Super-Resolution on Degraded Low-Resolution Images Using Convolutional Neural Networks

Conference Paper · October 2019

DOI: 10.23919/EUSIPCO.2019.8903000

3 authors:

Fatma Albluwi
Trinity College Dublin

Vladimir A. Krylov
Trinity College Dublin

Rozenn Dahyot
National University of Ireland, Maynooth

Super-Resolution on Degraded Low-Resolution Images Using Convolutional Neural Networks

Fatma Albluwi
Computer Science & Statistics
Trinity College Dublin
Dublin, Ireland
albluwif@tcd.ie

Vladimir A. Krylov
Computer Science & Statistics
Trinity College Dublin
Dublin, Ireland
vladimir.krylov@tcd.ie

Rozenn Dahyot
Computer Science & Statistics
Trinity College Dublin
Dublin, Ireland
rozenn.dahyot@tcd.ie

***Abstract*—Single Image Super-Resolution (SISR) has witnessed a dramatic improvement in recent years through the use of deep learning and, in particular, convolutional neural networks (CNN). In this work we address reconstruction from low-resolution images and consider as well degrading factors in images such as blurring. To address this challenging problem, we propose a new architecture to tackle blur with the down-sampling of images by extending the DBSRCNN architecture [1]. We validate our new architecture (DBSR) experimentally against several state of the art super-resolution techniques.**

***Index Terms*—Image super-resolution, image deblurring, deep learning, CNN.**

I. Introduction

Single image super-resolution (SISR) endeavours to estimate a super resolution (SR) image as an approximation to an unknown high-resolution (HR) image from a single low-resolution (LR) input image. It is a classical problem in computer vision aiming at increasing image resolution thus providing efficient zooming tool for applications such as surveillance [2], etc. It is still an active challenging task due to its complexity and ill-posed nature [3]. Tsai [4] presented the first work that discussed the topic of image super-resolution in the 1980s, and the area remains active ever since. The state-of-the-art methods are example-based approaches which learn prior information to alleviate the problem of solution ambiguity [5]. The example-based methods are categorised into two classes: internal example-based methods [6]–[8] and external example-based methods [9]–[14].

Recently, SISR performance witnessed a dramatic boost due to the introduction of CNNs. The latter have many advantages such as efficiency, accuracy, fast speed using parallel computing, allowing improvements for designing and training the networks [15]–[18]. Moreover CNN architectures attempt to learn an end-to-end mapping function linking Low Resolution (LR) images with the corresponding High Resolution (HR) images. Several CNN models proposed for SISR have achieved excellent results when they deal with simple degradation reduced to down-sampling only. The limitation of these models is that they do not take into account other frequently occurring image degradation factors. In a more realistic scenario the performance of these learned CNNs will be substantially lower when the true degradation is more complex [5]. An important image degradation factor can be modeled via convolution with a blurring kernel. By not taking blurring into consideration or via simple mismatch of blurring kernels, most SISR methods are of reduced interest practically, see [19]. Despite the importance of considering blur as a degradation factor in SISR methods, there is little work on designing CNN models to handle this problem.

The first author was supported by King Abdullah Scholarship Program from Saudi Arabian Government. The second author was supported by the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No.713567. This work was partly funded by the ADAPT Centre for Digital Content Technology funded under the SFI Research Centres Programme Grant 13/RC/2106 and co-funded under the European Regional Development Fund.

This paper aims to design a CNN to perform SISR that is capable of handling low-resolution and input image blurring simultaneously. We propose a new architecture called DeBlurring Super-Resolution (DBSR) that is applied to both non-blind and blind SR scenarios. We also provide the program implementation of the architecture online. In the remainder of the paper we start by presenting the related work in Sec. II. We then present DBSR in Sec. III, followed by its experimental evaluation Sec. IV. We finally draw conclusions in Sec. V.

