The Visual Computer (2022) 38:31–50
https://doi.org/10.1007/s00371-020-01998-z

ORIGINAL ARTICLE

Joint restoration convolutional neural network for low-quality image super resolution

Gadipudi Amaranageswarao[1] · S. Deivalakshmi[1] · Seok-Bum Ko[2]

Accepted: 13 October 2020 / Published online: 1 November 2020

Abstract
In this paper, a joint restoration convolutional neural network (JRCNN) is proposed to produce a visually pleasing super resolution (SR) image from a single low-quality (LQ) image. The LQ image is a low resolution (LR) image with ringing, blocking and blurring artifacts arising due to compression. JRCNN consists of three deep dense residual blocks (DRB). Each DRB comprises of parallel convolutional layers with cross residual connections. The representational power of JRCNN is improved by depth-wise concatenation of feature representations from each of the DRBs. Moreover, these connections mitigate the problem of vanishing of gradients. Different from the previous networks, JRCNN exploits the contextual information directly in the LR image space without using any interpolation. This strategy improves the training efficiency of the network. The exhaustive experimentation on different datasets show that the proposed JRCNN produces state-of-the-art performance. Furthermore, ablation experiments are performed to assess the effectiveness of JRCNN. In addition, individual experiments are conducted for SR and compression artifact removal on benchmark datasets.

Keywords Blocking artifacts · Cross residual connections · Dense residual blocks · Ringing · Skip connections

1 Introduction

In this era of information explosion, transfer of multimedia content among different networks and devices have become more popular. Compression and down sampling are the most common image degradations. Images are available in degraded form on the web due to the availability of low storage space. Furthermore, images are compressed while being transmitted through low bandwidth channels. Down sampling exploits the spatial redundancy (duplicate pixels) in an image while compression further exploits the correlation in frequency (DCT coefficients) and temporal domains for frames in a video. Even though the degraded image requires less storage space, it contains unpleasant visual artifacts, i.e., ringing, blocking and blurring. These complex visual artifacts severely affect the user experience (e.g., Fig. 1). Accurate reconstruction of the super resolution (SR) image from a single LQ image is a challenging task. There is a huge requirement for improvement in visual quality of reconstructed SR image.

The lossy Joint Photographic Experts Group (JPEG) compression [32] standard introduces blocking artifacts (e.g., Fig. 2) due to discontinuities between the adjacent 8×8 pixel blocks, while the ringing and blurring artifacts occur during the coarse quantization of high frequency details. The JPEG 2000 compression standard uses wavelet transform and avoids the blocking artifacts but still suffers from the ringing and the blurring artifacts. The main objective of this paper is to address super resolution (SR), compression artifact removal (CAR) and compressed image super resolution (CISR) from a single low-quality image. SR focuses mainly on the estimation of missing high frequency details (e.g., Fig. 3), while CAR focuses on the removal of ringing and blocking artifacts from the images compressed using the JPEG standard that's widely adopted on the internet. CISR focuses on the SR of low resolution (LR) compressed images. In this paper, a joint restoration convolutional neural network (JRCNN) architecture is proposed to address all the three tasks, i.e., SR, CAR and CISR.

Figure 1 represents the SR of a LQ image for a JPEG quality factor (QF) of 10 and an up scaling factor of 2 ($\times2$).

✉ Gadipudi Amaranageswarao
amar.nag9@gmail.com

[1] Department of Electronics and Communication Engineering, National Institute of Technology, Thiruchirappalli, India

[2] Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada

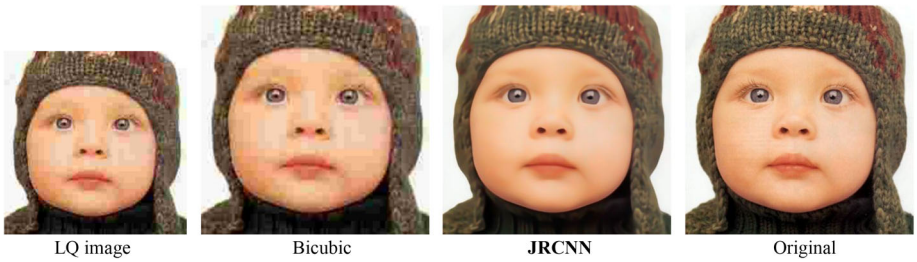

Fig. 1 Super resolution by $\times 2$ for a JPEG compressed image with a quality factor of 10

Fig. 2 Artifacts removal from a JPEG compressed image with a quality factor of 10

Fig. 3 Super resolution by $\times 4$

From Fig. 1, one can notice that the LQ image is an LR and compressed image. Bicubic interpolation amplifies the compression artifacts and produces visually unpleasant outputs, whereas the image produced by JRCNN is visually pleasant. Figure 2 represents the CAR of a JPEG compressed image with a quality factor of 10. The lower quality factors represent the higher compression. From Fig. 2, one can observe that the visually annoying artifacts are present in the JPEG decompressed image. These artifacts are removed in the image produced by the JRCNN. Similarly, Fig. 3 represents the SR of a LR image for $\times 4$. Blurring artifacts can be clearly observed in the bicubic interpolated image. These blurring artifacts are removed using the JRCNN. In all Figures 1, 2 and 3, it can be observed that the outputs of the JRCNN are much more visually pleasant than the LR input image and the bicubic interpolated image.

Nowadays deep learning-based methods are able to produce state-of-the-art performance. An end-to-end learning strategy is adopted in this paper using deep convolutional neural network (DNN). The CISR involves two different tasks, i.e., removal of complex artifacts and increasing the spatial resolution of the image. If these operations are performed one after the other, i.e., SR followed by CAR or CAR followed by SR, perceptually unpleasant images will be produced. In SR followed by CAR strategy, the complex artifacts are amplified in the SR stage and CAR stage is not able to remove these artifacts. Similarly, in CAR followed by SR, the CAR stage removes some of the essential information like edges and textures which are required in SR stage for accurate reconstruction. So, in this paper a joint restoration DNN is proposed to learn an end-to-end mapping function between LQ image I_{LQ} and the corresponding residual image I_{Res} formed by subtracting bicubic interpolated image I_{Bic} [15] from the original image I_{HR}. The I_{LQ} can be obtained from compression and down sampling of I_{HR} and is formulated as follows:

$$I_{\mathrm{LQ}} = C H I_{\mathrm{HR}} \tag{1}$$

where C is the composite operator of compression and decompression and H is the down sampling operator. The aim of CISR is to obtain I_{HR} from I_{LQ}. Out of many existing compression standards for still images, the JPEG is one of the widely used compression standard. In this paper, the JPEG compression standard is taken as an example. In JPEG compression standard, quality factor (QF) decides the amount of compression and perceptual quality of an image. Lower value of QF corresponds to higher compression and reduced storage size of the image.

In Eq. 1, if C is an identity matrix then I_{LQ} is just the down sampled version of I_{HR}. SR of I_{LQ} produces I_{HR}. If H is an identity matrix then I_{LQ} is just the compressed and decompressed version of I_{HR}. In this case different artifacts must be removed from I_{LQ} occurred during compression and decompression operations on I_{HR}. If none of C and H are identity matrices then the operation is simultaneous removal of compression artifacts and SR. To perform all these operations individually, a joint restoration convolutional neural network (JRCNN) is proposed in this paper. The novel contributions in this paper are as follows;

- Deep dense residual blocks (DRB) with parallel convolutional layers (PCL) are proposed for efficient training of the JRCNN.
- Cross residual connections (CRC) are introduced in each dense residual block to ease the training and to improve the training efficiency. Moreover, these connections mitigate the problem of gradient explosion.
- Skip connections are introduced among different deep dense residual blocks to improve the representational power of the JRCNN. Furthermore, skip connections avoid the problem of vanishing of gradients by creating short paths for efficient flow of gradients. In addition, feature redundancy is avoided by using skip connections.
- Different ablation experiments have been conducted on benchmark datasets to access the superiority of the JRCNN.
- Different experiments like super resolution (SR), compression artifact removal (CAR) and compressed image super resolution (CISR) have been performed using the JRCNN on standard datasets.

The rest of the paper is organized as follows: Sect. 2 reviews the related works for SR, CAR and CISR. Section 3 presents the network architecture of JRCNN and the training methodology. Section 4 shows the extensive experiments on different datasets along with ablation studies. Finally, Sect. 5 is the conclusion.

2 Related work

Recently, deep learning-based methods have gain popularity due to the availability of modern GPUs and they can produce state-of-the-art performance. Different methods have been proposed for SR, CAR and CISR using DNNs. A three-layered DNN named SRCNN has been proposed for SR [8] with lightweight. The SRCNN cannot perform multi-scale SR. Moreover, SRCNN takes interpolated LR image as input. To accelerate the SRCNN, a network named FSRCNN [9] has been proposed. The FSRCNN improved the speed of training and testing by directly performing SR of LR image using deconvolution layer as output layer without needing any interpolation. The FSRCNN also cannot provide multi-scale SR. To provide multi-scale SR, a very deep network named VDSR [16] has been proposed with 20 layers. However, the VDSR requires bicubic interpolation of LR image to support multi-scale SR. Similar to the VDSR, a denoising network named DnCNN [36] has been proposed with batch-normalization layers. The DnCNN can perform multiple tasks like SR, CAR and denoising. Densely connected DNN with deep dilation filters has been proposed in DenseDbSR [2] to perform SR, deblocking and simultaneous deblocking, SR. To control the number of parameters, recursive learning has been proposed with local and global residual connections in DRRN [28] for SR. The DRRN contains up to 52 convolutional layers. In LapSRN [18], the sub-band residuals of HR images are reconstructed at multiple pyramid levels. Furthermore, the LapSRN uses recursive learning for parameter sharing across as well as within pyramid levels.

