BYNET: IMAGE SUPER RESOLUTION WITH A BYPASS CONNECTION NETWORK

Jiu Xu *Yeongnam Chae* *Björn Stenger*

Rakuten Institute of Technology

ABSTRACT

This paper proposes a deep residual network, *ByNet*, for the single image super resolution task. The main innovation is the introduction of two effective components, bypass connections and a feature scaling layer. Bypass connections are formed either by skip connections that jump multiple layers or by adding a convolution layer in such a jump. The final feature scaling layer enables more robust convergence. Experiments on standard benchmarks show that the proposed method achieves state of the art results over multiple scales in terms of PSNR and structural similarity (SSIM).

Index Terms— super resolution, deep convolutional neural networks, residual learning, image enhancement

1 Introduction

Image quality improvement includes denoising, dehazing, and super resolution, and continues to be an active research topic. In particular single image super resolution has seen significant advances [1, 2, 3] and has been applied in areas such as video enhancement [4], text image improvement [5, 6], and medical image processing [7, 8]. Given a single low-resolution (LR) image as input, super-resolution methods learn a mapping to a high-resolution (HR) image. Impressive results were shown using a single image to learn the mapping by self-similarity [3, 9]. In general, however, it is beneficial to use a larger training set, easily obtained by downsampling HR images, to capture the nonlinearity of the upscaling function. Convolutional neural networks (CNN) are known for their ability to learn highly nonlinear functions, and CNN-based methods are currently the top performers in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Various network architectures have been proposed, and this paper introduces an architecture with two new network components, which increase performance compared to VDSR [10] and are easy to implement. The first component is a combination of bypass connection blocks, which can either be simple skip connections or with an additional convolution layer. The second component is a final feature scaling layer that leads to faster convergence to a better solution.

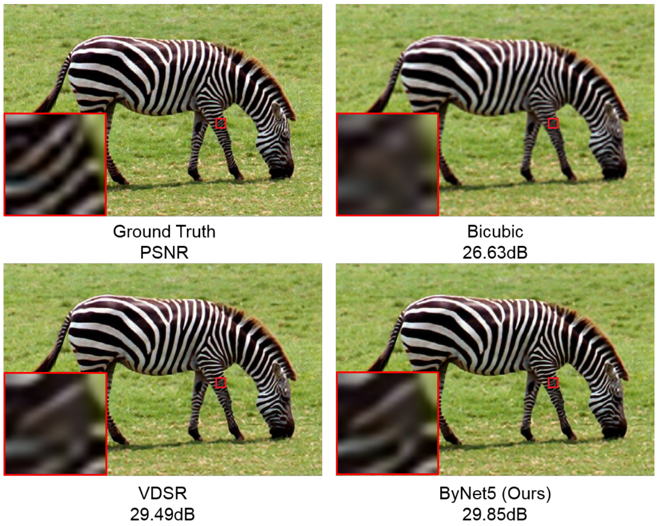

Fig. 1: Example result. *A region in the* zebra *image from* Set14 *[11] shows improved recovery of detail (scale factor of 3).*

2 Prior work

Super resolution methods can be divided into several approaches. Dictionary methods using sparse coding encode image patches based on sparse signal representation [12, 13, 11]. Another dictionary-based approach is neighborhood embedding, which interpolates low-resolution image patches using a nonlinear embedding to obtain high-resolution patches [14, 15]. Recently, convolutional neural networks (CNN) have been successfully applied to the super resolution task. The CNN directly learns an end-to-end mapping between the low/high-resolution images. The early *Super Resolution CNN* (SRCNN) [16], has a simple architecture, feeding a low-resolution input image into two stacked convolution and rectified linear unit (ReLU) layers. The convolution layer performs a similar function to a sparse coding dictionary and selects the best feature during the learning phase. Following this work, a number of improvements have been proposed. One line of work combines a sparse coding model together with a neural network [17]. Kim *et al.* proposed a deeply-recursive convolutional network, which is able to better preserve image context [18], but has relatively high computational complexity. The *Sub-Pixel CNN* in [19], introduces a sub-pixel convolution layer to substitute the deconvolution layer for up-sampling. Kim *et al.* [10] introduced a 20-layer network (VDSR) to learn the residual image between the LR and HR image. This method represents the state of the art in terms of PSNR over multiple scales. Note that other reconstruction cost functions have been proposed, for example in terms of a *perceptual loss* [20]. The SRResNet method in [21] optimizes this loss, achieving high PSNR values on standard test datasets for a 4 × upscaling factor when trained on ImageNet. Generative adversarial networks, such as SRGAN [21], produce results with a lower PSNR, but with a higher perceptual score.

