Advanced Super-Resolution using Lossless Pooling Convolutional Networks

Farzad Toutounchi, Ebroul Izquierdo
Multimedia and Vision Research Group
Queen Mary University of London

{f.toutounchi,e.izquierdo}@qmul.ac.uk

Abstract

In this paper, we present a novel deep learning-based approach for still image super-resolution, that unlike the mainstream models does not rely solely on the input low resolution image for high quality upsampling, and takes advantage of a set of artificially created auxiliary self-replicas of the input image that are incorporated in the neural network to create an enhanced and accurate upscaling scheme. Inclusion of the proposed lossless pooling layers, and the fusion of the input self-replicas enable the model to exploit the high correlation between multiple instances of the same content, and eventually result in significant improvements in the quality of the super-resolution, which is confirmed by extensive evaluations.

1. Introduction

Super-resolution (SR) has received significant attention in recent years, and the learning-based approaches in designing SR solutions have been providing promising results for spatial upsampling of still images and videos. In particular, application of deep learning and Convolutional Neural Networks (CNN) has become a trendy approach for SR, leading to major improvements in the performance of the state-of-the-art SR solutions.

Dong et al. [3, 4] presented the first concrete deep learning-based SR approach for still images, exploiting deep CNNs, capable of upsampling still images with high visual quality and reasonable complexity. Dong et al. improved their work further in [5] by introduction of transposed convolutional layers to their framework for upsampling from source to target resolution. Shi et al. [13] used a similar concept as [5] for reducing the complexity of the framework, except they introduced a sub-pixel upscaling layer instead of the transposed convolution layer, with which they achieved an efficient performance.

Other major contributions in still image SR using deep learning have been made by Kim et al. [7, 8]. In [7], the concept of residual learning was introduced to deep SR mod-
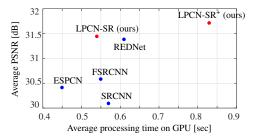

Figure 1. Performance comparison of the proposed SR approach with the state-of-the-art models on Set 5 images.

els, which led to quicker convergence of the network and high visual quality reconstruction. In [8], a deep and recursive architecture was employed, that achieved very good performance in terms of image quality, although expensive in terms of computation cost. In other deep learning-related efforts, Wang et al. [15] introduced a deep joint SR model to exploit both external and self-similarities for reconstruction, in which a stacked denoising convolutional auto-encoder was first trained on external training data, then it was fine-tuned with multi-scale self-examples from the input data.

Another interesting endeavor in devising deep learning models for image SR was done by Mao et al. [12]. They introduced a deep hourglass-shaped CNN that included multiple convolution and transposed convolution layers with symmetric skip connections and was designed originally for image denoising. However, the model was also applied for SR and image upscaling, and it was reported as one of the best SR models in the literature for high quality image upscaling. Ledig et al. [10] also made a contribution to SR by using a generative adversarial network for performing high quality SR on images, and they showed promising results in highly textured images, and high subjective quality, although the method falls behind the state-of-the-art approaches in terms of the objective metrics.

While major improvements are reported in the still image SR domain, video SR has also been an active research field, and several interesting contributions have been made in recent years by applying deep learning-based solutions to

multi-frame SR [2, 6, 14, 11]. In multi-frame SR, unlike the typical still image SR problem, several input frames that are highly correlated contribute in generation of one high quality upsampled target image, hence a better objective and subjective quality can be expected when compared to the single image SR. Inspired by this approach, we aimed at devising a model that utilizes the multi-frame concept within the still image frameworks for an enhanced SR experience, using artificially generated self-replicas of the input image.

In this paper, we propose a novel approach for still image SR which exploits the incorporation of several auxiliary downscaled versions of the input image in CNNs, that results in a significant improvement in the quality of the reconstructed output image (Figure 1). The followings summarize the main contributions of this paper:

- Introduction of lossless pooling layers that create several highly correlated downsampled replicas from the input image, that can fuse in the SR network.

- Designing a lossless pooling convolutional network SR (LPCN-SR) model, that only exploits the downsampled self-replicas, and can provide faster performance than the existing high quality approaches.

- Designing a lossless pooling convolutional network SR model that also incorporates the original input image in addition to the self-replicas (LPCN-SR⁺), and can outperform the existing approaches in terms of quality.