A persistent memory network [29] consisting of recursive units and gate units has been proposed for three image restoration tasks, i.e., JPEG deblocking, SR and denoising. The optimal balance between the accuracy and the speed for SR has been achieved with balanced two-stage residual network (BTSRN) [10] with constrained depth. Different from other methods, SRMD network [37] has been proposed for SR with multiple degradations by considering blur kernel and noise level. A multi-level wavelet DNN (MWCNN) has been proposed [21] to reduce the computational cost by adopting U-Net architecture for SR. The MWCNN consists of contracting sub-network for reducing the size of feature maps by applying wavelet transform and expanding the sub-network for constructing the HR images by deploying inverse wavelet transform. A cascading residual network (CARN) has been proposed [1] for accurate SR with multiple local and global cascading connections which allow efficient flow of data and gradients. An enhanced deep SR (EDSR) network has been proposed [20] for performance improvement by removing the batch-normalization layer in the residual block.

A combination of cascading residual network (CRN) and an enhanced residual network (ERN) has been proposed [19] for SR. The CRN contains several locally sharing groups (LSGs) to promote propagation of information and gradients. The ERN enhances the image resolution. A residual dense network (RDN) has been proposed [39] for SR to fully exploit the hierarchical features from all the layers. The RDN uses a contiguous memory (CM) mechanism to improve the SR performance. A residual channel attention network (RCAN) has been proposed [38] for SR to learn high frequency information and low frequency information is bypassed through multiple skip connections with residual in residual (RIR) structure. Very recently, progressive growing methodologies have been proposed [22] for SR using a generative adversarial network (PG-GAN). In the PG-GAN, the training has been stabilized and accelerated by adopting progressive growing methodologies. An embedded block residual network (EBRN) has been proposed [26] with shallower modules for low frequency estimation and deeper modules for estimation of high frequency information.

In the past decade, several methods have been proposed for the reduction of JPEG compression artifacts. Most of the existing methods have been proposed to remove blocking artifacts. Recently, the DNN-based methods have been proposed to remove visually dominating artifacts occurred during JPEG compression and decompression, i.e., blocking, ringing and blurring. Shape adaptive discrete cosine transformation (SA-DCT) in conjunction with local polynomial approximation (LPA) has been used [11] for deblocking and deringing. To address JPEG compression artifacts a DNN named ARCNN has been proposed [7] with large-stride convolutional and deconvolutional layers. The ARCNN has a compact structure with just four convolutional layers but multi-scale CAR is not possible. A filter-based deblocking method [14] has been proposed by treating the blocking artifacts as an outlier random variable. This method aims at preserving the edge structures while removing the artifact outliers. CONstrained non-COnvex LOw-Rank (CONCOLOR) model has been proposed [35] by formulating deblocking as an optimization problem within maximum a posteriori framework.

A flexible learning framework has been proposed [6] based on trainable nonlinear reaction diffusion (TNRD) for various image restoration problems, i.e., SR, CAR and denoising. A 12-layer DNN named CAS-CNN has been proposed [4] with hierarchical skip connections and a multi-scale loss function to suppress compression artifacts. A slight modified version of CONCOLOR has been proposed with two new priors, i.e., structural sparse representation (SSR) prior and quantization constraint (QC) prior [40] for efficient deblocking. In the SSRQC method, the SSR prior is used for simultaneous enforcement of the intrinsic local sparsity and the nonlocal self-similarity of natural images and QC is used to ensure a more reliable and robust estimation. An adaptive distribution estimation and QC in DCT domain has been used in addition to patch sparse modeling in PCA domains on external images for robust deblocking [27]. A residual encoder–decoder (RED) network has been proposed [23] with dense skip connections among different convolutional and deconvolutional layers for multiple image restoration tasks. A modified inception module-based artifact removal DNN (IACNN) has been proposed [17] for blind and non-blind CAR. In IACNN, the compression quality factor is estimated first from the real compressed image and then the corresponding trained model has been used for CAR.

A highly compressed image is usually not only of low resolution but also contains visually annoying complex artifacts. Direct SR of these images would also magnify the artifacts. To solve this problem, a learning-based joint SR and deblocking (LJSRDB) method has been proposed [13]. In the LJSRDB method, sparse representation of LR and HR image patches with and without blocking artifacts have been exploited for joint SR and deblocking. Furthermore, morphological component analysis (MCA)-based image decomposition is also employed. Adjusted Anchored Neighborhood Regression (A+) [30] method has been proposed to increase the spatial resolution of an image with compression noise. SR in compressed domain based on field of experts (SRCDFOE) has been proposed [33] to produce visually pleasing images from compressed and LR images. In SRCDFOE method, HR image is modeled as a high-order Markov random field while compression is modeled as an additive and spatially correlated Gaussian noise. Recently, an end-to-end learning-based DNN has been proposed [5] with the name CISRDCNN to address SR of a compressed as well as LR image. The proposed JRCNN is also an end-to-end learning framework which is used to enhance the spatial resolution of the compressed low resolution images.

3 Methodology

In this section, the architecture and the structural components proposed for compressed image SR are discussed. The architecture of the JRCNN is shown in Fig. 4. The JRCNN takes I_{LQ} image as input and produces the corresponding I_{Res} of I_{HR} image. In SR, the input and the output images are highly correlated. Moreover, residual learning eases the training of the JRCNN. As shown in Fig. 4, the JRCNN consists of 47 convolutional layers and 2 transposed convolutional layers. Each of these layers are followed by a LeakyReLu layer with a scaling factor of 0.2 except the last convolutional layer which reconstructs I_{Res} of I_{HR} image. As shown in Fig. 4, the JRCNN mainly consists of four parts: first convolutional layer (Conv1) for low-level feature extraction, three dense residual blocks for high level feature extraction, two trans-

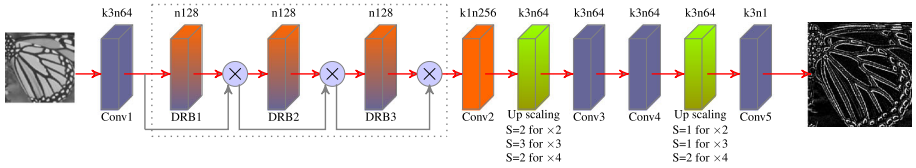

Fig. 4 Architecture of the JRCNN

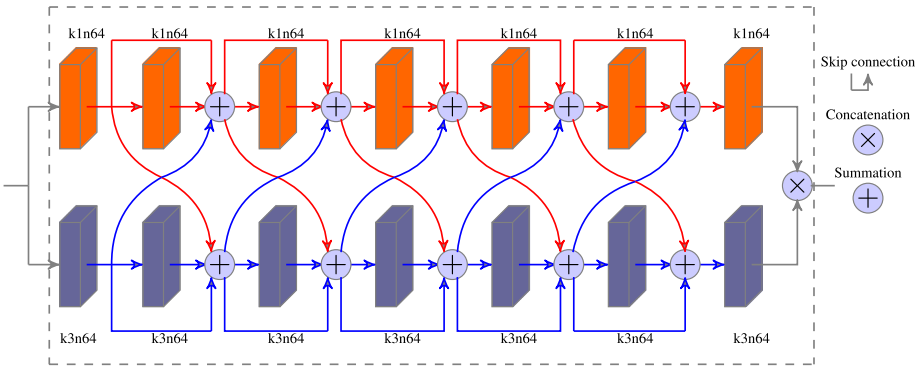

Fig. 5 Structure of dense residual block

posed convolutional layers for up scaling operation and the final convolutional layer for reconstruction.

3.1 Low-level feature representations

The input to the Conv1 layer is a single channel I_{LQ} image. The Conv1 layer extracts the low-level features like edges and blobs from the I_{LQ} image. The Conv1 layer has 64 filters each of size 3×3. The output of Conv1 layer is 64 low-level feature representations (F_O) of I_{LQ} image. The learned F_O is represented as follows:

$$F_O = H_{\mathrm{LF}}(I_{\mathrm{LQ}}) \tag{2}$$

where $H_{\mathrm{LF}}(.)$ is the convolution operation. F_O is then used for high-level feature extraction with DRB.

3.2 Dense residual block

Dense residual block (DRB) is shown in Fig. 5. The DRB takes F_O as input and produces high-level feature representations (F_H). The F_H can be formulated as follows:

$$F_H = H_{\mathrm{DRB},3}(H_{\mathrm{DRB},2}(H_{\mathrm{DRB},1}(F_O))) \tag{3}$$

where $H_{\mathrm{DRB}}(.)$ is the proposed DRB structure which contains parallel convolutional layers (PCL). The PCL contains two categories of convolutional layers. The first category contains seven convolutional layers each consisting of 64 filters of size 1×1. Similarly, the second category contains seven convolutional layers each consisting of 64 filters of size 3×3. There are residual connections between each convolutional layers in the two categories of PCL. Furthermore, cross residual connections (CRC) are provided between two categories to share feature representations from different convolutional layers. This type of structure eases the training and can help to increase the depth of DNN. A total of three DRBs are used in the JRCNN. Moreover, skip connections are used between each DRB to improve the representational power of the JRCNN by learning diverse sets of features. The feature representations learned from the previous layer are skipped to the current layer and then transferred to the next layer. All the feature representations are concatenated channel-wise. This strategy mitigates the feature redundancy by avoiding feature relearning.

In each of the DRBs, the feature representations from the previous convolutional layers are added element-wise in PCL to ease the training. All the feature representations from each of the DRBs are concatenated channel-wise to extract hierarchical features. All these features are extracted in the LR space. The goal of SR is to recover more useful information from the available abundant information in the LR inputs and the features. The skip connections make short paths for information and gradient flow thereby avoiding vanishing of gradients. Furthermore, feature representations from different layers are concatenated channel-wise to learn diverse set of features and improves the representational power of the model. In addition, feature redundancy is also avoided using skip connections.

