

Hierarchical Back Projection Network for Image Super-Resolution

Zhi-Song Liu, Li-Wen Wang, Chu-Tak Li, and Wan-Chi Siu
The Hong Kong Polytechnic University
Hung Hom, Hong Kong

{zhisong.liu, liwen.wang, ron.li}@connect.polyu.hk, enwcsiu@polyu.edu.hk

Abstract

Deep learning based single image super-resolution methods use a large number of training datasets and have recently achieved great quality progress both quantitatively and qualitatively. Most deep networks focus on nonlinear mapping from low-resolution inputs to high-resolution outputs via residual learning without exploring the feature abstraction and analysis. We propose a Hierarchical Back Projection Network (HBPN), that cascades multiple Hour-Glass (HG) modules to bottom-up and top-down process features across all scales to capture various spatial correlations and then consolidates the best representation for reconstruction. We adopt the back projection blocks in our proposed network to provide the error correlated up- and down-sampling process to replace simple deconvolution and pooling process for better estimation. A new Soft-max based Weighted Reconstruction (WR) process is used to combine the outputs of HG modules to further improve super-resolution. Experimental results on various datasets (including the validation dataset, NTIRE2019, of the Real Image Super-resolution Challenge) show that our proposed approach can achieve and improve the performance of the state-of-the-art methods for different scaling factors.

1. Introduction

Single Image Super-Resolution (SISR) attracts a lot of attention in the research community in the past few years. It is a fundamental low-level vision problem where the aim is to form a high-resolution (HR) image \mathbf{Y} from a low-resolution (LR) image \mathbf{X} . Usually, SISR is described as an ill-posed problem $\mathbf{X} = \mathbf{H}\mathbf{Y} + \mu$, where \mathbf{H} is a down-sampling operator, μ is additive white Gaussian noise with standard deviation σ .

To resolve the ill-posed problem, Super-Resolution (SR) images can be obtained in the perspective of model-based optimization [35, 8, 7, 5, 9] and discriminative learning methods [6, 16, 29, 18, 20, 10]. The model-based opti-

mization can be formulated as,

$$\hat{\mathbf{Y}} = \arg \min_{\mathbf{Y}} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\mathbf{Y}\|^2 + \lambda \Omega(\mathbf{Y}) \quad (1)$$

where λ is the regularization factor that controls the significance of the regularization term $\Omega(\mathbf{Y})$. Though model-based optimization methods are flexible to handle different SR condition and noise, they are usually time-consuming and require various priors.

On the contrary, discriminative approaches use external or internal paired LR-HR training samples to directly learn the nonlinear relationship. The objective is given by

$$\min_{\hat{\mathbf{Y}}, \mathbf{W}} \ell(\hat{\mathbf{Y}}, \mathbf{Y}) \text{ s.t. } \hat{\mathbf{Y}} = \arg \min_{\mathbf{Y}} \frac{1}{2} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|^2 + \lambda \Omega(\mathbf{W}) \quad (2)$$

where \mathbf{W} is the mapping model for reconstruction. The fidelity term $\arg \min_{\mathbf{Y}} \frac{1}{2} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|^2$ determines the distortion of reconstruction and similarly, the regularization term $\Omega(\mathbf{W})$ controls the complexity of the mapping model. In the previous research works, patch-based approaches use classification tools, like kNN [4], to classify the patches from natural images and capture the mapping relationship for clustered patches. Taking the advantage of non-local statistical priors from external datasets, there are many successful approaches that achieve good SR performance by off-line training classifiers and regressors for efficient on-line reconstruction. For example, Timofte *et al.* [30, 31] proposed the adjusted anchored neighbor regression (ANR and A+) which uses clustering on encoded sparse dictionary to search nearest neighbor dictionary atoms for LR patch reconstruction. Siu *et al.* [15, 14, 21, 22] proposed random forests for binary classification to obtain fast and high qualified image SR and hierarchical decision trees to further boost up image SR performance.

Since Dong *et al.* [6] proposed the first deep convolution neural network (CNN) for image SR, a large number of CNN based SR approaches have been proposed to significantly improve the image SR performance. Along with the development of other computing vision fields, i.e., image classification, object detection and so on, more deep and

complex models are adopted in image SR. For example, VDSR [16] uses a 20-layer convolution network for different up-sampling factors. Tai *et al.* [29] proposed a deep recursive residual network by using recursive blocks to explore long-term correlations between LR and HR images. LapSRN [18] uses Laplacian pyramid networks to gradually super-resolve LR with different up-sampling factors. Most recently, Haris *et al.* [10] proposed Deep Back Projection Network (DBPN) for image SR by iteratively computing reconstruction errors then, fusing them back for model tuning.