II. Related Work

Dong et al.'s architecture (SRCNN) [20] was the first one to solve single image super-resolution using a three-layers convolutional neural network. The Very Deep Super-Resolution model (VDSR) [21] has been proposed by Kim et al. using residual learning to be able to train deep networks. Using residual learning strategy, Zhang et al. [22] proposed to train a single CNN architecture to tackle several general tasks such as Gaussian denoising, single image super-resolution and JPEG image deblocking. Their latest iteration of their architecture (SRMD [23]) is compared with ours in the experimental section (Sec.IV). Using transposed convolutions, a Laplacian pyramid super-resolution network (LapSRN) takes the input LR image and progressively produces the sub-band residuals [24]. Many other CNN models have been proposed likewise for SISR [25]–[33]. Several CNN models have been proposed for SISR based on bicubic interpolated inputs. Recently practical scenarios where Gaussian blurring occur as additional degradation to low resolution have been addressed with a CNN denoiser plugged as a learned prior into a model-based method to solve image deblurring and super-resolution (IRCNN) [34].

On the other hand, end-to-end learning can be performed by super-resolution network trained for multiple degradations (SRMD) [23] which takes the concatenated LR image and degradation maps in a single network and these degradation maps are collected by a dimensionality stretching method of the degradation parameters (i.e., blur kernel and noise level). Alternatively we recently proposed deblurring super-resolution convolutional neural network (DBSRCNN) [1] to handle noisy and low res images in non-blind and blind SR scenarios. DBSRCNN tackles blurring and SISR together in a single network (see Sec.III-A). We propose here to improve this network by enhancing the features inside the network (see Sec. III-B).

III. Super-Resolution and de-blurring

A. DeBlurring Super-Resolution Convolutional Neural Network (DBSRCNN)

The DBSRCNN [1] aims to recover the deblurred high-resolution image $F(\mathbf{X})$ from the blurred low-resolution image $\mathbf{X}$ (input) after interpolation, by learning an end-to-end mapping F. This network includes five convolutional layers. The first and the second layers are merged by using the concatenation operation. The objective is to learn a mapping function F containing three main operations: patch extraction and representation operation to extract the noisy feature maps from the LR input image; reconstruction operation to form the reconstructed image (output); and finally, a sequence of non-linear mapping operations between the input and the output. The model can be written as:

$$F_i(\mathbf{X}) = \max\left(0, W_i * F_{i-1}(\mathbf{X}) + b_i\right) \quad i \in \{1,2,4\} \tag{1}$$

$$F_{12}(\mathbf{X}) = \text{merge}\big(F_1(\mathbf{X}), F_2(\mathbf{X})\big) \tag{2}$$

$$F_3(\mathbf{X}) = \max\left(0, W_3 * F_{12}(\mathbf{X}) + b_3\right) \tag{3}$$

$$F(\mathbf{X}) = W_5 * F_4(\mathbf{X}) + b_5 \qquad \textit{(output layer)} \tag{4}$$

where W_i and b_i are the weights of the filters and biases of the ith layer respectively. The W_i comprises of n_i filters which supports $n_{i-1} \times f_i \times f_i$, where n_i is the number of filter (number of feature maps), and n_0 is the number of channels in the input image. $F_i(\mathbf{X})$ is the output feature maps and $F_{i-1}(\mathbf{X})$ is the input feature maps. The activation function used is Rectified Linear Unit (ReLU, max(0,X)) [35]. **The structure of DBSRCNN:** The number of feature maps (and filter size) of each layer is as following $32(9)$, $32(5)$, $32(5)$, $32(5)$ and $1(5)$, to be easier we can write it as (32-32-32-32-1)(9-5-5-5-5). The third layer concatenate feature maps of the first two layers together to form a vector containing low-level features and enhanced features together. This layer comprises 64-features maps.