The rest of this paper is organized as follows. In Section 2, we introduce the novelties including the lossless pooling mechanism, as well as incorporation of the self-replicas in SR networks. Section 3 covers the technical specifications of the proposed CNN architecture and the proposed SR model. Section 4 summarizes the results of our evaluations and comparisons with state-of-the-art approaches, followed by conclusions in Section 5.

2. Proposed Approach

Our proposed approach is inspired by the multi-frame SR models that take several highly correlated low resolution inputs and generate a high resolution version of a target image. In order to extend this concept to still image SR,

we require multiple versions of the low resolution image as the input. Hence a mechanism is needed to create different replicas of the input image. We propose a particular pooling process which downsamples the input image to several lower resolution versions by reshuffling the pixel positions and without any information loss. The resulting replicas of the input image can be incorporated in a multi-frame model for upsampling the target image. In the following, the proposed pooling operation is described in details. The process of self-replicas fusion and their inclusion in the SR network are described next.

2.1. Lossless Pooling Layer

Normal pooling layers in CNNs result in the loss of data, however we propose a pooling layer that downscales a single-channel image to a multi-channel image with lower spatial resolution. Lossless pooling process is performed by rearranging a $H \times W$ matrix $\mathbf{M}$ into a $\frac{H}{r} \times \frac{W}{r} \times r^2$ tensor $\mathbf{T}$. This operation can be considered as a reverse sub-pixel upscaling defined in [13], and can be described mathematically as the following $\mathcal{LP}$ function:

$$\mathbf{T} = \mathcal{LP}(\mathbf{M})_{x, \, y, \, c} = \mathbf{M}_{r \cdot x - \bmod \, (r^2 - c, r), \; r \cdot y - \left\lfloor \frac{r^2 - c}{r} \right\rfloor} \tag{1}$$

where x, y, and c represent the coordinates of a pixel in the output tensor. Although the above description aims at lossless pooling of single-channel (grayscale) images, the concept can be easily extended to multi-channel images. Figure 2 demonstrates the lossless pooling operation, along with a visual example.

Application of lossless pooling layer reduces the spatial size of the input data without losing any information, and reduces the computation cost of convolution operation by a factor of r^2. Moreover, the process results in several replicas of the input image with very high correlations, which can eventually act as a data augmentation process that can enhance the still image SR.

2.2. Self-Replicas Fusion

In multi-frame SR, application of multiple input frames for creating a high quality upsampled target frame is well studied, and Caballero et al. [2] and Kapperler et al. [6] present different ways of fusing the inputs within the CNN
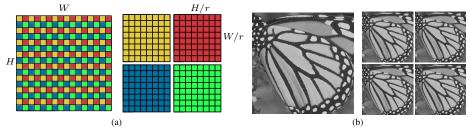

Figure 2. The lossless pooling process: (a) Input and output of the layer for $r = 2$, and (b) a sample lossless pooling on a grayscale image.
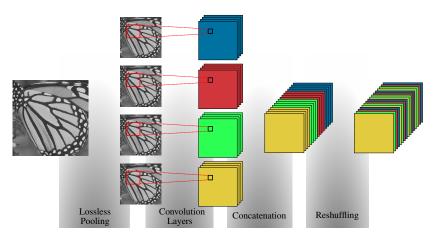
Figure 3. The process of fusing the self-replicas in CNNs by concatenation and reshuffling of the convolution outputs.

architectures to obtain a single-image output from a multi-frame input, while fully exploiting the correlations between the input frames. We adopt a similar approach, namely early fusion, for coping with the downsampled replicas created by the lossless pooling layer.

If an input frame of size $H \times W$ is fed to the lossless pooling layer with parameter r, r^2 downsampled replicas are generated, and each replica is fed to a convolutional layer with n filters. The resulting output is an ensemble of r^2 feature maps, each with the size $(\frac{H}{r}, \frac{W}{r}, n)$. The feature maps can then be concatenated and create one set of feature maps $\mathbf{F}$ with the size $(\frac{H}{r}, \frac{W}{r}, n \times r^2)$. This is similar to the early fusion concept introduced in [2, 6], also demonstrated in Figure 3 for the case of $r = 2$ and $n = 4$.

In addition to the concatenation of the feature maps, which is a widely used approach in multi-frame processing, we perform a reshuffling of the feature maps in order to create a better mix of the features produced by different replicas. The reshuffling is performed by rearranging the order of the features in the depth dimension as the following $\mathcal{RS}$ function:

$$\mathcal{RS}(\mathbf{F})_{x,\,y,\,c} = \mathbf{F}_{x,\,y,\,\lceil \frac{c}{r^2}\rceil + n \cdot \bmod (c-1, r^2)} \qquad (2)$$

where the reshuffled output of the process is depicted in Figure 3. This output can be treated as a normal feature map and be further processed in a CNN with different layers. Similar to the lossless pooling process, the concatenation and reshuffling processes can also be easily extended to multi-channel images.