The features learned from the DRB are then feed into bottleneck layer (1×1 convolutional layer) introduced after the final DRB. The bottleneck layer reduces the number of feature representations that are extracted from the previous layers and reduces the burden on the transposed convolutional layer. The features extracted from bottleneck layer are formulated as follows:

$$F_{\mathrm{BNeck}} = H_{\mathrm{BNeck}}(F_H) \tag{4}$$

where $H_{\text{BNeck}}(.)$ denotes the convolution operation in the bottleneck layer. These features are then upscaled using a upscale module.

3.3 Upscaling

The transposed convolutional layer upsamples the features extracted from the bottleneck layer to increase the spatial resolution of features. In the JRCNN, two transposed convolutional layers are used. The upscaled features extracted from the first transposed convolutional layer are formulated as follows:

$$F_{UP_1} = H_{UP_1}(F_{\text{BNeck}}) \tag{5}$$

where $H_{UP_1}(.)$ and F_{UP_1} denote the upscale module and the upscaled features, respectively. Two additional convolutional layers are used after the first transposed convolutional layer to enhance the upscaled features. Different from the other networks [38,39], the JRCNN has two transposed convolutional layers. Different upscaling factors are produced using two transposed convolutional layers as represented in Fig. 4. The stride of the first transposed convolutional layer is 2, 3 and 2 and the second transposed convolutional layer is 1, 1 and 2 for $\times 2$, $\times 3$ and $\times 4$, respectively. The output of feature enhancement layers Conv3 and Conv4 are formulated as follows:

$$F_{\text{Enhance}} = H_{\text{Conv4}}(H_{\text{Conv3}}(F_{UP_1})) \tag{6}$$

where $H_{\text{Conv3}}(.)$ and $H_{\text{Conv4}}(.)$ represent the convolution of features from enhancement layers Conv3 and Conv4, respectively. F_{Enhance} is the enhanced feature representation of the enhancement layers. The first and second transposed convolutional layers have 64 filters each of size 3×3. Similarly, the two convolutional layers used for feature enhancement also have 64 filters each of size 3×3. To increase the spatial resolution further, one more transposed convolutional layer is introduced after the enhancement layers. The feature representations of second transposed convolutional layer are formulated as follows:

$$F_{UP_2} = H_{UP_2}(F_{\text{Enhance}}) \tag{7}$$

where $H_{UP_2}(.)$ and F_{UP_2} denote the second upscale module and the upscaled features using second transposed convolutional layer, respectively. These features are then fed to the final reconstruction layer to extract the final reconstructed I_{Res} image.

3.4 Reconstruction

The reconstruction layer is the final convolutional layer which has a single filter of size 3×3 to reconstruct the required output. The reconstruction layer produces the residual of the required SR image. The reconstructed I_{Res} image is formulated as follows:

$$I_{\text{Res}} = H_{Res}(F_{UP_2}) \tag{8}$$

where $H_{\text{Res}}(.)$ and I_{Res} represent the reconstruction layer and the reconstructed image. The loss of high frequency information while interpolating an image results in a blurred image. The bicubic interpolation introduces blurring artifacts. Subtracting I_{Bic} from I_{HR} produces I_{Res}. So, I_{Res} image contains only the high frequency information like edges, textures and complex patterns. The final SR image is obtained by adding I_{Res} to the bicubic interpolated I_{Bic} image of I_{LQ} image. The I_{SR} image is formulated as follows:

$$I_{\text{SR}} = I_{\text{Res}} + I_{\text{Bic}} \tag{9}$$

3.5 Training

An end-to-end mapping function is learned between the I_{LQ} image and the I_{Res} image by minimizing the mean squared error (MSE) using the JRCNN. Let $\{{I_{\text{LQ}}}^i, {I_{\text{Res}}}^i\}$ represent the i^{th} training pair of N training samples and Θ represent the network weight parameters of K^{th} layer. The MSE is formulated as follows:

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left\| H_{\text{JRCNN}}(\boldsymbol{I_{\text{LQ}}}^{\boldsymbol{i}}, \Theta) - \boldsymbol{I_{\text{Res}}^{i}}\right\|^2 + \lambda \left\|\Theta\right\|^2 \tag{10}$$

where λ represents the weight decay (regularization) factor used to avoid under-or-over fitting. The MSE in Eq. 10 is minimized using Adam optimizer with 32 samples in each minibatch. Standard gradient descent method is used on each minibatch to optimize the error. The parameters β_1 and β_2 of Adam optimizer are set to 0.9 and 0.999, respectively.

4 Experimental analysis

In this section, a complete analysis of extensive experimental results on different datasets are provided quantitatively and qualitatively. Furthermore, the experimental settings and the structural comparison of the JRCNN with other methods are also provided.

4.1 Experimental settings

Number of filters and filter size are key factors which decide the performance of the JRCNN. The JRCNN has a total of 47 convolutional layers and 2 transposed convolutional layers all with 64 filters of size 3×3. The convolutional layer which serves as a bottleneck layer that has filters of size 1×1 only. The residual connections and the skip connections also improve the performance. Moreover in the JRCNN structure, cross residual connections are provided between the parallel convolutional layers (PCL) for efficient flow of feature representations.

Training is performed by optimizing MSE using Adam optimizer on minibatches of size 32. l_2-norm regularization with weight decay factor (λ) of 10^{-5} is used. The JRCNN is trained for 100 epochs with initial learning rate set to 10^{-4}. After 50 epochs, the learning rate is reduced by 10%. Gradient clipping is not used as initial learning rate is very small. Patches of size 41×41 are extracted from DIV2K dataset [31] images. A total of 16000 training pairs are extracted from 800 images of DIV2K training dataset. NVIDIA Tesla K40c GPU is used for training. For training and testing, MATLAB 2019b framework is used.

4.2 Degradation model

Bicubic degradation is used for SR. JPEG compression is used for CAR. In addition to bicubic degradation, JPEG compression is also used for CISR. The synthetic degraded images are formed by reducing the spatial resolution of image using bicubic interpolation and compressing the output image with JPEG coder. The color image is first transformed into YCbCr space. Only the luminance component (Y) is used for training. All the degradations are present in the Y channel only as Cb and Cr represent the chrome components. The following are the experiments performed on the synthesized image:

- The SR with three different up scaling factors 2, 3 and 4.
- The CAR with four different JPEG quality factors QF = {10, 20, 30 and 40}.
- The CISR with three different JPEG quality factors QF = {10, 20 and 30} and an up scaling factor of 2.

4.3 Datasets

For comparison with state-of-the-art methods, different benchmark datasets have been used for different applications.

4.3.1 SR datasets

For training, 800 images from DIV2K [31] images are considered as all the methods [20,37,39] used the same dataset. For validation 100 images from validation set of DIV2K dataset are used. For testing, the benchmark datasets are: Set5 [3], Set14 [34], B100 [24], Urban100 [12] and Manga109 [25].

Training is performed with 400 images by combining 200 training and 200 testing images from BSDS500 [24] dataset. Validation is performed with 100 images from the validation set of BSDS500 dataset. For testing benchmark datasets are: classic5 [7], LIVE1 [7] and B100 [24]

4.3.2 CAR datasets

Training is performed with 400 images by combining 200 training and 200 testing images from BSDS500 ([24]) dataset. Validation is performed with 100 images from the validation set of BSDS500 dataset. For testing benchmark datasets are: classic5 ([7]), LIVE1 ([7]) and B100 ([24])

4.3.3 CISR datasets

For CISR training and validation are performed on same datasets used for CAR. The testing is performed on Set10 [5] images considered in the CISRDCNN. Furthermore, three more datasets Set5 [3], Set14 [34] and B100 [24] are also used to assess the performance of the JRCNN.

4.4 Compared methods and metrics

The SR performance of the JRCNN is compared with SRCNN [8], FSRCNN [9], VDSR [16], DnCNN [36], DenseDbSR [2], DRRN [28], LapSRN [18], MemNet [29], BTSRN [10], SRMD [37], MWCNN [21], CARN [1], EDSR [20], CRN [19], ERN [19], RDN [39], RCAN [38], PG-GAN [22] and EBRN [26]. The CAR performance of the JRCNN is compared with SA-DCT [11], ARCNN [7], CONCOLOR [35], TNRD [6], CAS-CNN [4], SSRQC [40], EPCA [27], RED30 [23], DnCNN [36], IACNN [17] and DenseDbSR [2]. Similarly, the baseline methods used for CISR performance comparison of the JRCNN are LJSRDB [13], A+ [30], SRCDFOE [33], CONCOLOR-VDSR [16,35], ARCNN-TNRD [6,7], ARCNN-VDSR [7,16], ARCNN-DnCNN [7,36], FSRCNN [9], VDSR [16], DenseDbSR [2] and CISRDCNN [5]. The CONCOLOR-VDSR [16,35], ARCNN-TNRD [6,7], ARCNN-VDSR [6,7] and ARCNN-DnCNN [7,36] are cascading methods which consist of state-of-the-art CAR and SR methods.

In this paper, the widely used Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index [41] are used for objective performance comparison with the other state-of-the-art methods.

