Image Deblurring and Super-Resolution Using Deep Convolutional Neural Networks

Conference Paper · September 2018

DOI: 10.1109/MLSP.2018.8516983

3 authors:

Fatma Albluwi
Trinity College Dublin
4 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Vladimir A. Krylov
Trinity College Dublin
32 PUBLICATIONS **282** CITATIONS
SEE PROFILE

Rozenn Dahyot
National University of Ireland, Maynooth
124 PUBLICATIONS **1,555** CITATIONS
SEE PROFILE

2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

IMAGE DEBLURRING AND SUPER-RESOLUTION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Fatma Albluwi, Vladimir A. Krylov & Rozenn Dahyot

School of Computer Science & Statistics, Trinity College Dublin, Ireland

ABSTRACT

Recently multiple high performance algorithms have been developed to infer high-resolution images from low-resolution image input using deep learning algorithms. The related problem of super-resolution from blurred or corrupted low-resolution images has however received much less attention. In this work, we propose a new deep learning approach that simultaneously addresses deblurring and super-resolution from blurred low resolution images. We evaluate the state-of-the-art super-resolution convolutional neural network (SRCNN) architecture proposed in [1] for the blurred reconstruction scenario and propose a revised deeper architecture that proves its superiority experimentally both when the levels of blur are known and unknown a priori.

Index Terms— Image super-resolution, deblurring, deep learning, convolutional neural networks.

1. INTRODUCTION

Single image super-resolution (SR) is an essential application in computer vision that proves useful in multiple areas such as remote sensing image processing, security systems, medical imaging, etc. The SR task is an ill-posed problem, where one LR image has many solutions for HR image. In the early 1980s, Tsai [2] has addressed the topic of image super-resolution for the first time. Since then the problem has received a lot of attention, and typical state-of-the-art methods perform as example-based approaches where learned prior information alleviates the problem of multiple solutions [3]. The example-based methods are divided into two kinds: internal methods [4, 5, 6] and external example-based methods [7, 8, 9, 10, 11, 12]. The pipeline of most external example-based approaches is shared, where the focus is on learning and optimizing the dictionaries or learning mapping functions. Despite some similarities convolutional neural networks (CNNs) based technique are different because they attempt to learn an end-to-end mapping function between LR and HR images without addressing explicitly the problem of selection and composition of dictionaries and / or manifolds.

Currently, the majority of the best performing state-of-the-art methods for SR are based on deep neural networks [13, 1, 14, 15, 16, 17, 18, 19, 20, 21]. The pioneering SRCNN model of Dong et al. [1] introduced a simple yet efficient deep architecture for SR. All these algorithms assume that a constant amount of noising or blurring is applied to all training and testing images. In this study we address an additional factor of unknown amount of blurring applied to images that are received by the SR pipeline. It is thus necessary to tackle simultaneously deblurring and SR reconstruction in a unified procedure. This paper aims at designing a smart system for tackling this dual problem. We propose a new architecture for deblurring SR application inspired by SRCNN. We refer to our model as deblurring super-resolution convolutional neural network (DBSRCNN) which we apply to SR on blurred images with a priori known (non-blind) and unknown (blind) amount of blurring. We experimentally validate our model and show that our architecture is better suited for reconstructing blurred images than SRCNN in both blind and non-blind scenarios.

2. SUPER RESOLUTION AND DEBLURRING

We begin by briefly reviewing the SRCNN architecture proposed by Dong et al. [1] and its optimization procedure in Sec. 2.1. To tackle efficiently simultaneous deblurring and SR we propose a novel DBSRCNN architecture presented in Section 2.2.

2.1. SRCNN architecture

The (9-1-5) architecture of SRCNN corresponds to a relatively small network that contains 8,032 parameters and is composed of 2 hidden layers: added to the input and the output layers there are 4 layers in total. The objective is to learn a mapping function F performing three tasks: patch extraction, non-linear mapping, and reconstruction. The structure of the network is defined as follows [1]:

- **Input Layer:** the input $\mathbf{x}$ is 2-dimensional representation of the sub-image with $c = 1$ for grey level image

The first author was supported by King Abdullah Scholarship Program from Saudi Arabian Government. The second author was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No.713567.