Inspired by [10], we design a Hierarchical Back Projection Network (HBPN) for image SR. As shown in Figure 1, our work have following contributions:

- **Enhanced back projection blocks.** We propose an enhanced back projection block, including the new Up-sampling Back Projection block (UBP) and Down-sampling Back Projection block (DBP). Both UBPs and DBPs embed the back projection mechanism in the residual block to update up-sampling and down-sampling errors for better results. The key modification is two 1×1 convolution layers within the back projection block to fine tune the LR and HR features. Details are explained in Section 3.
- **Hierarchical SR HourGlass (SR-HG) module.** We stack multiple stages of SR-HG modules to capture various spatial correlations by repeated bottom-up and top-down process across all scales. Different from HG structure used in other applications [24, 25], we replace the pooling and deconvolution layer by enhanced back projection blocks for better feature down- and up-sampling process.
- **Softmax based Weighted Reconstruction (WR).** To encourage different SR-HG modules super-resolve LR images in a hierarchical order, each SR-HG module outputs one coarse SR result and one weighting map. At the final WR stage, we propose to use Softmax layer to normalize weighting maps from different SR-HG modules to obtain the global weighting map. Finally, we consolidate all coarse SR results by using the global weighting map to output the final SR image.

2. Related Work

In order to compare the different SR reconstruction measurements, we can divide the convolutional neural network based SR approaches into distortion based SR and perception based SR.

2.1. Distortion based image super-resolution

As discussed in Section 1, to resolve Equation 2, the end-to-end CNN model is a very direct and efficient method.

By inputting LR images, we can define a mean squared errors based loss function to target on optimizing the convolutional parameters to obtain the SR outputs with minimal distortion. Considering the mismatch of dimension between LR and HR images, there are different designs of CNN models for SR. In the early stage of CNN for image SR, researchers inherited the knowledge on traditional machine learning based SR approaches by initially up-sampling LR images to the desired size by simple interpolation, i.e., Bicubic, and then learn the mapping model between the up-sampled LR and HR images. SRCNN [6] and many other CNN approaches [16, 29, 18, 20, 10] use this idea to build networks using cascaded convolution process. In order to grasp long-term correlation of pixels for reconstruction, we need to stack more convolution layers to cover a larger receptive field. However, building deeper convolution networks can encounter computation exploding and gradient vanishing problems. To resolve the former problem, Kim *et al.* [17] and Tai *et al.* [29] proposed to use recursive convolution networks to increase recursion depth rather than convolution depth without introducing new parameter for computation. For the latter problem, residual learning [11] is introduced in CNN models to add shortcuts to avoid gradient vanishing. In recent SR works, Lim *et al.* [20] proposed a state-of-the-art CNN network using residual blocks to achieve good SR performance on various datasets.

Rather than using initial interpolation to up-sample LR image to feed into CNN for training, there have also been some novel CNN works that build up-sampling process into CNN models. The deconvolution with stride larger than 1 is used in CNN working as an up-sampling process, Lai *et al.* [18] proposed a Laplacian Pyramid network to gradually super-resolve LR image by different scales. Shi *et al.* [27], on the other hand, proposed the sub-pixel convolution process to work as a pixel based interpolation for enlargement. Recently, Haris *et al.* [10] further studied the residual learning on image SR and proposed the back projection based residual block that can efficiently learn LR and HR feature maps iteratively to feedback residual errors.

2.2. Perception based image super-resolution

Rather than targeting on minimizing mean squared errors based loss function, perception based image SR focuses on visual quality over data fidelity. Since a pioneer work on using the Generative Adversarial Network (GAN) for image SR [19], there are a lot of studies on using adversarial loss as a measurement for SR performance. By replacing the l_n -norm minimization by distribution divergence, we force the SR networks to learn the meaningful features rather than pixel differences. The idea of using GAN for image SR can be described as: the generator and discriminator learn from each other to generate a “fake” SR image that gives mini-

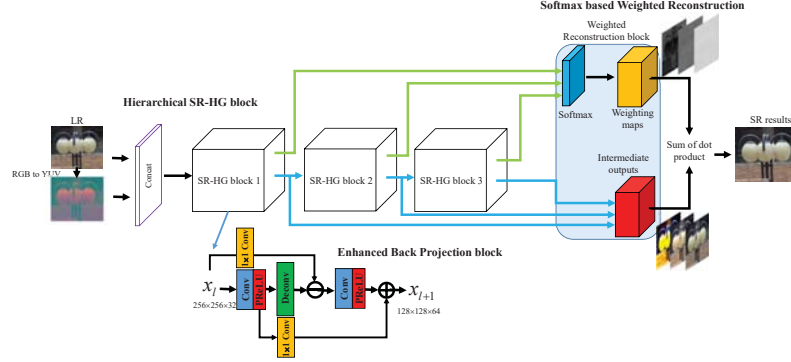


Figure 1. Proposed HBPN structure. In a bottom-up and top-down manner, it can explore various scales to extract hierarchical features for image SR.

mal distance on the high-level feature space (features used commonly extracted from VGG19 [28]). Wang *et al.* [33] further investigated this study. They modified the generator by using Residual-in-Residual Dense Block to improve SR performance in terms of PSNR (one measurement of distortion) and then they fine tuned the network by using adversarial loss to generate SR image with better visual quality.

From recent studies of GAN for image SR, one of the key issues is still the design of generators. A good generator should be able to extract rich feature maps for estimation by any criteria. Our proposed network can also be considered as a perception based image SR by using adversarial loss. However, the measurement of visual quality was only used in $4\times$ image SR [19, 26, 33]. To make a good comparison, we still use distortion based evaluation (PSNR, SSIM, etc.) to make analysis among different approaches.

