Wide Activation for Efficient and Accurate Image Super-Resolution

Jiahui Yu
jyu79@illinois.edu

Yuchen Fan
yuchenf4@illinois.edu

Jianchao Yang
yangjianchao@bytedance.com

Ning Xu
ning.xu@snap.com

Zhaowen Wang
zhawang@adobe.com

Xinchao Wang
xwang135@stevens.edu

Thomas Huang
t-huang1@illinois.edu

Abstract

In this report we demonstrate that with same parameters and computational budgets, models with wider features before ReLU activation have significantly better performance for single image super-resolution (SISR). The resulted SR residual network has a slim identity mapping pathway with wider ($2\times$ to $4\times$) channels before activation in each residual block. To further widen activation ($6\times$ to $9\times$) without computational overhead, we introduce linear low-rank convolution into SR networks and achieve even better accuracy-efficiency tradeoffs. In addition, compared with batch normalization or no normalization, we find training with weight normalization leads to better accuracy for deep super-resolution networks. Our proposed SR network *WDSR* achieves better results on large-scale DIV2K image super-resolution benchmark in terms of PSNR with same or lower computational complexity. Based on WDSR, our method also won 1st places in NTIRE 2018 Challenge on Single Image Super-Resolution in all three realistic tracks. Experiments and ablation studies support the importance of wide activation for image super-resolution. Code is released at: `https://github.com/JiahuiYu/wdsr_ntire2018`.

1 Introduction

Deep convolutional neural networks (CNNs) have been successfully applied to the task of single image super-resolution (SISR) [14, 19, 20, 42]. SISR aims at recovery of a high resolution (HR) image from its low resolution (LR) counterpart (typically a bicubic downsampled version of HR). It has many applications in security, surveillance, satellite, medical imaging [24, 34] and can serve as a built-in module for other image restoration or recognition tasks [6, 21, 37, 40, 41].

Previous image super-resolution networks including SRCNN [4], FSRCNN [3], ESPCN [29] utilized relatively shallow convolutional neural networks (with its depth from 3 to 5). They are inferior in accuracy compared with later proposed deep SR networks (e.g., VDSR [14], SRResNet [17] and EDSR [19]). The increasing of depth brings benefits to representation power [2, 5, 18, 28] but meanwhile under-use the feature information from shallow layers (usually represent low-level features). To address this issue, methods including SRDenseNet [36], RDN [42], MemNet [33] introduce various skip connections and concatenation operations between shallow layers and deep layers, formalizing holistic structures for image super-resolution.

In this work we address this issue in a different perspective. Instead of adding various shortcut connections, we conjecture that the non-linear ReLUs impede information flow from shallow layers to deeper ones [26]. Based on residual SR network, we demonstrate that without additional parameters

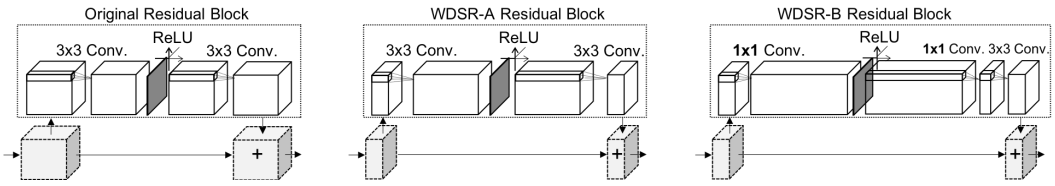

Figure 1: **Left:** vanilla residual block. **Middle *WDSR-A*:** residual block with wide activation. **Right *WDSR-B*:** residual block with wider activation and linear low-rank convolution. We demonstrate different residual building blocks for image super-resolution networks. Compared with vanilla residual blocks used in EDSR [19], we introduce *WDSR-A* which has a slim identity mapping pathway with wider ($2\times$ to $4\times$) channels before activation in each residual block. We further introduce *WDSR-B* with linear low-rank convolution stack and even widen activation ($6\times$ to $9\times$) without computational overhead. In *WDSR-A* and *WDSR-B*, all ReLU activation layers are only applied between two wide features (features with larger channel numbers).

