Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 139–147

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

A Light Weight Convolutional Neural Network for Single Image Super-Resolution

Kalpesh Prajapati, Vishal Chudasama, Kishor Upla*

Sardar Vallabhbhai National Institute of Technology, Surat, India

Abstract

Recently, many convolutional neural network based models obtain remarkable performance in single-image super-resolution task by stacking more number of convolution layers. However, those models require a huge amount of network parameters which increases the computational complexity of their single image super-resolution models. Due to this, they are no longer appropriate for many real-world applications. Hence, to design a network which can obtain better super-resolution performance with less number of network parameters is always an active area of research in the computer vision community. In this paper, we propose a light weight convolutional neural network based SR model called LWSRNet for the upscaling factor ×4. In LWSRNet, we introduce a novel basic block which helps to extract complex features of the given low-resolution observation. Additionally, we use a weighted L_2 loss function in order to train the network which is more effective than L_1 and L_2 loss functions. Various experiments have been carried out to validate the proposed method and observe that the super-resolution results obtained using the proposed LWSRNet method are better than that of the other existing single image super-resolution methods. Also, the proposed LWSRNet outperforms to the recently proposed state-of-the-art methods with approximately 20% to 60% less number of training parameters.

Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).

Keywords: Super-Resolution; Convolution Neural Network; Residual Network; Light weight

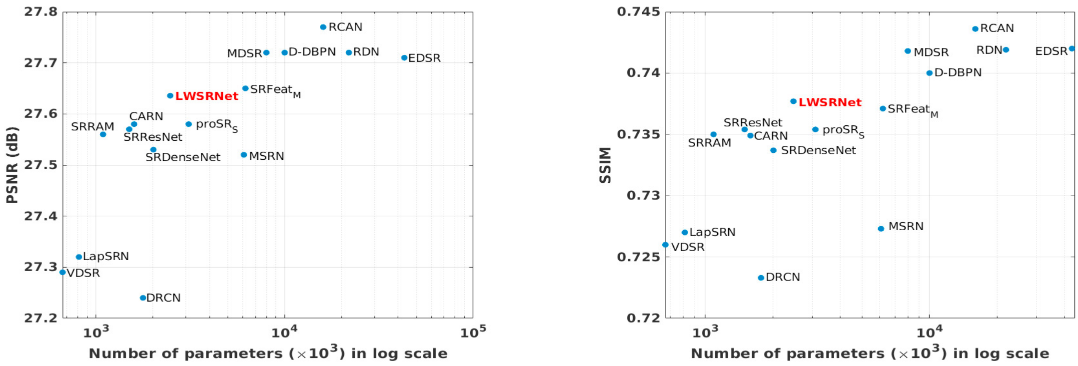
1. Introduction

Super-resolution (SR) is a low-level computer vision task in which the high-resolution (HR) image is reconstructed from its low-resolution (LR) version. The SR techniques can be categorized into two ways: multi images SR (MISR) and single-image SR (SISR). In MISR, additional tasks such as image registration and fusion of input images are necessary in order to obtain SR image which also impacts the quality of reconstructed SR image. Due to this, MISR approaches are less effective in SR task [20, 14, 3, 21]. While in SISR, the HR image is reconstructed directly from

* Corresponding author. Tel.: +91-261-220-1726.
E-mail address: kishorupla@gmail.com

1877-0509 © 2020 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).
10.1016/j.procs.2020.04.015



(a) PSNR vs. Number of parameters

(b) SSIM vs. Number of parameters

Fig. 1: The effect of PSNR and SSIM values of state-of-the-art SISR models along with the proposed LWSRNet model with respect to the number of training parameters on BSD100 testing dataset.

its LR observation [3, 21]. Here, the given LR image is upsampled to the spatial resolution of the HR image and the missing high-frequency details in the LR observation are learned through the SR approach.