3. Architecture and Implementation

With the introduction of the lossless pooling and the concatenation and reshuffling mechanisms, a CNN architecture can be devised to incorporate the abovementioned processes in the still image SR problem. Our proposed network is illustrated in Figure 4. The first step in the process is a bicu-

bic interpolation similar to many state-of-the-art approaches to reach the target resolution. This step also ensures that one architecture can be applied for upsampling with different scaling factors. Hence given an input low resolution image, a high resolution version using bicubic upscaling is created, and then the lossless pooling, individual convolutional layers on downsampled replicas, concatenation of the feature maps, and reshuffling is performed. Depending on the selected architecture, the network can work on two different modes, each described in details in the following.

3.1. LPCN-SR

The proposed CNN architecture for still image SR can be operational in two modes. The first mode is a basic version of the model, namely LPCN-SR, that only takes the downsampled replicas as the key inputs for SR, and ignores the original low quality input image generated by bicubic interpolation. This model is depicted by solid lines and connections in Figure 4. The output to this model is denoted as *SR Image A* in the figure. As mentioned earlier, an input grayscale image of size $(\frac{H}{s}, \frac{W}{s})$ is interpolated to the target resolution of (H, W) using bicubic filtering. The first step after the bicubic interpolation is the lossless pooling with $r = 2$, which results in four downsampled replicas, each with the size of $(\frac{H}{2}, \frac{W}{2})$. Each of the replicas goes through separate convolutional layers with 16 filters, resulting in four tensors of size $(\frac{H}{2}, \frac{W}{2}, 16)$. The four tensors are concatenated and reshuffled according to Section 2.2 to create a feature map of size $(\frac{H}{2}, \frac{W}{2}, 64)$.

The resulting feature map is then fed to a typical encoder-decoder architecture consisting of 10 layers with the kernel specifications presented in Figure 4. The encoder-decoder structure contains coupled convolution and transposed convolution layers with stride of 2, that are also connected using skip connections. The output of this stage is a tensor with a similar size as the input feature map to the encoder-decoder network. The generated feature map
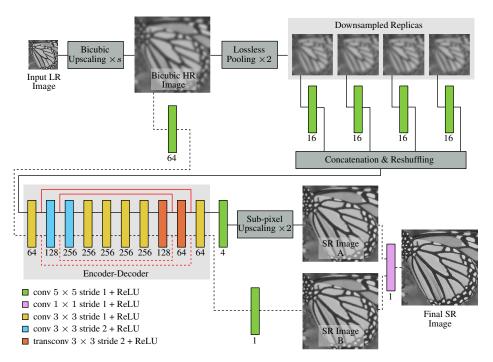
Figure 4. The architecture of the proposed CNN for still image SR. The solid connections depict the LPCN-SR model. Inclusion of the dashed connections in the model results in the LPCN-SR⁺. The red lines present the skip connections between layers.

is then fed to one last convolution layer resulting in a tensor of size $(\frac{H}{2}, \frac{W}{2}, 4)$. We then use a sub-pixel upscaling operation as proposed in [13] to upsample the tensor to the target resolution and create an image of the target size (H, W). This image is denoted as *SR Image A*, and is a high quality reconstruction of the input image created by the downsampled replicas as the only input to the model.

3.2. LPCN-SR⁺

In addition to the downsampled replicas created by the lossless pooling, the original bicubic version of the image can also be incorporated in the network, and provide more information for creating a higher quality SR image. This process is depicted using the dashed lines and connections in Figure 4, complementing the model described previously. The first step after the bicubic filtering is a convolution layer with 64 filters that results in a feature map of size $(H, W, 64)$. This feature map is then fed to the same encoder-decoder network described earlier with the same weights. We apply weight sharing for this section of the model in order to avoid extra complexity and having too many network parameters. The output of the encoder-decoder network will be a feature map of size $(H, W, 64)$, which is then fed to a single kernel convolution layer, that results in a high quality SR image, denoted as *SR Image B* in the figure.