978-1-5386-5477-4/18/$31.00

(channel Y) and $c = 3$ for colour images (YCbCr).

- **Patch extraction and representation** The first hidden layer extracts overlapping patches from the input sub-image and represents each patch as a high-dimensional vector. It uses a Rectified Linear Unit (ReLU) activation function F_1, kernel size $f_1 = 9$ and contains $n_1 = 64$ feature maps:

$$F_1(\mathbf{x}) = \max\left(0, W^{(1)} * \mathbf{x} + b^{(1)}\right), \quad (1)$$

where $W^{(1)}$ contains n_1 filters of size $c \times f_1 \times f_1$ that produce n_1 feature maps. $b^{(1)}$ is a n_1-dimensional bias vector; '*' denotes the convolution operation.

- **Second order mapping** The second hidden layer maps each high-dimensional vector of the previous layer to another high-dimensional vector, which is the representation of a high-resolution patch. ReLU activation function is employed with $n_2 = 32$ feature maps and filter size $f_2 = 1$:

$$F_2(\mathbf{x}) = \max\left(0, W^{(2)} * F_1(\mathbf{x}) + b^{(2)}\right), \quad (2)$$

where $W^{(2)}$ involves n_2 filters of size $n_1 \times f_2 \times f_2$ to produces n_2 feature maps. $b^{(2)}$ is the bias vector.

- **Reconstruction operation** The output layer produces the reconstructed SR image by aggregating the patch-wise representations:

$$F(\mathbf{x}) = W^{(3)} * F_2(\mathbf{x}) + b^{(3)} \quad (3)$$

where $W^{(3)}$ includes c filters of size $n_2 \times f_3 \times f_3$ with filter size $f_3 = 5$. $b^{(2)}$ is a c-dimensional bias vector, associated to the number of image channels.

The structure of SRCNN can be written as a network with 3 layers (9-1-5)(64-32-1).

SRCNN Optimization The filter weights in each layer are initialized by drawing randomly from a Gaussian distribution with zero mean and standard deviation 0.05, the biases set to 0, and the learning rate equal to 0.001. The aim is to recover SR image $F(\mathbf{x})$ from LR $(\mathbf{x})$ that is as similar as possible to the original HR image $\mathbf{y}$. The estimation of $\Theta = \{W^{(1)}, W^{(2)}, W^{(3)}, b^{(1)}, b^{(2)}, b^{(3)}\}$ is required to specify the end-to-end mapping function F. To this end, the cost minimization between the reconstructed images $F(\mathbf{x}, \Theta)$ and its original HR images $\mathbf{y}$ is performed. The MSE function $C(\theta)$ is employed as the cost function:

$$C(\theta) = \frac{1}{n} \sum_{i=1}^{n} ||F(\mathbf{x}_i; \theta) - \mathbf{y}_i||^2 \quad (4)$$

This cost function is minimized using Adam [22], which is used to optimize the network at faster convergence rates.

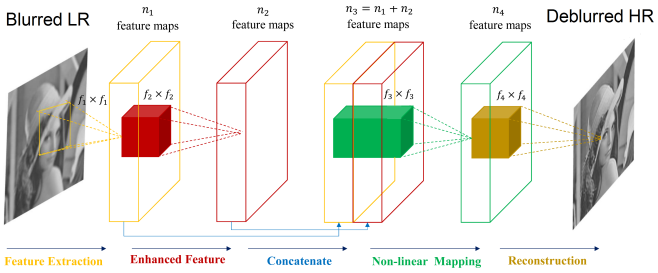

Fig. 1. Proposed architecture DBSRCNN

2.2. De-blurring with DBSRCNN

The proposed network aims to learn an end-to-end mapping F, which takes the blurred LR image $\mathbf{x}$ as input, and directly provides the deblurred HR reconstruction $F(\mathbf{x})$. Our architecture includes four convolutional layers in addition to a concatenation layer as demonstrated in Figure 1.

