Deep Back-Projection Networks For Super-Resolution

Muhammad Haris[1], Greg Shakhnarovich[2], and Norimichi Ukita[1],
[1]Toyota Technological Institute, Japan [2]Toyota Technological Institute at Chicago, United States
{mharis, ukita}@toyota-ti.ac.jp, greg@ttic.edu

Abstract

The feed-forward architectures of recently proposed deep super-resolution networks learn representations of low-resolution inputs, and the non-linear mapping from those to high-resolution output. However, this approach does not fully address the mutual dependencies of low- and high-resolution images. We propose Deep Back-Projection Networks (DBPN), that exploit iterative up- and down-sampling layers, providing an error feedback mechanism for projection errors at each stage. We construct mutually-connected up- and down-sampling stages each of which represents different types of image degradation and high-resolution components. We show that extending this idea to allow concatenation of features across up- and down-sampling stages (Dense DBPN) allows us to reconstruct further improve super-resolution, yielding superior results and in particular establishing new state of the art results for large scaling factors such as 8× across multiple data sets.

1. Introduction

Significant progress in deep learning for vision [15, 13, 5, 39, 26, 33, 17] has recently been propagating to the field of super-resolution (SR) [19, 29, 6, 12, 20, 21, 24, 42].

Single image SR is an ill-posed inverse problem where the aim is to recover a high-resolution (HR) image from a low-resolution (LR) image. A currently typical approach is to construct an HR image by learning non-linear LR-to-HR mapping, implemented as a deep neural network [6, 7, 37, 24, 21, 22, 42]. These networks compute a sequence of feature maps from the LR image, culminating with one or more upsampling layers to increase resolution and finally construct the HR image. In contrast to this purely feed-forward approach, human visual system is believed to use a feedback connection to simply guide the task for the relevant results [9, 23, 25]. Perhaps hampered by lack of such feedback, the current SR networks with only feed-forward connections have difficulty in representing the LR to HR relation, especially for large scaling factors.

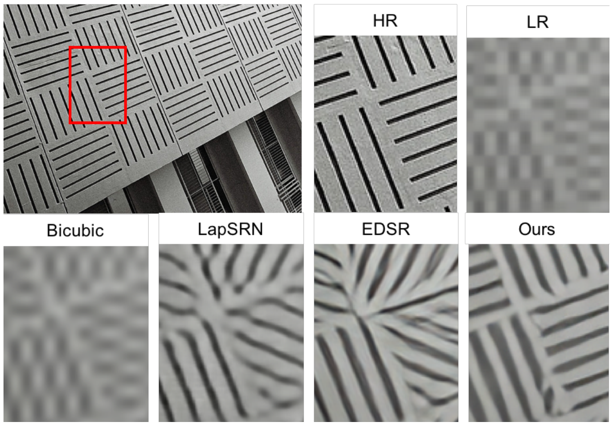

Figure 1. Super-resolution result on 8× enlargement. PSNR: LapSRN [24] (15.25 dB), EDSR [30] (15.33 dB), and Ours (16.63 dB)

On the other hand, feedback connections were used effectively by one of the early SR algorithms, the iterative back-projection [18]. It iteratively computes the reconstruction error then fuses it back to tune the HR image intensity. Although it has been proven to improve the image quality, the result still suffers from ringing effect and chessboard effect [4]. Moreover, this method is sensitive to choices of parameters such as the number of iterations and the blur operator, leading to variability in results.

Inspired by [18], we construct an end-to-end trainable architecture based on the idea of iterative up- and down-sampling: Deep Back-Projection Networks (DBPN). Our networks successfully perform large scaling factors, as shown in Fig. 1. Our work provides the following contributions:

(1) **Error feedback**. We propose an iterative error-correcting feedback mechanism for SR, which calculates both up- and down-projection errors to guide the reconstruction for obtaining better results. Here, the projection errors are used to characterize or constraint the features in early layers. Detailed explanation can be seen in Section 3.

(2) **Mutually connected up- and down-sampling stages**. Feed-forward architectures, which is considered as a one-way mapping, only map rich representations of the input to

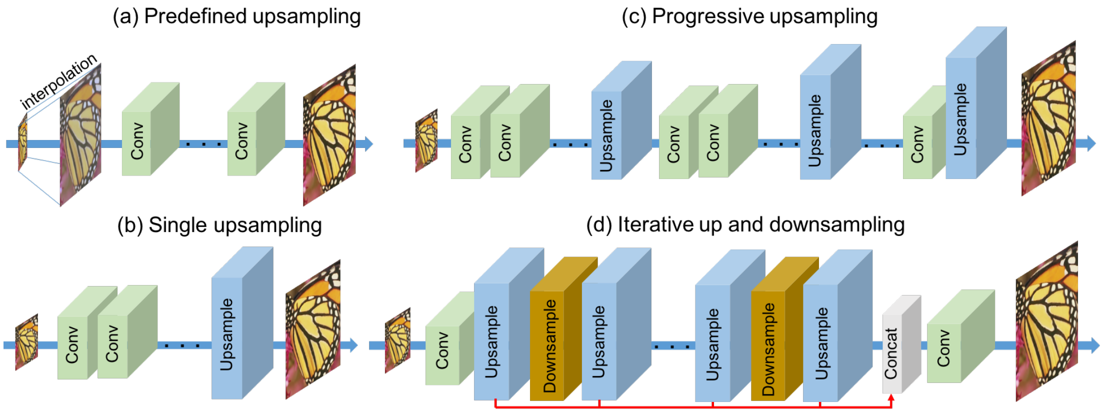

Figure 2. Comparisons of Deep Network SR. (a) Predefined upsampling (e.g., SRCNN [6], VDSR [21], DRRN [42]) commonly uses the conventional interpolation, such as Bicubic, to upscale LR input images before entering the network. (b) Single upsampling (e.g., FSRCNN [7], ESPCN [37]) propagates the LR features, then construct the SR image at the last step. (c) Progressive upsampling uses a Laplacian pyramid network which gradually predicts SR images [24]. (d) Iterative up and downsampling approach is proposed by our DBPN which exploit the mutually connected up- (blue box) and down-sampling (gold box) stages to obtain numerous HR features in different depths.

the output space. This approach is unsuccessful to map LR and HR image, especially in large scaling factors, due to limited features available in the LR spaces. Therefore, our networks focus not only generating variants of the HR features using upsampling layers but also projecting it back to the LR spaces using downsampling layers. This connection is shown in Fig. 2 (d), alternating between up- (blue box) and down-sampling (gold box) stages, which represent the mutual relation of LR and HR image.