3. Hierarchical Back Projection Network

Before introducing our proposed work, let us first define some terms. As defined in Section 1, given a RGB LR image \mathbf{X} with size $h \times w$, we want to super-resolve it by $\alpha \times$ to the dimension $\alpha h \times \alpha w$, the HR image \mathbf{Y} . The super-resolved image is the SR image $\hat{\mathbf{Y}}$.

3.1. Back projection

Let us first revisit the back projection approach that has been commonly used in image SR. Back projection was first proposed to utilize multiple LR images to estimate one SR image. [10] comes up with using back projection to refine SR image to improve the quality. It is an efficient iterative process to improve the data fidelity of SR by minimizing the loss between the original LR image and the down-sampled SR image. Mathematically, description of the back projection is

$$\hat{\mathbf{Y}}_{t+1} = \hat{\mathbf{Y}}_t - \lambda \mathbf{H}^{-1}(\mathbf{H}\hat{\mathbf{Y}}_t - \mathbf{X}) \quad (3)$$

where \mathbf{H}^{-1} is the inverse operator of \mathbf{H} which represents the up-sampling operation process. For estimating the SR residues, we need to assume a certain known down-

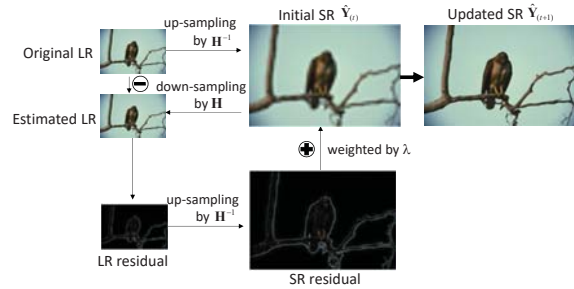


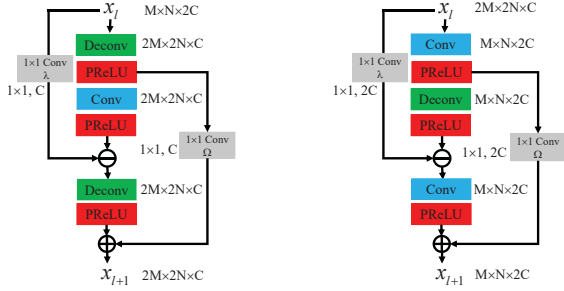
Figure 2. Back Projection procedure.

sampling and up-sampling operators. λ is the trade-off parameter to control the ratio of the residual information to gradually improve the SR quality. t is the iteration number. A simple back projection process is shown in Figure 2. Back projection has been widely used in many SR approaches as a final refinement to reduce the distortion in terms of PSNR. However, it is observed that the down- and up-sampling operators need to be pre-determined as fixed parameters for estimation which may not obtain optimal results. To resolve this problem, [10] proposes to embed the back projection into CNN model to learn the unknown parameters by training. By using multiple proposed back projection blocks, it can expand the iterative process as a cascading process using more parameters to minimize the SR residual information. Our study further develops this work by coming up with a hierarchical back projection network to learn LR and HR features across different scales to extract more compact and robust features for reconstruction.

3.2. Enhanced back projection blocks

Let us propose our Enhanced back projection blocks, which contain both new Up-sampling Back Projection (UBP) and Down-sampling Back Projection (DBP) blocks. The UBP is the forward back projection process that estimates HR residues while the DBP is the backward back projection process that estimates LR residues. The details

of two blocks are shown in Figure 3.



(a) Up-sampling Back Projection (UBP) block (b) Down-sampling Back Projection (DBP) block

Figure 3. Proposed Enhanced back projection blocks: (a) the UBP block that up-samples the LR feature maps by $2\times$ and reduces the number of feature maps by half, and (b) the DBP block that down-samples the HR feature maps by half and increases the number of feature maps by $2\times$.

The process of UBP block can be described by rewriting Equation 3 as,

$$x_{l+1} = \Omega D x_l + D (\lambda x_l - C D x_l) \quad (4)$$

Similarly, the process of DBP block can be considered as the backward of UBP that estimates the LR residues as Equation 5,

$$x_{l+1} = \Omega C x_l + C (\lambda x_l - D C x_l) \quad (5)$$

There are two key modifications between our proposed Enhanced back projection blocks and that in DBPN [10]: global weighting model Ω and residual weighting model λ . They all use 1×1 convolution layers to work as the weighting process.

For the residual weighting model λ , it resembles the trade-off parameter in Equation 3 that provides the regularization on the update of SR residues. Without followed by any activation function, this 1×1 convolution layer is a linear weighted model that can tune the residual information without increasing any computation burden.

For the global weighting model, it has two jobs: first, to work as a weighted model to tune the down- and up-sampled features for update so that we can introduce one extra freedom of parameters for training; second, to adjust the channel (number) of feature maps for addition. For example, from (a) in Figure 3, the global weighting model reduces the number of feature maps by half. From (b) in Figure 3, the global weighting model doubles the number of feature maps for addition.