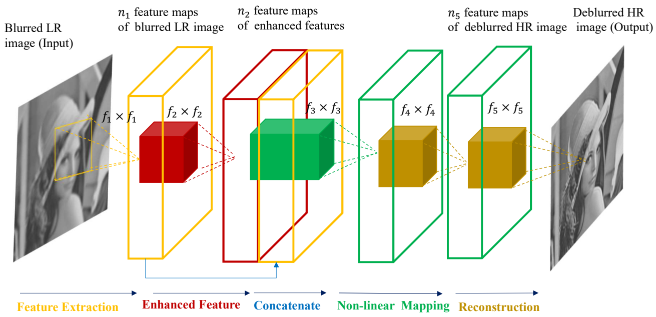

Fig. 1: DBSRCNN architecture [1] which consists 5 layers.

B. DeBlurring Super-Resolution (DBSR)

In DBSRCNN, feature extraction is the first step for determining what should be extracted and restored in the following steps. The enhanced layer is used to enhance the noisy extracted features. As the image blurring level increases, the output image quality is decreased, because the extracted and enhanced features are incapable of dealing with the additional noise. Inspired by the step of feature enhancement used in super-resolution [36] and JPEG compression artifacts reduction [37], we propose to introduce three feature enhancement layers after the merged layer in DBSRCNN to create a more efficient and deeper network (DBSR). Indeed, a single layer has a limited capacity to enhance the noisy extracted features in complex applications like blurred SISR. Therefore, we increase this number to improve the capacity to suppress blur (noise) in the features. Indeed, we have added more than three layers to improve the performance of the network, but the enhancement was marginal.

DBSR is a deeper version of DBSRCNN which incorporates more non-linear mapping layers. A deeper network allows for enhanced non-linearity mapping, thus supporting a more robust regressor between the low-level features and the output. Indeed, applications like deblurring or denoising are complex which leads to noisy low-level features extracted by a single layer. In such case, the performance of the network depends on the features, not on the regressor. DBSR improves the mapping efficiency by enhancing the low-level extracted features using additional layers after concatenation of the low-level feature maps of the first layer and enhanced feature maps of the second layer. All these layers put together constitute a better feature extractor.

a) Formulation: The new model DBSR is shown in Figure 2. Overall it consists of 8 layers. The five layers of DBSRCNN remain unchanged in the new network. The first enhanced feature layer located after the first layer is to extract new features form the extracted noisy features, and then merged these features together using concatenate layer to map them together. While in DBSRCNN we mapped these features directly, in DBSR these features are further processed by three layers before the final mapping. Similar to DBSRCNN, DBSR adopts Rectified Linear Unit (Relu) as the activation function.

b) Model Learning: Consider a set $\{\mathbf{Y_i}, \mathbf{X_i}\}_{i=1}^{m}$, where $\mathbf{Y}$ is a high-resolution image and $\mathbf{X}$ is its corresponding interpolated blurred low-resolution image. Mean Squared Error (MSE) is used as the loss function to find the optimal parameters Θ of the model. This is achieved by minimizing

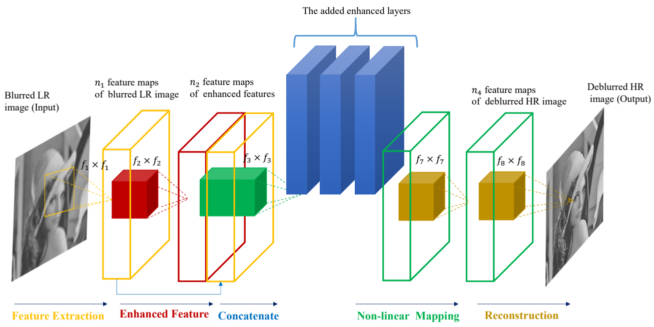

Fig. 2: Proposed DBSR architecture. This network comprises eight layers: the five convolutional layers of DBSRCNN, in addition to extra 3 enhanced layers inserted after the concatenated layer to further refine the merged feature maps.