Recently, deep learning has obtained break-through results in SISR tasks over traditional machine learning algorithms. Starting from SRCNN [4] with three convolution layers, many deep learning models have been proposed in the literature in which more number of convolution layers are added to improve the SR performance. This includes EDSR [13], DBPN [5], RDN [24] and RCAN [23] methods. Fig. 1 shows the quantitative comparison in terms of peak-signal-to-noise ratio (PSNR) and structural similarity (SSIM) measures with respect to the number of training parameters required to train the existing models of the state-of-the-art SISR methods for BSD100 testing dataset. Here, one can observe that EDSR [13], DBPN [5], RDN [24] and RCAN [23] methods obtain better SR performance; however, those methods require a large number of training parameters (i.e., more than 10M number of training parameters) in order to obtain SR result which makes them unsuitable for real-world applications. Hence, it is an active area of research to design light-weight (i.e., with fewer training parameters) deep learning models which can be useful in real-world applications. In this paper, we propose a light weight deep learning based SISR model called LWSRNet for upscaling factor ×4. From Fig. 1, one can notice that the proposed LWSRNet obtains better PSNR and SSIM measures on BSD100 testing dataset than that of recently proposed MSRN [12] and proSR$_s$ [19] methods while it obtains comparable SR performance with recently proposed SRFeat$_M$ [15] model in the light weight category. However, the proposed LWSRNet model obtains this performance with significantly less number of training parameters (i.e., approximately 20% to 60% less number of training parameters) than that of the recently proposed methods such as proSR$_s$ [19], MSRN [12] and SRFeat$_M$ [15] methods.

The main contributions in this paper are as follows:

- We propose a novel light weight SISR model called LWSRNet for upscaling factor ×4. The proposed LWSRNet obtains better SR results than that of the recently proposed state-of-the-art SISR methods with a significantly less number of training parameters.
- We use a weighted L_2 loss function (i.e., $L_{2-weighted}$) to train the proposed LWSRNet model. The effectiveness of the $L_{2-weighted}$ loss function is discussed separately in the experimental section and it exhibits better performance when compared to L_1 and L_2 loss functions.

2. Related Work

Recent advancements in terms of powerful graphical processing units (GPUs) and larger datasets increase the efficiency of deep learning based models in various computer vision tasks than the traditional algorithms. In the SISR task, deep learning (especially convolutional neural network (CNN)) based methods have obtained remarkable

Table 1: The comparison of different CNN based SISR algorithms in terms of upsampling strategy, number of filters, number of training parameters, implementation framework and loss function required to train for upscaling factor ×4. Here, the details of the proposed model are mentioned in bold font texts.

Method	Up-sampling	Filters	Parameters	Framework	Loss
SRCNN [4]	pre	64	57k	Caffe	L_2
VDSR [7]	pre	64	665k	Caffe	L_2
LapSRN [10]	progressive	64	812k	PyTorch	Charbonnier
SRRAM [9]	post	64	1,090k	TensorFlow	L_1
CMSC [6]	pre	64	1,220k	PyTorch	L_2
SRMD [22]	post	128	1,482k	MatConvNet	L_2
SRResNet [11]	post	64	1,500k	Theano	L_2
CARN [2]	post	64	1,592k	PyTorch	L_1
DRCN [8]	pre	256	1,775k	Caffe	L_2
SRDenseNet [18]	post	128	2,015k	TensorFlow	L_2
LWSRNet	**post**	**64**	**2,475k**	**PyTorch**	$L_{2-weighted}$
proSR$_s$ [19]	progressive	—	3,100k	PyTorch	L_1
MSRN [12]	post	64	6,078k	PyTorch	L_1
SRFeat$_M$ [15]	post	128	6,196k	TensorFlow	L_2
D-DBPN [5]	post	64	10,000k	Caffe	L_2
proSR$_l$ [19]	progressive	—	15,500k	PyTorch	L_1
RCAN [23]	post	64	16,000k	PyTorch	L_1
RDN [24]	post	64	21,900k	Torch	L_1
EDSR [13]	post	256	43,000k	Torch	L_1

performance in the SISR task. A detailed analysis of these methods have been discussed in [3, 21]. The pioneer work in this category referred as the Super-Resolution Convolutional Neural Network (SRCNN) is carried out by Dong et al. [4] which consists of three convolution layers. After that, many methods have been proposed to improve the performance of SR images by stacking more number of convolution layers [7, 10, 11, 18, 13, 5, 12, 15, 19, 24, 23].