(3) **Deep concatenation**. Our networks represent different types of image degradation and HR components. This ability enables the networks to reconstruct the HR image using deep concatenation of the HR feature maps from all of the up-sampling steps. Unlike other networks, our reconstruction directly utilizes different types of LR-to-HR features without propagating them through the sampling layers as shown by the red arrow in Fig. 2 (d).

(4) **Improvement with dense connection**. We improve the accuracy of our network by densely connected [15] each up- and down-sampling stage to encourage feature reuse.

2. Related Work

2.1. Image super-resolution using deep networks

Deep Networks SR can be primarily divided into four types as shown in Fig. 2.

(a) **Predefined upsampling** commonly uses interpolation as the upsampling operator to produce middle resolution (MR) image. This schema was firstly proposed by SRCNN [6] to learn MR-to-HR non-linear mapping with simple convolutional layers. Later, the improved networks exploited residual learning [21, 42] and recursive layers [22]. However, this approach might produce new noise from the MR image.

(b) **Single upsampling** offers simple yet effective way to increase the spatial resolution. This approach was proposed by FSRCNN [7] and ESPCN [37]. These methods have been proven effective to increase the spatial resolution and replace predefined operators. However, they fail to learn complicated mapping due to limited capacity of the networks. EDSR [30], the winner of NTIRE2017 [43], belongs to this type. However, it requires a large number of filters in each layer and lengthy training time, around eight days as stated by the authors. These problems open the opportunities to propose lighter networks that can preserve HR components better.

(c) **Progressive upsampling** was recently proposed in LapSRN [24]. It progressively reconstructs the multiple SR images with different scales in one feed-forward network. For the sake of simplification, we can say that this network is the stacked of single upsampling networks which only relies on limited LR features. Due to this fact, LapSRN is outperformed even by our shallow networks especially for large scaling factors such as 8× in experimental results.

(d) **Iterative up and downsampling** is proposed by our networks. We focus on increasing the sampling rate of SR features in different depths and distribute the tasks to calculate the reconstruction error to each stage. This schema enables the networks to preserve the HR components by learning various up- and down-sampling operators while generating deeper features.

2.2. Feedback networks

Rather than learning a non-linear mapping of input-to-target space in one step, the feedback networks compose the prediction process into multiple steps which allow the model to have a self-correcting procedure. Feedback procedure has been implemented in various computing tasks [3, 34, 46, 28, 48, 38, 31].

In the context of human pose estimation, Carreira et al. [3] proposed an iterative error feedback by iteratively estimating and applying a correction to the current estimation. PredNet [31] is an unsupervised recurrent network to predictively code the future frames by recursively feeding the predictions back into the model. For image segmentation, Li et al. [28] learn implicit shape priors and use them to improve the prediction. However, to our knowledge, feedback procedures have not been implemented to SR.

2.3. Adversarial training

Adversarial training, such as with Generative Adversarial Networks (GANs) [10] has been applied to various image reconstruction problems [27, 36, 33, 5, 19]. For the SR task, Johnson et al. [19] introduced perceptual losses based on high-level features extracted from pre-trained networks. Ledig et al. [27] proposed SRGAN which is considered as a single upsampling method. It proposed the natural image manifold that is able to create photo-realistic images by specifically formulating a loss function based on the euclidian distance between feature maps extracted from VGG19 [40] and SRResNet.

Our networks can be extended with the adversarial loss as generator network. However, we optimize our network only using an objective function such as mean square root error (MSE). Therefore, instead of training DBPN with the adversarial loss, we can compare DBPN with SRResNet which is also optimized by MSE.

2.4. Back-projection

Back-projection [18] is well known as the efficient iterative procedure to minimize the reconstruction error. Previous studies have proven the effectivity of back-projection [50, 11, 8, 45]. Originally, back-projection is designed for the case with multiple LR inputs. However, given only one LR input image, the updating procedure can be obtained by upsampling the LR image using multiple upsampling operators and calculate the reconstruction error iteratively [4]. Timofte et al. [45] mentioned that back-projection can improve the quality of SR image. Zhao et al. [50] proposed a method to refine high-frequency texture details with an iterative projection process. However, the initialization which leads to an optimal solution remains unknown. Most of the previous studies involve constant and unlearnable predefined parameters such as blur operator and number of iteration.

To extend this algorithm, we develop an end-to-end trainable architecture which focuses to guide the SR task using mutually connected up- and down-sampling stages to learn non-linear relation of LR and HR image. The mutual relation between HR and LR image is constructed by creating iterative up and down-projection unit where the up-projection unit generates HR features, then the down-projection unit projects it back to the LR spaces as shown in Fig. 2 (d). This schema enables the networks to preserve the HR components by learned various up- and down-sampling operators and generates deeper features to construct numerous LR and HR features.