3.3. Hierarchical SR HourGlass (SR-HG) module

For the proposed SR-HG module, we adopt the HourGlass structure to cascade multiple enhanced back projection blocks in bottom-up and top-down manner. The HourGlass structure is commonly used in many computing vision fields. By down-sampling the size of feature maps

while increasing the number of feature maps, we can extract denser and deeper features for various applications. The key differences of our proposed SR-HG module are three folds: 1) replacing pooling process by DBP blocks to avoid information loss, 2) replacing the single convolution process by DBP blocks to down-sample the feature maps and 3) output a coarse SR image and a weighting map.

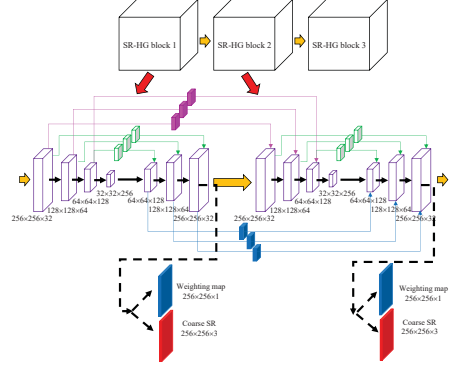


Figure 4. Proposed SR-HG module structure.

The complete structure of SR-HG is shown in Figure 4. For each SR-HG module, it contains 3 DBP blocks for down-sampling process and 3 UBP blocks for up-sampling process. For DBP and UBP blocks with same feature dimension, we use 1×1 convolution as local shortcuts (green blocks) to share the features. For different SR-HG blocks, we use 1×1 convolution as global shortcuts (pink and blue blocks) to share features across different modules. For each SR-HG module, there are two branches (dash lines in Figure 4) to generate one coarse SR result and one weighting map to describe the contribution of the coarse SR. There are global and local shortcuts that share the features across different HourGlass modules and spatial scales. Each SR-HG module contains 3 UBP blocks for up-sampling and 3 DBP blocks for down-sampling and each UBP/DBP block up-/down-samples the input data by $2\times$. Totally, the input data are first down-sampled by $8\times$ and then up-sampled by $8\times$. In the meantime, the number of features are first increased by $8\times$ and then decreased by $8\times$ so that the network can learn denser and more compact features for reconstruction.

3.4. Softmax based Weighted Reconstruction (WR)

For the final reconstruction, instead of concatenating coarse SR results from different SR-HG modules to generate the final SR by one convolution layer, we propose a Softmax based Weighted Reconstruction (WR) that makes use of the weighting maps to estimate the contribution of coarse SR results. It can be regarded as an adaptive weighted addition of coarse SR results. The comparison between WR process and plain process is shown in Figure 5. It concatenates the weighting maps from SR-HG modules and learns a global probability map using a Softmax normalization. The

coarse SR results are weighted by the probability map to generate the final SR image.

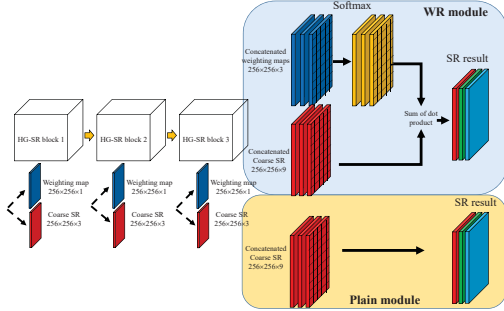


Figure 5. Proposed Weighted Reconstruction module.

For the plain process, it simply concatenates the coarse SR results together and learns one convolution layer to output the SR results without considering the internal correlation between coarse SR results. In the WR module, the Softmax layer is used to normalize the weighting maps from SR-HG modules in the range of $[0, 1]$. Then the final SR image is the weighted sum of the coarse SR results. By using Softmax normalization, we force each SR-HG module to learn the SR image at different scales.

4. Experimental Results

4.1. Implementation and training setups

Different from DBPN [10] which has different structures and configurations for different up-sampling enlargement, the proposed HBPN network uses the same structure as shown in Figure 1. In UBP and DBP blocks, we use 6×6 convolution filters with two striding and two padding for down- and up-sampling. For shortcut connections, we use 3×3 convolution filters with one striding and 1 padding. We initialize the weights based on [12]. The testing data include Set5 [3], Set14 [34], BSD100 [2], Urban100 [13] and Manga109 [23] on $2\times$, $4\times$ and $8\times$ SR enlargement.

The training data include 800 2K images from DIV2K [32] and 2650 2K images from Flickr [20]. Each image was rotated and flipped for augmentation to increase the images by $8\times$. The LR images were down-sampled and initially up-sampled by *bicubic* function in MATLAB on different scaling factors. We extracted LR-HR patch pairs from images of size 256×256 . In order to achieve better SR performance, for different SR scaling factors, we trained our model by using different LR-HR training patches. The learning rate is set to 0.0001 for all layers. The batch size is 8 for every 5×10^5 iterations and 32 for the rest 5×10^5 iterations to achieve better results. For optimization, we used Adam with the momentum to 0.9 and the weight decay of 0.0001. All experiments were conducted using Caffe, MATLAB R2016b on two NVIDIA GTX1080Ti GPUs.