The first layer is feature extraction to compute the low-level features and contains 32-feature maps (32 filters) with filter size 9×9. The second layer is a feature enhancement layer which provides enhanced features from the output of the first layer, and contains 32-features maps with filters of size 5×5. The third layer concatenates features from the first two layers creating a merged vector with low-level and enhanced features. This layer contains 64-features maps or 32-features maps depending on the operation defining the merger procedure. Such operations include summation, maximum, subtraction, averaging, multiplication, concatenation. All these operations except the last one take the same size of inputs and return the same shape. Concatenation allows inputs of different sizes. We have empirically observed the best performance associated with the concatenation operation. The fourth layer performs the second-order mapping.The final fifth layer reconstructs the output HR image. The operations of the proposed network can be described as follow:

$$F_i(\mathbf{x}) = \max\left(0, W^{(i)} * F_{i-1}(\mathbf{x}) + b^{(i)}\right), \; i \in \{1, 2, 4\} \quad (5)$$

$$F_3(\mathbf{x}) = \text{merge}\left(F_1(\mathbf{x}), F_2(\mathbf{x})\right) \quad (6)$$

$$F(\mathbf{x}) = W^{(5)} * F_4(\mathbf{x}) + b^{(5)} \quad (7)$$

where $W^{(i)}$ and $b^{(i)}$ are the filters and biases. $W^{(i)}$ is comprised of n_i filters and n_0 is the number of channels in the input image. $F_i(\mathbf{x})$ are the feature maps and $F(\mathbf{x})$ is the reconstructed output image which has the same size as the input image. The activation functions used are ReLU. Mean Squared Error (MSE) is used as the cost function, and the cost function is minimized using Adam optimization.

3. EXPERIMENTAL RESULTS

Dataset & image degradation model The training dataset is composed of 91 images taken from Yang et al. [23]. The test datasets are denoted "Set5" (5 images) [8] and "Set14" (14

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	30.40	29.47	27.45	25.65	24.33
Original SRCNN[1]	$\sigma = 0$	31.95				
Re-trained SRCNN **non-blind**	custom σ		31.55	30.29	29.01	27.35
Re-trained SRCNN **blind**	mixed $\sigma = 1-3$	**28.49**	**30.05**	**30.02**	27.54	25.43
	mixed $\sigma = 1-4$	27.64	29.05	29.55	**27.80**	**25.91**

Table 1. Average of PSNR (dB) for SRCNN on "Set5"

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	27.54	26.86	25.37	24.04	23.05
Original SRCNN[1]	$\sigma = 0$	28.67				
Re-trained SRCNN **non-blind**	custom σ		28.40	27.41	26.33	25.19
Re-trained SRCNN **blind**	mixed $\sigma = 1-3$	**26.46**	**27.57**	**27.16**	25.34	23.85
	mixed $\sigma = 1-4$	25.85	26.92	26.93	**25.52**	**24.20**

Table 2. Average of PSNR (dB) for SRCNN on "Set14"

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	0.8589	0.8325	0.7660	0.6937	0.6337
Original SRCNN[1]	$\sigma = 0$	0.8852				
Re-trained SRCNN **non-blind**	custom σ		0.8816	0.8545	0.8047	0.7491
Re-trained SRCNN **blind**	mixed $\sigma = 1-3$	**0.8488**	**0.8707**	**0.8436**	0.7588	0.6771
	mixed $\sigma = 1-4$	0.8319	0.8561	0.8409	**0.7717**	**0.6969**

Table 3. Average of SSIM for SRCNN on "Set14"

images) [24]. We make this choice of training and test data to allow a fair comparison with [1] where they were employed. To fully exploit the available data we rely on augmentation: HR images from the training set are randomly cropped to obtain $f_{sub} \times f_{sub} \times c$ pixel sub-images. We employ sub-images of size $f_{sub} = 33$, thus the 91 HR images can be divided into 21,824 training sub-images with stride 14. The model is trained on sub-images, and the inference on the whole image.

To create a single blurred LR sub-images $\mathbf{x}_i$ (input) for training and testing, the HR sub-images $\mathbf{y}_i$ are first blurred using a Gaussian kernel noted $blur_i$ (with standard deviation $\sigma = i$). Secondly images are down-sampled using the down-scaling factor, and then up-sampled using bicubic interpolation to the HR input resolution. Thus, the sizes of the input and output images of our network are equal. The down/up scaling factor employed in this study is $s = 3$.