3. Deep Back-Projection Networks

Let I^h and I^l be HR and LR image with $(M \times N)$ and $(M^{'} \times N^{'})$, respectively, where $M^{'} < M$ and $N^{'} < N$. The main building block of our proposed DBPN architecture is the projection unit, which is trained (as part of the end-to-end training of the SR system) to map either an LR feature map to an HR map (up-projection), or an HR map to an LR map (down-projection).

3.1. Projection units

The up-projection unit is defined as follows:

$$\text{scale up:} \qquad H_0^t = (L^{t-1} * p_t) \uparrow_s, \tag{1}$$

$$\text{scale down:} \qquad L_0^t = (H_0^t * g_t) \downarrow_s, \tag{2}$$

$$\text{residual:} \qquad e_t^l = L_0^t - L^{t-1}, \tag{3}$$

$$\text{scale residual up:} \qquad H_1^t = (e_t^l * q_t) \uparrow_s, \tag{4}$$

$$\text{output feature map:} \qquad H^t = H_0^t + H_1^t \tag{5}$$

where * is the spatial convolution operator, $\uparrow_s$ and $\downarrow_s$ are, respectively, the up- and down-sampling operator with scaling factor s, and p_t, g_t, q_t are (de)convolutional layers at stage t.

This projection unit takes the previously computed LR feature map L^{t-1} as input, and maps it to an (intermediate) HR map H_0^t; then it attempts to map it back to LR map L_0^t ("back-project"). The residual (difference) e_t^l between the observed LR map L^{t-1} and the reconstructed L_0^t is mapped to HR again, producing a new intermediate (residual) map H_1^t; the final output of the unit, the HR map H^t, is obtained by summing the two intermediate HR maps. This step is illustrated in the upper part of Fig. 3.

The down-projection unit is defined very similarly, but now its job is to map its input HR map H^t to the LR map L^t as illustrated in the lower part of Fig. 3.

$$\text{scale down:} \qquad L_0^t = (H^t * g_t^{\prime}) \downarrow_s, \tag{6}$$

$$\text{scale up:} \qquad H_0^t = (L_0^t * p_t^{\prime}) \uparrow_s, \tag{7}$$

$$\text{residual:} \qquad e_t^h = H_0^t - H^t, \tag{8}$$

$$\text{scale residual down:} \qquad L_1^t = (e_t^h * g_t^{\prime}) \downarrow_s, \tag{9}$$

$$\text{output feature map:} \qquad L^t = L_0^t + L_1^t \tag{10}$$

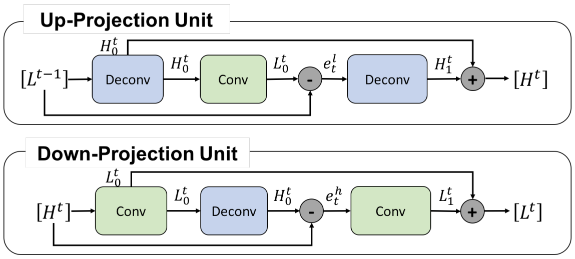

Figure 3. Proposed up- and down-projection unit in the DBPN.

We organize projection units in a series of *stages*, alternating between H and L. These projection units can be understood as a self-correcting procedure which feeds a projection error to the sampling layer and iteratively changes the solution by feeding back the projection error.

The projection unit uses large sized filters such as 8×8 and 12×12. In other existing networks, the use of large-sized filter is avoided because it slows down the convergence speed and might produce sub-optimal results. However, iterative utilization of our projection units enables the network to suppress this limitation and to perform better performance on large scaling factor even with shallow networks.

3.2. Dense projection units

The dense inter-layer connectivity pattern in DenseNets [15] has been shown to alleviate the vanishing-gradient problem, produce improved feature, and encourage feature reuse. Inspired by this we propose to improve DBPN, by introducing dense connections in the projection units called, yielding Dense DBPN (D-DBPN).

Unlike the original DenseNets, we avoid dropout and batch norm, which are not suitable for SR, because they remove the range flexibility of the features [30]. Instead, we use 1×1 convolution layer as feature pooling and dimensional reduction [41, 12] before entering the projection unit.

In D-DBPN, the input for each unit is the concatenation of the outputs from all previous units. Let the $L^{\tilde{t}}$ and $H^{\tilde{t}}$ be the input for dense up- and down-projection unit, respectively. They are generated using $conv(1, n_R)$ which is used to merge all previous outputs from each unit as shown in Fig. 4. This improvement enables us to generate the feature maps effectively, as shown in the experimental results.

3.3. Network architecture

The proposed D-DBPN is illustrated in Fig. 5. It can be divided into three parts: initial feature extraction, projection, and reconstruction, as described below. Here, let $conv(f, n)$ be a convolutional layer, where f is the filter size and n is the number of filters.

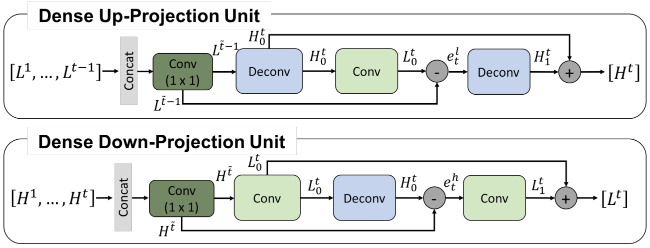

Figure 4. Proposed up- and down-projection unit in the D-DBPN. The feature maps of all preceding units (i.e., $[L^1, ..., L^{t-1}]$ and $[H^1, ..., H^t]$ in up- and down-projections units, respectively) are concatenated and used as inputs, and its own feature maps are used as inputs into all subsequent units.

1. **Initial feature extraction**. We construct initial LR feature-maps L^0 from the input using $conv(3, n_0)$. Then $conv(1, n_R)$ is used to reduce the dimension from n_0 to n_R before entering projection step where n_0 is the number of filters used in the initial LR features extraction and n_R is the number of filters used in each projection unit.