the difference between the reconstructed images $F(\mathbf{X}, \Theta)$ and its ground truth high-resolution images $\mathbf{Y}$:

$$L(\Theta) = \frac{1}{m}\sum_{i=1}^{m} ||F(\mathbf{X}_i; \Theta) - \mathbf{Y}_i||^2, \tag{5}$$

where $\Theta = \{W_1, W_2, \cdots, W_8, b_1, b_2, \cdots, b_8\}$, and m is the number of training samples. The lost function is minimized using Adam optimization [38]. We train all experiments for 60 epochs with a batch size 64. In each layer, the filter weights are initialized by the initialization method described in He et al. [39], considered a robust method for ReLU. The learning rate is 0.001. The structure of the proposed network is 8 layers (64-32-32-32-32-32-32-1)(9-5-5-5-5-5-5-5).

IV. Experiments

A. Datasets and Degradation Model

a) Training and Testing dataset: Different training datasets have been used for different networks. For example, SRCNN [20] uses 91 images from Yang et al. [9] which is a relatively small dataset. In the VDSR model [21] 291 images are used, 91 images from Yang et al. in addition to 200 images from Berkeley Segmentation Dataset [40]. In our work we employ 291 images as in [21], with data augmentation (flip and rotation) resulting in a total of 2,328 images. The training dataset is divided into sub-images with size $f_{sub} = 31$ resulting in 573,632 sub-images by employing a stride of 21. The model is trained on sub-images, whereas the inference is carried out on the whole image. 'Set5' [10] and 'Set14' [41] are two datasets commonly used for testing.

b) Degradation Model: A standard model for degradation is formulated by the as a linear combination

$$\mathbf{I}_{LR} = \mathrm{D}_s \; \mathrm{H}_\sigma \; \mathbf{I}_{HR} + n, \tag{6}$$

where the HR image $\mathbf{I}_{HR}$ is first blurred by operator H_σ and then down-sampled with D_s. The bicubic downsampling method with fixed down-sampling factor s is used here. The noise n is additive noise.

Before generating LR images according to Eq.(6), the blur kernels should be defined. The degradation model assumes that an HR image can be degraded into many LR images depending on the blur kernel and the noise. To produce blur we applied the Gaussian kernel model with a fixed kernel width (the value of σ). In non-blind CNN models, the standard deviation σ is set to 1, 2, and 3. For blind CNN models, the standard deviation takes values in $[0.5, 3]$, which is assumed unknown. To generate a single blurred LR image $\mathbf{X}_i$ (input) for training and testing, the HR image $\mathbf{Y}_i$ are first blurred using a Gaussian kernel with standard deviation $\sigma = i$. Secondly, images are down-sampled using the down-scaling factor s, and then up-sampled using bicubic interpolation to the HR size. The down-scaling and up-scaling factors used here are 2, 3, 4. Note that padding may be required.

B. Comparison with the state-of-the-art

The proposed DBSR model is compared with several CNN models designed to handle down-sampling without blurring. Table I shows the PSNR and SSIM [42] results of state-of-the-art CNN models. While our proposed method may not always perform best, our DBSR pipeline still achieves competitive results with a smaller number of parameters in comparison. The results of the DBSR model are better than those obtained by SRCNN and DBSRCNN and this is achieved by adding more layers in DBSRCNN model to enhance the low level features before mapping.

C. Non-Blind and Blind Scenarios

Table II shows the evaluation of the performance of DBSR on images with different degrees of blur. We have considered two different scenarios: non-blind and blind scenario. The non-blind scenario corresponds to the case when the network is trained and tested on images with the same σ in $N(0, \sigma)$; $\sigma = 1, 2$ or 3. While, in the blind scenario the network is trained on images with kernel width $N(0, \sigma)$ with σ ranging between [0.5-3]. The blind models are tested on images at any kernel width value. The quantitative results (PSNR/ SSIM) for Set5 and Set14 point out that DBSR enhances the quality of images over SRCNN and DBSRCNN models, for both non-blind and blind scenarios. A possible explanation is that the added enhanced layers led to improved results relying on cleaner features with less noise.