Computation Time The Python implementation of SRCNN [25] uses Keras-1 with Theano library as a backend. We converted the code to keras-2 and switched the backend to TensorFlow. To render the SRCNN training more computationally efficient, our implementation relies on Adam optimization. The training time of SRCNN in this implementation was 8.33 minutes with NVIDIA GTX 1050 GPU. Dong et al. [1] indicate that the (9-5-5) SRCNN network achieved better performance than (9-1-5) SRCNN network but at the cost of training time. In our implementation of (9-5-5) SRCNN network, the training time was around 11 minutes.

3.1. Evaluation of SRCNN

To fully evaluate the performance of SRCNN on SR of blurred images we test several different scenarios:

- *Non-blind scenario*: four pipelines are trained on images with blur $N(0, \sigma)$ with $\sigma = 1, 2, 3, 4$ and tested on images having the same level of blur.
- *Blind scenario*: two pipelines are trained on images with blur $N(0, \sigma)$ with σ varying between 1-3 and 1-4. Testing performed on images with various levels of blur.

The average of peak signal-to-noise ratio (PSNR) in dB between the blurred LR input (degraded images) and the original HR images on Set5 and Set14 is used to evaluate the performance of all pipelines and results are reported in Tables 1 and 2. The default baseline comparison is with the bicubic interpolation. In the non-blind scenario, all AI pipelines improve the PSNR. The same behavior is observed for the structural similarity index measure (SSIM) as well, see Table 3 for "Set14". The performance improvement becomes less pronounced in the blind scenarios.

Table 4 presents the average PSNR over the merged test sets. Here we test all pipelines on all possible levels of the input blurring. As expected non-blind pipelines perform best on images presenting the correct level of blurring (the one used for their training), see diagonal PSNR in red, and outperform blind pipelines trained to tackle a range of blurring levels. But when information about the blurring level is unavailable or a wrong non-blind model is used, the blind networks allow to achieve the best performance.

3.2. Evaluation of DBSRCNN

Tables 5, 6 and 7 report average PSNR for "Set5", "Set14", and SSIM for "Set14", respectively. One observes a clear improvement of DBSRCNN's performance over SRCNN on blurred images. DBSRCNN architecture allows to improve the quality of the images as measured with PSNR and SSIM. A possible explanation of this performance is that the concatenation of features extracted at an early stage acts similarly to traditional image processing techniques such as unsharp masking that boosts relevant (high) frequencies partially lost in the blurring stage, to enhance the reconstructed image.

Reconstruction examples with various levels of blur are shown in Figure 2 for qualitative comparison: DBSRCNN al-

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	28.29	27.55	25.92	24.46	23.39
Original SRCNN[1]	$\sigma = 0$	**29.53**	28.90	26.53	24.63	23.42
Re-trained SRCNN **non-blind**	$\sigma = 1$	28.18	**29.23**	27.11	24.87	23.54
	$\sigma = 2$	20.66	23.64	**28.16**	25.78	23.92
	$\sigma = 3$	16.86	18.80	23.98	**27.04**	24.88
	$\sigma = 4$	14.75	16.04	19.59	24.35	**25.76**
Re-trained SRCNN **blind**	mixed $\sigma = 1-3$	**26.99**	**28.22**	**27.92**	25.92	24.27
	mixed $\sigma = 1-4$	26.32	27.48	27.62	**26.12**	**24.65**

Table 4. Average of PSNR (dB) for SRCNN on "Set5+Set14"

