DENSE BYNET: RESIDUAL DENSE NETWORK FOR IMAGE SUPER RESOLUTION

Jiu Xu[1] *Yeongnam Chae*[2] *Björn Stenger*[2] *Ankur Datta*[1]

[1]Rakuten Institute of Technology, Boston
[2]Rakuten Institute of Technology, Tokyo

ABSTRACT

This paper proposes a method, *Dense ByNet*, for single image super-resolution based on a convolutional neural network (CNN). The main innovation is a new architecture that combines several CNN design choices. Using a residual network as a basis, it introduces dense connections inside residual blocks, significantly reducing the number of parameters. Second, we apply dilation convolutions to increase the spatial context. Lastly, we propose modifications to the activation and cost functions. We evaluate the method on benchmark datasets and show that it achieves state-of-the-art results over multiple upscaling factors in terms of peak SNR and structural similarity (SSIM).

Index Terms— image super resolution, convolutional neural networks, dense convolutional network, image enhancement

1 Introduction

The goal of single image super-resolution is to learn a mapping between a low-resolution (LR) and a high-resolution (HR) image. Methods including self-similarity [1, 2], sparse coding [3, 4], and neighborhood embedding [5, 6] have been applied to learn the nonlinear mapping. More recently, convolutional neural networks (CNN) have been applied to the super-resolution task, leading to better performance owing to their outstanding ability to learn highly nonlinear functions. CNNs directly learn an end-to-end mapping between low and high-resolution images. The first *Super Resolution CNN* (SRCNN) [7], introduced an architecture consisting of only two stacked convolution and activation layers. The convolution layer acts as a sparse coding dictionary and learns the best features during training. Various network architectures have been proposed since, and this paper introduces a new CNN, *Dense ByNet*, based on modern residual networks [8], that combines several components, including dense blocks, dilated convolutions, and updated activation and cost functions. In an ablative study we evaluate the contribution of each component. In combination these components lead to increased PSRN and SSIM performance compared to ByNet [9] and LapSRN [10], reduce the number of parameters, and are easy to implement.

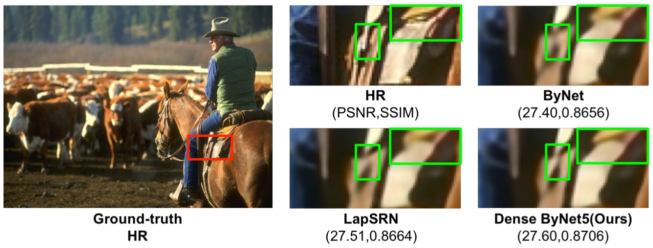

Fig. 1: Example result. *Visual comparison for* ×4 *SR on an example from the* BSD100 *[11] dataset. Regions in green boxes highlight improved recovery of detail.*

2 Prior work

Given the success of CNNs applied to the super-resolution task, prior work has explored various modifications.

Network architectures. Deep recursive convolutional networks, which better preserve image context were proposed in [12]. In [13] a 20-layer network, *VDSR*, was introduced to learn the residual image of LR and HR images. The ByNet architecture [9] uses two types of bypass connections, standard skip connections to jump multiple layers, and skip-connections with an additional convolution. It achieved improved performance over VDSR and is used as baseline model in our comparisons. Laplacian pyramid networks (LapSRN) that progressively predict high-frequency residual were proposed in [10]. It shows good reconstruction performance, but due to its deconvolution operation it has been applied to even scale factors only.

Convolution layers. The *Sub-Pixel CNN* in [14], introduced a sub-pixel convolution layer to substitute the deconvolution layer for up-sampling. Another concept that has been widely applied in CNNs is that of dilated convolutions [15, 16], which increases the size of receptive fields, thereby better taking spatial context into account. Dilated convolutions have been adopted to the super-resolution task in [17, 18], leading to good performance, particularly for large upscaling factors.

Generative adversarial networks. Methods based on Generative adversarial networks (GAN), such as SRGAN [19] and DeblurGAN [20] aim to increase the perceptual score of the reconstructions. The resulting images introduce realism by producing a high-resolution image that is close to the distribution of natural images. This leads to perceptually better results, in particular for large upscaling factors, but results in lower PSNR scores.