2. **Back-projection stages**. Following initial feature extraction is a sequence of projection units, alternating between construction of LR and HR feature maps H^t, L^t; each unit has access to the outputs of all previous units.

3. **Reconstruction**. Finally, the target HR image is reconstructed as $I^{sr} = f_{Rec}([H^1, H^2, ..., H^t])$, where f_{Rec} use $conv(3, 3)$ as reconstruction and $[H^1, H^2, ..., H^t]$ refers to the concatenation of the feature-maps produced in each up-projection unit.

Due to the definitions of these building blocks, our network architecture is modular. We can easily define and train networks with different numbers of stages, controlling the depth. For a network with T stages, we have the initial extraction stage (2 layers), and then T up-projection units and $T - 1$ down-projection units, each with 3 layers, followed by the reconstruction (one more layer). However, for the dense network, we add $conv(1, n_R)$ in each projection unit, except the first three units.

4. Experimental Results

4.1. Implementation and training details

In the proposed networks, the filter size in the projection unit is various with respect to the scaling factor. For $2\times$ enlargement, we use 6×6 convolutional layer with two striding and two padding. Then, $4\times$ enlargement use 8×8 convolutional layer with four striding and two padding. Finally, the $8\times$ enlargement use 12×12 convolutional layer with eight striding and two padding.[1]

[1] We found these settings to work well based on general intuition and preliminary experiments.

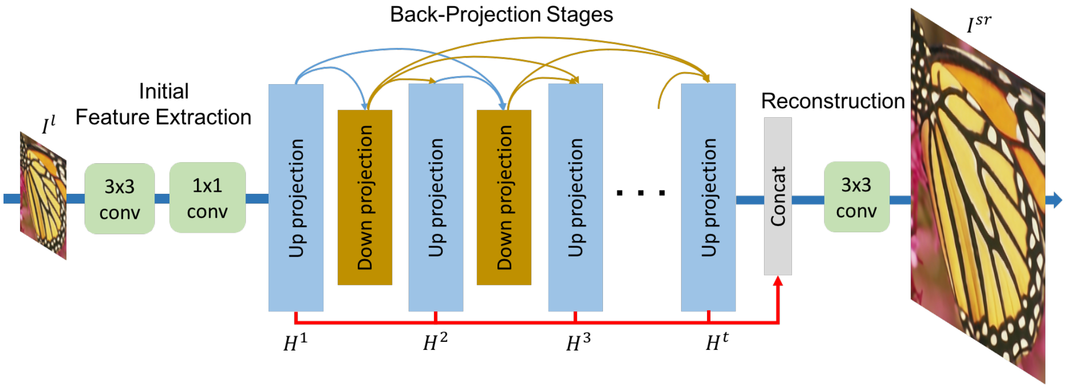

Figure 5. An implementation of D-DBPN for super-resolution. Unlike the original DBPN, D-DBPN exploits densely connected projection unit to encourage feature reuse.

We initialize the weights based on [14]. Here, std is computed by $(\sqrt{2/n_l})$ where $n_l = f_t^2 n_t$, f_t is the filter size, and n_t is the number of filters. For example, with $f_t = 3$ and $n_t = 8$, the std is 0.111. All convolutional and deconvolutional layers are followed by parametric rectified linear units (PReLUs).

We trained all networks using images from DIV2K [43], Flickr [30], and ImageNet dataset [35] without augmentation.[2] To produce LR images, we downscale the HR images on particular scaling factors using Bicubic. We use batch size of 20 with size 32×32 for LR image, while HR image size corresponds to the scaling factors. The learning rate is initialized to $1e-4$ for all layers and decrease by a factor of 10 for every 5×10^5 iterations for total 10^6 iterations. For optimization, we use Adam with momentum to 0.9 and weight decay to $1e-4$. All experiments were conducted using Caffe, MATLAB R2017a on NVIDIA TITAN X GPUs.

4.2. Model analysis

Depth analysis. To demonstrate the capability of our projection unit, we construct multiple networks S $(T = 2)$, M $(T = 4)$, and L $(T = 6)$ from the original DBPN. In the feature extraction, we use $conv(3, 128)$ followed by $conv(1, 32)$. Then, we use $conv(1, 1)$ for the reconstruction. The input and output image are luminance only.

The results on $4\times$ enlargement are shown in Fig. 6. DBPN outperforms the state-of-the-art methods. Starting from our shallow network, the S network gives the higher PSNR than VDSR, DRCN, and LapSRN. The S network uses only 12 convolutional layers with smaller number of filters than VDSR, DRCN, and LapSRN. At the best performance, S networks can achieve 31.59 dB which better 0.24 dB, 0.06 dB, 0.05 dB than VDSR, DRCN, and LapSRN, respectively. The M network shows performance improvement which better than all four existing state-of-the-art methods (VDSR, DRCN, LapSRN, and DRRN). At the best performance, the M network can achieve 31.74 dB which better 0.39 dB, 0.21 dB, 0.20 dB, 0.06 dB than VDSR, DRCN, LapSRN, and DRRN respectively. In total, the M network use 24 convolutional layers which has the same depth as LapSRN. Compare to DRRN (up to 52 convolutional layers), the M network undeniable shows the effectiveness of our projection unit. Finally, the L network outperforms all methods with 31.86 dB which better 0.51 dB, 0.33 dB, 0.32 dB, 0.18 dB than VDSR, DRCN, LapSRN, and DRRN, respectively.