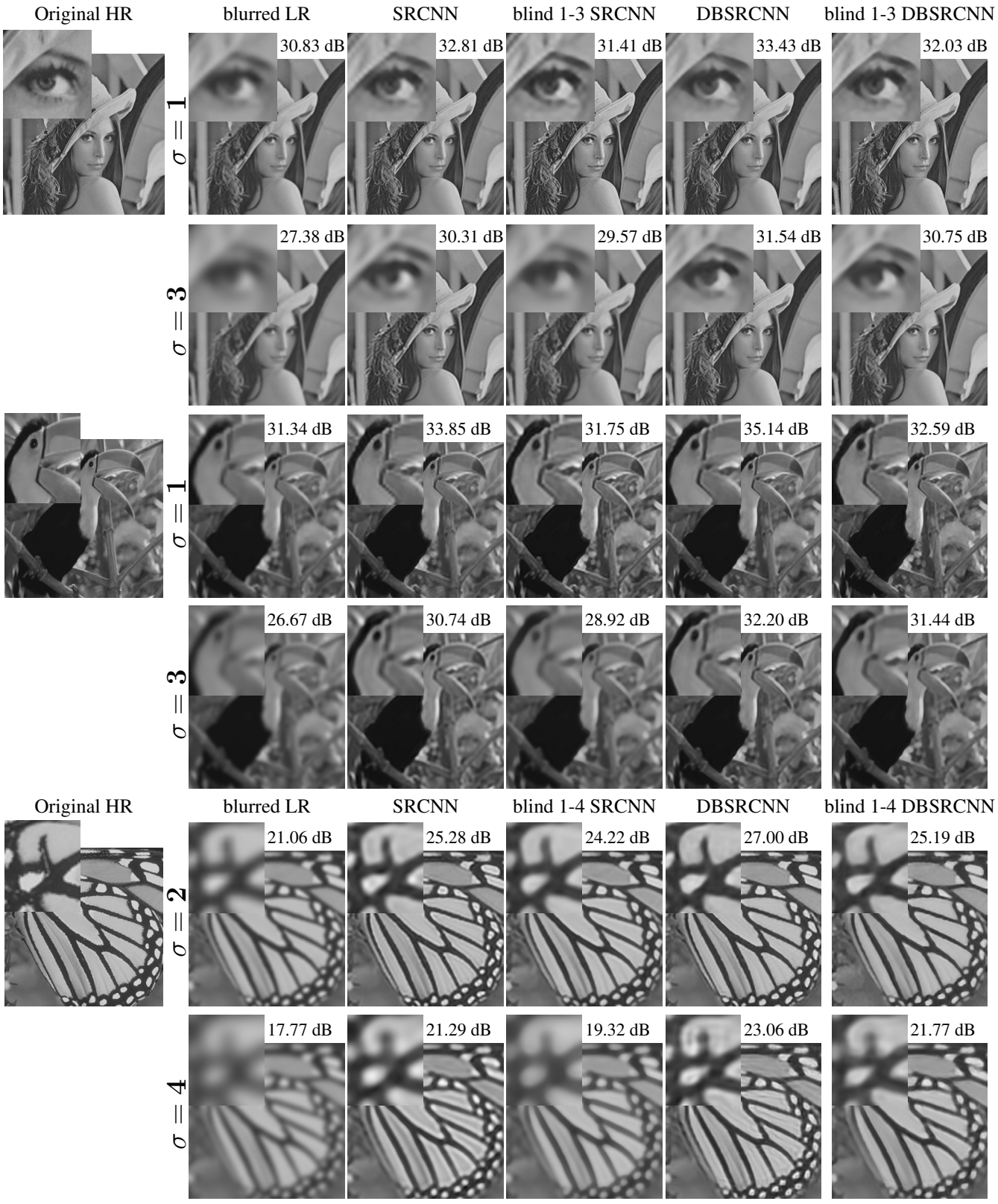

Fig. 2. SR with SRCNN and DBSRCNN on grey-scale images after Gaussian blur with different σ. Third and fifth column show non-blind scenario, and fourth and sixth correspond to blind scenario. Each result is accompanied by zoom and PSNR.