4.2. Model analysis

Scaling factors of UBP and DBP. For each SR-HG module, we used DBP blocks to down-sample the feature maps to the smallest size and UBP blocks as mirror reflection to up-sample feature maps to the original size. For input data with size $M \times N \times 3$, we used T DBP blocks to down-sample the input to obtain feature maps with size $\frac{M}{2^T} \times \frac{N}{2^T} \times (64 \cdot 2^{T-1})$. To demonstrate the capability of this bottom-up and top-down structure, we conducted multiple networks *HG-1*, *HG-2*, *HG-3* (which is the proposed HBPN model) and *HG-4* for $4\times$ enlargement on Set5 to make comparison.

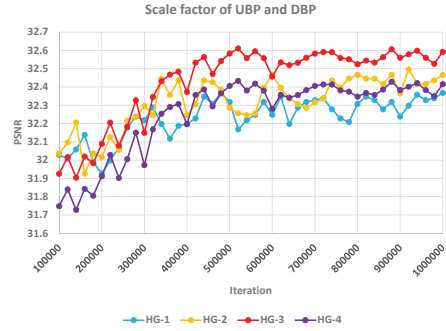


Figure 6. Back projection blocks analysis with different networks.

The results are shown in Figure 6. We compare different SR-HG blocks using different numbers of UBP and DBP to down- and up-scale features. Using *HG-3* shows the best performance comparing with other networks. Due to the model complexity, *HG-1* and *HG-2* can converge faster than *HG-3* and *HG-4*. As the best performance, *HG-3* achieves 32.66 dB in terms of PSNR which is 0.2 dB and 0.4 dB better than *HG-2* and *HG-4*.

Number of SR-HG modules. Generally, a deeper network can train more parameters to learn deeper feature representation for good performance. By stacking more and more SR-HG modules, [10] shows that the network with more HG blocks can produce a better prediction. In our experiments, we conduct multiple networks with different number of SR-HG modules: *S* (2 SR-HG modules), *M* (3 SR-HG modules, which is the proposed HBPN model) and *L* (4 SR-HG modules).

From Figure 7, we can see that network *L* (4 SR-HG module) gives the highest PSNR result. For network *S* (2 SR-HG module), its performance is lower than network *M* and network *L*. For network *L* (4 SR-HG module), it requires extra 33% parameters as compared with network *M* but only achieves slight (0.1 dB) improvement in PSNR. This result shows that our proposed HBPN has the best trade-off between performance and number of parameters. To further study the significance of each SR-HG module, let us visualize the activation maps of the output of each SR-HG module in our HBPN network.

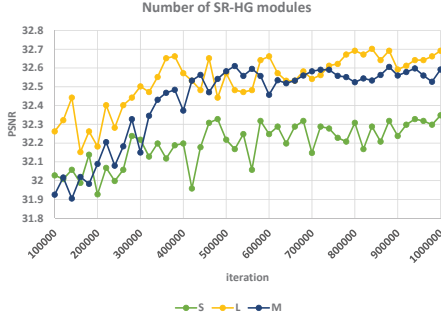


Figure 7. The number of SR-HG modules analysis. Note that we compare networks using different numbers of SR-HG modules for repeated feature extraction.

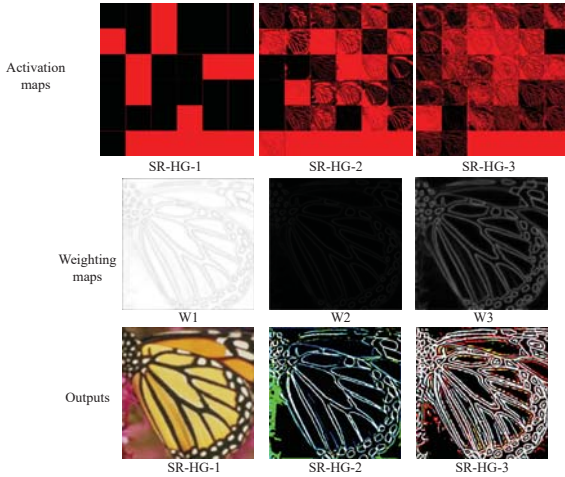


Figure 8. Visualization of activation maps, weighting maps and intermediate outputs. For better observation, please check the electronic version of this figure.

In the Figure 8, the first row shows three activation maps of each SR-HG output on image *butterfly*. We believe that the reason why CNNs outperform other patch-based learning approaches is that CNNs use activation layers to introduce the nonlinearity in the network to improve the feature representation power of filters. Hence, we show the activation maps rather than the output feature maps to show how activation layer works. In our design, we use the *PReLU* function that assigns weight 1 to non-zero values and very small weights to negative values. We can visualize the weights as the activation maps. In our experiments, we chose the last *PReLU* layer of each SR-HG module to make comparison. We can observe that the activation map of the *SR-HG-1* module has high activation across some of the feature maps while zero activation on others because the first layer only focuses on reconstructing the low-frequency information on averaging the whole image. This can be observed on the output of *SR-HG-1* of Figure 8. For *SR-HG-2* and *SR-HG-3*, there are more activated values on the activation maps, that focus on edge and texture regions. We calculated the percentage of activated values on *SR-HG-1*,

Table 1. Comparison of the network using plain concatenation block or WR reconstruction block, including PSNR and SSIM for scale $2\times$, $4\times$ and $8\times$ SR on Set5 and Set14. Red indicates the best results.