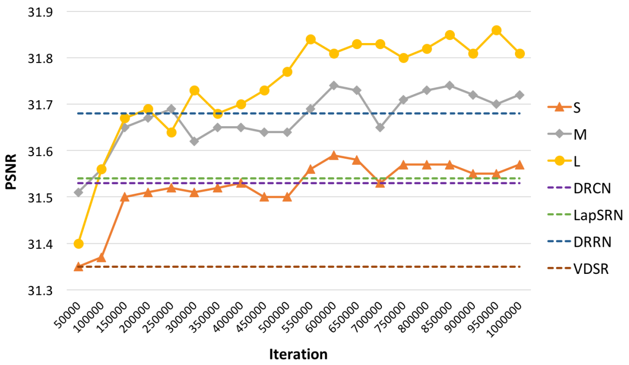

Figure 6. The depth analysis of DBPNs compare to other networks (VDSR [21], DRCN [22], DRRN [42], LapSRN [24]) on Set5 dataset for $4\times$ enlargement.

The results of $8\times$ enlargement are shown in Fig. 7. The S, M, L networks outperform the current state-of-the-art for $8\times$ enlargement which clearly show the effectiveness of our proposed networks on large scaling factors. However, we found that there is no significant performance gain from each proposed network especially for L and M networks where the difference only 0.04 dB.

Number of parameters. We show the tradeoff between performance and number of network parameters from our networks and existing deep network SR in Fig. 8 and 9.

[2] The comparison with only DIV2K dataset are available in the supplementary material.

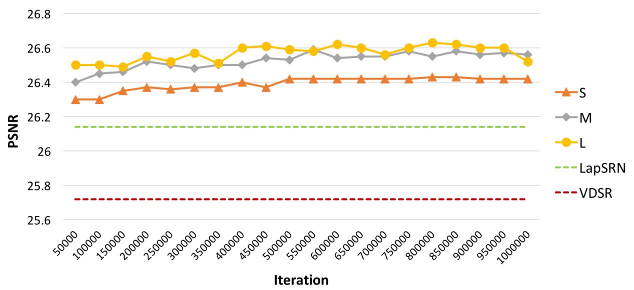

Figure 7. The depth analysis of DBPN on Set5 dataset for 8× enlargement. S ($T = 2$), M ($T = 4$), and L ($T = 6$)

For the sake of low computation for real-time processing, we construct SS network which is the lighter version of the S network, ($T = 2$). We only use $conv(3, 64)$ followed by $conv(1, 18)$ for the initial feature extraction. However, the results outperform SRCNN, FSRCNN, and VDSR on both 4× and 8× enlargement. Moreover, our SS network performs better than VDSR with 72% and 37% fewer parameters on 4× and 8× enlargement, respectively.

Our S network has about 27% fewer parameters and higher PSNR than LapSRN on 4× enlargement. Finally, D-DBPN has about 76% fewer parameters, and approximately the same PSNR, compared to EDSR on 4× enlargement. On the 8× enlargement, D-DBPN has about 47% fewer parameters with better PSNR compare to EDSR. This evidence show that our networks has the best trade-off between performance and number of parameter.

Deep concatenation. Each projection unit is used to distribute the reconstruction step by constructing features which represent different details of the HR components. Deep concatenation is also well-related with the number of T (back-projection stage), which shows more detailed features generated from the projection units will also increase the quality of the results. In Fig. 10, it is shown that each stage successfully generates diverse features to reconstruct SR image.

Dense connection. We implement D-DBPN-L which is a dense connection of the L network to show how dense connection can improve the network's performance in all cases as shown in Table 1. On 4× enlargement, the dense network, D-DBPN-L, gains 0.13 dB and 0.05 dB higher than DBPN-L on the Set5 and Set14, respectively. On 8×, the gaps are even larger. The D-DBPN-L has 0.23 dB and 0.19 dB higher that DBPN-L on the Set5 and Set14, respectively.

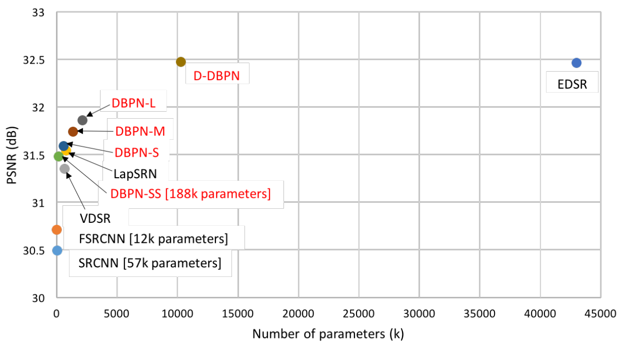

Figure 8. Performance vs number of parameters. The results are evaluated with Set5 dataset for 4× enlargement.

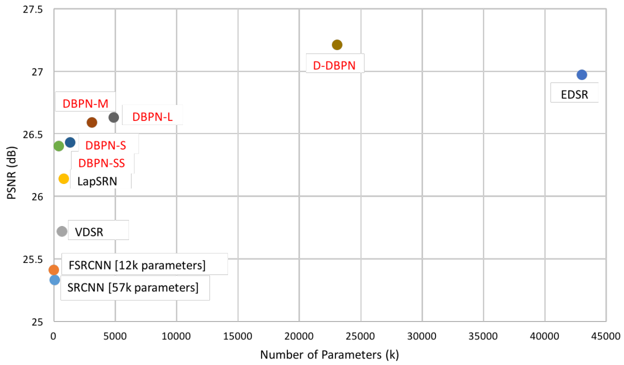

Figure 9. Performance vs number of parameters. The results are evaluated with Set5 dataset for 8× enlargement.

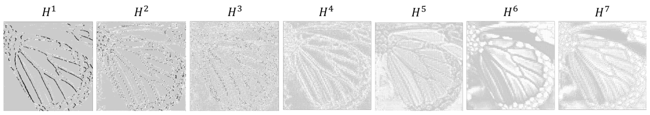

Figure 10. Sample of activation maps from up-projection units in D-DBPN where $t = 7$. Each feature has been enhanced using the same grayscale colormap for visibility.