and computation, simply expanding features before ReLU activation leads to significant improvements for single image super-resolution, beating SR networks with complicated skip connections and concatenations including SRDenseNet [36] and MemNet [33]. The intuition of our work is that expanding features before ReLU allows more information pass through while still keeps highly non-linearity of deep neural networks. Thus low-level SR features from shallow layers may be easier to propagate to the final layer for better dense pixel value predictions.

The central idea of wide activation leads us to explore efficient ways to expand features before ReLU, since simply adding more parameters is inefficient for real-time image SR scenarios [8]. We first introduce SR residual network *WDSR-A*, which has a slim identity mapping pathway with wider ($2\times$ to $4\times$) channels before activation in each residual block. However when the expansion ratio is above 4, channels of the identity mapping pathway have to be further slimmed and we find it dramatically deteriorates accuracy. Thus as the second step, we keep constant channel numbers of identity mapping pathway, and explore more efficient ways to expand features. We first consider group convolution [38] and depthwise separable convolution [1]. However, we find both of them have unsatisfactory performance for the task of image super-resolution. To this end, we propose *linear low-rank convolution* that factorizes a large convolution kernel into two low-rank convolution kernels. With wider activation and *linear low-rank convolutions*, we construct our SR network *WDSR-B*. It has even wider activation ($6\times$ to $9\times$) without additional parameters or computation, and boosts accuracy further for image super-resolution. The illustration of *WDSR-A* and *WDSR-B* is shown in Figure 1. Experiments show that wider activation consistently beats their baselines under different parameter budgets.

Additionally, compared with batch normalization [12] or no normalization, we find training with weight normalization [25] leads to better accuracy for deep super-resolution networks. Previous works including EDSR [19], BTSRN [7] and RDN [42] found that batch normalization [12] deteriorates the accuracy of image super-resolution, which is also confirmed in our experiments. We provide three intuitions and related experiments showing that batch normalization, due to 1) mini-batch dependency, 2) different formulations in training and inference and 3) strong regularization side-effects, is not suitable for training SR networks. However, with the increasing depth of neural networks for SR (e.g. MDSR [19] has depth around 180), the networks without batch normalization become difficult to train. To this end, we introduce weight normalization for training deep SR networks. The weight normalization enables us to train SR network with an order of magnitude higher learning rate, leading to both faster convergence and better performance.

In summary, our contributions are as follows. 1) We demonstrate that in residual networks for SISR, wider activation has better performance with same parameter complexity. Without additional computational overhead, we propose network *WDSR-A* which has wider ($2\times$ to $4\times$) activation for better performance. 2) To further improve efficiency, we also propose *linear low-rank convolution* as basic building block for construction of our SR network *WDSR-B*. It enables even wider activation ($6\times$ to $9\times$) without additional parameters or computation, and boosts accuracy further. 3) We suggest batch normalization [12] is not suitable for training deep SR networks, and introduce weight

normalization [25] for faster convergence and better accuracy. 4) We train proposed *WDSR-A* and *WDSR-B* built on the principle of wide activation with weight normalization, and achieve better results on large-scale DIV2K image super-resolution benchmark. Our method also won 1st places in NTIRE 2018 Challenge on Single Image Super-Resolution in all three realistic tracks.

2 Related Work

2.1 Super-Resolution Networks

Deep learning-based methods for single image super-resolution significantly outperform conventional ones [23, 39] in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). SRCNN [4] was the first work utilizing an end-to-end convolutional neural network as a mapping function from LR images to their HR counterparts. Since then, various convolutional neural network architectures were proposed for improving the accuracy and efficiency. In this section, we review these approaches under several groups.