Table 1 shows a brief summary of CNN based state-of-the-art SISR methods. Here, we compare different CNN based state-of-the-art SISR methods in terms of their upsampling strategy, number of filters required in the network, number of trainable parameters, implementation framework and loss function required to train their model for upscaling factor ×4. In Table 1, all methods are sorted in ascending order as per their number of training parameters. The details of the proposed LWSRNet model is also mentioned on the same table along with the other SISR methods. From Table 1, one can observe that EDSR [13], D-DBPN [5], proSR$_l$ [19], RDN [24] and RCAN [23] methods obtained their SR performance with large number of training parameters (i.e., more than 10M parameters). However, the proposed LWSRNet model requires approximately 75% to 95% less number of training parameters when compared to those methods [13, 5, 19, 24, 23]. Hence, in this paper, we have compared the SR results obtained using the proposed model with that of those methods which have less than 7M number of training parameters. This comparison is explained in detail in the experimental section 4.2.

3. Proposed Method: LWSRNet

In this section, we discuss the design of the network architecture of the proposed SR model called LWSRNet for the upscaling factor ×4. Fig. 2 depicts the architecture design of the proposed LWSRNet. Here, the LR image (I^{LR}) is applied to network as input and then it passed through the network and finally SR image with ×4 upscaling factor (i.e., $I^{SR}_{\times 4}$) is obtained as output. The network design mainly consists of three modules:

- low-frequency feature extraction,
- high-frequency feature extraction and

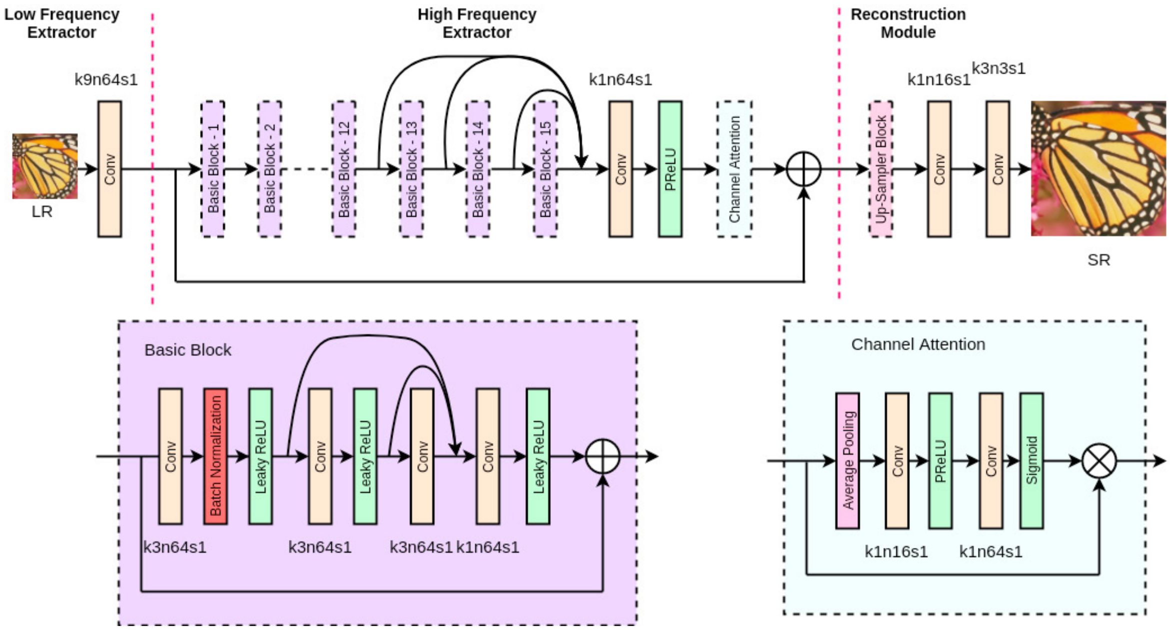


Fig. 2: The network architecture of the proposed model. Here, k, n and s denote the filter size, number of filters and stride value, respectively.