3 Proposed Method

This section introduces our proposed super resolution method, which uses a novel CNN architecture that includes bypass connection blocks. Starting from a baseline model, which is a lean version of VDSR, we describe the new components, which in combination constitute the proposed ByNet SR model.

3.1 Network Architecture

3.1.1 Baseline Network

Our baseline model is related to the VDSR model [10], which is composed of 20 convolution and ReLU layers. Convolution layers have a 3×3 filter size with 64 channels except for the last layer, which has a single 3×3 filter. The network takes a $64 \times 64 \times 1$ image X, obtained from the LR image via bicubic interpolation, as input and each layer computes the function

$$h_{t+1}(x) = \max(0, w_{t+1} * h_t(x) + b_{t+1}), \quad (1)$$

where layer index t is in the range from 1 to 19, $h_1(x) = X$, and residual image estimate $\widehat{R} = h_{20}(x)$. Instead of calculating the mean squared error (MSE) between HR ground truth Y and its estimation $\widehat{Y}$, the model aims to minimize the loss of residual $\frac{1}{2}||R - \widehat{R}||^2$, improving both accuracy and runtime, where $R = Y - X$ is the residual image.

Our baseline network analogously consists of 17 weight layers with 3×3 filters. The first 16 layers contain 64 filters while the last layer contains a single filter. As in VDSR, we apply image residual learning for calculating the MSE loss. In contrast to VDSR, none of the layers has a bias. We select this lean version of VDSR for our baseline as it allows easy extension into our proposed network design, while ensuring that its performance is still comparable to the published VDSR results.

3.1.2 Bypass Connections

Here we introduce two types of residual blocks that we integrate into the CNN architecture. Figure 2 shows these two blocks, which both include bypass connections. The first block, see Fig. 2(a), contains a *skip connection*, defined as:

$$y = f\left(h_t(x)\right) + h_t(x), \quad (2)$$

where $h_t(x)$ and y are input and output vectors, respectively, and $f()$ is the feature to be learned, which in our case is

$$f(h_t(x)) = w_{t+3} * \big(\max\big(0, w_{t+2} * \max\big(0, w_{t+1} * h_t(x)\big)\big)\big). \quad (3)$$

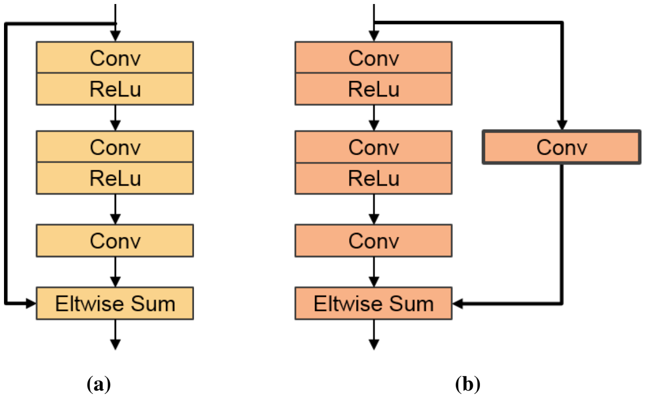

Fig. 2: New residual blocks used in ByNet: (a) *feature bypass with skip connection* **(b)** *feature bypass with convolutional connection.*

As in the baseline case, all biases are omitted. Skip connections (sometimes called *shortcut* connections) have been used in ResNet[22] and Highway Networks[23] for image classification. Note that skip connections introduce no additional parameters or computational complexity, except for negligible element-wise addition. As shown in [22], skip connections improve convergence properties and achieve higher accuracy within the same training epoch. However, when introducing skip connections in the super-resolution task, training becomes prone to the vanishing gradient problem. Unlike in object classification, the difference between the input and output of the network is small, and many values in the residual image are close to zero [10]. We therefore define a second block type with an additional convolution layer to prevent the gradient from vanishing, see Fig. 2(b):

$$y = f\left(h_t(x)\right) + w_k * h_t(x). \quad (4)$$

The element-wise addition layer sums the features learned from different receptive fields of input help to further improve the model robustness in terms of multi-scale capability. We stack residual blocks with alternating types in sequence (see Section 4.3 for this choice). Figure 3 shows ByNet5, which contains three skip connection blocks (for the first, third, and fifth residual block) and two convolution blocks (for the second and fourth residual block).