Upsampling layers Super-resolution involves upsampling operation of image resolution. The first super-resolution network SRCNN [4] applied convolution layers on the pre-upscaled LR image. It is inefficient because all convolutional layers have to compute on high-resolution feature space, yielding S^2 times computation than on low-resolution space, where S is the upscaling factor. To accelerate processing speed without loss of accuracy, FSRCNN [3] utilized parametric deconvolution layer at the end of SR network [3], making all convolution layers compute on LR feature space. Another non-parametric efficient alternative is pixel shuffling [29] (a.k.a., sub-pixel convolution). Pixel shuffling is also believed to introduce less checkerboard artifacts [22] than the deconvolutional layer.

Very deep and recursive neural networks The depth of neural networks is of central importance for deep learning [9, 30, 31]. It is also experimentally proved in single image super-resolution task [7, 14, 17, 19, 33, 36, 42]. These very deep networks (usually more than 10 layers) stack many small-kernel (i.e., 3×3) convolutions and have higher accuracy than shallow ones [3, 29]. However, the increasing depth of convolutional neural networks introduces over-parameterization and difficulty of training. To address these issues, recursive neural networks [15, 32] are proposed by re-using weights repeatedly.

Skip connections On one hand, deeper neural networks have better performance in various tasks [30], on the other hand low-level features are also important for image super-resolution task [42]. To address this contradictory, VDSR [14] proposed a very deep VGG-like [30] network with global residual connection (i.e. identity skip connection) for SISR. SRResNet [17] proposed a ResNet-like [9] network. Densely connected networks [11] are also adapted for SISR in SRDenseNet [36]. MemNet [33] integrated skip connections and recursive unit for low-level image restoration tasks. To further exploit the hierarchical features from all the convolutional layers, residual dense networks (RDN) [42] are proposed. All these works benefit from additional skip connections between different levels of features in deep neural networks.

Normalization layers As image super-resolution networks going deeper and deeper (from 3-layer SRCNN [4] to 160-layer MDSR [19]), training becomes more difficult. Batch normalization layers are one of the cures for this problem in many tasks [9, 31]. It is also introduced in SISR networks in SRResNet [17]. However, empirically it is found that batch normalization [12] hinders the accuracy of image super-resolution. Thus, in recent image SR networks [7, 19, 42], batch normalization is abandoned.

2.2 Parameter-Efficient Convolutions

In this subsection, we also review several related methods proposed for improving efficiency of convolutions.

Flattened convolution Flattened convolutions [13] consist of consecutive sequence of one-dimensional filters across all directions in 3D space (lateral, vertical and horizontal) to approximate

conventional convolutions. The number of parameters in flattened convolution decreases from XYC to $X + Y + C$, where C is the number of input planes, X and Y denote filter width and height.

Group convolution Group convolutions [38] divide features into groups channel-wisely and perform convolutions inside the group individually, followed by a concatenation to form the final output. In group convolutions, the number of parameters can be reduced by g times, where g is the group number. Group convolutions are the key components to many efficient models (e.g. ResNeXt [38]).

Depthwise separable convolution Depthwise separable convolution is a stack of depthwise convolution (i.e. a spatial convolution performed independently over each channel of an input) followed by a pointwise convolution (i.e. a 1x1 convolution) without non-linearities. It can also be viewed as a specific type of group convolution where the number of groups g is the number of channels. The depthwise separable convolution formulates the basic architecture in many efficient models including Xception [1], MobileNet [10] and MobileNetV2 [27].

Inverted residuals Another work [27] expands features before activation for image recognition tasks (named inverted residuals). The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. The inverted residual shares similar merits with our proposed wide activation, however we found the inverted residual proposed in [27] has unsatisfactory performance on the task of image SR. In this work we mainly explore different network architectures to improve the accuracy and efficiency for the task of image super-resolution with the central idea of wide activation.