Table 1. Comparison of the DBPN-L and D-DBPN-L on 4× and 8× enlargement. Red indicates the best performance.

		Set5		Set14	
Algorithm	Scale	PSNR	SSIM	PSNR	SSIM
DBPN-L	4	31.86	0.891	28.47	0.777
D-DBPN-L	4	31.99	0.893	28.52	0.778
DBPN-L	8	26.63	0.761	24.73	0.631
D-DBPN-L	8	26.86	0.773	24.92	0.638

4.3. Comparison with the-state-of-the-arts

To confirm the ability of the proposed network, we performed several experiments and analysis. We compare our network with eight state-of-the-art SR algorithms: A+ [44], SRCNN [6], FSRCNN [7], VDSR [21], DRCN [22], DRRN [42], LapSRN [24], and EDSR [30]. We carry out extensive experiments using 5 datasets: Set5 [2], Set14 [49], BSDS100 [1], Urban100 [16] and Manga109 [32]. Each dataset has different characteristics. Set5, Set14 and BSDS100 consist of natural scenes; Urban100 contains urban scenes with details in different frequency bands; and Manga109 is a dataset of Japanese

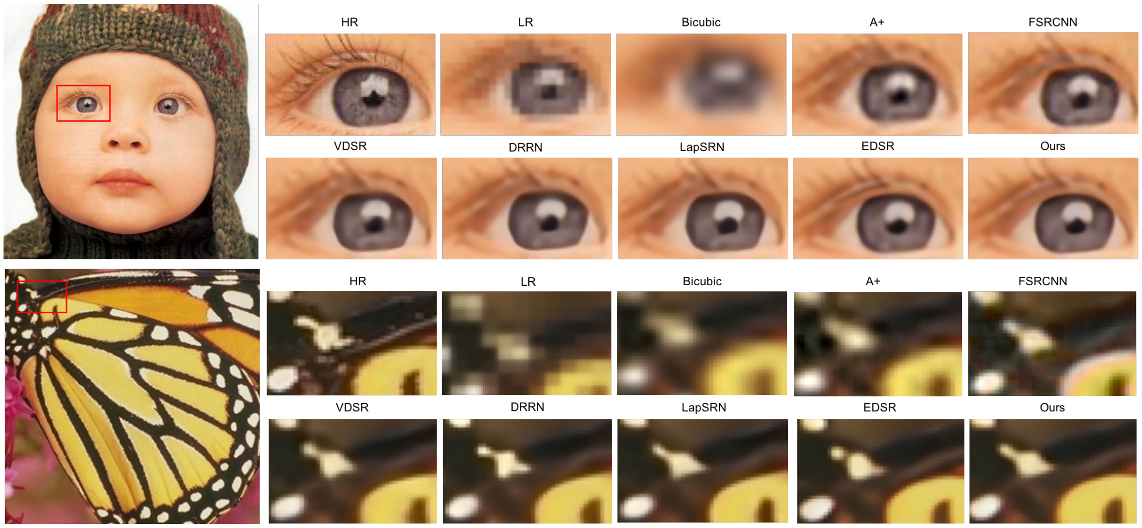

Figure 11. Qualitative comparison of our models with other works on $4\times$ super-resolution.

manga. Due to computation limit of Caffe, we have to divide each image in Urban100 and Manga109 into four parts and then calculate PSNR separately.

Our final network, D-DBPN, uses $conv(3, 256)$ then $conv(1, 64)$ for the initial feature extraction and $t = 7$ for the back-projection stages. In the reconstruction, we use $conv(3, 3)$. RGB color channels are used for input and output image. It takes less than four days to train.

PSNR and structural similarity (SSIM) [47] were used to quantitatively evaluate the proposed method. Note that higher PSNR and SSIM values indicate better quality. As used by existing networks, all measurements used only the luminance channel (Y). For SR by factor s, we crop s pixels near image boundary before evaluation as in [30, 7]. Some of the existing networks such as SRCNN, FSRCNN, VDSR, and EDSR did not perform $8\times$ enlargement. To this end, we retrained the existing networks by using author's code with the recommended parameters.

We show the quantitative results in the Table 2. D-DBPN outperforms the existing methods by a large margin in all scales except EDSR. For the $2\times$ and $4\times$ enlargement, we have comparable PSNR with EDSR. However, EDSR tends to generate stronger edge than the ground truth and lead to misleading information in several cases. The result of EDSR for eyelashes in Fig. 11 shows that it was interpreted as a stripe pattern. On the other hand, our result generates softer patterns which subjectively closer to the ground truth. On the butterfly image, EDSR separates the white pattern which shows that EDSR tends to construct regular pattern such ac circle and stripe, while D-DBPN constructs the same pattern as the ground truth. The previous statement is strengthened by the results from the Urban100 dataset which consist of many regular patterns from buildings. In Urban100, EDSR has 0.54 dB higher than D-DBPN.

Our network shows it's effectiveness in the $8\times$ enlargement. The D-DBPN outperforms all of the existing methods by a large margin. Interesting results are shown on Manga109 dataset where D-DBPN obtains 25.50 dB which is 0.61 dB better than EDSR. While on the Urban100 dataset, D-DBPN achieves 23.25 which is only 0.13 dB better than EDSR. The results show that our networks perform better on fine-structures images such as manga characters, even though we do not use any animation images in the training.

The results of $8\times$ enlargement are visually shown in Fig. 12. Qualitatively, D-DBPN is able to preserve the HR components better than other networks. It shows that our networks can extract not only features but also create contextual information from the LR input to generate HR components in the case of large scaling factors, such as $8\times$ enlargement.