3.1.3 Feature Scaling

In multi-scale training, disparately downscaled images make training prone to divergence. Moreover, due to the large variation in each training mini-batch, the deeper the network is, the harder it is for the network to converge. Gradient clipping and image residual learning have been proposed as remedies [10], yet we have still observed cases of divergence during training. We insert a scaling layer before the element-wise sum layer of the baseline network, scaling the layers by a value that is learned during training. Feature scaling improves convergence by scaling the network output to fit the distribution

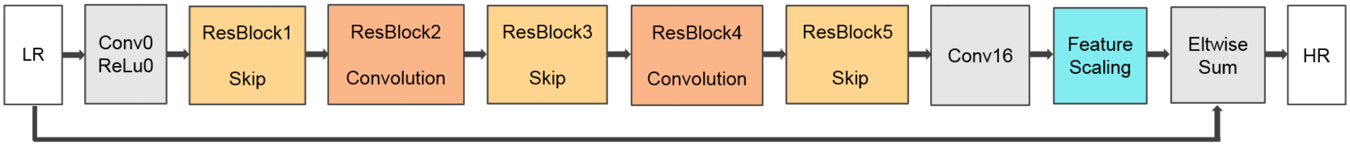

Fig. 3: ByNet5 architecture. *The proposed CNN learns the non-linear mapping between LR/HR images. New elements of the architecture (in color) are two types of residual blocks, which are alternated in sequence. Convolution layers before and after these blocks ensure correct dimensions of the image features. An additional feature scaling layer is introduced before taking the element-wise sum.*

Table 1: Number of parameters in the ByNet5 model.

layer	kernel	plane-in	plane-out	#layers	#params
Conv0	3x3	1	64	1	576
ResBlock 1,3,5	3x3	64	64	3	3 × 110,592
ResBlock 2,4	3x3	64	64	4	2 × 147,456
Conv16	3x3	64	1	1	576
Scaling	-	1	1	0	1
Total					627,841

of image residuals. Similar to the functionality of the CNN initializer, the scaling layer takes the role of overall feature weighting, allowing the network to converge by focusing on the best gradient direction in the early optimization stages.

3.2 Training

Given an LR image x as input, the model learns to predict the HR residual $\widehat{y}$ by minimizing the MSE, averaged over each training mini-batch. The number of parameters in each layer and block is shown in Table 1. LR images are obtained by down-sampling HR images with the scale factors of 2,3, and 4. These sets of LR images are merged and shuffled for training which allows our model to naturally handle multiple scale factors.

4 Results

Here we describe the datasets and provide the parameters required for model training to make the results reproducible. We compare the method with bicubic scaling as well as with state-of-the-art methods. Finally, we assess the contribution of each new network component in an ablative study.

4.1 Datasets

To be consistent with prior work we use the same training dataset as in [10, 24, 25], consisting of 291 images, which is the combination of the 91 images from the dataset of Yang *et al.* [12] and the 200 images from the *BSD* dataset [26]. Data augmentation includes mirroring, rotating (90, 180, and 270 degrees), and scaling (factors 0.6 and 0.8) each image, providing a total of 6,984 training images. We evaluate on three public datasets used in prior work [10, 16], *Set5* [14], *Set14* [11], and *BSD100* [24]. To generate HR/LR image pairs we down-sample all HR images using Matlab's bicubic interpolation function.

4.2 Parameter Settings

We train the CNN using stochastic gradient descent and a momentum of 0.9 and weight decay of 10^{-4} . A maximum of 40 training epochs with a batch size of 128 (2 × 64) is run on two Nvidia GeForce GTX1080 GPUs. Following [10], gradient clipping is utilized with a limit of 1.0. The learning rate is initialized to 0.1 and divided by 10 every tenth epoch. The initial parameter for the feature scaling layer is set to 0.1. Xavier initialization [27] is used where all weights are drawn from a zero-mean Gaussian with variance $1/n_{in}$, where n_{in} is the number of input units.

4.3 Analysis of Residual Block Arrangement

The two types of residual blocks can be arranged in various ways. In order to limit the search space we evaluate different symmetric arrangements of five bypass blocks in sequence, with zero to five convolution bypass blocks, respectively, with skip connections constituting the other blocks. The results show that performance increases most from zero to one convolution blocks, and that there is a local PSNR maximum for two convolution blocks, which we use in ByNet5. We also test performance for deeper networks, ByNet7 and ByNet9. For each extension we add two more blocks, one convolution and one skip connection block.