The two created SR images, *SR Image A* and *SR Image B*, are then combined using a single kernel convolution layer

with the kernel size of 1×1 to create the *Final SR Image*. The final convolution layer operates as an averaging mechanism to mix the two created SR images. It is worth noting, that the dashed section of the model in Figure 4 is essentially a similar concept to the REDNet model proposed in [12].

3.3. Training

Training the proposed SR model is performed by solving an optimization problem to minimize the error between the ground truth labels and the network outputs. The training for the basic and the full models can be performed separately. For both cases, the training labels are a set of high resolution image samples $\mathbf{X}$, and the network outputs are the high quality high resolution images $\mathbf{X}^*$ generated by the model (*SR Image A* for LPCN-SR and *Final SR Image* for LPCN-SR⁺). The network outputs can be formulated as the following:

$$\mathbf{X}^* = \boldsymbol{\theta}_m(\mathbf{Y}) \qquad (3)$$

where $\mathbf{Y}$ is the input low resolution image, $\boldsymbol{\theta}$ is the end-to-end SR model encompassing all the parameters, and m defines the network model of operation with is either LPCN-SR or LPCN-SR⁺.

The mean squared error, defined as the following, is employed as the cost function for the training process.

$$J_{MSE}(\mathbf{X}, \mathbf{Y}, \boldsymbol{\theta}_m) = \frac{1}{N} \sum_{i=0}^{N} (\mathbf{X}_i - \boldsymbol{\theta}_m(\mathbf{Y}_i))^2 \qquad (4)$$

with N denoting the total number of training samples. The training input samples are created by downsampling the high resolution samples, and upscaling them back to the original resolution by bicubic interpolation. The scaling factor can be fixed to focus on training a particular scaling, or alternatively can include several values to cover a wide range of scaling.

4. Experiments

We focused the experiments on the SR with a scaling factor of 4, which is a challenging factor in image scaling, and it is also the basis for most of the SR evaluations. We used the DIV2K data set [1] as our training set, which comprises 800 high quality images. The images were partitioned into 96×96 samples with a stride of 80, which led to around 325,000 training sample pairs. We used Adam optimizer [9] for training the model with a learning rate of 0.0001, β_1 of 0.9, β_2 of 0.999, ϵ of 10^{-8} and a training batch size of 128. The proposed model, along with the existing state-of-the-art approaches, were implemented using TensorFlow[1] library.

4.1. Quantitative Evaluations

To compare the presented model with the existing solutions, we used Set 5, Set 14 and BSD 100 data sets, which are widely used in SR task. We compared the approach with well-known deep learning-based SR models including SR-CNN [3], ESPCN [13], FSRCNN [5], and REDNet [12]. All the models were implemented and trained according to the provided information in the literature. All the trainings were performed on Tesla K80 NVIDIA GPUs, and all the tests were performed on a machine with a generic Intel Core i7-6700 CPU with a 3.40GHz clock and 16GB RAM, and a GeForce GTX 1070 GPU.

Table 1 summarizes the performance results of the presented model, in comparison with state-of-the-art methods. The quality metrics are the Peak Signal-to-Noise Ratio

(PSNR) and Structural SIMilarity (SSIM) index. According to the results, the proposed approach can outperform the baseline models in all datasets, promising a high quality enhancement for still images.

The basic LPCN-SR model can perform as good as the best existing model, REDNet, and in some cases outperform it slightly. The strength of this model however is in the lower computation cost on GPU, which is due to using lossless pooling, that results in downscaling of the input, and consequently performing all the convolutional processes in a lower resolution than the native target resolution. The full LPCN-SR+ model, on the other hand, shows a solid outperformance on all data sets and provides major improvements in PSNR and SSIM results. The computation cost is however higher than the existing models due to the structure of the model, and incorporating multiple input signals in the approach.

4.2. Qualitative Evaluations

We also examined the subjective quality of the enhanced images on the test content, as objective metrics cannot always grasp the intricacies detected by human eyes. Application of the proposed approach resulted in clear visual improvements in the reconstructed high resolution images, some of which are presented in Figure 5 and Figure 6. In these examples only the luma signal is upsampled by the proposed SR approach, and the color components are scaled using bicubic interpolation.

One of the most challenging scenarios in SR is coping with the low resolution images with major aliasing distortions. The *Barbara* image from Set 14 of the test sets is an extreme example for this case. Upscaling the downsampled version of this image can result in magnifying those aliasing artifacts in most cases. However, as depicted in Figure 5, our proposed model does a good job in restoring some of the content that is highly distorted by the downsampling process, and is outperforming the other SR approaches. Another example is in the *PPT3* image, depicted

---

[1]https://www.tensorflow.org/

Table 1. Quality and complexity analysis of the proposed method and state-of-the-art approaches for the scaling factor of 4.