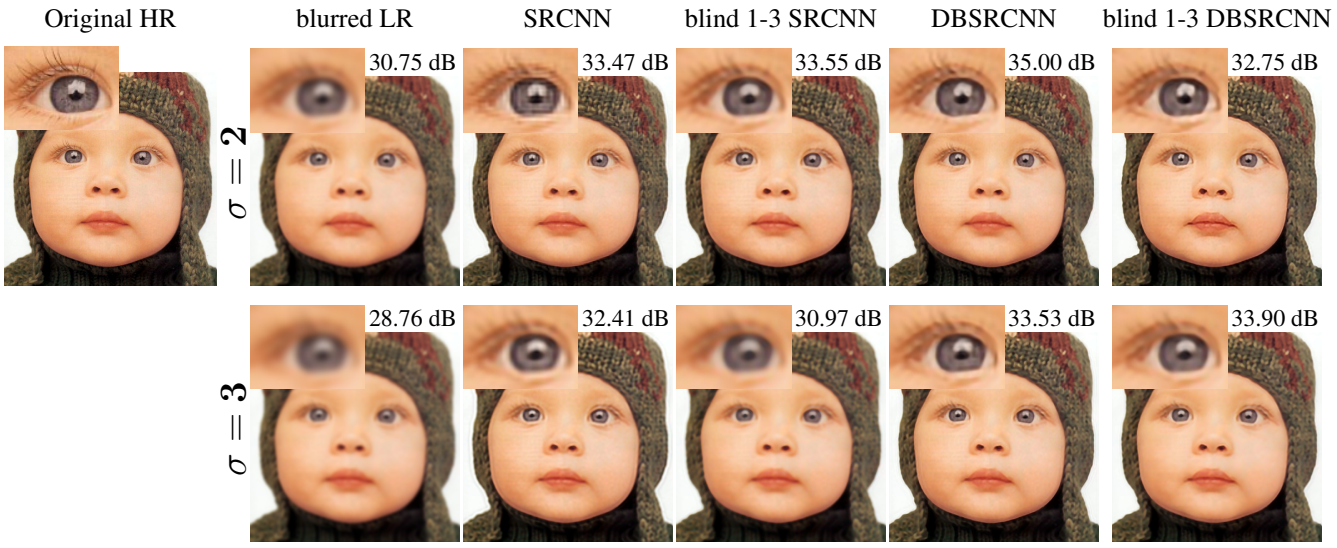

Fig. 3. SR with SRCNN and DBSRCNN on a colour image after Gaussian blur with different σ. Third and fifth column show non-blind scenario, and fourth and sixth correspond to blind scenario. Each result is accompanied by zoom and PSNR.

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	30.40	29.47	27.45	25.65	24.33
DBSRCNN **non-blind**	$\boldsymbol{\sigma = 0}$	32.60				
	custom $\boldsymbol{\sigma}$		32.65	32.09	30.48	28.69
	improvement over non-blind SRCNN	*+0.65*	*+1.10*	*+1.80*	*+1.47*	*+1.34*
DBSRCNN **blind**	mixed $\boldsymbol{\sigma = 1-3}$	**29.89**	**31.24**	**30.14**	**29.51**	26.28
	mixed $\boldsymbol{\sigma = 1-4}$	29.63	30.85	30.08	28.40	**27.72**
	improvement over blind SRCNN	*+1.4*	*+1.19*	*+0.12*	*+1.71*	*+1.81*

Table 5. Average of PSNR (dB) for DBSRCNN on "Set5"

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	27.54	26.86	25.37	24.04	23.05
DBSRCNN **non-blind**	$\boldsymbol{\sigma = 0}$	29.11				
	custom $\boldsymbol{\sigma}$		29.11	28.80	27.50	26.28
	improvement over non-blind SRCNN	*+0.44*	*+0.71*	*+1.39*	*+1.17*	*+1.09*
DBSRCNN **blind**	mixed $\boldsymbol{\sigma = 1-3}$	**27.20**	**28.38**	**27.43**	**26.68**	24.55
	mixed $\boldsymbol{\sigma = 1-4}$	27.10	28.09	27.28	25.99	**25.46**
	improvement over blind SRCNN	*+0.74*	*+0.81*	*+0.27*	*+1.16*	*+1.26*

Table 6. Average of PSNR (dB) for DBSRCNN on "Set14"

lows for a visually better reconstruction than SRCNN confirming the quantitative assessment results reported by PSNR and SSIM. SRCNN has been compared with the earlier state-of-the-art methods (SC, NE+LLE, KK, ANR, A+) in [1], and hence DBSRCNN compares favourably with these as well.