Dense connections. *DenseNet* [10] introduced densely connected layers, where feature maps are channel-wise concatenated instead of element-wise summed. Features from all preceding layers are input into subsequent layers. *DenseNet* significantly reduces the total number of parameters by encouraging feature reuse throughout the network. It showed good performance on the ImageNet classification task. Dense skip connection networks were applied to the super-resolution task in [21]. The network applies densely connected layers, but concatenates all preceding layer outputs, resulting in large output feature channels. It also does not support multiple upscaling factors.

This paper combines dense connection block with a bypass residual network, such as the one proposed in [9]. The key modification is replacing the convolutions in the residual bypass connections with convolutional blocks with dense connections. The resulting architecture is shown to improve performance over *ByNet*, while making use of the parameter reduction effect of *DenseNet*.

3 Dense ByNet

The base architecture of Dense ByNet follows that of the ByNet5 model [9]. ByNet5 has one input convolution layer, five residual blocks (two bypass convolutional connection blocks and three bypass skip connection blocks), one output convolution layer, one scaling layer, and one element-wise sum layer. The network is composed of 19 convolution and ReLU layers in total. It takes a gray-scale image X as input, obtained from the LR image via bicubic interpolation. During training ground truth HR image, Y is provided. The network outputs $\widehat{Y}$ with $\widehat{Y} = X + \widehat{R}$, where $\widehat{R}$ is the residual image.

3.1 Residual dense blocks

The original convolution layers in the residual blocks feed input features into a convolutional layer with a 3×3 filter with 64 channels. Thus, each layer has a total of $3 \times 3 \times 64 \times 64 = 36,864$ parameters. We modify the base network by replacing the convolutional layer in the residual blocks with a dense connection block, forming a *residual dense block*, see Fig. 2(c). Each dense connection block is composed of n convolutional layers ($n = 4$ in our case). Given the input feature vector X_t, the output y of a dense connection block is defined as:

$$y = f\left(\left[h_t(X_t), h_{t+1}(X_{t+1}), ..., h_{t+n-1}(X_{t+n-1})\right]\right), \quad (1)$$

where

$$X_{t+1} = f\left(h_t(X_t)\right), \quad (2)$$

$$X_{t+n} = f\left(\left[h_t(X_t), ..., h_{t+n-1}(X_{t+n-1})\right]\right), \quad (3)$$

with activation function f, and

$$h_t(X_t) = w_t * X_t. \quad (4)$$

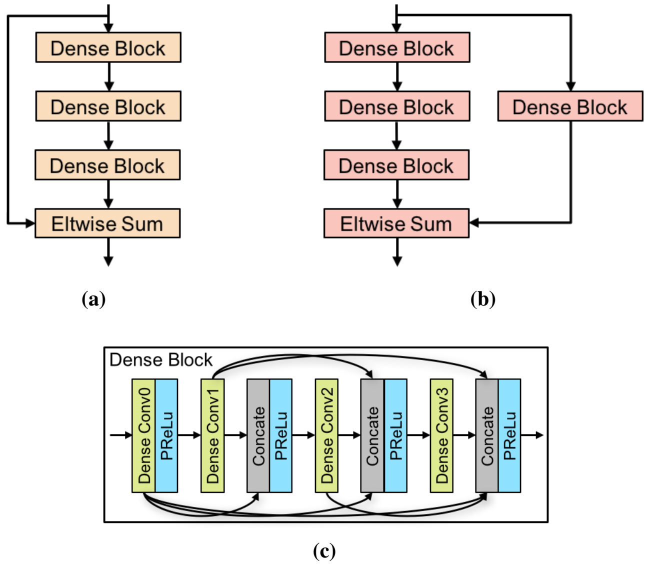

Fig. 2: Residual dense blocks used in Dense ByNet: (a) *feature bypass with dense skip connection* **(b)** *feature bypass with dense convolutional connection* **(c)** *dense connection block.*

Table 1: Number of parameters in a dense connection block.

layer	kernel	plane-in	plane-out	#params
Dense Conv0	3x3	64	16	9,216
Dense Conv1	3x3	16	16	2,304
Dense Conv2	3x3	32	16	4,608
Dense Conv3	3x3	48	16	6,912
Total				23,040

The two types of residual dense blocks are shown in Fig. 2(a) and (b). The input and output dimensions of the residual blocks remain unchanged with respec to *ByNet*. The total number of network parameters is reduced from approximately 37K to 23K, see Table 1.