		Bicubic	SRCNN	ESPCN	FSRCNN	REDNet	LPCN-SR	LPCN-SR+
Set 5	PSNR	28.44	30.09	30.41	30.58	31.38	31.44	**31.71**
	SSIM	0.8110	0.8520	0.8590	0.8658	0.8820	0.8833	**0.8872**
	GPU time	-	0.57	0.45	0.55	0.61	0.54	0.83
	CPU time	-	0.11	0.02	0.02	0.27	0.27	1.17
Set 14	PSNR	26.00	27.18	27.37	27.52	27.98	27.98	**28.12**
	SSIM	0.7009	0.7385	0.7457	0.7506	0.7636	0.7643	**0.7686**
	GPU time	-	0.60	0.46	0.58	0.70	0.56	1.03
	CPU time	-	0.22	0.03	0.03	0.55	0.53	2.39
BSD 100	PSNR	25.89	26.64	26.77	26.85	27.16	27.14	**27.26**
	SSIM	0.6651	0.6994	0.7073	0.7100	0.7207	0.7215	**0.7253**
	GPU time	-	0.57	0.44	0.57	0.68	0.59	0.96
	CPU time	-	0.15	0.02	0.02	0.36	0.36	1.57

Figure 5. Upscaling *Barbara* and *Comic* from Set 14 with a factor of 4.

| Bicubic: 25.11 dB | ESPCN: 25.72 dB | REDNet: 25.68 dB | LPCN-SR$^+$: 25.80 dB | Ground Truth |

| Bicubic: 21.63 dB | ESPCN: 22.58 dB | REDNet: 22.86 dB | LPCN-SR$^+$: 22.93 dB | Ground Truth |

Figure 6. Upscaling *Monarch* and *PPT3* from Set 14 with a factor of 4.

| Bicubic: 27.32 dB | ESPCN: 29.79 dB | REDNet: 31.27 dB | LPCN-SR$^+$: 31.82 dB | Ground Truth |

| Bicubic: 21.76 dB | ESPCN: 23.74 dB | REDNet: 24.91 dB | LPCN-SR$^+$: 25.14 dB | Ground Truth |

in Figure 6, which reconstruction of a distorted patterned texture is shown, and LPCN-SR$^+$ provides a better prediction of the original image compared to the state-of-the-art models.

5. Conclusions

Still image SR can benefit from application of auxiliary correlated inputs derived from the original low resolution

source input. The process of creating self-replicas and their incorporation in the CNNs is performed by a novel lossless pooling layer, that generates multiple downscaled versions of the input image, which are later fused in the general SR framework. The presented LPCN-SR model can perform as good as the state of the art with a reduced computation cost on GPUs, and the full LPCN-SR[+] outperforms the existing approaches, and promises high quality and accurate image upscaling.

Acknowledgments

References

[1] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[2] J. Caballero, C. Ledig, A. P. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2848–2857, 2017.

[3] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a Deep Convolutional Network for Image Super-Resolution. In *European Conference on Computer Vision (ECCV) 2014*, pages 184–199, 2014.

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):295–307, Feb. 2016.

[5] C. Dong, C. C. Loy, and X. Tang. Accelerating the Super-Resolution Convolutional Neural Network. In *European Conference on Computer Vision (ECCV) 2016*, pages 184–199, 2016.

[6] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos. Video Super-Resolution With Convolutional Neural Networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, June 2016.

[7] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-Recursive Convolutional Network for Image Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 12 2014.

[10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.

[11] O. Makansi, E. Ilg, and T. Brox. End-to-End Learning of Video Super-Resolution with Motion Compensation. In *German Conference on Pattern Recognition (GCPR) 2017*, 2017.

[12] X. Mao, C. Shen, and Y. Yang. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2802–2810, 2016.

[13] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 1874–1883, 2016.

[14] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia. Detail-Revealing Deep Video Super-Resolution. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[15] Z. Wang, Y. Yang, Z. Wang, S. Chang, W. Han, J. Yang, and T. S. Huang. Self-Tuned Deep Super Resolution. In *CVPR Workshops*, pages 1–8. IEEE Computer Society, 2015.