In Table III, we follow the comparison presented in Zhang et al. [23], where Gaussian blur with $\sigma = 1.3$ and $\sigma = 2.6$ and scale factor $s = 3$ was considered on Set5 dataset. We

TABLE I: Average PSNR and SSIM results for $\sigma = 0$ (without adding any blur) on datasets Set5 and Set14.

Dataset	Scale Factor	LR Input	SRCNN [43]	DBSRCNN [1]	LapSRN [24]	SRMD [23]	DBSR
		PSNR/ SSIM					
Set5	$s = 2$	33.66 / 0.930	35.68/ 0.948	-	37.52 / 0.959	37.53 / 0.959	37.23/ 0.957
	$s = 3$	30.40 / 0.868	31.95/ 0.845	32.60/ 0.908	33.82 / 0.922	33.86 / 0.923	33.24/ 0.917
	$s = 4$	28.42 / 0.810	29.79/ 0.844	-	31.54 / 0.885	31.59 / 0.887	30.84/ 0.873
Set14	$s = 2$	30.24 / 0.869	31.74/ 0.899	-	33.08 / 0.913	33.12 / 0.914	32.83/ 0.911
	$s = 3$	27.54 / 0.774	28.67/ 0.806	29.11/ 0.817	29.89 / 0.834	29.84 / 0.833	29.56/ 0.827
	$s = 4$	25.99 / 0.703	27.00/ 0.735	-	28.19 / 0.772	28.15 / 0.772	27.69/ 0.759
No of parameters		-	8k	105k	813K	1,478k	236k

The results for LapSRN and SRMD are taken from Zhang et al. [23].

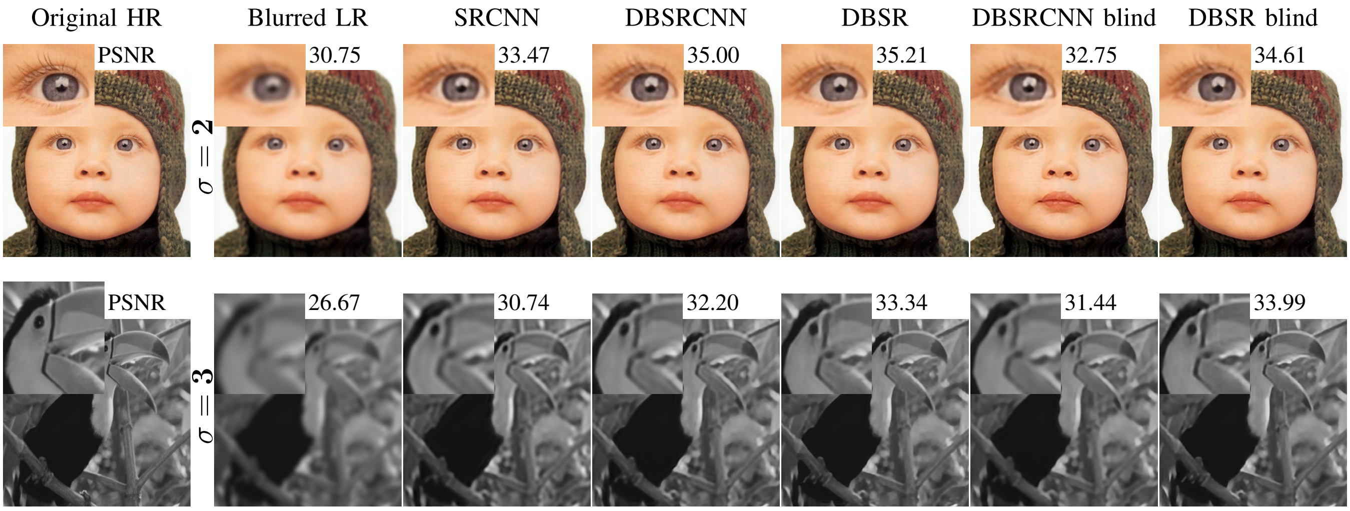

Fig. 3: SR with different models on images after Gaussian blur with different $\sigma = 2, 3$. The results show the non-blind and blind scenarios. Each result is accompanied by zoom and PSNR(dB). In blind scenarios $\sigma \in [0.5, 3]$.