Algorithm	Scale	Set5		Set14	
		PSNR	SSIM	PSNR	SSIM
Plain model	2	37.95	0.959	33.61	0.917
WR model	2	38.13	0.961	33.78	0.921
Plain model	4	32.33	0.889	28.55	0.731
WR model	4	32.55	0.900	28.67	0.785
Plain model	8	26.89	0.761	24.81	0.632
WR model	8	27.17	0.785	24.96	0.642

SR-HG-2 and *SR-HG-3* and found the value decreases from 30.55%, 25.46% to 22.35%, which explains that the convolutional filters focus more on the edge and texture reconstruction.

The effect of WR process. Finally, we compare the WR process and the plain concatenated process in Table 1. We design the plain concatenated process and WR process with the structure as shown in Figure 5. They use the same SR-HG modules for feature extraction and the only difference is the final reconstruction process. The results were conducted on Set5, Set14 dataset of $2\times$, $4\times$ and $8\times$ enlargement.

From Table 1, we can see that using WR process can significantly improve the PSNR by at least 0.11 dB. The effectiveness of WR process can be further explained in Figure 8. In the second and third rows of Figure 8, we visualize the weighting maps of each SR-HG module and coarse SR outputs. For the first SR-HG module, the weighting map focuses on the low-frequency domain that reconstructs the main components of the image. For the second and third SR-HG modules, the weighting maps give high attentions to the edge regions. From the coarse SR output of each SR-HG module, we can also match the results with their weighting maps. Note that the output of *SR-HG-2* focuses on the edge reconstruction on G and B channels and the output of *SR-HG-3* focuses on the edge reconstruction on the R channel. From the aspect of gradient based edge detection, *SR-HG-2* focuses on the first-order edge reconstruction (see the single-line edges on the output of *SR-HG-2*) while *SR-HG-3* pays attention on the second-order edge reconstruction (see the double-line edges on the output of *SR-HG-3*). This can prove that using more SR-HG modules can explore deeper features in terms of the order of the pixel gradient.

From Table 1, it can be found that using WR process is very efficient that can gain 0.2 dB and 0.1 higher than the plain process in terms of PSNR and SSIM, respectively. We also show the weighting maps to possibly indicate the contribution of coarse SR results. We name the weighting maps at different stages of SR-HG modules as *W1*, *W2* and *W3*. The weighting maps are visualized by normalizing the pixel values in the range of $[0, 255]$. The weighting map corresponds to the SR-HG results giving different weights to the pixel values. The first weighting map gives a large

Table 2. Quantitative evaluation of state-of-the-art SR approaches, including PSNR and SSIM for scale $2\times$, $4\times$ and $8\times$. **Red** indicates the best and **blue** indicates the second best results.

Algorithm	Scale	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic		33.65	0.930	30.34	0.870	29.56	0.844	27.39	0.841	31.05	0.935
A+ [31]		36.54	0.954	32.40	0.906	31.22	0.887	29.23	0.894	35.33	0.967
CRFSR [22]		37.29	0.957	32.61	0.909	31.61	0.891	30.48	0.907	36.78	0.970
SRCNN [6]		36.65	0.954	32.29	0.903	31.36	0.888	29.52	0.895	35.72	0.968
VDSR [16]		37.53	0.958	32.97	0.913	31.90	0.896	30.77	0.914	37.16	0.974
DRRN [29]	$2\times$	37.74	0.959	33.23	0.913	32.05	0.897	31.23	0.919	37.92	0.976
SRResNet [19]		-	-	-	-	-	-	-	-	-	-
LapSRN [18]		37.52	0.959	33.08	0.913	31.80	0.895	30.41	0.910	37.27	0.974
EDSR [20]		38.11	0.960	33.92	0.919	32.32	0.901	32.93	0.935	39.10	0.977
DBPN [10]		38.09	0.960	33.85	0.919	32.27	0.900	32.96	0.931	39.10	0.978
HBPN(Ours)		38.13	0.961	33.78	0.921	32.33	0.902	33.12	0.938	39.30	0.979
Bicubic		28.42	0.810	26.10	0.704	25.96	0.669	23.64	0.659	25.15	0.789
A+ [31]		30.30	0.859	27.43	0.752	26.82	0.710	24.34	0.720	27.02	0.850
CRFSR [22]		31.10	0.871	27.87	0.765	27.05	0.719	24.89	0.744	28.12	0.872
SRCNN [6]		30.49	0.862	27.61	0.754	26.91	0.712	24.53	0.724	27.66	0.858
VDSR [16]		31.35	0.882	28.03	0.770	27.29	0.726	25.18	0.753	28.82	0.886
DRRN [29]	$4\times$	31.68	0.888	28.21	0.772	27.38	0.728	25.44	0.764	29.46	0.896
SRResNet [19]		32.05	0.891	28.53	0.780	27.57	0.735	26.07	0.784	-	-
LapSRN [18]		31.54	0.885	28.19	0.772	27.32	0.728	25.21	0.756	29.09	0.890
EDSR [20]		32.46	0.897	28.80	0.788	27.71	0.742	26.64	0.803	31.02	0.915
DBPN [10]		32.47	0.898	28.82	0.786	27.72	0.740	26.60	0.795	31.13	0.914
HBPN(Ours)		32.55	0.900	28.67	0.785	27.77	0.743	27.30	0.818	31.57	0.920
Bicubic		24.39	0.657	23.19	0.568	23.67	0.547	21.24	0.516	21.68	0.647
A+ [31]		25.52	0.692	23.98	0.597	24.20	0.568	21.37	0.545	22.39	0.680
CRFSR [22]		26.07	0.732	23.97	0.600	24.20	0.569	21.36	0.550	22.59	0.688
SRCNN [6]		25.33	0.689	23.85	0.593	24.13	0.565	21.29	0.543	22.37	0.682
VDSR [16]		25.72	0.711	24.21	0.609	24.37	0.576	21.54	0.560	22.83	0.707
DRRN [29]	$8\times$	26.18	0.738	24.42	0.622	24.59	0.587	21.88	0.583	23.60	0.742
SRResNet [19]		-	-	-	-	-	-	-	-	-	-
LapSRN [18]		26.15	0.738	24.35	0.620	24.54	0.586	21.81	0.582	23.39	0.735
EDSR [20]		26.97	0.775	24.94	0.640	24.80	0.596	22.47	0.620	24.58	0.778
DBPN [10]		27.21	0.784	25.13	0.648	24.88	0.601	22.69	0.622	24.96	0.799
HBPN(Ours)		27.17	0.785	24.96	0.642	24.93	0.602	23.04	0.647	25.24	0.802
NTIRE2019 Validation											
Algorithm		PSNR				SSIM					
Bicubic		29.548				0.844					
Using the proposed plain HBPN		33.41				0.889					
HBPN with Weighted Reconstruction		33.88				0.920					