Figure 3 shows the complete Dense-ByNet5 architecture. Analogous to *ByNet5*, it contains three dense skip connection blocks and two dense convolutional connection blocks. Compared to *ByNet*, dense connection blocks enables efficient reuse of features via channel-wise concatenation.

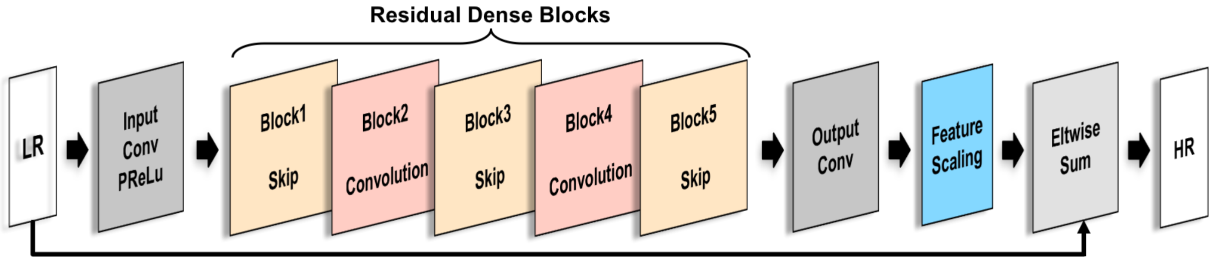

Fig. 3: Network architecture of Dense ByNet5. The proposed CNN learns the non-linear mapping between LR/HR images. The key elements of the architecture are two types of residual dense blocks, which are alternated in sequence.

Dilated convolutions. In order to better take advantage of spatial context we apply a 2-dilated convolution to every convolutional layer in the dense bypass blocks. This allows the features to be computed over receptive fields of multiple sizes. Note that the receptive field size of 2-dilated 3×3 convolutions is the same as 5×5 convolutions, without any additional parameters.

Activation function. Replacing the residual blocks with dense connection blocks increases the number of convolution and attached activation function layers. One issue in learning the residual image for image super-resolution is that many values will be close to zero. The increased number of activation functions exacerbates the vanishing gradient problem, particularly for dense connections. To address this issue, we replace the rectified linear unit (ReLU) activation functions to parametric ReLU (PReLU) [22], which is able to adaptively learns the parameters of the rectifiers.

Loss function A common loss function for image super-resolution is the mean squared error (MSE) [7, 9, 13]. It was previously pointed out [19] that MSE leads to reduced high frequency components, resulting in perceptually unsatisfying solutions with overly smooth textures. This has previously been addressed by using adversarial [24] or perceptual loss functions [25]. However, these loss functions tend to reduce PSNR and SSIM performance, which we aim to increase. Here we use the Charbonnier loss function [26], which has been suggested for use in image restoration tasks in [23]. The loss l_C can be considered a differentiable version of the L_1 norm, which is a robust convex function:

$$l_C(\widehat{Y}, Y) = \left((\widehat{Y} - Y)^2 + \epsilon^2\right)^{\frac{1}{2}}, \qquad (5)$$

where we set the constant ϵ to 10^{-3}. Compared to the MSE loss, this function penalizes outliers less, and is therefore better suited for textured image regions.

4 Results

In this section, we first introduce the datasets and implementation details in order to make the results reproducible. We then compare the proposed method with several recent super-resolution methods. Finally, we present an ablative study to quanitfy the contributions of the different components.

4.1 Datasets

Training data: The training dataset is a combination of the 91 images from the dataset of Yang *et al.* [27] and the 200 images from the *BSD* dataset [28], so that in total of 291 images are used for training. This is in line with prior work [9, 11, 13, 29]. Data augmentation includes mirroring, rotating, and scaling. All training images are partitioned into non-overlapping 41×41 patches, which is consistent with [9, 13].

Test data: We evaluate on three public datasets used in prior work [7, 13], *Set5* [5], *Set14* [4], and *BSD100* [11]. To generate HR/LR image pairs, all HR images are down-sampled using bicubic interpolation with scale factors of 2, 3, and 4, and then up-sampled to obtain pairs of the same size.