- reconstruction module.

In the low-frequency feature extraction module, one convolution layer with kernel size (9, 9) is used which extracts the low-level information from the LR observations. Larger kernel size provides a larger receptive field to extract information. The low-frequency feature maps are extracted as,

$$I_{low-freq} = LFE(I^{LR}), \tag{1}$$

where $LFE(\cdot)$ denotes the operation of the low-frequency extraction module.

The high-frequency extraction module consists of several basic blocks and one channel attention module. Along with the short-range skip connections, we also adopt the long-range skip connection in order to reduce the vanishing gradient problems. The basic block as displayed in Fig. 2 which is the key component of this module. Each basic block consists of four convolution layers followed by leakyReLU activation functions, one batch-normalization layer, skip connection and concatenation connections. Such basic block helps to extract more complex high-frequency features. In order to concentrate on only high-frequency components, we concatenate the feature maps of the last four basic blocks (see Fig. 2). The concatenated feature maps are passed through one convolution layer which maps the feature maps to the desired number of feature maps. Finally, one channel attention module is used to re-scale the feature maps adaptively. Here, we use the same channel attention module as described in [23] (which is also depicted in Fig. 2) with parametric ReLU (PReLU) as activation function instead of ReLU activation function which was used by authors in [23]. The output feature maps of high-frequency extraction module are described as,

$$\begin{aligned} I_{HFE} &= HFE(I_{low-freq}), \\ I_{high-freq} &= I_{HFE} + I_{low-freq}. \end{aligned} \tag{2}$$

Here, HFE denotes the function of the high-frequency extraction module.

In the reconstruction module, the output feature maps of the high-frequency feature extraction module are upsampled to the desired resolution using the upsampler block. Here, we use an efficient sub-pixel convolution neural network [16] as an upsampler block. The upsampled feature maps are followed by two convolution layers to obtain final SR image (i.e., $I^{SR}_{\times 4}$) as

$$I^{SR}_{\times 4} = R(I_{high-freq}), \tag{3}$$

where R indicates the function of the reconstruction module which consists of one upsampling function followed by two convolution operations.

3.1. Loss function

In this subsection, we describe the loss function which is used to train the proposed LWSRNet model. Most of the recent SISR methods use either L_1 or L_2 loss function to train their models. In order to improve the efficiency of the loss function, we use a weighted L_2 loss function (i.e, $L_{2-weighted}$) which is based on classical L_2 loss function as,

$$L_{2-weighted} = 100 \times \left[\frac{1}{NWH} \times \sum_{k=1}^{N} \sum_{i=1}^{H} \sum_{j=1}^{W} [I^{HR}_{i,j,k} - I^{SR}_{i,j,k}]^2\right], \tag{4}$$

where H and W indicate the height and width of corresponding images. Here, N denotes the batch-size. The effectiveness of this weighted loss function is evaluated and tested in result section 4.2.

4. Result analysis

In this section, we present the detailed description of the result obtained using the proposed LWSRNet along with the other existing state-of-the-art SISR methods for upscaling factor ×4. All the experiments have been performed on a computer with the following specifications: Intel Xeon(R) CPU E5-2620 v4 processor @2.10$GHz \times 32$, 128GB RAM and NVIDIA Quadro P5000 16GB GPU.