weights on the whole image and small weights on the edges. The second and third weighting maps give higher weights to the non-edge regions (first-order edge detection) and edge regions (second-order edge detection), respectively.

4.3. Comparison with the state-of-the-art SR approaches

To prove the effectiveness of the proposed methods, we conducted experiments by comparing with most (if not all) state-of-the-art SR algorithms: Bicubic, A+ [31], CRFSR [22], SRCNN [6], VDSR [16], DRCN [17], LapSRN [18], SRResNet [19], EDSR [20] and DBPN [10]. PSNR and SSIM are used to evaluate the proposed method and others. Generally, PSNR and SSIM are calculated by converting RGB image to YUV and only the Y-channel image taken for consideration. During the testing, we rotated and flipped LR images for augmentation to gener-

ate several augmented inputs, and then applied the inverse transform and averaged all the outputs together to form the final SR results. For different scaling factors s , we exclude s pixels at boundaries to avoid boundary effect. For SR results, SRCNN, VDSR, SRResNet, EDSR and DBPN were reimplemented and provided by the authors of [10] and LapSRN was provided by the authors of [18]. Note that, this of our proposed approach also participated in the NTIRE2019 Real Image Super-resolution Challenge [1]. Table 2 also includes the validation testing results of this dataset. For this competition, it targets at real daily images, with down-sampling process using different degradation and distortions, and all images were taken by DSLR cameras in natural environments. However, all the state-of-the-art SR algorithms in the literature have been trained by using bicubic down-sampled images. It would then be inappropriate to use our HBPN model to make compari-

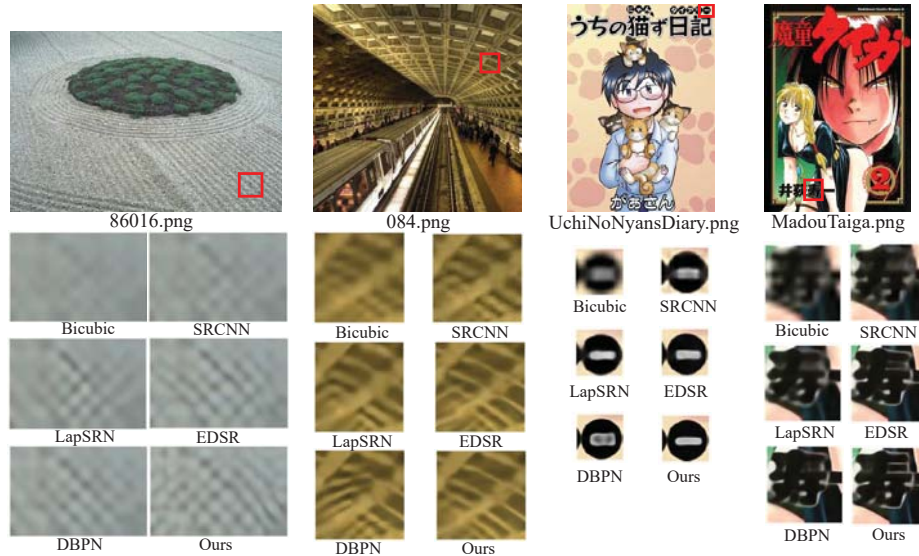


Figure 9. Visual quality comparison among different SR algorithms on $8\times$ super-resolution.

son with approaches in the literature with the NTIRE2019 validation dataset. Hence we just mainly listed out the results of our model using or without using the final stage of the proposed Weighted Reconstruction model for comparison. For more visual quality comparison, it is available at <https://github.com/Holmes-Alan/HBPN>.