3.3. Extension to colour images

The majority of existing SR methods concentrate on single-band or grey-scale input imagery. There are several main approaches to perform super-resolution on colour images. The straight forward approach is to perform SR separately on the input colour channels and then merge them together into a colour image. Another approach consists in dealing with all three channels in a unified manner by expanding the sizes of layers in the deep architecture ($c = 3$). Finally, colour images can be super-resolved by casting the colour images to YCbCr colour space where the SR is performed solely on the luminance channel Y. Chroma components Cb, and Cr are upscaled by bicubic interpolation, and then all channels (Y, Cb, Cr) are combined again to produce output.

In this work we follow the third strategy and perform SR on Y channel and bicubic interpolation on chroma components. Figure 3 demonstrates an example output of processing a colour input along with relevant comparisons with SRCNN.

Method	Training images	LR	Blur $\sigma = 1$	Blur $\sigma = 2$	Blur $\sigma = 3$	Blur $\sigma = 4$
Upsampling (bicubic)	-	0.8589	0.8325	0.7660	0.6937	0.6337
DBSRCNN **non-blind**	$\boldsymbol{\sigma = 0}$	0.8927				
	custom $\boldsymbol{\sigma}$		0.8930	0.8860	0.8416	0.7882
	improvement over non-blind SRCNN	*+0.0075*	*+0.0114*	*+0.0315*	*+0.0369*	*+0.0391*
DBSRCNN **blind**	mixed $\boldsymbol{\sigma = 1-3}$	**0.8692**	**0.8832**	**0.8488**	**0.8115**	0.7130
	mixed $\boldsymbol{\sigma = 1-4}$	0.8628	0.8775	0.8422	0.7983	**0.7561**
	improvement over blind SRCNN	*+0.0204*	*+0.0125*	*+0.0052*	*+0.0398*	*+0.0592*

Table 7. Average of SSIM for DBSRCNN on "Set14"

4. CONCLUSION

In this work we have extensively evaluated performance of the recently proposed SRCNN architecture for recovering high resolution images from low resolution corrupted by blur / noise. We have proposed a new architecture DBSRCNN that enhances the reconstruction by boosting the relevant features that were originally lost in the SRCNN pipeline. Our experimental study with different levels of Gaussian blur demonstrates that our revised deeper architecture performs better in both blind and non-blind testing scenarios.

5. REFERENCES

[1] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[2] R. Tsai, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing.*, vol. 1, no. 2, pp. 317–339, 1984.

[3] C. Yang, C. Ma, and M. Yang, "Single-image super-resolution: A benchmark," in *Proc. ECCV*. Springer, 2014, pp. 372–386.

[4] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. ICCV*. IEEE, 2009, pp. 349–356.

[5] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Proc. ECCV*. Springer, 2014, pp. 49–64.

[6] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. CVPR*, 2015, pp. 5197–5206.

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

[8] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. BMVC*.

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

[10] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. ICCV*, 2013, pp. 1920–1927.

[11] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. ACCV*. Springer, 2014, pp. 111–126.

[12] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. CVPR*, 2015, pp. 3791–3799.

[13] C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*. Springer, 2014, pp. 184–199.

[14] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. ICCV*, 2015, pp. 370–378.

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

[16] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, 2016, pp. 1646–1654.

[17] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. CVPR*, 2016, pp. 1637–1645.

[18] W. Shi, J. Caballero, F. Huszár, J. Totz, R. Aitken, A.and Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc.CVPR*, 2016, pp. 1874–1883.

[19] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," *arXiv preprint arXiv:1704.03915*, 2017.

[20] B. Lim, S. Son, H. Kim, S. Nah, and K. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. CVPR workshops*. IEEE, 2017, pp. 1132–1140.

[21] F. Abluwi, R. Dahyot, and V. Krylov, "Artifacts reduction in JPEG-Compressed Images using CNNs," in *Proc. Irish Machine Vision and Image Processing Conf. IMVIP*, 2018.

[22] D. P. Kingma and L. Ba, "ADAM: a method for stochastic optimization," in *Proc. ICLR*, 2015.

[23] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. CVPR*. IEEE, 2008, pp. 1–8.

[24] 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.

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