4.1. Training details and hyper-parameter settings

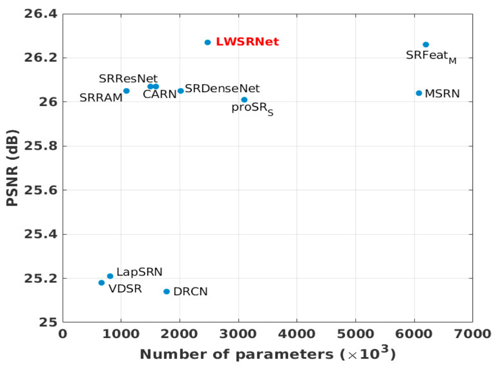
We train our proposed model up to 10^6 number of iterations with a batch size of 16 on the DF2K dataset. This dataset consists of 800 images of DIV2K [1] and 2500 images of Flickr2k [17] datasets. The training images are augmented using random rotation (i.e., 0^0 and 90^0), random cropping and horizontal flipping operations. The proposed model LWSRNet is optimized with Adam optimizer with an initial learning rate of 2×10^{-4} and is evaluated and tested on four testing benchmark datasets: Set5, Set14, BSD100 and Urban100. The PSNR and SSIM are used as evaluation measurements for quantitative comparison. These measurements are calculated after removing the four boundary pixels of Y-channel images in YCbCr color space as suggested in [11, 15, 13, 23]. As mentioned earlier, different methods, such as EDSR [13], DBPN [5], RDN [24] and RCAN [23] obtained better SR performance than that of the proposed LWSRNet, however, these methods require a large number of training parameters. Hence, in this paper, we choose only those SISR methods which require less than 7M number of training parameters and compare their SR results with the SR results obtained using the proposed LWSRNet method.

4.2. Result analysis

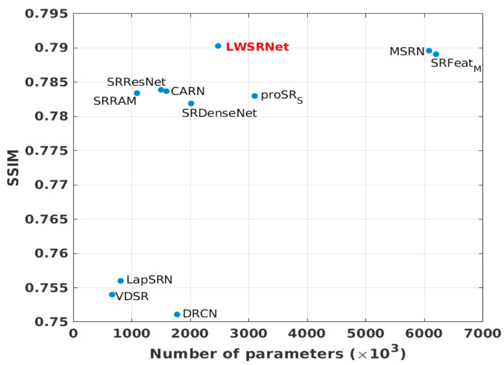
In this subsection, we analyze the qualitative and quantitative SR performance of the proposed model with that of other existing state-of-the-art SISR methods. The quantitative comparison in terms of PSNR and SSIM measures

Table 2: The quantitative comparison in terms of PSNR and SSIM measures for the upscaling factor ×4. Here, the highest and second-highest values are mentioned with red and blue color fonts, respectively.

Methods	PSNR / SSIM			
	Set5	Set14	BSD100	Urban100
Bicubic	28.43/0.8111	26.25/0.7187	25.96/0.6680	23.14/0.6580
SRCNN [4]	30.48/0.8628	27.50/0.7513	26.90/0.7103	24.52/0.7226
VDSR [7]	31.35/0.8838	28.02/0.7678	27.29/0.7252	25.18/0.7525
LapSRN [10]	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560
CMSC [6]	31.91/0.8923	28.35/0.7751	27.46/0.7308	25.64/0.7692
SRMD [22]	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731
SRDenseNet [18]	32.02/0.8934	28.50/0.7782	27.53/0.7337	26.05/0.7819
SRResNet [11]	32.05/0.8910	28.53/0.7804	27.57/0.7354	26.07/0.7839
MSRN [12]	32.07/0.8903	28.60/0.7750	27.52/0.7273	26.04/0.7896
SRRAM [9]	32.13/0.8932	28.54/0.7800	27.56/0.7350	26.05/0.7834
CARN [2]	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
proSR$_s$ [19]	32.13/0.8941	28.62/0.7815	27.58/0.7354	26.01/0.7830
SRFeat$_M$ [15]	32.29/0.8957	28.72/0.7834	27.65/0.7371	26.26/0.7891
LWSRNet	32.22/0.8952	28.68/0.7837	27.64/0.7377	26.27/0.7903