3 Proposed Methods

3.1 Wide Activation: *WDSR-A*

In this part, we mainly describe how we expand features before ReLU activation layer without computational overhead. We consider the effects of wide activation inside a residual block. A naive way is to directly add channel numbers of all features. However, it proves nothing except that more parameters lead to better performance. Thus, in this section, we design our SR network to study the importance of wide features before activation with *same parameter and computational budgets*. Our first step towards wide activation is extremely simple: we slim the features of residual identity mapping pathway while expand the features before activation, as shown in Figure 1.

Two-layer residual blocks are specifically studied following baseline EDSR [19]. Assume the width of identity mapping pathway (Fig. 2) is w_1 and width before activation inside residual block is w_2. We introduce expansion factor before activation as r thus $w_2 = r \times w_1$. In the vanilla residual networks (e.g., used in EDSR and MDSR) we have $w_2 = w_1$ and the number of parameters are $2 \times w_1^2 \times k^2$ in each residual block. The computational (Mult-Add operations) complexity is a constant scaling of parameter numbers when we fix the input patch size. To have same complexity $w_1^2 = \hat{w}_1 \times \hat{w}_2 = r \times {\hat{w}_1}^2$, the residual identity mapping pathway need to be slimmed as a factor of $\sqrt{r}$ and the activation can be expanded with $\sqrt{r}$ times meanwhile.

This simple idea forms our first widely-activated SR network *WDSR-A*. Experiments show that *WDSR-A* is extremely effective for improving accuracy of SISR when r is between 2 to 4. However, for r larger than this threshold the performance drops quickly. This is likely due to the identity mapping pathway becoming too slim. For example, in our baseline EDSR (16 residual blocks with 64 filters) for $\times 3$ super-resolution, when r is beyond 6, w_1 will be even smaller than the final HR image representation space $S^2 * 3$ (we use pixel shuffle as upsampling layer) where S is the scaling factor and 3 represents RGB. Thus we seek for parameter-efficient convolution to further improve accuracy and efficiency with wider activation.

3.2 Efficient Wider Activation: *WDSR-B*

To address the above limitation, we keep constant channel numbers of identity mapping pathway, and explore more efficient ways to expand features. Specifically we consider 1×1 convolutions.

1×1 convolutions are widely used for channel number expansion or reduction in ResNets [9], ResNeXts [38] and MobileNetV2 [27]. In *WDSR-B* (Fig. 1) we first expand channel numbers by using 1×1 and then apply non-linearity (ReLUs) after the convolution layer. We further propose an efficient *linear low-rank convolution* which factorizes a large convolution kernel to two low-rank convolution kernels. It is a stack of one 1×1 convolution to reduce number of channels and one 3×3 convolution to perform spatial-wise feature extraction. We find adding ReLU activation in *linear low-rank convolutions* significantly reduces accuracy, which also supports wide activation hypothesis.

3.3 Weight Normalization vs. Batch Normalization

In this part, we mainly analyze the different purposes and effects of batch normalization (BN) [12] and weight normalization (WN) [25]. We offer three intuitions why batch normalization is not appropriate for image SR tasks. Then we demonstrate that weight normalization does not have these drawbacks like BN, and it can be effectively used to ease the training difficulty of deep SR networks.

Batch normalization BN re-calibrates the mean and variance of intermediate features to solve the problem of *internal covariate shift* [12] in training deep neural networks. It has different formulations in training and testing. For simplicity, here we ignore the re-scaling and re-centering learnable parameters of BN. During training, features in each layer are normalized with mean and variance of the current training mini-batch:

$$\hat{x}_B = \frac{x_B - E_B[x_B]}{\sqrt{Var_B[x_B] + \epsilon}}, \tag{1}$$

where x_B is the features of current training batch, ϵ is a small value (e.g. 1e-5) to avoid zero-division. The first order and second order statistics are then updated to global statistics in a moving average way:

$$E[x] \leftarrow E_B[x_B], \tag{2}$$

$$Var[x] \leftarrow \frac{m}{m-1} Var_B[x_B], \tag{3}$$

where m is the mini-batch size, the $\frac{m}{m-1}$ is for estimating un-biased variance, $\leftarrow$ means assigning moving average. During inference, these global statistics are used instead to normalize the features:

$$\hat{x}_{test} = \frac{x_{test} - E[x]}{\sqrt{Var[x] + \epsilon}}. \tag{4}$$

As shown in the formulations of BN, it will cause following problems. 1) For image super-resolution, commonly only small image patches (e.g. 48×48) and small mini-batch size (e.g. 16) are used to speedup training [7, 14, 17, 19, 33, 36, 42], thus the mean and variance of small image patches differ a lot among mini-batches, making theses statistics unstable, which is demonstrated in the section of experiments. 2) BN is also believed to act as a regularizer and in some cases can eliminate the need for Dropout [12]. However, it is rarely observed that SR networks overfit on training datasets. Instead, many kinds of regularizers, for examples, weight decaying and dropout, are not adopted in SR networks [7, 14, 17, 19, 33, 36, 42]. 3) Unlike image classification tasks where softmax (scale-invariant) is used at the end of networks to make prediction, for image SR, the different formulations of training and testing may deteriorate the accuracy for dense pixel value predictions.

Weight normalization Weight normalization, on the other hand, is a reparameterization of the weight vectors in a neural network that decouples the length of those weight vectors from their direction. It does not introduce dependencies between the examples in a mini-batch, and has the same formulation in training and testing. Assume the output $\mathbf{y}$ is with the form:

$$\mathbf{y} = \mathbf{w} \cdot \mathbf{x} + b, \tag{5}$$

where $\mathbf{w}$ is a k-dimensional weight vector, b is a scalar bias term, $\mathbf{x}$ is a k-dimensional vector of input features. WN re-parameterizes the weight vectors in terms of the new parameters using

$$\mathbf{w} = \frac{g}{||\mathbf{v}||}\mathbf{v}, \tag{6}$$

where v is a k-dimensional vector, g is a scalar, and $||\mathbf{v}||$ denotes the Euclidean norm of $\mathbf{v}$. With this formalization, we will have $||\mathbf{w}|| = g$, independent of parameters $\mathbf{v}$. As shown in [25], the decouples of length and direction speed up convergence of deep neural networks. And more importantly, for image SR, it does not introduce troubles of BN as described above, since it is just a reparameterization technique and has exact same representation ability.

It is also noteworthy that introducing WN allows training with higher learning rate (i.e. $10\times$), and improves both training and testing accuracy.

3.4 Network Structure

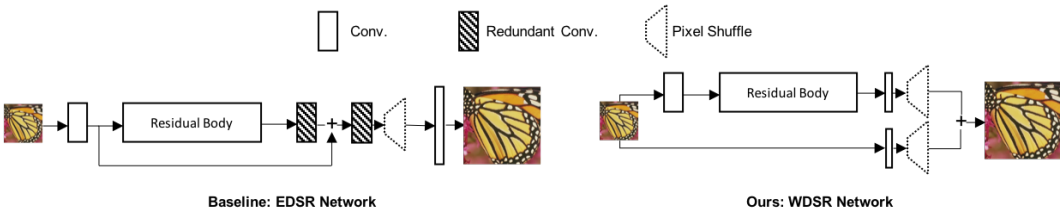

Figure 2: Demonstration of our simplified SR network compared with EDSR [19].

In this part, we overview the *WDSR* network architectures. We made two major modifications based on EDSR [19] super-resolution network.

Global residual pathway Firstly we find that the global residual pathway is a linear stack of several convolution layers, which is computational expensive. We argue that these linear convolutions are redundant (Fig. 2) and can be absorbed into residual body to some extent. Thus, we slightly modify the network structure and use single convolution layer with kernel size 5×5 that directly take $3 \times H \times W$ LR RGB image/patch as input and output $3S^2 \times H \times W$ HR counterparts, where S is the scale. This results in less parameters and computation. In our experiments we have not found any accuracy drop with our simpler form.