Table 1 The average of performance metrics on different datasets for SR. The metrics in bold indicates the best values

Method	Scale	Set5 [3]	Set14 [34]	B100 [24]	Urban100 [12]	Manga109 [25]
Bicubic [15]	×2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
SRCNN [8]		36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
FSRCNN [9]		37.05/0.9560	32.66/0.9090	31.53/0.8920	29.88/0.9020	36.67/0.9710
VDSR [16]		37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140	37.22/0.9750
DnCNN [36]		37.58/0.9590	33.03/0.9128	31.90/0.8961	30.74/0.9139	–/–
DenseDbSR [2]		37.65/0.9605	33.09/0.9130	31.97/0.8965	30.79/0.9145	–/–
DRRN [28]		37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	–/–
LapSRN [18]		37.52/0.9591	33.08/0.9130	31.08/0.8950	30.41/0.9101	37.27/0.9740
MemNet [29]		37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
BTSRN [10]		37.75/0.9243	33.20/0.8322	32.05/0.8005	31.63/0.8389	–/–
SRMD [37]		37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9761
MWCNN [21]		37.91/0.9600	33.70/0.9182	32.23/0.8999	32.30/0.9296	–/–
CARN [1]		37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	–/–
EDSR [20]		38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773
CRN [19]		38.17/0.9610	33.84/0.9203	32.30/0.9012	32.69/0.9334	–/–
ERN [19]		38.18/0.9610	33.88/0.9195	32.30/0.9011	32.66/0.9332	–/–
RDN [39]		38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	39.18/0.9780
RCAN [38]		38.27/0.9614	34.12/0.9216	32.41/0.9027	33.34/0.9384	39.44/0.9786
PG-GAN [22]		38.33/0.9630	34.05/0.9190	32.44/0.9010	33.12/0.9380	39.33/0.9760
EBRN [26]		38.35/0.9620	34.24/0.9226	**32.47/0.9033**	33.52/0.9402	39.62/0.9802
JRCNN		**38.36/0.9623**	**34.29/0.9226**	32.43/0.9031	**33.56/0.9405**	**39.66/0.9805**
Bicubic [15]	×3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN [8]		32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
FSRCNN [9]		33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
VDSR [16]		33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290	32.01/0.9340
DnCNN [36]		33.75/0.9222	29.81/0.8321	28.85/0.7981	27.15/0.8276	32.03/0.9342
DenseDbSR [2]		33.80/0.9226	29.86/0.8323	28.89/0.7984	27.22/0.8286	–/–
DRRN [28]		34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.74/0.9390
LapSRN [18]		33.82/0.9227	29.87/0.8320	28.82/0.7980	27.07/0.8280	32.21/0.9350
MemNet [29]		34.09/0.9248	30.01/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
BTSRN [10]		34.03/0.9243	29.90/0.8322	28.97/0.8005	27.75/0.8389	–/–
SRMD [37]		34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
MWCNN [21]		34.17/0.927	30.16/0.841	29.12/0.8060	28.13/0.851	–/–
CARN [1]		34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	–/–
EDSR [20]		34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
CRN [19]		34.60/0.9286	30.48/0.8455	29.20/0.8081	28.62/0.8620	–/–
ERN [19]		34.62/0.9285	30.51/0.8450	29.21/0.8080	28.61/0.8614	–/–
RDN [39]		34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484
RCAN [38]		34.74/0.9299	30.64/0.8481	29.32/0.8111	29.08/0.8702	34.43/0.9498
JRCNN		**34.67/0.9293**	**30.61/0.8476**	**29.35/0.8110**	**28.93/0.8671**	**34.30/0.9491**
Bicubic [15]	×4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [8]		30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCNN [9]		30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [16]		31.35/0.8830	28.02/0.7680	27.29/0.7238	25.18/0.7540	28.83/0.8870
DnCNN [36]		31.40/0.8845	28.04/0.7672	27.29/0.7253	25.20/0.7521	28.87/0.8873
DenseDbSR [2]		31.48/0.8876	28.12/0.7679	27.34/0.7258	25.24/0.7531	–/–
DRRN [28]		31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	23.60/0.7420

Table 1 continued

Method	Scale	Set5 [3]	Set14 [34]	B100 [24]	Urban100 [12]	Manga109 [25]
LapSRN [18]		31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560	29.09/0.8900
MemNet [29]		31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
BTSRN [10]		31.85/0.8898	28.20/0.7721	27.47/0.7286	25.74/0.7738	–/–
SRMD [37]		31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
MWCNN [21]		32.12/0.894	28.41/0.7816	27.62/0.735	28.13/0.8514	–/–
CARN [1]		32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	–/–
EDSR [20]		32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
CRN [19]		32.34/0.8971	28.74/0.7855	27.66/0.7395	26.44/0.7967	–/–
ERN [19]		32.39/0.8975	28.75/0.7853	27.70/0.7398	26.43/0.7966	–/–
RDN [39]		32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151
RCAN [38]		32.62/0.9001	28.86/0.7888	27.76/0.7435	26.82/0.8087	31.21/0.9172
PG-GAN [22]		32.63/0.8980	28.90/0.7900	27.77/0.7410	26.82/0.8050	31.44/0.9150
EBRN [26]		32.79/0.9032	**29.01/0.7903**	27.85/0.7464	27.03/0.8114	31.53/0.9198
JRCNN		**32.84/0.9033**	28.92/0.7888	**27.88/0.7476**	**27.09/0.8131**	**31.58/0.9203**

Table 2 The average of quantitative metrics on different datasets for compression artifacts reduction

QF	Method	classic5 [7] PSNR/PSNR-B/SSIM	LIVE1 [7] PSNR/PSNR-B/SSIM	BSDS100 [24] PSNR/PSNR-B/SSIM
10	JPEG [32]	27.82/25.21/0.7800	27.77/25.33/0.7905	27.58/24.97/0.7694
	SA-DCT [11]	28.88/28.16/0.8071	28.65/28.01/0.8093	–
	ARCNN [7]	29.03/28.75/0.8111	28.96/28.69/0.8220	28.73/28.40/0.7967
	CONCOLOR [35]	29.20/29.14/0.8140	28.77/–/0.7991	28.49/28.43/0.7864
	TNRD [6]	29.28/29.00/0.8174	29.15/28.80/0.8255	28.80/28.47/0.7990
	CAS-CNN [4]	–	29.36/28.92/0.8300	–
	SSRQC [40]	29.08/29.03/0.8112	–	28.37/28.31/0.7824
	EPCA [27]	29.37/–/0.8019	28.94/–/0.8071	–
	RED30 [23]	29.35/–/0.8041	29.28/–/0.8177	–
	DnCNN [36]	29.40/29.13/0.8201	29.19/28.90/0.8262	28.84/28.44/0.8007
	IACNN [17]	29.43/–/0.8070	29.34/–/0.8199	28.87/–/0.7816
	DenseDbSR [2]	29.52/29.19/0.8231	29.30/28.95/0.8297	28.92/28.51/0.8032
	JRCNN	**30.01/29.27/0.8289**	**29.41/29.10/0.8335**	**29.18/28.67/0.8137**
20	JPEG [32]	30.12/27.50/0.8541	30.07/27.57/0.8683	29.73/26.97/0.8519
	SA-DCT [11]	30.92/29.75/0.8663	30.81/29.82/0.8781	–
	ARCNN [7]	31.15/30.60/0.8694	31.29/30.69/0.8875	30.81/30.16/0.8682
	CONCOLOR [35]	31.34/31.24/0.8709	31.00/–/0.8669	30.54/30.24/0.8598
	TNRD [6]	31.47/31.06/0.8749	31.46/31.02/0.8910	30.94/30.26/0.8709
	CAS-CNN [4]	–	31.67/30.84/0.8940	–
	SSRQC [40]	31.23/31.13/0.8696	–	30.43/30.23/0.8578
	EPCA [27]	31.58/–/0.8583	32.45/–/0.9008	–
	RED30 [23]	31.61/–/0.8611	31.65/–/0.8833	–
	DnCNN [36]	31.63/30.67/0.8775	31.59/31.07/0.8936	31.05/30.29/0.8741
	IACNN [17]	31.64/–/0.8628	31.73/–/0.8848	31.08/–/0.8621
	DenseDbSR [2]	31.71/31.20/0.8793	31.64/31.08/0.8953	31.09/30.31/0.8756
	JRCNN	**31.83/31.39/0.8809**	**31.76/31.14/0.9003**	**31.21/30.47/0.8839**

Table 2 continued

QF	Method	classic5 [7] PSNR/PSNR-B/SSIM	LIVE1 [7] PSNR/PSNR-B/SSIM	BSDS100 [24] PSNR/PSNR-B/SSIM
30	JPEG [32]	31.48/28.94/0.8844	31.41/28.92/0.9000	30.98/28.23/0.8865
	SA-DCT [11]	32.14/30.83/0.8914	32.08/30.92/0.9078	–
	ARCNN [7]	32.51/31.99/0.8967	32.69/32.15/0.9166	32.13/31.39/0.9011
	CONCOLOR [35]	32.66/32.05/0.8962	32.35/–/0.8977	31.81/31.37/0.8930
	TNRD [6]	32.78/32.25/0.8994	32.84/32.29/0.9181	32.24/31.30/0.9020
	SSRQC [40]	32.50/32.36/0.8951	–	31.68/31.35/0.8918
	EPCA [27]	32.87/–/0.8834	32.45/–/0.9008	–
	DnCNN [36]	32.91/32.33/0.9011	32.98/32.31/0.9204	32.36/31.40/0.9049
	IACNN [17]	32.93/–/0.8874	33.19/–/0.9132	–
	DenseDbSR [2]	32.97/32.36/0.9025	33.03/32.33/0.9214	32.40/31.41/0.9059
	JRCNN	**33.14/32.49/0.9082**	**33.17/32.48/0.9245**	**32.57/31.56/0.9075**
40	JPEG [32]	32.43/29.92/0.9011	32.35/29.96/0.9173	31.88/29.13/0.9057
	SA-DCT [11]	33.00/31.59/0.9055	32.99/31.79/0.9240	–
	ARCNN [7]	33.34/32.80/0.9101	33.63/33.12/0.9306	32.99/32.14/0.9176
	CONCOLOR [35]	33.56/33.15/0.9102	33.33/–/0.9153	32.71/32.14/0.9116
	CAS-CNN [4]	–	33.98/32.83/0.9350	–
	SSRQC [40]	33.37/33.18/0.9090	–	32.55/32.19/0.9107
	EPCA [27]	33.58/–/0.8962	33.40/–/0.9170	–
	DnCNN [36]	33.77/33.19/0.9141	33.96/33.25/0.9346	33.27/32.21/0.9215
	IACNN [17]	33.79/–/0.9014	34.18/–/0.9283	–
	DenseDbSR [2]	33.80/33.20/0.9150	33.98/33.27/0.9351	33.30/32.22/0.9221
	JRCNN	**33.92/33.32/0.9171**	**34.09/33.38/0.9383**	**33.48/32.45/0.9289**