4.2 Implementation details

We use Xavier initializion [30] and train the network using mini-batches of size 256. Training is conducted for 40 epochs using stochastic gradient descent with standard momentum of 0.9. The learning rate is initially set to 0.1 and is step-wise annealed using division by 10 every tenth epoch. The parameter for the final feature scaling layer is initialized to 0.1. Following [9, 13], gradient clipping with a threshold value 0.5 is used to avoid the exploding gradient problem. Dense ByNet is implemented in PyTorch and trained on two Nvidia Quadro M6000 GPUs.

4.3 Benchmark comparisons

We re-implemented VDSR [13], LapSRN [23], and ByNet [9][1], and compare the results with our proposed Dense ByNet on the same training and test sets under same conditions. Multi-scale training is applied in VDSR [13], ByNet [9], and Dense ByNet, which means that the input and

[1] The code for VDSR and LapSRN is publicly available [31].

Table 2: Performance comparison on benchmark datasets: *PSNR and SSIM are averaged over all images for each scale. The proposed Dense ByNet model consistently achieves the best PSNR and SSIM results.*

Dataset	Scale	Bicubic PSNR	VDSR[13] PSNR	LapSRN[23] PSNR	ByNet5[9] PSNR	Dense ByNet5 PSNR	Dense ByNet7 PSNR	Bicubic SSIM	VDSR[13] SSIM	LapSRN[23] SSIM	ByNet5[9] SSIM	Dense ByNet5 SSIM	Dense ByNet7 SSIM
Set5	×2	33.66	37.53	37.48	37.61	37.70	**37.76**	0.9299	0.9587	0.9591	0.9597	0.9600	**0.9605**
	×3	30.39	33.66	-	33.79	34.03	**34.09**	0.8682	0.9213	-	0.9235	0.9254	**0.9261**
	×4	28.42	31.35	31.65	31.44	31.66	**31.74**	0.8104	0.8838	0.8889	0.8860	0.8891	**0.8902**
Set14	×2	30.24	33.13	33.08	33.20	33.25	**33.34**	0.9056	0.9124	0.9127	0.9143	0.9147	**0.9153**
	×3	27.55	29.92	-	29.95	30.04	**30.07**	0.8188	0.8314	-	0.8348	0.8363	**0.8370**
	×4	26.00	28.20	28.26	28.18	28.27	**28.29**	0.7491	0.7674	0.7735	0.7711	0.7738	**0.7742**
BSD100	×2	29.56	31.92	31.85	31.93	31.99	**32.04**	0.8431	0.8960	0.8948	0.8970	0.8978	**0.8984**
	×3	27.21	28.86	-	28.86	28.92	**28.96**	0.7385	0.7976	-	0.7994	0.8010	**0.8017**
	×4	25.96	27.31	27.36	27.30	27.37	**27.40**	0.6675	0.7251	0.7291	0.7269	0.7293	**0.7301**

Table 3: Ablation Study. We evaluate the contribution of the different components in terms of PSNR performance on the *Set5* dataset. The baseline model is ByNet5 [9]. We separately and jointly include dilated convolutions, parametric ReLU (PReLU), Charbonnier loss, and dense blocks.

Dilated Conv	PReLU	Charb Loss	Dense Block	PSNR 2×	 3×	 4×
				37.61	33.79	31.44
✓				37.62	33.87	31.57
	✓			37.60	33.80	31.45
		✓		37.70	33.81	31.45
			✓	37.60	33.76	31.43
✓	✓			37.63	33.92	31.56
✓		✓		37.74	33.97	31.59
✓			✓	37.55	33.89	31.56
	✓	✓		37.71	33.81	31.45
	✓		✓	37.70	33.88	31.52
		✓	✓	37.63	33.72	31.35
✓	✓	✓		37.72	33.93	31.56
✓	✓		✓	37.63	33.90	31.59
✓		✓	✓	37.58	33.89	31.56
	✓	✓	✓	**37.77**	33.89	31.56
✓	✓	✓	✓	37.70	**34.03**	**31.66**

output images have the same size, 41×41 in our implementation. LapSRN [23] takes 32×32 patches as input and outputs 64×64 and 128×128 patches for scales 2 and 4, respectively.

Table 2 shows quantitative results on the luminance channel, in terms of PSNR (in dB) and structural similarity (SSIM) for different upscaling factors. Note that due to the deconvolution operation, LapSRN does yield results for scale factor 3 directly. In [32] these are obtained by down-sampling results for scale factor 4. Dense ByNet5 consistently achieves better results than VDSR, LapSRN, and ByNet5. By adding two more residual dense blocks the performance can be improved further, as shown in the columns *Dense ByNet7*.