(a) PSNR vs. Number of parameters



(b) SSIM vs. Number of parameters

Fig. 3: The effect of PSNR and SSIM values of state-of-the-art SISR models along with the proposed LWSRNet model with respect to the number of training parameters on Urban100 dataset.

are detailed in Table 2. Here, the highest and second-highest measured values are written in red and blue color text, respectively. Also, in this table, we compare the PSNR and SSIM values of the proposed LWSRNet with that of those existing state-of-the-art SISR methods which have less than 7M number of the training parameters. From the Table 2, one can notice that the proposed LWSRNet outperforms to the other state-of-the-art SISR methods except recently proposed SRFeat$_M$ [15] in which the proposed LWSRNet obtains comparable quantitative performance.

Fig. 3 shows the graph of quantitative measurements (i.e., PSNR and SSIM) obtained using the state-of-the-art SISR methods on Urban100 dataset with respect to the number of training parameters required to train their models. One can observe from the Fig. 3(a) that the proposed model LWSRNet obtains better PSNR value than the recently proposed MSRN [12] and proSR$_s$ [19] methods, while it obtains comparable performance with recently proposed SRFeat$_M$ [15] method. In case of Fig. 3(b), the LWSRNet obtains better SSIM value than that of MSRN [12], proSR$_s$ [19] and SRFeat$_M$ [15] methods. However, the LWSRNet exhibits this performance with significantly less number of training parameters. To observe the effectiveness of the proposed loss function (i.e., $L_{2-weighted}$), we also train the proposed LWSRNet model using L_1 and L_2 loss functions for upscaling factor ×4. Table 3 shows the quantitative

Table 3: The comparison of different loss functions of the proposed LWSRNet in terms of PSNR and SSIM measurements on four testing datasets for the upscaling factor ×4. Here, the highest value among all is mentioned in bold font text.

Methods	Set5	Set14	BSD100	Urban100
L_2	32.20/0.8949	28.66/0.7834	27.62/0.7375	26.26/0.7902
L_1	**32.23/0.8954**	28.64/0.7823	27.62/0.7368	26.20/0.7884
$L_{2-weighted}$	32.22/0.8952	**28.68/0.7837**	**27.64/0.7377**	**26.27/0.7903**

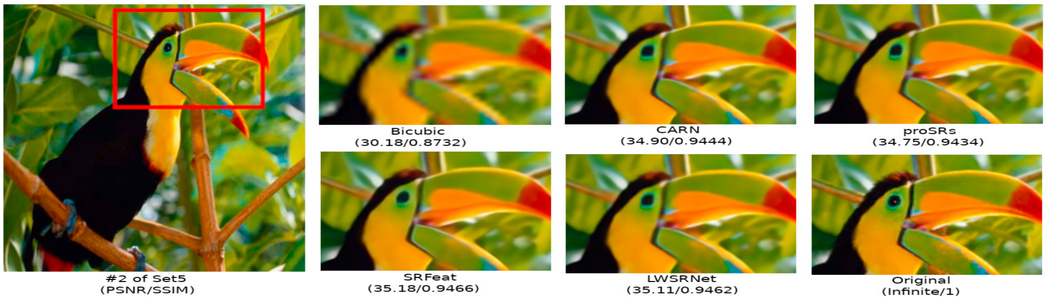
comparison in terms of PSNR and SSIM measures obtained using the different loss functions for the upscaling factor ×4. Here, the bold font text indicates the highest measures. One can notice from Table 3 that the proposed LWSRNet model with weighted loss function (i.e., $L_{2-weighted}$) obtains better quantitative measures than that of obtained using L_1 and L_2 loss functions on Set14, BSD100 and Urban100 testing dataset. In case of Set5 dataset, the weighted loss function (i.e., $L_{2-weighted}$) obtains comparable performance with measurements obtained using L_1 loss function; L_1 loss function is marginally better than that of $L_{2-weighted}$ loss function. In addition to the quantitative comparison, we carried out the qualitative comparison of SR results obtained using the proposed LWSRNet as well as other existing state-of-the-art SISR methods. This comparison is depicted in Fig. 4(a-d) for a single image of Set5, Set14, BSD100 and Urban100 datasets, respectively for the upscaling factor ×4. Here, the SR results obtained using the proposed LWSRNet method are compared with the bicubic interpolation method and other three recently proposed state-of-the-art SR methods such as CARN [2], proSR$_s$ [19] and SRFeat$_M$ [15] whose number of training parameters are less than 7M. These methods are chosen based on their SR performance and the number of trainable parameters required in their model. To see the preservation of high-frequency details in the SR results, instead of the complete image, only zoomed-in patches of those methods are displayed in Fig. 4(a-d). Along with the zoomed-in patches of SR results, the corresponding quantitative measurements (i.e., PSNR and SSIM values) are also mentioned at the bottom of each SR results. From Fig. 4(a), one can find that the proposed LWSRNet model outperforms to the state-of-the-art CARN [2] and proSR$_s$ [19] methods in terms of qualitative as well as quantitative measurements while it obtains comparable performance with the recently proposed SRFeat$_M$ [15] model. However, in case of Set14, BSD100 and Urban100 dataset (see Fig. 4(b-d)), the proposed LWSRNet model obtains better qualitative as well as quantitative measurements than that of CARN [2], proSR$_s$ [19] and SRFeat$_M$ [15] methods. Moreover, the proposed LWSRNet model obtains such performance with a significantly less number of training parameters than that of CARN [2], proSR$_s$ [19] and SRFeat$_M$ models.