The metrics in bold indicates the best values

4.5 Objective evaluation on synthetic images

4.5.1 Quantitative analysis for SR

Table 1 represents the comparison of SR objective metrics on different benchmark datasets with the state-of-the-art methods. B100 [24] dataset consists of natural scenes. The JRCNN produces better performance on B100 [24] dataset. Urban100 dataset [12] contains 100 urban images with complex patterns and restoring these patterns while up scaling the images is a very challenging task. The proposed JRCNN outperforms the existing state-of-the-art methods on Urban100 dataset. The performance of the JRCNN is competitive with other recent state-of-the-art methods. For all the up scaling factors, i.e., $\times 2$, $\times 3$ and $\times 4$, the JRCNN outperforms all the state-of-the-art methods.

The proposed JRCNN achieves 1.7dB improvement in the PSNR on Set5 dataset for $\times 2$ when compared to SRCNN [8] method. The proposed JRCNN outperforms the recent state-of-the-art methods RDN [39], RCAN [38] and produces the competitive performance when compared to PG-GAN [22] and EBRN [26].

4.5.2 Quantitative analysis for CAR

Table 2 represents the comparison of CAR objective metrics on different benchmark datasets with the state-of-the-art methods. Overall, the DenseDbSR produces second best performance. The ARCNNN [7], CONCOLOR [35] and TNRD [6] are superior to SA-DCT [11] but the gains are limited to some extent. The DnCNN [36] and IACNN [17] are competitive to each other. The proposed JRCNN achieves the best performance for all the quality factors. A PSNR gain of 0.98dB is achieved with the JRCNN on classic5 dataset for QF = 10 when compared to the ARCNN [7].

4.5.3 Quantitative analysis for CISR

Table 3 represents the PSNR(dB) comparison of CISR objective metrics on Set10 dataset with the state-of-the-art methods. Overall, the CISRDCNN [5] generates the second-best results. The FSRCNN [9], ARCNN-VDSR [7,16] and ARCNN-DnCNN [7,36] achieve similar performance, and all of them are slightly inferior to the VDSR [16]. The A+ [30], CONCOLOR-VDSR [16,35] and SRCDFOE [33] are superior to bicubic, but the gains are limited to some extent.

Table 3 The average of PSNR in dB on Set10 dataset

Test images	Butterfly	House	Parrot	Woman	Circuit	Leaves	Foreman	Zebra	Peppers	Ppt3	Average
						QF = 10					
Bicubic [15]	22.69	27.39	26.23	26.28	23.37	21.46	27.76	25.65	28.49	23.26	25.26
LJSRDB [13]	22.67	27.59	26.33	26.34	23.39	21.45	28.13	25.77	28.80	23.20	25.37
A + [30]	23.18	27.92	26.64	27.05	23.83	21.91	28.80	26.38	29.33	23.16	25.82
SRCDFOE [33]	23.17	28.09	26.65	26.88	23.90	21.84	28.37	26.27	29.35	23.72	25.83
CONCOLOR-VDSR [16,35]	23.45	28.29	26.92	27.32	24.40	22.37	29.56	26.80	29.80	24.20	26.31
ARCNN-TNRD [6,7]	24.01	28.11	26.98	27.41	24.49	22.98	29.47	26.57	29.78	24.47	26.43
ARCNN-VDSR [7,16]	24.12	28.39	27.15	27.53	24.59	23.22	29.60	26.67	29.85	24.79	26.59
ARCNN-DnCNN [7,36]	24.21	28.43	27.31	27.49	24.65	23.45	29.49	26.54	29.81	24.82	26.62
FSRCNN [9]	23.91	28.68	27.21	27.52	24.64	22.97	29.30	26.78	29.81	24.58	26.54
VDSR [16]	24.19	29.44	26.50	27.87	24.44	22.68	29.74	26.89	30.14	24.75	26.66
DenseDbSR [2]	24.31	29.47	26.71	27.92	24.98	23.56	30.05	27.24	30.03	25.25	26.95
CISRDCNN [5]	24.53	30.03	27.55	28.18	25.13	23.76	30.27	27.34	30.38	25.19	27.24
JRCNN	**24.69**	**30.15**	**27.69**	**28.38**	**25.29**	**23.92**	**30.46**	**27.55**	**30.69**	**25.64**	**27.45**
						QF = 20					
Bicubic [15]	23.89	29.12	27.64	27.92	25.00	23.07	29.74	27.24	30.45	24.51	26.86
LJSRDB [13]	23.86	28.99	27.62	27.83	24.69	22.78	29.92	27.15	30.43	24.46	26.77
A + [30]	24.57	29.89	27.97	28.80	25.71	23.89	31.07	28.19	31.41	24.89	27.64
SRCDFOE [33]	24.42	29.81	27.98	28.51	25.52	23.59	30.34	27.82	31.14	25.19	27.43
CONCOLOR-VDSR [16,35]	24.63	29.34	28.09	28.80	25.78	23.97	31.68	28.11	31.42	25.94	27.78
ARCNN-TNRD [6,7]	25.21	29.48	28.41	28.75	26.03	24.58	31.56	28.34	31.54	26.25	28.02
ARCNN-VDSR [7,16]	25.43	29.54	28.53	28.99	26.24	24.95	31.65	28.24	31.65	26.47	28.17
ARCNN-DnCNN [7,36]	25.56	30.21	28.69	29.11	26.51	24.89	31.51	28.36	31.78	26.54	28.32
FSRCNN [9]	25.37	30.53	28.59	29.22	26.43	24.91	31.64	28.48	31.71	26.20	28.31
VDSR [16]	25.49	31.29	28.16	29.45	26.17	24.84	31.87	28.37	31.97	26.39	28.40
DenseDbSR [2]	25.87	31.45	28.78	29.89	26.87	25.99	32.14	28.67	32.06	27.06	28.88
CISRDCNN [5]	25.93	31.65	28.98	30.03	26.95	26.10	32.25	28.95	32.19	27.18	29.02

Table 3 continued

Test images	Butterfly	House	Parrot	Woman	Circuit	Leaves	Foreman	Zebra	Peppers	Ppt3	Average
JRCNN	**26.03**	**31.85**	**29.05**	**30.67**	**27.08**	**26.35**	**32.37**	**29.12**	**32.26**	**27.38**	**29.22**
						QF = 30					
Bicubic [15]	24.51	29.66	28.51	28.81	25.74	23.96	30.72	28.13	31.36	25.13	27.65
LJSRDB [13]	24.53	29.32	28.55	28.75	24.85	23.29	30.62	27.99	31.24	24.84	27.40
A + [30]	25.29	30.43	29.07	29.74	26.57	24.98	32.19	29.16	32.34	25.84	28.56
SRCDFOE [33]	25.11	30.35	28.93	29.41	26.25	24.60	31.28	28.79	31.98	26.05	28.28
CONCOLOR-VDSR [16,35]	25.32	29.99	29.09	29.63	26.46	24.97	32.41	28.98	32.25	27.09	28.62
ARCNN-TNRD [6,7]	26.14	29.87	29.44	30.02	26.89	25.69	32.47	29.08	32.54	27.45	28.96
ARCNN-VDSR [7,16]	26.26	29.82	29.52	30.22	27.05	26.08	32.59	29.38	32.66	27.64	29.12
ARCNN-DnCNN [7,36]	26.34	30.89	29.47	30.15	27.14	26.17	32.47	29.44	32.74	27.34	29.22
FSRCNN [9]	26.09	31.10	29.55	30.20	27.31	26.15	32.61	29.33	32.59	27.14	29.21
VDSR [16]	26.30	31.88	29.35	30.50	27.29	26.15	32.99	29.47	32.86	27.75	29.45
DenseDbSR [2]	26.47	32.14	29.75	30.79	27.94	27.21	33.15	29.86	33.11	28.42	29.88
CISRDCNN [5]	26.65	32.21	29.89	30.92	27.85	27.31	33.26	29.92	33.07	28.48	29.96
JRCNN	**26.85**	**32.41**	**29.97**	**31.13**	**28.02**	**27.82**	**33.58**	**30.18**	**33.27**	**28.98**	**30.22**