5. Conclusion

We have proposed Deep Back-Projection Networks for Single Image Super-resolution. Unlike the previous methods which predict the SR image in a feed-forward manner, our proposed networks focus to directly increase the SR features using multiple up- and down-sampling stages and feed the error predictions on each depth in the networks to revise the sampling results, then, accumulates the self-correcting features from each upsampling stage to create SR image. We use error feedbacks from the up- and down-scaling steps to guide the network to achieve a better result. The results show the effectiveness of the proposed network compares to other state-of-the-art methods. Moreover, our proposed network successfully outperforms other state-of-the-art methods on large scaling factors such as $8\times$ enlargement.

This work was partly supported by FCRAL.

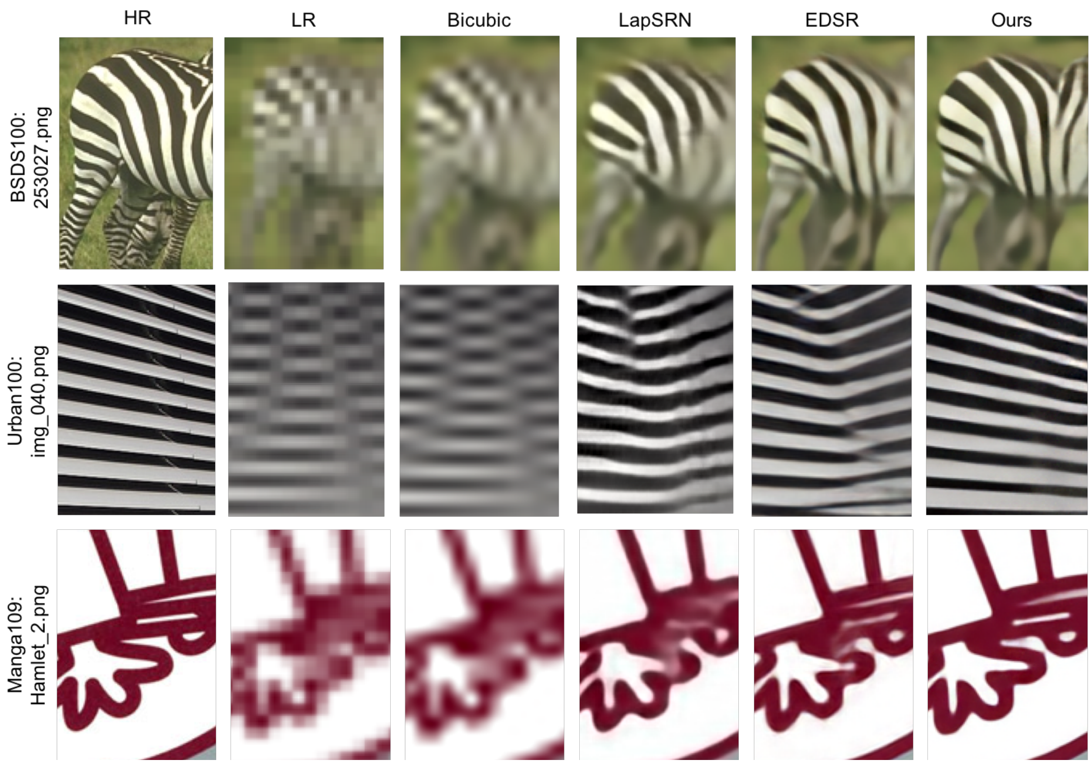

Figure 12. Qualitative comparison of our models with other works on 8× super-resolution. 1^{st} line: LapSRN [24] (19.77 dB), EDSR [30] (19.79 dB), and Ours (19.82 dB). 2^{nd} line: LapSRN [24] (16.45 dB), EDSR [30] (19.1 dB), and Ours (23.1 dB). 3^{rd} line: LapSRN [24] (24.34 dB), EDSR [30] (25.29 dB), and Ours (28.84 dB)

Table 2. Quantitative evaluation of state-of-the-art SR algorithms: average PSNR/SSIM for scale factors 2×, 4× and 8×. Red indicates the best and blue indicates the second best performance. (* indicates that the input is divided into four parts and calculated separately due to computation limitation of Caffe)