We show the quantitative results in Table 2. Our proposed HBPN method outperforms other state-of-the-art approaches in all scales. Among these approaches, our proposed work can outperform EDSR and DBPN by large improvement (0.1-0.6 dB) on $8\times$ enlargement and improve the SR quality about 0.1-0.4 dB on $2\times$ and $4\times$ enlargement. Note that the PSNR and SSIM on BSD100, Urban100 and Manga109 using DBPN are different from [10] because we calculated the results on the whole image (rather than dividing images into four parts and calculating separately) by running their released code for fair comparison. For visual comparison, $2\times$ and $4\times$ enlargement are difficult to distinguish the improvement of the proposed method. We show $8\times$ enlargement in Figure 9, including the *86016.png* image from BSD100, *084.png* image from Urban100 and *UchiNoNyansDiary.png* and *MadouTaiga.png* images from Manga109. Figure 9 shows that both DBPN and EDSR cannot reconstruct well the fine texture of *86016.png*. On the other hand, our result can predict a clearer pattern of the sand. On the edge pattern of the roof on *084.png*, DBPN fails to reconstruct the concrete texture. Our approach can predict the horizontal and diagonal strides of the roof. The last two images of our approach on Manga109 give better visual quality in comparison with different approaches. On *UchiNoNyansDiary.png*, there is a Japanese character on the right upper corner that cannot be clearly reconstructed by SRCNN and LapSRN. DBPN, on the other hand, gives a

result containing holes on that stride that is misunderstanding. Our result actually can predict a sharper character. Similarly on *MadouTaiga.png*, the Japanese character inside the red box can better be observed on our result. Other SR approaches either generate blur edges on the strides or miss the stride pattern.

From all the results, we can see that our proposed HBPN approach can achieve better SR performance both quantitatively and qualitatively. It not only preserves the edge components, but also reconstructs the fine textures at different scaling factors.

5. Conclusion

We have proposed a Hierarchical Back Projection Network for image Super-Resolution on different up-scaling factors. Different from the previous SR study, we focus on feature extraction by conducting a HourGlass structure to learn the features in a bottom-up and top-down manner. The back projection mechanism is embedded into the network to update the low-resolution and high-resolution feature maps to reduce the errors. Meanwhile, we propose a self-weighting process that each HourGlass module generates one intermediate SR result along with its weighting map. By using the proposed Weighted Reconstruction block, we normalize the weighting maps to tune the contribution of each intermediate SR results for generating the final SR images. Results on quantitative and quality evaluation show its advantages over other approaches. Furthermore, we have also visualized the trained feature maps to illustrate the power of feature representation of each HourGlass module.

References

- [1] NTIRE 2019 Real Super-Resolution Challenge. <http://www.vision.ee.ethz.ch/ntire19/>.
- [2] 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, May 2011.
- [3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi Morel. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *British Machine Vision Conference (BMVC)*, Guildford, Surrey, United Kingdom, Sept. 2012.
- [4] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, June 2004.
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug 2007.
- [6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015.
- [7] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, April 2013.
- [8] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec 2006.
- [9] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, June 2014.
- [10] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. *CoRR*, abs/1803.02735, 2018.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.
- [13] J. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, June 2015.
- [14] J. Huang and W. Siu. Learning hierarchical decision trees for single-image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(5):937–950, May 2017.
- [15] J. Huang, W. Siu, and T. Liu. Fast image interpolation via random forests. *IEEE Transactions on Image Processing*, 24(10):3232–3245, Oct 2015.
- [16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015.
- [17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. *CoRR*, abs/1511.04491, 2015.
- [18] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [19] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.
- [20] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017.
- [21] Z. Liu, W. Siu, and Y. Chan. Fast image super-resolution via randomized multi-split forests. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, May 2017.
- [22] Zhi-Song Liu and Wan-Chi Siu. Cascaded random forests for fast image super-resolution, 10 2018.
- [23] Yusuke Matsui, Kota Ito, Yuji Aramaki, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *CoRR*, abs/1510.04389, 2015.
- [24] Alejandra Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016.
- [25] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *arXiv preprint arXiv:1505.04366*, 2015.
- [26] Mehdi S. M. Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. *CoRR*, abs/1612.07919, 2016.
- [27] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CoRR*, abs/1609.05158, 2016.
- [28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [29] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [30] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *2013 IEEE International Conference on Computer Vision*, pages 1920–1927, Dec 2013.
- [31] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. volume 9006, pages 111–126, 04 2015.
- [32] Radu Timofte, Shuhang Gu, Jiqing Wu, and Luc Van Gool. Ntire 2018 challenge on single image super-resolution:

Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

- [33] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018.
- [34] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proceedings of the 7th International Conference on Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer-Verlag.
- [35] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486, Nov 2011.