Table 4 The average of SSIM on Set10 dataset

Test images	Butterfly	House	Parrot	Woman	Circuit	Leaves	Foreman	Zebra	Peppers	Ppt3	Average
						QF = 10					
Bicubic [15]	0.6790	0.7499	0.7742	0.7684	0.6956	0.6680	0.7771	0.7996	0.7550	0.7876	0.7454
LJSRDB [13]	0.6888	0.7706	0.7921	0.7827	0.7029	0.6728	0.8013	0.8204	0.7781	0.8039	0.7614
A + [30]	0.7180	0.7870	0.8054	0.8098	0.7317	0.7172	0.8224	0.8342	0.7914	0.8231	0.7840
SRCDFOE [33]	0.7213	0.7936	0.8102	0.8113	0.7404	0.7164	0.8191	0.8369	0.7971	0.8451	0.7891
CONCOLOR-VDSR [16,35]	0.7422	0.8057	0.8255	0.8317	0.7705	0.7549	0.8486	0.8556	0.8118	0.8697	0.8116
ARCNN-TNRD [6,7]	0.7478	0.8047	0.8264	0.8305	0.7697	0.8011	0.8479	0.8546	0.8096	0.8709	0.8613
ARCNN-VDSR [7,16]	0.7494	0.8050	0.8278	0.8317	0.7678	0.8023	0.8477	0.8466	0.8089	0.8720	0.8159
ARCNN-DnCNN [7,36]	0.7486	0.8074	0.8264	0.8306	0.7689	0.8015	0.8467	0.8496	0.8097	0.8714	0.8161
FSRCNN [9]	0.7437	0.8084	0.8250	0.8319	0.7694	0.7953	0.8450	0.8513	0.8081	0.8721	0.8150
VDSR [16]	0.7575	0.8232	0.8279	0.8406	0.7741	0.7976	0.8551	0.8562	0.8150	0.8831	0.8230
DenseDbSR [2]	0.7687	0.8297	0.8386	0.8423	0.7895	0.8274	0.8612	0.8621	0.8208	0.8912	0.8332
CISRDCNN [5]	0.7699	0.8310	0.8410	0.8494	0.7937	0.8326	0.8640	0.8655	0.8222	0.8958	0.8365
JRCNN	**0.7727**	**0.8322**	**0.8476**	**0.8408**	**0.7996**	**0.8337**	**0.8689**	**0.8670**	**0.8288**	**0.9063**	**0.8398**
						QF = 20					
Bicubic [15]	0.7369	0.7999	0.8211	0.8178	0.7596	0.7404	0.8273	0.8479	0.8029	0.8299	0.7984
LJSRDB [13]	0.7444	0.8164	0.8321	0.8265	0.7638	0.7409	0.8446	0.8562	0.8141	0.8433	0.8082
A + [30]	0.7704	0.8263	0.8414	0.8500	0.7996	0.8059	0.8652	0.8718	0.8283	0.8748	0.8334
SRCDFOE [33]	0.7694	0.8317	0.8427	0.8488	0.7980	0.7939	0.8582	0.8701	0.8280	0.8820	0.8323
CONCOLOR-VDSR [16,35]	0.7835	0.8301	0.8535	0.8634	0.8159	0.8166	0.8863	0.8815	0.8369	0.9036	0.8471
ARCNN-TNRD [6,7]	0.7956	0.8324	0.8498	0.8607	0.8167	0.8489	0.8794	0.8805	0.8298	0.9028	0.8500
ARCNN-VDSR [7,16]	0.7902	0.8370	0.8548	0.8616	0.8186	0.8550	0.8795	0.8762	0.8365	0.9046	0.8514
ARCNN-DnCNN [7,36]	0.7921	0.8407	0.8536	0.8647	0.8208	0.8560	0.8804	0.8807	0.8378	0.9048	0.8532
FSRCNN [9]	0.7916	0.8410	0.8547	0.8666	0.8234	0.8570	0.8818	0.8819	0.8380	0.9056	0.8542
VDSR [16]	0.8014	0.8478	0.8576	0.8735	0.8288	0.8676	0.8868	0.8856	0.8431	0.9202	0.8612
DenseDbSR [2]	0.8109	0.8517	0.8654	0.8797	0.8412	0.8937	0.8896	0.8917	0.8498	0.9337	0.8707
CISRDCNN [5]	0.8135	0.8533	0.8678	0.8822	0.8429	0.8949	0.8937	0.8938	0.8476	0.9356	0.8725

Table 4 continued

Test images	Butterfly	House	Parrot	Woman	Circuit	Leaves	Foreman	Zebra	Peppers	Ppt3	Average
JRCNN	**0.8198**	**0.8601**	**0.8712**	**0.8895**	**0.8489**	**0.9004**	**0.9012**	**0.9004**	**0.8499**	**0.9431**	**0.8785**
						QF = 30					
Bicubic [15]	0.7656	0.8200	0.8440	0.8440	0.7889	0.7792	0.8527	0.8698	0.8225	0.8539	0.8241
LJSRDB [13]	0.7720	0.8269	0.8543	0.8538	0.7842	0.7745	0.8664	0.8738	0.8304	0.8673	0.8304
A + [30]	0.7966	0.8416	0.8638	0.8715	0.8263	0.8441	0.8847	0.8896	0.8431	0.8973	0.8559
SRCDFOE [33]	0.7946	0.8428	0.8625	0.8688	0.8216	0.8318	0.8756	0.8879	0.8411	0.9034	0.8530
CONCOLOR-VDSR [16,35]	0.8047	0.8507	0.8734	0.8823	0.8359	0.8505	0.8979	0.8960	0.8490	0.9257	0.8666
ARCNN-TNRD [6,7]	0.8150	0.8489	0.8741	0.8834	0.8398	0.8798	0.8965	0.8957	0.8498	0.9238	0.8707
ARCNN-VDSR [7,16]	0.8165	0.8509	0.8748	0.8853	0.8412	0.8840	0.8948	0.8961	0.8504	0.9249	0.8719
ARCNN-DnCNN [7,36]	0.8157	0.8498	0.8754	0.8859	0.8457	0.8836	0.8939	0.8958	0.8498	0.9228	0.8718
FSRCNN [9]	0.8135	0.8513	0.8749	0.8846	0.8468	0.8845	0.8953	0.8966	0.8501	0.9232	0.8721
VDSR [16]	0.8243	0.8574	0.8798	0.8929	0.8544	0.8965	0.9021	0.9023	0.8556	0.9404	0.8806
DenseDbSR [2]	0.8318	0.8612	0.8849	0.8978	0.8648	0.9147	0.9078	0.9068	0.8579	0.9498	0.8878
CISRDCNN [5]	0.8327	0.8633	0.8853	0.8999	0.8638	0.9158	0.9081	0.9082	0.8588	0.9515	0.8887
JRCNN	**0.8398**	**0.8689**	**0.8912**	**0.9010**	**0.8697**	**0.9198**	**0.9101**	**0.9091**	**0.8612**	**0.9525**	**0.8923**

Table 5 The average of performance metrics for CISR on different datasets for QF = 10 and ×2

Datasets	Set5 [3]	Set14 [34]	B100 [24]	Urban100 [12]
		PSNR (dB)		
Bicubic [15]	26.60	25.04	24.29	23.00
SRCDFOE [33]	27.24	25.45	24.60	23.34
ARCNN-TNRD [6,7]	27.63	25.61	24.87	23.79
ARCNN-VDSR [7,16]	27.83	25.86	24.90	23.93
ARCNN-DnCNN [7,36]	27.92	25.62	24.97	23.97
FSRCNN [9]	27.74	25.76	24.79	23.68
VDSR [16]	27.81	25.90	24.87	23.93
DenseDbSR [2]	27.96	25.93	24.98	24.21
CISRDCNN [5]	28.15	26.13	25.02	24.37
JRCNN	**28.47**	**26.34**	**25.32**	**24.71**
		SSIM		
Bicubic [15]	0.7239	0.6433	0.5863	0.6224
SRCDFOE [33]	0.7661	0.6663	0.6025	0.6512
ARCNN-TNRD [6,7]	0.7861	0.6789	0.6128	0.6698
ARCNN-VDSR [7,16]	0.7878	0.6803	0.6153	0.6774
ARCNN-DnCNN [7,36]	0.7927	0.6847	0.6162	0.6821
FSRCNN [9]	0.7915	0.6831	0.6157	0.6817
VDSR [16]	0.7931	0.6853	0.6179	0.6859
DenseDbSR [2]	0.7992	0.6887	0.6217	0.6978
CISRDCNN [5]	0.8039	0.6926	0.6238	0.7043
JRCNN	**0.8051**	**0.6976**	**0.6276**	**0.7076**

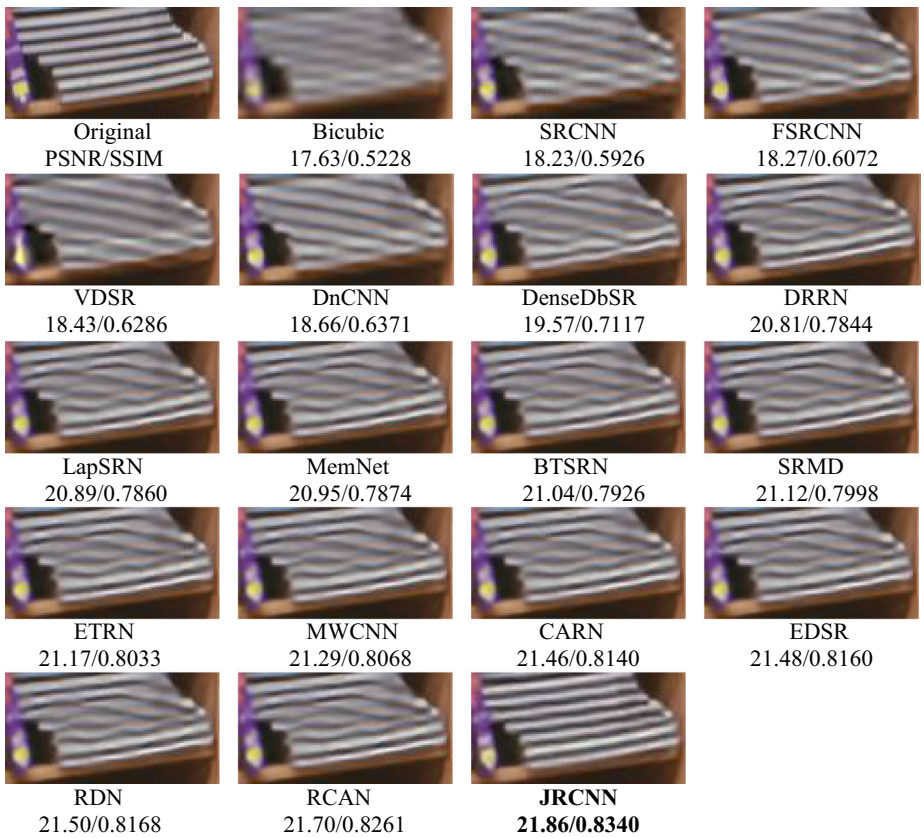

Fig. 6 Visual quality comparison for SR (×4)

Fig. 7 Visual quality comparison for CAR (QF = 10

Table 4 represents the SSIM comparison of CISR objective metrics on Set10 dataset with the state-of-the-art methods. The SSIM measures the structural similarity between the restored image and the I_{HR} image. The SSIM metric of the JRCNN is the highest when compared to the other methods for all the images of Set10 dataset. From Table 4, one can clearly observe that the structural content restoration of the JRCNN is the best.