Upsampling layer Different from previous state-of-the-arts [19, 42] where one or more convolutional layers are inserted after upsampling, our proposed *WDSR* extracts all features in low-resolution stage (Fig. 2). Empirically we find it does not affect accuracy of SR networks while improves speed by a large margin.

4 Experimental Results

We train our models on DIV2K dataset [35] since the dataset is relatively large and contains high-quality (2K resolution) images. The default splits of DIV2K dataset consist 800 training images, 100 validation images and 100 testing images. We use 800 training images for training and 10 validation images for validation during training. The trained models are evaluated on 100 validation images (testing images are not publicly available) of DIV2K dataset. We mainly measure PSNR on RGB space. ADAM optimizer [16] is used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The batch size is set to 16. The learning rate is initialized the maximum convergent value (10-4 for models without weight normalization and 10-3 for models with weight normalization). The learning rate is halved at every 2×10^5 iterations.

We crop 96×96 RGB input patches from HR image and its bicubic downsampled image as training output-input pairs. Training data is augmented with random horizontal flips and rotations following common data augmentation methods [7, 19]. During training, the input images are also subtracted with the mean RGB values of the DIV2K training images.

Residual Blocks	1			3		
Networks	EDSR	WDSR-A	WDSR-B	EDSR	WDSR-A	WDSR-B
Parameters	2.6M	**0.8M**	**0.8M**	4.1M	**2.3M**	**2.3M**
DIV2K (val) PSNR	33.210	33.323	33.434	34.043	34.163	34.205

Residual Blocks	5			8		
Networks	EDSR	WDSR-A	WDSR-B	EDSR	WDSR-A	WDSR-B
Parameters	5.6M	**3.7M**	**3.7M**	7.8M	**6.0M**	**6.0M**
DIV2K (val) PSNR	34.284	34.388	34.409	34.457	34.541	34.536

Table 1: Model comparisons at different parameters budgets by controlling the number of residual blocks with fixed number of channels. We mainly compare the number of parameters and validation PSNR to measure efficiency and accuracy.

4.1 Wide and Efficient Wider Activation:

In this part, we show results of baseline model EDSR [19] and our proposed *WDSR-A* and *WDSR-B* for the task of image bicubic x2 super-resolution on DIV2K dataset. To ensure fairness, each model is evaluated at different parameters and computational budgets by controlling the number of residual blocks with fixed number of channels. The results are shown in Table 1. We compare each model with its number of residual blocks. The results suggest that our proposed *WDSR-A* and *WDSR-B* have better accuracy and efficiency than EDSR [19]. *WDSR-B* with wider activation also has better or similar performance compared with *WDSR-A*, which supports our wide activation hypothesis and demonstrates the effectiveness of our proposed *linear low-rank convolution*.

4.2 Normalization layers:

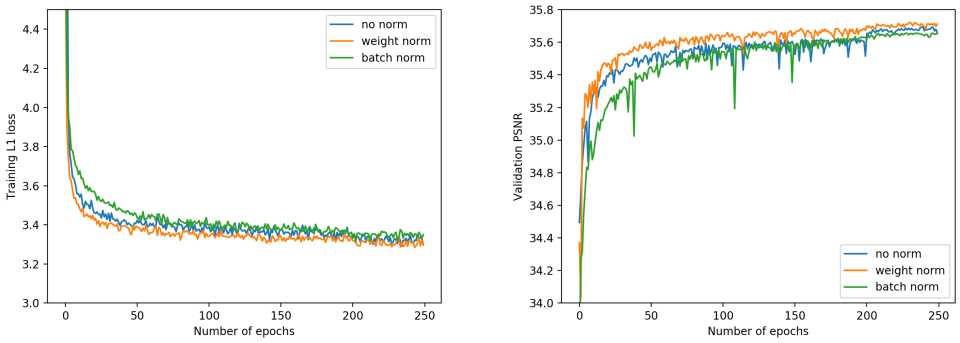

Figure 3: Training L1 loss and validation PSNR of same model trained with weight normalization, batch normalization or no normalization.