4.4 Patch Size Analysis

All training images are partitioned into non-overlapping patches of fixed size. This means that the larger the patch size, the larger the receptive fields, but the smaller the number of training images. We evaluate the baseline model using three different patch sizes, of side length 41, 52, and 62 pixels, respectively. The resulting PSNR values, averaged over three scales, are 34.20dB, 34.21dB, and 34.19dB, respectively. The effects of larger receptive field and smaller training size seem to balance out. Consistent with VDSR [10] we use 41 × 41 patches in the ablation study (in Section 4.6), and 52 × 52 patches for the final ByNet models that we compare to other methods.

4.5 Comparison with State of the Art

We compare ByNet with the reported results of the following methods on the same training and test sets: A+ [24], RFL [25], SRCNN [16], and VDSR [10]. Table 2 shows the quantitative results on the luminance channel, both PSNR (in

Table 2: Performance Comparison on benchmark datasets: *PSNR and SSIM are averaged over all images for each scale. The proposed ByNet model consistently achieves the best PSNR and SSIM results. Adding more blocks improves performance. Results for the recent VDSR method [10] are highlighted in blue for easy visual comparison.*

Dataset	Scale	Bicubic PSNR	A+[24] PSNR	RFL[25] PSNR	SRCNN[16] PSNR	VDSR[10] PSNR	ByNet5 PSNR	ByNet7 PSNR	ByNet9 PSNR	Bicubic SSIM	A+[24] SSIM	RFL[25] SSIM	SRCNN[16] SSIM	VDSR[10] SSIM	ByNet5 SSIM	ByNet7 SSIM	ByNet9 SSIM
Set5	×2	33.66	36.54	36.54	36.66	37.53	37.60	37.69	**37.74**	0.9299	0.9544	0.9537	0.9542	0.9587	0.9594	0.9598	**0.9599**
	×3	30.39	32.58	32.43	32.75	33.66	33.85	33.89	**33.96**	0.8682	0.9088	0.9057	0.9090	0.9213	0.9237	0.9241	**0.9246**
	×4	28.42	30.28	30.14	30.48	31.35	31.49	31.54	**31.60**	0.8104	0.8603	0.8548	0.8628	0.8838	0.8862	0.8871	**0.8886**
Set14	×2	30.24	32.28	32.26	32.42	33.03	33.13	33.21	**33.24**	0.8688	0.9056	0.9040	0.9063	0.9124	0.9138	0.9144	**0.9147**
	×3	27.55	29.13	29.05	29.28	29.77	29.92	29.94	**29.96**	0.7742	0.8188	0.8164	0.8209	0.8314	0.8342	0.8350	**0.8354**
	×4	26.00	27.32	27.24	27.49	28.01	28.20	28.20	**28.24**	0.7027	0.7491	0.7451	0.7503	0.7674	0.7716	0.7722	**0.7732**
BSD100	×2	29.56	31.21	31.16	31.36	31.90	31.92	31.96	**32.00**	0.8431	0.8863	0.8840	0.8879	0.8960	0.8967	0.8973	**0.8977**
	×3	27.21	28.29	28.22	28.41	28.82	28.86	28.89	**28.91**	0.7385	0.7835	0.7806	0.7863	0.7976	0.7993	0.8001	**0.8808**
	×4	25.96	26.82	26.75	26.90	27.29	27.31	27.34	**27.37**	0.6675	0.7087	0.7054	0.7101	0.7251	0.7267	0.7277	**0.7285**