4.6 CISR performance on benchmark datasets

To better assess the stability and robustness of the JRCNN, the performance metrics for four widely used benchmark datasets, i.e., Set5 [3], Set14 [34], B100 [24] and Urban100 [12] are shown in Table 5. The images in B100 [24] are cropped to generate test images of size 256×256. Similarly, the images in Urban100 [12] dataset are cropped to generate test images of size 512×512. From Table 5, one can see that the JRCNN consistently outperforms all of the compared methods. On an average the JRCNN achieves a PNSR gain of 0.3dB on all the datasets when compared to the next best CISRDCNN [5] method.

4.7 Subjective evaluation on synthetic images

In this section, the visual comparison analysis of images from standard datasets are performed. Different images from the different benchmark datasets for SR, CAR and CISR are considered. Figures 6, 7 and 8 represent the visual comparison for SR, CAR and CISR, respectively.

4.7.1 Qualitative analysis for SR

The visual comparison of restored images from different state-of-the-art methods for SR is shown in Fig. 6. A small patch from the barbara image of the Set14 [34] dataset is cropped. The cropped portion is super resolved by a factor of 4. The cropped portion has regular structures and complex patterns. The regular structures restored from all the state-of-the-art methods are distorted. The reconstructed patch by the JRCNN is visually pleasing and all the regular structures

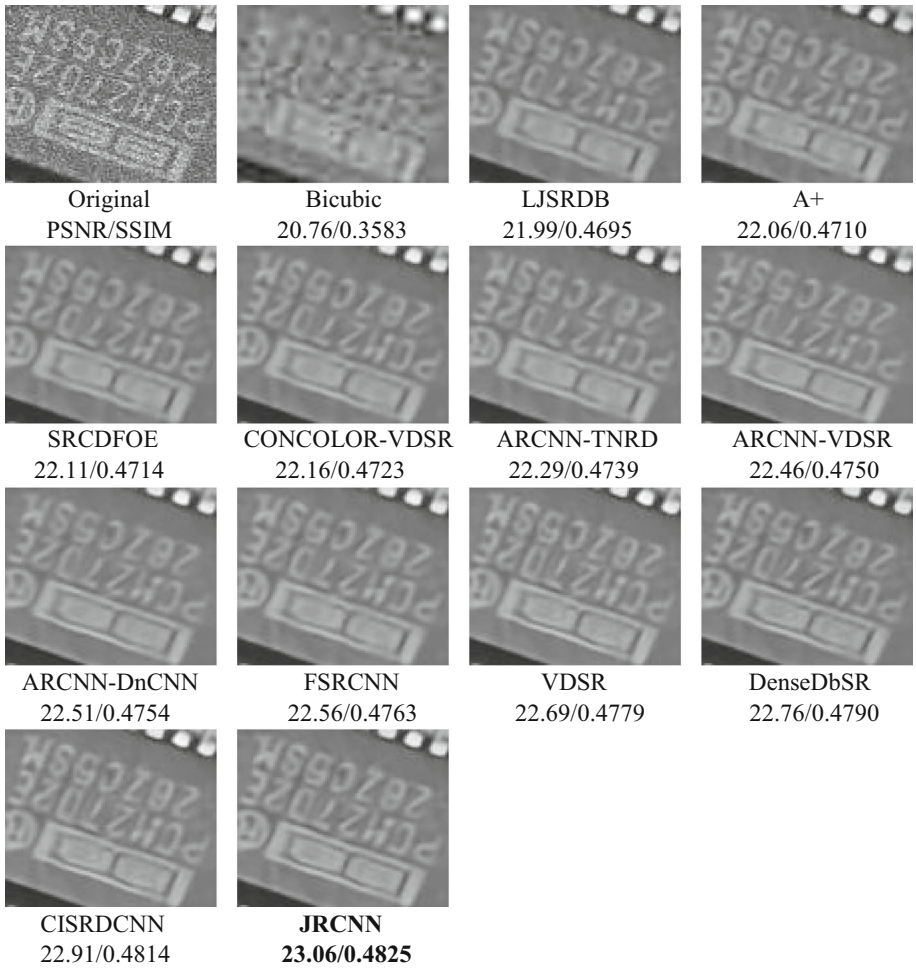

Fig. 8 Visual quality comparison for CISR (×2 and QF = 20)

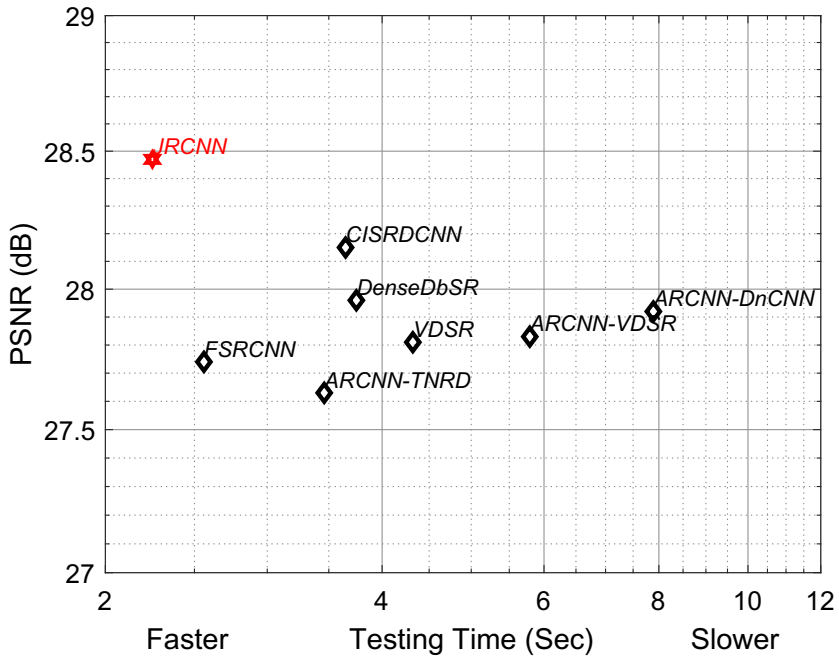

Fig. 9 Test time comparison on Set5 dataset for QF = 10 and ×2

are restored accurately. The PSNR and SSIM values are also shown in Fig. 6. The EDSR [20], RDN [39] and RCAN [38] achieve better PNSR but the visual quality of the restored image is not good as it contains regular structure distortions. Quantitatively and qualitatively the JRCNN produces the best results for SR.

4.7.2 Qualitative analysis for CAR

The deblocking and deringing capability of the JRCNN is visually analyzed by extracting a small portion from the parrots image of LIVE1 [7] dataset. The zoomed version of the extracted patch is shown in Fig. 7. The JPEG decompressed patch contains blocking and ringing artifacts. A Small 8×8 blocks can be observed in the restored patch. Similarly, ringing artifacts can be clearly observed around the edges. The JRCNN seamlessly attenuates the blocking and ringing artifacts which can be clearly observed in Fig. 7. The visual quality of the patches restored by the IACNN [17], DenseDbSR [2] and JRCNN are competitive.

4.7.3 Qualitative analysis for CISR

The visual quality of the restored image patches using different methods corrupted by compression and bicubic degradation is shown in Fig. 7. The visual quality of the image produced by the JRCNN is visually pleasing. Moreover, the JRCNN restores textures accurately. The visual quality of the CISRDCNN [5] and JRCNN are competitive. The bicubic interpolated image contains blurring, ringing and blocking artifacts. The DNN-based cascaded methods, i.e., ARCNN-TNRD [6,7], ARCNN-VDSR [6,7] and ARCNN-DnCNN [7,36] produces blurred outputs. Basically, in cascaded methods, the CAR stage removes some of the information required for SR stage and produces blurred images. Similarly, the SR stage amplifies compression artifacts and CAR stage cannot suppress all the amplified artifacts and produces visually annoying output. The end-to-end learning frameworks, i.e., CISRDCNN [5] and the proposed JRCNN produces visually plausible images.

4.8 Time analysis

For real time applications testing time is also one of the key factors which decides the robustness of any network. Test time comparisons of different DNN-based methods is shown in Fig. 9. For fair comparison with the JRCNN only DNN-based methods are considered. The test time plot shown in Fig. 9 represents the average PSNR in dB and the average test time in Sec of Set5 dataset images for QF = 10 and $\times2$. Out of all compared methods, the JRCNN is the fastest method. The JRCNN outperforms the next best CISRDNN [5] method by a margin of 0.32dB. When it comes to speed the FSRCNN [9] is the second best method after the JRCNN.