5. Conclusion

In this paper, we propose a light-weight CNN based SISR model called LWSRNet for upscaling factor ×4. The proposed LWSRNet perform better in qualitative as well as quantitative inspections than that of the recently proposed state-of-the-art SISR methods with a significantly less number of training parameters. In LWSRNet, a novel basic block is introduced which helps to learn more complex features from the LR observation. We also suggest a weighted loss function $L_{2-weighted}$ to train the proposed model. We compare the SR results obtained using the proposed LWSRNet method with that of the other existing SR methods and it shows that the proposed LWSRNet outperforms to the recently proposed state-of-the-art MSRN [12], proSR$_s$ [19] and SRFeat$_M$ [15] methods. However, we obtain such performance with approximately 20% to 60% less number of training parameters.

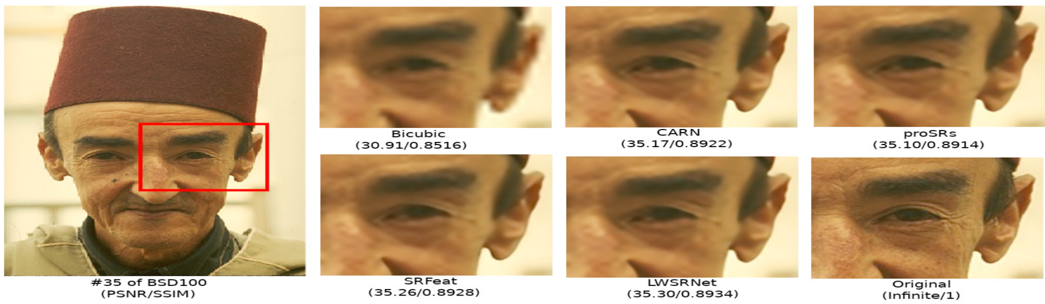
Acknolwdgement

This work was supported by the Science and Engineering Research Board (SERB), a statutory body of the Department of Science and Technology (DST), Government of India [Grant: ECR/2017/003268].