In terms of efficiency, Dense ByNet5 contains 393K network parameters, and Dense ByNet7 contains 554K. Both are lower than ByNet5 with 628K, VDSR with 665K, or LapSRN with 812K parameters. Training Dense ByNet5 takes three days, while at test time processing an image from BSD100 dataset takes 10ms on average.

4.4 Ablation study

We carry out an ablation study to evaluate the contribution of each modification over the baseline network. These factors include dense block, dilated convolution, PReLU activation, and Charbonnier loss. In this experiment, all models are tested on the *Set5* dataset with a patch size of 41×41.

Table 3 lists the PSNR scores of all combinations of the four components. Compared with the baseline model (ByNet5, results are shown in the first row of the table), using dilated convolution yields better performance, particularly for larger scale factors. Charbonnier loss, on the other hand, consistently improves the scores. Applying dense connection blocks together with PReLU leads to PSNR improvement. Interestingly, we observe that changing activation functions from ReLU to PReLU alone does affect performance significantly. Introducing dense connection blocks alone even sightly harms the performance. The reason may be that the dense blocks increase the number of activation functions. The parameterized ReLU leads to more specialized activation functions which benefit the proposed dense-in-residual structure. Finally, by adding all components, Dense ByNet5 outperforms ByNet5 with an average PSNR score of 0.18, at a 38% reduction in the number of parameters.

5 Conclusion

This paper introduced *Dense ByNet*, a network architecture for image super-resolution. The key contribution is the combination of a residual block network architecture as in *ByNet* [9] and dense connection layes as in *DenseNet* [10]. This leads to a significant reduction in the number of network parameters, however at slightly reduced performance. We show that by replacing ReLU with parametric ReLU (PReLU) [22], the performance can actually be increased. Other modifications include dilated convolutions for larger receptive fields and a more suitable loss function, the Charbonnier loss, further increase performance. Experiments on standard benchmarks show that *Dense ByNet* achieves state-of-the-art performance in terms of PSNR and SSIM on the image super-resolution task, while being highly computationally efficient.

6 References

[1] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *ICCV*, 2009.

[2] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *CVPR*, 2015.

[3] J. Yang, Z. Wang, Z. Lin, and S. Cohen, "Coupled dictionary training for image super-resolution," in *TIP*, 2012, vol. 21, pp. 3467–3478.

[4] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, 2012, vol. 6920, pp. 711–730.

[5] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi, "Low-complexity single-image super-resolution based on non-negative neighbor embedding," in *BMVC*, 2012.

[6] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *CVPR*, 2016.

[7] C. Dong, C.C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," in *TPAMI*, 2015, vol. 38, pp. 295–307.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[9] J. Xu, Y. Chae, and B. Stenger, "ByNet-SR: Image super resolution with a bypass connection network," in *ICIP*, 2017.

[10] G. Huang, Z. Liu, L. Van der Maaten, and K. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

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

[12] J. Kim, J.K. Lee, and K.M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *CVPR*, 2016.

[13] J. Kim, J.K. Lee, and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016.

[14] W. Shi, J. Caballero, F. Huszár, 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 *CVPR*, 2016.

[15] Y. Fisher and K. Vladlen, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[16] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *CVPR*, 2017.

[17] W. Shi, F. Jiang, and D. Zhao, "Single image super-resolution with dilated convolution based multi-scale information learning inception module," in *ICIP*, 2017.

[18] Z. Huang, L. Wang, G. Meng, and C. Pan, "Image super-resolution via deep dilated convolutional networks," in *ICIP*, 2017.

[19] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.

[20] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," *arXiv 1711.07064*, 2017.

[21] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *ICCV*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015.

[23] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *CVPR*, 2017.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[25] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016.

[26] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *ICIP*, 1994.

[27] J. Yang, J. Wright, T. Huang, and Ma. Y., "Image super resolution via sparse representation," in *TIP*, 2010, vol. 19, pp. 2861–2873.

[28] 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 *ICCV*, 2001.

[29] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *CVPR*, 2015.

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[31] J. Xu, "https://github.com/twtygqyy," accessed on Feb 12, 2018.

[32] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *arXiv:1710.01992*, 2017.