5 Conclusion

A deep learning-based joint restoration DNN is proposed with cross residual connections, dense skip connections and parallel convolutional layers for an efficient super resolution of compressed and low resolution images. The cross residual connections and dense skip connections helped to ease the training by mitigating the problem of gradient vanishing/exploding. The parallel convolutional layer structure improved the efficiency of the proposed network by performing parallel convolution operations. A notable gain is achieved in performance metrics when compared to the state-of-the-art methods. The visual quality of the restored images by the proposed network is good when compared to other existing methods. Using the proposed network, three different restoration tasks, i.e., super resolution, compression artifacts removal and compressed low resolution image super resolution are also performed. The robustness and the stability of the proposed network is assessed by performing restoration tasks on different benchmark datasets.

References

1. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision—ECCV 2018, pp. 256–272. Springer International Publishing, Cham (2018)
2. Amaranageswarao, G., Deivalakshmi, S., Ko, S.-B.: Residual learning based densely connected deep dilated network for joint deblocking and super resolution. Appl. Intell. (2020). https://doi.org/10.1007/s10489-020-01670-y
3. Bevilacqua, M., Roumy, A., Guillemot, C.: line Alberi Morel M (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proceedings of the British Machine Vision Conference, BMVA Press, pp 135.1–135.10, https://doi.org/10.5244/C.26.135
4. Cavigelli, L., Hager, P., Benini, L.: Cas-cnn: A deep convolutional neural network for image compression artifact suppression. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp 752–759 (2017) https://doi.org/10.1109/IJCNN.2017.7965927
5. Chen, H., He, X., Ren, C., Qing, L., Teng, Q.: Cisrdcnn: super-resolution of compressed images using deep convolutional neural networks. Neurocomputing **285**, 204–219 (2018). https://doi.org/10.1016/j.neucom.2018.01.043
6. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1256–1272 (2017). https://doi.org/10.1109/TPAMI.2016.2596743
7. Dong, C., Deng, Y., Loy, C.C., Tang, X.: Compression artifacts reduction by a deep convolutional network. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 576–584 (2015) https://doi.org/10.1109/ICCV.2015.73

8. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016). https://doi.org/10.1109/TPAMI.2015.2439281
9. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision—ECCV 2016, pp. 391–407. Springer International Publishing, Cham (2016)
10. Fan, Y., Shi, H., Yu, J., Liu, D., Han, W., Yu, H., Wang, Z., Wang, X., Huang, T.S.: Balanced two-stage residual networks for image super-resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1157–1164 (2017) https://doi.org/10.1109/CVPRW.2017.154
11. Foi, A., Katkovnik, V., Egiazarian, K.: Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. IEEE Trans. Image Process. **16**(5), 1395–1411 (2007). https://doi.org/10.1109/TIP.2007.891788
12. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5197–5206 (2015) https://doi.org/10.1109/CVPR.2015.7299156
13. Kang, L., Hsu, C., Zhuang, B., Lin, C., Yeh, C.: Learning-based joint super-resolution and deblocking for a highly compressed image. IEEE Trans. Multimed. **17**(7), 921–934 (2015)
14. Kayhan, S.K.: Efficient robust filtering technique for blocking artifacts reduction. Vis. Comput. (2015). https://doi.org/10.1007/s00371-015-1068-0
15. Keys, R.: Cubic convolution interpolation for digital image processing. IEEE Trans. Acoust. Speech Signal Process. **29**(6), 1153–1160 (1981)
16. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1646–1654 (2016) https://doi.org/10.1109/CVPR.2016.182
17. Kim, Y., Soh, J.W., Park, J., Ahn, B., Lee, H., Moon, Y., Cho, N.I.: A pseudo-blind convolutional neural network for the reduction of compression artifacts. IEEE Trans. Circuits Syst. Video Technol. (2019). https://doi.org/10.1109/TCSVT.2019.2901919
18. Lai, W., Huang, J., Ahuja, N., Yang, M.: Fast and accurate image super-resolution with deep laplacian pyramid networks. IEEE Trans. Pattern Anal. Mach. Intell. **41**(11), 2599–2613 (2019). https://doi.org/10.1109/TPAMI.2018.2865304
19. Lan, R., Sun, L., Liu, Z., Lu, H., Su, Z., Pang, C., Luo, X.: Cascading and enhanced residual networks for accurate single-image super-resolution. IEEE Trans. Cybern. (2020)
20. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140 (2017) https://doi.org/10.1109/CVPRW.2017.151
21. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-cnn for image restoration. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 886–88609 (2018) https://doi.org/10.1109/CVPRW.2018.00121
22. Ma, T., Tian, W.: Back-projection-based progressive growing generative adversarial network for single image super-resolution. In: Vis Comput (2020)
23. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, Curran Associates, Inc., pp. 2802–2810 (2016)
24. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol 2, vol. 2, pp. 416–423 (2001) https://doi.org/10.1109/ICCV.2001.937655
25. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimed. Tools Appl. **76**(20), 21811–21838 (2017). https://doi.org/10.1007/s11042-016-4020-z
26. Qiu, Y., Wang, R., Tao, D., Cheng, J.: Embedded block residual network: A recursive restoration model for single-image super-resolution. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4179–4188 (2019)
27. Song, Q., Xiong, R., Fan, X., Liu, X., Huang, T., Gao, W.: Compressed image restoration via external-image assisted band adaptive pca model learning. In: 2018 Data Compression Conference, pp. 97–106 (2018) https://doi.org/10.1109/DCC.2018.00018
28. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2790–2798 (2017) https://doi.org/10.1109/CVPR.2017.298
29. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4549–4557 (2017b) https://doi.org/10.1109/ICCV.2017.486
30. Timofte, R., DéSmet, V., Van Gool, L.: A+: adjusted anchored neighborhood regression for fast super-resolution. ACCV **9006**, 111–126 (2015). https://doi.org/10.1007/978-3-319-16817-3_8
31. Timofte, R., Agustsson, E., Gool, L.V., Yang, M., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., Wang, X., Tian, Y., Yu, K., Zhang, Y., Wu, S., Dong, C., Lin, L,, Qiao, Y., Loy, C.C., Bae, W., Yoo, J., Han, Y,, Ye, J.C., Choi, J., Kim, M., Fan, Y,, Yu, J., Han, W., Liu, D., Yu, H., Wang, Z., Shi, H., Wang, X,, Huang, T.S., Chen, Y., Zhang, K., Zuo, W., Tang, Z., Luo, L., Li. S., Fu, M., Cao, L., Heng, W., Bui, G., Le, T., Duan, Y., Tao, D., Wang, R., Lin, X., Pang, J., Xu, J., Zhao, Y., Xu, X., Pan, J., Sun, D., Zhang, Y., Song, X., Dai, Y., Qin, X., Huynh, X., Guo, T., Mousavi, H.S., Vu, T.H., Monga, V., Cruz, C., Egiazarian, K., Katkovnik, V., Mehta, R., Jain, A.K., Agarwalla, A., Praveen, C.V.S., Zhou, R., Wen, H., Zhu, C., Xia, Z., Wang, Z., Guo, Q.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1110–1121 (2017) https://doi.org/10.1109/CVPRW.2017.149
32. Wallace, G.K.: The jpeg still picture compression standard. IEEE Trans. Consum. Electron. **38**(1), xviii–xxxiv (1992)
33. Xiao, J., Wang, C., Hu, X.: Single image super-resolution in compressed domain based on field of expert prior. In: 2012 5th International Congress on Image and Signal Processing, pp. 607–611 (2012)
34. Zeyde, R., Elad, M., Protter M.: On single image scale-up using sparse-representations. **6920**, 711–730 (2010). https://doi.org/10.1007/978-3-642-27413-8_47
35. Zhang, J., Xiong, R., Zhao, C., Zhang, Y., Ma, S., Gao, W.: Concolor: Constrained non-convex low-rank model for image deblocking. IEEE Trans. Image Process. **25**(3), 1246–1259 (2016). https://doi.org/10.1109/TIP.2016.2515985
36. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: residual learning of deep cnn for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017). https://doi.org/10.1109/TIP.2017.2662206
37. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3262–3271 (2018) https://doi.org/10.1109/CVPR.2018.00344
38. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Com-

puter Vision—ECCV 2018, pp. 294–310. Springer International Publishing, Cham (2018)
39. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2472–2481 (2018) https://doi.org/10.1109/CVPR.2018.00262
40. Zhao, C., Zhang, J., Ma, S., Fan, X., Zhang, Y., Gao, W.: Reducing image compression artifacts by structural sparse representation and quantization constraint prior. IEEE Trans. Circuits Syst. Video Technol. **27**(10), 2057–2071 (2017). https://doi.org/10.1109/TCSVT.2016.2580399
41. Wang, Zhou, Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861

Gadipudi Amaranageswarao is currently pursuing his Ph.D. degree at National Institute of Technology, Thiruchirappalli, India. He received his M.Tech. degree from Jawaharlal Nehru Technological University, Anantapur in Digital Electronics and Communication Systems and B.Tech. degree from Jawaharlal Nehru Technological University, Pulivendula in the Department of Electronics and Communication Engineering. His research interests include Machine Learning, Deep Learning, Image Restoration and Medical Image Analysis.

S. Deivalakshmi currently an Assistant Professor of Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli, India. She received the Ph.D. degree from Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli, India. Her research interests include deep networks for various image processing applications that include image super-resolution, denoising, dehazing and segmentation.

Seok-Bum Ko is currently a Professor at the Department of Electrical and Computer Engineering and the Division of Biomedical Engineering, University of Saskatchewan, Canada. He got his Ph.D. degree from the University of Rhode Island, USA in 2002. His research interests include computer architecture/arithmetic, efficient hardware implementation of compute-intensive applications, deep learning processor architecture and biomedical engineering. He is a senior member of IEEE circuits and systems society and associate editors of IEEE TCAS I and IEEE Access.