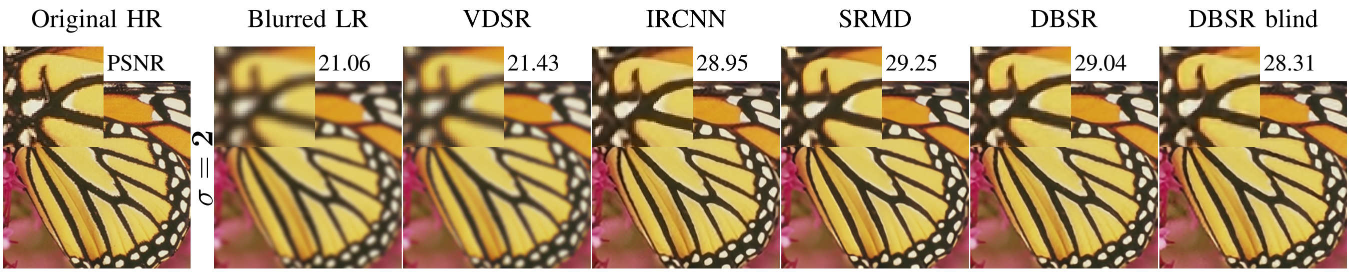

Fig. 4: SISR performance of different models on Butterfly image after Gaussian blur at $\sigma = 2$. In the blind scenario $\sigma \in [0.5, 3]$.

TABLE II: Average PSNR and SSIM results with different blur levels $\sigma = 1, 2, 3$, scale factor $s = 3$ on Set5 and Set14.

Dataset	Kernel Width	LR Input	Non-blind Networks			Blind Networks	
			SRCNN [43]	DBSRCNN [1]	DBSR	DBSRCNN [1] $\sigma \in [1, 3]$	DBSR $\sigma \in [0.5, 3]$
			PSNR/ SSIM				
Set5	$\sigma = 1$	29.47/ 0.847	31.55/ 0.892	32.65/ 0.907	33.24/ 0.917	31.24/ 0.888	32.60/ 0.909
	$\sigma = 2$	27.45/ 0.789	30.29/ 0.873	32.09/ 0.897	33.05/ 0.914	30.14/ 0.868	31.78/ 0.900
	$\sigma = 3$	25.65/ 0.724	29.03/ 0.819	30.48/ 0.858	31.36/ 0.879	29.51/ 0.840	31.67/ 0.886
Set14	$\sigma = 1$	26.86/ 0.745	28.40/ 0.805	29.11/ 0.818	29.56/ 0.827	28.38/ 0.805	29.10/ 0.820
	$\sigma = 2$	25.37/ 0.679	27.41/ 0.780	28.80/ 0.808	29.47/ 0.825	27.43/ 0.765	28.80/ 0.813
	$\sigma = 3$	24.04/ 0.617	26.33/ 0.712	27.50/ 0.753	28.00/ 0.782	26.68/ 0.722	28.29/ 0.788

compare our model with VDSR [21], SRMD [23], also model-based methods such as IRCNN [34]. Our model provides good performance compared to other models. Examples of qualitative comparison of reconstruction are shown in Figure 3 and 4. In particular, it can be observed that SRMD, the best performing model in terms of PSNR, reports SR results of comparable visual quality with the proposed DBSR [1], whereas the more realistic DBSR blind pipeline achieves slightly worse sharpness (see zoom). Note that the higher the PSNR and

TABLE III: Average PSNR and SSIM results with different kernel width of blur kernel with scale factor $s = 3$ on Set5.