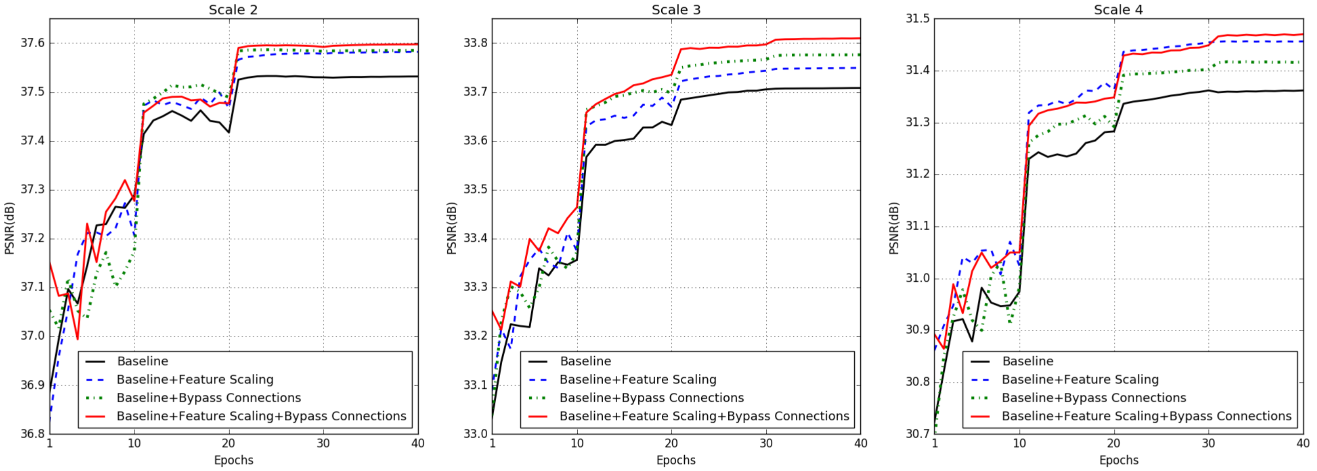

Fig. 4: Ablation study. *The convergence curves for the Set5 training set show that adding both feature scaling and bypass connections consistently lead to increased PSNR across different scales.*

dB) and structural similarity (SSIM), for different upscaling factors. ByNet9 consistently achieves the top result, achieving a mean improvement of 0.25dB PSNR on the *Set5* dataset. It is impossible to directly compare published run-times as the methods were implemented with different libraries on different machines. For comparison we re-implemented the 20-layer VDSR method [10] and test on the *BSD100* dataset (100 images at 3 scales each). The mean time per image is 37ms for VDSR compared to 32ms for ByNet5, 45ms for ByNet7, and 57ms for ByNet9. These numbers show the run-time *vs.* performance trade-off of the ByNet architecture.

4.5.1 Increasing Training Set Size

We extend the training set by adding the recent *General-100* dataset [28], to the 291 images, and evaluate ByNet5 trained using the extended set on the *Set5* data set. The resulting PSNR values are 37.68, 33.89, and 31.54dB for scales 2, 3, and 4, respectively, confirming that additional data further increases performance.

4.6 Ablation Study

We carry out an ablation study to evaluate the contribution of each proposed component, feature scaling and bypass connection blocks. Models are tested on the *Set5* dataset with a patch size of 41 × 41. Figure 4 shows the evolution of PSNR values during training until convergence, separately for the three different scale values. Points worth noting are (1) the combination of both components consistently yields the best performance, (2) the improvement is not additive, and (3) their relative contribution can differ for different scales, e.g. for a scale value of 3 the bypass convolution is more effective, for a scale value of 4 the feature scaling layer is more effective. We also observe that when using skip connections without additional convolution layers training does not converge to a good solution due to vanishing gradients.

5 Conclusion

In this paper we proposed ByNet, a new network architecture for image super resolution, with state-of-the-art performance in terms of PSNR and SSIM. The two main contributions consist of bypass connection blocks as well as feature scaling. Both are simple to implement, yet effective in increasing performance as demonstrated in an ablative study and comparisons with competing methods on standard super resolution benchmarks. For future work, we consider training on large-scale data, *e.g.* ImageNet, to be a promising avenue [21].

6 References

[1] K. Nasrollahi and T. B. Moeslund, "Super-resolution: A comprehensive survey," in *Machine Vision and Applications*, 2014, vol. 25, pp. 1423–1468.

[2] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," in *IJCV*, 2000.

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

[4] X. Zhang, M. Tang, and R. Tong, "Robust super resolution of compressed video," in *Vis. Comput.*, 2012, vol. 28, pp. 1167–1180.

[5] D. Capel and A. Zisserman, "Super-resolution enhancement of text image sequences," in *ICPR*, 2000.

[6] Q. Yuan, L. Zhang, and H. Shen, "Multiframe super-resolution employing a spatially weighted total variation model," in *TCSVT*, 2012, vol. 22, pp. 379–392.

[7] Y. Zhang, G. Wu, P.-T. Yap, Q. Feng, J. Lian, W. Chen, and D. Shen, "Reconstruction of super-resolution lung 4d-CT using patch based sparse representation," in *CVPR*, 2012.

[8] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. Marvao, T. Dawes, D. O'Regan, and D Rueckert, "Cardiac image super-resolution with global correspondence using multi-atlas patch match," in *MICCAI*, 2013, vol. 8151, p. 916.

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

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

[11] 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.

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

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

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

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

[16] 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.

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

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

[19] 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.

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

[21] 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," *arXiv:1609.04802*, 2016.

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

[23] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *NIPS*, 2015.

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

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

[26] 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.

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

[28] C. Dong, C.C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *ECCV*, 2016.