We also demonstrate the effectiveness of weight normalization for improved training of SR networks. We compare the training and testing accuracy (PSNR) when train the same model with different normalization methods, i.e. weight normalization, batch normalization or no normalization. The results in Figure 3 show that the model trained with weight normalization has faster convergence and better accuracy. The model trained with batch normalization is unstable during testing, which is likely due to different formulations of BN in training and testing.

To further study whether this is because the learning rate is too large for models trained with batch normalization, we also train the same model with different learning rates. The results are shown in Figure 4. Even with $lr = 10^{-4}$ when the training curves are stable, the validation PSNR is still not stable across training.

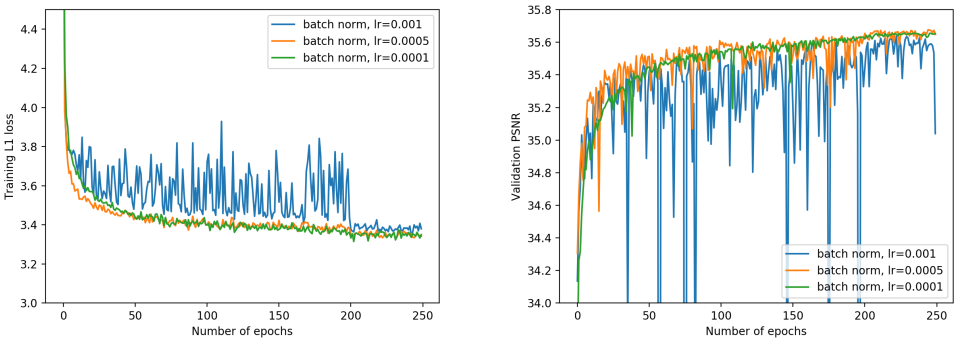

Figure 4: Training L1 loss and validation PSNR of model trained with batch normalization but different learning rates.

5 Conclusions

In this report, we introduce two super-resolution networks *WDSR-A* and *WDSR-B* based on the central idea of wide activation. We demonstrate in our experiments that with same parameter and computation complexity, models with wider features before ReLU activation have better accuracy for single image super-resolution. We also find training with weight normalization leads to better accuracy for deep super-resolution networks comparing to batch normalization or no normalization. The proposed methods may help to other low-level image restoration tasks like denoising and dehazing.

References

[1] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: ().

[2] Nadav Cohen, Or Sharir, and Amnon Shashua. "On the expressive power of deep learning: A tensor analysis". In: *Conference on Learning Theory*. 2016, pp. 698–728.

[3] Chao Dong, Chen Change Loy, and Xiaoou Tang. "Accelerating the super-resolution convolutional neural network". In: *European Conference on Computer Vision*. Springer. 2016, pp. 391–407.

[4] Chao Dong et al. "Learning a deep convolutional network for image super-resolution". In: *European Conference on Computer Vision*. Springer. 2014, pp. 184–199.

[5] Ronen Eldan and Ohad Shamir. "The power of depth for feedforward neural networks". In: *Conference on Learning Theory*. 2016, pp. 907–940.

[6] Yuchen Fan, Jiahui Yu, and Thomas S Huang. "Wide-activated Deep Residual Networks based Restoration for BPG-compressed Images". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[7] Yuchen Fan et al. "Balanced two-stage residual networks for image super-resolution". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE. 2017, pp. 1157–1164.

[8] Tomio Goto et al. "Super-resolution System for 4K-HDTV". In: *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE. 2014, pp. 4453–4458.

[9] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[10] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[11] Gao Huang and Zhuang Liu. "Densely connected convolutional networks". In:

[12] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. 2015, pp. 448–456.