Kernel Width	LR Input	VDSR [21]	IRCNN [34]	SRMD [23]	DBSR $\sigma \in [0.5, 3]$
$\sigma = 0.2$	30.39/ 0.8680	33.67/ 0.9213	33.39/ 0.9393	33.86/ 0.9232	-
$\sigma = 1.3$	28.84/ 0.8308	-	33.31/ 0.9186	33.77/ 0.9214	32.70/ 0.9094
$\sigma = 2.6$	26.17/ 0.7444	-	31.48/ 0.8624	32.59/ 0.8999	31.85/ 0.8951

The results of VDSR model is taken from Zhang et al. [23].

SSIM values are, the better the performance is.

V. Conclusion

In this paper, we have presented DBSR, an extension model for DBSRCNN model, where we have proposed adding convolutional layers to enhance the extracted features. We have reported a panel of comparisons with the state-of-the-art deep learning and model-based methods to highlight the competitive performance of the proposed model for super resolution on blurred images. Importantly, these results are obtained by a model with 3 to 6 times less parameters. This constitutes a strong advantage of DBSR and facilitates its deployment and accelerates the training process [44].

[1]The code is available at: https://github.com/Fatma-ALbluwi/DBSR.git

REFERENCES

[1] F. Albluwi, V. Krylov, and R. Dahyot, "Image deblurring and super-resolution using deep convolutional neural networks," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Aalborg, Denmark, 2018.

[2] P. Shamsolmoali, M. Zareapoor, D. K. Jain, V. K. Jain, and J. Yang, "Deep convolution network for surveillance records super-resolution," *Multimedia Tools and Applications*, Apr 2018. [Online]. Available: https://doi.org/10.1007/s11042-018-5915-7

[3] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.

[4] R. Tsai, "Multiframe image restoration and registration," *Advance Computer Visual and Image Processing*, vol. 1, pp. 317–339, 1984.

[5] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *European Conference on Computer Vision*. Springer, 2014, pp. 372–386.

[6] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 349–356.

[7] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 49–64.

[8] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[10] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[11] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE transactions on image processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

[12] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.

[13] ——, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian conference on computer vision*. Springer, 2014, pp. 111–126.

[14] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[19] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, "Accurate blur models vs. image priors in single image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2832–2839.

[20] 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*. Springer, 2014, pp. 184–199.

[21] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[22] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. 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.

[23] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 6, 2018.

[24] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate superresolution," in *IEEE Conference on CVPR*, vol. 2, no. 3, 2017, p. 5.

[25] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 370–378.

[26] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637–1645.

[27] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[28] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust single image super-resolution via deep networks with sparse prior," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, 2016.

[29] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee *et al.*, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1110–1121.

[30] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*, vol. 1, no. 2, 2017, p. 4.

[31] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, 2017, p. 5.

[33] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4539–4547.

[34] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[35] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[36] Z. Xiong, X. Sun, and F. Wu, "Image hallucination with feature enhancement," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2074–2081.

[37] K. Yu, C. Dong, C. C. Loy, and X. Tang, "Deep convolution networks for compression artifacts reduction," *arXiv preprint arXiv:1608.02778*, 2016.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[40] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.

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

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[43] YapengTian, "Srcnn-keras," https://github.com/YapengTian/SRCNN-Keras, 2017.

[44] T. Llewellynn, M. M. Fernández-Carrobles, O. Deniz, S. Fricker, A. Storkey, N. Pazos, G. Velikic, K. Leufgen, R. Dahyot, S. Koller, G. Goumas, P. Leitner, G. Dasika, L. Wang, and K. Tutschku, "Bonseyes: Platform for open development of systems of artificial intelligence," in *Proceedings of the Computing Frontiers Conference*, ser. CF'17. New York, NY, USA: ACM, 2017, pp. 299–304. [Online]. Available: http://doi.acm.org/10.1145/3075564.3076259