.80	25.89	26.18	**26.35**
	×3	**24.77**	24.73	24.67	24.70
	×4	22.85	**23.17**	22.04	21.93
Urban100	×2	28.55	**28.67**	27.82	28.42
	×3	21.03	**21.14**	20.99	21.06
	×4	23.07	23.22	23.09	**23.28**
Manga109	×2	27.34	27.38	27.71	**27.87**
	×3	26.18	**26.31**	26.18	26.19
	×4	22.40	22.45	**22.47**	22.46

Table 3. Performance comparison between ShuffleNet and ($\mathcal{S}$, R_L, $\neg R_B$). Bold red indicates the best performance. Brown indicates statistical significance between the model and the best performance, and black indicates no statistical significance.

Conclusion

We present a comparative study on wavelets and residuals in deep super resolution. We show that super resolution networks achieve a higher performance and converge faster using residual connections. Additionally, we don't find a significant impact between using spatial or wavelet inputs. However, wavelet inputs reduce the memory requirements of the network. Lastly, it is good to note that all current super resolution neural networks are trained on bicubic downsampling and upsampling. This does not actually fit a real camera model, and as such the networks are learning to reverse the degradation incurred on the image. This is also another explanation of the performance of the residual networks, as they are in fact trying to learn this degradation compared to non-residual networks who learn the natural image space. Finally, we propose a new network architecture for spatial image input based on PixelShuffle that reduces the memory requirements of residual blocks, and demonstrate empirically that it still achieves competitive super resolution results despite the memory reduction.

References

[1] Alistair N Boettiger, Bogdan Bintu, Jeffrey R Moffitt, Siyuan Wang, Brian J Beliveau, Geoffrey Fudenberg, Maxim Imakaev, Leonid A Mirny, Chao-ting Wu, and Xiaowei Zhuang, "Super-resolution imaging reveals distinct chromatin folding for different epigenetic states," *Nature*, vol. 529, no. 7586, pp. 418, 2016.

[2] Linyi Li, Yun Chen, Tingbao Xu, Rui Liu, Kaifang Shi, and Chang Huang, "Super-resolution mapping of wetland inundation from remote sensing imagery based on integration of back-propagation neural network and genetic algorithm," *Remote Sensing of Environment*, vol. 164, pp. 142–154, 2015.

[3] Matt W Thornton, Peter M Atkinson, and DA Holland, "Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping," *International Journal of Remote Sensing*, 2006.

[4] Patrick Vandewalle, Sabine Susstrunk, and Martin Vetterli, "Superresolution images reconstructed from aliased images," in *Visual Communications and Image Processing 2003*. International Society for Optics and Photonics, 2003, vol. 5150, pp. 1398–1406.

[5] Shubin Zhao, Hua Han, and Silong Peng, "Wavelet-domain hmt-based image super-resolution," in *In Proc. ICIP 2003*. IEEE, 2003.

[6] Hasan Demirel and Gholamreza Anbarjafari, "Image resolution enhancement by using discrete and stationary wavelet decomposition," *IEEE Transactions on Image Processing*, 2011.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, 2014.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[9] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks.," in *Proc. of the IEEE CVPR*, 2017, vol. 1, p. 3.

[11] Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, and Vishal Monga, "Deep wavelet prediction for image super-resolution," in *Proc. of the IEEE CVPR Workshops*, 2017.

[12] Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al., "Wavelet-SRNet: A wavelet-based CNN for multi-scale face super resolution," in *Proc. of the IEEE CVPR*, 2017, pp. 1689–1697.

[13] 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 *Proc. of the IEEE CVPR*, 2016, pp. 1874–1883.

[14] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. of the IEEE CVPR*, 2016, pp. 1646–1654.

[15] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. of the IEEE CVPR*, 2017, vol. 1, p. 4.

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network.," in *Proc. of the IEEE CVPR*, 2017, vol. 2, p. 4.

[17] Eirikur Agustsson and Radu Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. of the IEEE CVPR Workshops*, July 2017.

[18] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[19] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

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

[21] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[22] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. of the IEEE CVPR*, 2015, pp. 5197–5206.

[23] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21811–21838, 2017.

[24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Author Biography

Ruofan Zhou received her BEng in Computer Science and Technology from Tsinghua University in 2015. Since then, she is a Research Assistant in Computational Photography, pursuing a PhD degree in the Image and Visual Representation Laboratory, EPFL. Her main interests are Computer Vision, Image Processing, and Machine Learning.

Fayez Lahoud received his B.E. in Computer and Communication Engineering and his minor in Mathematics from the American University of Beirut, Lebanon. He received his M.S. in Computer Science at EPFL where he is currently pursuing his Ph.D. at the Image and Visual Representation Laboratory. His work focuses on the development of computational and visual tools to help firefighters accomplish their tasks more efficiently.

Majed El Helou received his B.E. in Computer and Communications Engineering and Mathematics minor from the American University of Beirut. He is currently pursuing his PhD in the Image and Visual Representation Lab, at EPFL. His research interests include computational photography, AI (signal/image processing, classical machine learning, deep learning), computer vision, and stochastic estimation theory.

Sabine Süsstrunk leads the Images and Visual Representation Lab (IVRL) at EPFL, Switzerland. Her research areas are in computational photography, color computer vision and color image processing, image quality, and computational aesthetics. She has published over 150 scientific papers, of which 7 have received best paper/demos awards, and holds 10 patents. She received the IS&T/SPIE 2013 Electronic Imaging Scientist of the Year Award and IS&Ts 2018 Raymond C. Bowman Award. She is a Fellow of IEEE and IS&T.