[13] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. “Flattened convolutional neural networks for feedforward acceleration”. In: *arXiv preprint arXiv:1412.5474* (2014).

[14] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. “Accurate image super-resolution using very deep convolutional networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1646–1654.

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. “Deeply-recursive convolutional network for image super-resolution”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1637–1645.

[16] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[17] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: ().

[18] Shiyu Liang and R Srikant. “Why deep neural networks for function approximation?” In: *arXiv preprint arXiv:1610.04161* (2016).

[19] Bee Lim et al. “Enhanced deep residual networks for single image super-resolution”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Vol. 1. 2. 2017, p. 3.

[20] Ding Liu et al. “Robust single image super-resolution via deep networks with sparse prior”. In: *IEEE Transactions on Image Processing* 25.7 (2016), pp. 3194–3207.

[21] Ding Liu et al. “Robust video super-resolution with learned temporal dynamics”. In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE. 2017, pp. 2526–2534.

[22] Augustus Odena, Vincent Dumoulin, and Chris Olah. “Deconvolution and Checkerboard Artifacts”. In: *Distill* (2016). DOI: 10.23915/distill.00003. URL: http://distill.pub/2016/deconv-checkerboard.

[23] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. “Super-resolution image reconstruction: a technical overview”. In: *IEEE signal processing magazine* 20.3 (2003), pp. 21–36.

[24] Sharon Peled and Yehezkel Yeshurun. “Superresolution in MRI: application to human white matter fiber tract visualization by diffusion tensor imaging”. In: *Magnetic resonance in medicine* 45.1 (2001), pp. 29–35.

[25] Tim Salimans and Diederik P Kingma. “Weight normalization: A simple reparameterization to accelerate training of deep neural networks”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 901–909.

[26] Mark Sandler et al. “Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation”. In: *arXiv preprint arXiv:1801.04381* (2018).

[27] M. Sandler et al. “Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation”. In: *ArXiv e-prints* (Jan. 2018). arXiv: 1801.04381 [cs.CV].

[28] Franco Scarselli and Ah Chung Tsoi. “Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results”. In: *Neural networks* 11.1 (1998), pp. 15–37.

[29] Wenzhe Shi et al. “Real-time single image and video super-resolution using an efficient subpixel convolutional neural network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1874–1883.

[30] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[31] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning.” In: *AAAI*. Vol. 4. 2017, p. 12.

[32] Ying Tai, Jian Yang, and Xiaoming Liu. “Image super-resolution via deep recursive residual network”. In:

[33] Ying Tai et al. “Memnet: A persistent memory network for image restoration”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4539–4547.

[34] Matt W Thornton, Peter M Atkinson, and DA Holland. “Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping”. In: *International Journal of Remote Sensing* 27.3 (2006), pp. 473–491.

[35] Radu Timofte 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*. IEEE. 2017, pp. 1110–1121.

[36] Tong Tong et al. “Image Super-Resolution Using Dense Skip Connections”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 4809–4817.

[37] Zhangyang Wang et al. “Studying very low resolution recognition using deep networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4792–4800.

[38] Saining Xie et al. “Aggregated residual transformations for deep neural networks”. In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE. 2017, pp. 5987–5995.

[39] Jianchao Yang et al. “Image super-resolution via sparse representation”. In: *IEEE transactions on image processing* 19.11 (2010), pp. 2861–2873.

[40] Jiahui Yu et al. “Free-Form Image Inpainting with Gated Convolution”. In: *arXiv preprint arXiv:1806.03589* (2018).

[41] Jiahui Yu et al. “Generative Image Inpainting with Contextual Attention”. In: *arXiv preprint arXiv:1801.07892* (2018).

[42] Y. Zhang et al. “Residual Dense Network for Image Super-Resolution”. In: *ArXiv e-prints* (Feb. 2018). arXiv: `1802.08797 [cs.CV]`.