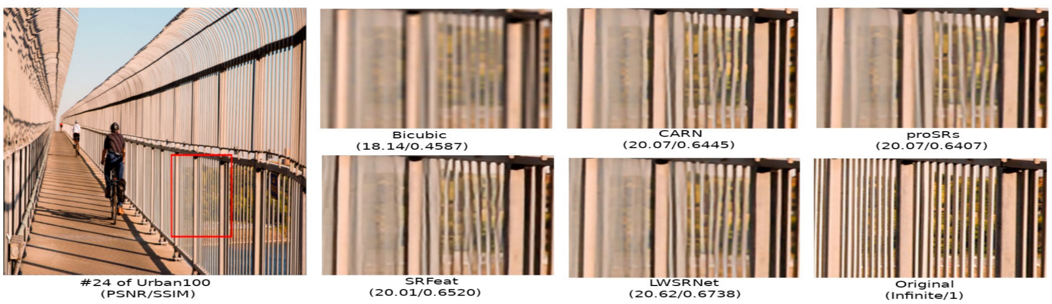


(a) Set5

(b) Set14



(c) BSD100



(d) Urban100

Fig. 4: The SR results obtained using the proposed along with the other existing state-of-the-art SISR methods on different datasets for the upscaling factor ×4.

References

[1] Agustsson, E., Timofte, R., 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, p. 2.

[2] Ahn, N., Kang, B., Sohn, K.A., 2018. Fast, accurate, and lightweight super-resolution with cascading residual network, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 252–268.
[3] Anwar, S., Khan, S., Barnes, N., 2019. A deep journey into super-resolution: A survey. CoRR abs/1904.07523.
[4] Dong, C., Loy, C.C., He, K., Tang, X., 2016. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38, 295–307.
[5] Haris, M., Shakhnarovich, G., Ukita, N., 2018. Deep back-projection networks for super-resolution, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1664–1673.
[6] Hu, Y., Gao, X., Li, J., Huang, Y., Wang, H., 2018. Single image super-resolution via cascaded multi-scale cross network. arXiv preprint arXiv:1802.08808 .
[7] Kim, J., Lee, J.K., Lee, K.M., 2016a. Accurate image super-resolution using very deep convolutional networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1646–1654.
[8] Kim, J., Lee, J.K., Lee, K.M., 2016b. Deeply-recursive convolutional network for image super-resolution. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 1637–1645.
[9] Kim, J.H., Choi, J.H., Cheon, M., Lee, J.S., 2018. Ram: Residual attention module for single image super-resolution. arXiv preprint arXiv:1811.12043 .
[10] Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H., 2018. Fast and accurate image super-resolution with deep laplacian pyramid networks. IEEE transactions on pattern analysis and machine intelligence .
[11] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4681–4690.
[12] Li, J., Fang, F., Mei, K., Zhang, G., 2018. Multi-scale residual network for image super-resolution, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 517–532.
[13] Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., 2017. Enhanced deep residual networks for single image super-resolution. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) , 1132–1140.
[14] Nasrollahi, K., Moeslund, T.B., 2014. Super-resolution: a comprehensive survey. Machine vision and applications 25, 1423–1468.
[15] Park, S.J., Son, H., Cho, S., Hong, K.S., Lee, S., 2018. Srfeat: Single image super-resolution with feature discrimination, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 439–455.
[16] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883.
[17] Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., 2017. Ntire 2017 challenge on single image super-resolution: Methods and results, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 114–125.
[18] Tong, T., Li, G., Liu, X., Gao, Q., 2017. Image super-resolution using dense skip connections, in: Computer Vision (ICCV), 2017 IEEE International Conference on, IEEE. pp. 4809–4817.
[19] Wang, Y., Perazzi, F., McWilliams, B., Sorkine-Hornung, A., Sorkine-Hornung, O., Schroers, C., 2018. A fully progressive approach to single-image super-resolution, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
[20] Yang, C.Y., Ma, C., Yang, M.H., 2014. Single-image super-resolution: A benchmark, in: European Conference on Computer Vision, Springer. pp. 372–386.
[21] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.H., Liao, Q., 2019. Deep learning for single image super-resolution: A brief review. IEEE Transactions on Multimedia .
[22] Zhang, K., Zuo, W., Zhang, L., 2018a. Learning a single convolutional super-resolution network for multiple degradations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3262–3271.
[23] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y., 2018b. Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 286–301.
[24] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y., 2018c. Residual dense network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2472–2481.