		Set5		Set14		BSDS100		Urban100		Manga109	
Algorithm	Scale	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	2	33.65	0.930	30.34	0.870	29.56	0.844	26.88 (27.39*)	0.841	30.84 (31.05*)	0.935
A+ [44]	2	36.54	0.954	32.40	0.906	31.22	0.887	29.23	0.894	35.33	0.967
SRCNN [6]	2	36.65	0.954	32.29	0.903	31.36	0.888	29.52	0.895	35.72	0.968
FSRCNN [7]	2	36.99	0.955	32.73	0.909	31.51	0.891	29.87	0.901	36.62	0.971
VDSR [21]	2	37.53	0.958	32.97	0.913	31.90	0.896	30.77	0.914	37.16	0.974
DRCN [22]	2	37.63	0.959	32.98	0.913	31.85	0.894	30.76	0.913	37.57	0.973
DRRN [42]	2	37.74	0.959	33.23	0.913	32.05	0.897	31.23	0.919	37.92	0.976
LapSRN [24]	2	37.52	0.959	33.08	0.913	31.80	0.895	30.41 (31.05*)	0.910	37.27 (37.53*)	0.974
EDSR [30]	2	38.11	0.960	33.92	0.919	32.32	0.901	32.93 (33.56*)	0.935	39.10 (39.33*)	0.977
D-DBPN	2	38.09	0.960	33.85	0.919	32.27	0.900	– (33.02*)	0.931	– (39.32*)	0.978
Bicubic	4	28.42	0.810	26.10	0.704	25.96	0.669	23.15 (23.64*)	0.659	24.92 (25.15*)	0.789
A+ [44]	4	30.30	0.859	27.43	0.752	26.82	0.710	24.34	0.720	27.02	0.850
SRCNN [6]	4	30.49	0.862	27.61	0.754	26.91	0.712	24.53	0.724	27.66	0.858
FSRCNN [7]	4	30.71	0.865	27.70	0.756	26.97	0.714	24.61	0.727	27.89	0.859
VDSR [21]	4	31.35	0.882	28.03	0.770	27.29	0.726	25.18	0.753	28.82	0.886
DRCN [22]	4	31.53	0.884	28.04	0.770	27.24	0.724	25.14	0.752	28.97	0.886
DRRN [42]	4	31.68	0.888	28.21	0.772	27.38	0.728	25.44	0.764	29.46	0.896
LapSRN [24]	4	31.54	0.885	28.19	0.772	27.32	0.728	25.21 (25.87*)	0.756	29.09 (29.44*)	0.890
EDSR [30]	4	32.46	0.897	28.80	0.788	27.71	0.742	26.64 (27.30*)	0.803	31.02 (31.41*)	0.915
D-DBPN	4	32.47	0.898	28.82	0.786	27.72	0.740	– (27.08*)	0.795	– (31.50*)	0.914
Bicubic	8	24.39	0.657	23.19	0.568	23.67	0.547	20.74 (21.24*)	0.516	21.47 (21.68*)	0.647
A+ [44]	8	25.52	0.692	23.98	0.597	24.20	0.568	21.37	0.545	22.39	0.680
SRCNN [6]	8	25.33	0.689	23.85	0.593	24.13	0.565	21.29	0.543	22.37	0.682
FSRCNN [7]	8	25.41	0.682	23.93	0.592	24.21	0.567	21.32	0.537	22.39	0.672
VDSR [21]	8	25.72	0.711	24.21	0.609	24.37	0.576	21.54	0.560	22.83	0.707
LapSRN [24]	8	26.14	0.738	24.44	0.623	24.54	0.586	21.81 (22.42*)	0.582	23.39 (23.67*)	0.735
EDSR [30]	8	26.97	0.775	24.94	0.640	24.80	0.596	22.47 (23.12*)	0.620	24.58 (24.89*)	0.778
D-DBPN	8	27.21	0.784	25.13	0.648	24.88	0.601	– (23.25*)	0.622	– (25.50*)	0.799

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 6

[2] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference (BMVC)*, 2012. 6

[3] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016. 3

[4] S. Dai, M. Han, Y. Wu, and Y. Gong. Bilateral back-projection for single image super resolution. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1039–1042. IEEE, 2007. 1, 3

[5] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 1, 3

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 1, 2, 6, 8

[7] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016. 1, 2, 6, 7, 8

[8] W. Dong, L. Zhang, G. Shi, and X. Wu. Nonlocal back-projection for adaptive image enlargement. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 349–352. IEEE, 2009. 3

[9] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991. 1

[10] 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*, pages 2672–2680, 2014. 3

[11] M. Haris, M. R. Widyanto, and H. Nobuhara. First-order derivative-based super-resolution. *Signal, Image and Video Processing*, 11(1):1–8, 2017. 3

[12] M. Haris, M. R. Widyanto, and H. Nobuhara. Inception learning super-resolution. *Appl. Opt.*, 56(22):6043–6048, Aug 2017. 1, 4

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1

[14] 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*, pages 1026–1034, 2015. 5

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 4

[16] 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*, pages 5197–5206, 2015. 6

[17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 1

[18] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991. 1, 3

[19] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 1, 3

[20] 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, 2016. 1

[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*, pages 1646–1654, June 2016. 1, 2, 5, 6, 8

[22] 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*, pages 1637–1645, 2016. 1, 2, 5, 6, 8

[23] D. J. Kravitz, K. S. Saleem, C. I. Baker, L. G. Ungerleider, and M. Mishkin. The ventral visual pathway: an expanded neural framework for the processing of object quality. *Trends in cognitive sciences*, 17(1):26–49, 2013. 1

[24] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conferene on Computer Vision and Pattern Recognition*, 2017. 1, 2, 5, 6, 8

[25] V. A. Lamme and P. R. Roelfsema. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in neurosciences*, 23(11):571–579, 2000. 1

[26] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016. 1

[27] C. Ledig, L. Theis, F. Huszár, 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 3

[28] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3659–3667, 2016. 3

[29] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video super-resolution via deep draft-ensemble learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 531–539, 2015. 1

[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, July 2017. 1, 2, 4, 5, 6, 7, 8

[31] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 3

[32] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, pages 1–28, 2016. 6

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1, 3

[34] S. Ross, D. Munoz, M. Hebert, and J. A. Bagnell. Learning message-passing inference machines for structured prediction. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2737–2744. IEEE, 2011. 3

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[36] M. S. Sajjadi, B. Schölkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. *arXiv preprint arXiv:1612.07919*, 2016. 3

[37] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 1, 2

[38] A. Shrivastava and A. Gupta. Contextual priming and feedback for faster r-cnn. In *European Conference on Computer Vision*, pages 330–348. Springer, 2016. 3

[39] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 4

[42] 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*, 2017. 1, 2, 5, 6, 8

[43] 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*, pages 1110–1121. IEEE, 2017. 2, 5

[44] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014. 6, 8

[45] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016. 3

[46] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1744–1757, 2010. 3

[47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004. 7

[48] A. R. Zamir, T.-L. Wu, L. Sun, W. Shen, J. Malik, and S. Savarese. Feedback networks. *arXiv preprint arXiv:1612.09508*, 2016. 3

[49] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. Springer, 2012. 6

[50] Y. Zhao, R.-G. Wang, W. Jia, W.-M. Wang, and W. Gao. Iterative projection reconstruction for fast and efficient image upsampling. *Neurocomputing*, 226:200–211, 2017. 3