Digital Heritage Reconstruction using deep learning-based Super-resolution

Prathmesh R. Madhu and Manjunath V. Joshi

Abstract Heritage sites and archival monuments have a great cultural significance. However, they suffer degradation due to several reasons. As a result, in order to preserve the cultural heritage, one has seen increased interest in research on digitally restoring the photographs of vandalized monuments. One may think of recreating the historical monuments by super-resolving the heritage images, an algorithmic approach to increase the spatial resolution of an image. This chapter presents a single image super-resolution (SR) method based on deep learning to obtain higher resolution photographs of the digitally reconstructed monuments. The resulting images can serve as the input to walkthrough systems. Given a low spatial resolution test image and a database consisting of low and high spatial resolution (LR - HR) images, we obtain super-resolution for the test image. We use the idea proposed in [5] to represent the mapping between LR and HR images by using a deep convolutional neural network (CNN). CNN filters are learned by standard back-propagation and stochastic gradient descent method. The novelty of our approach lies in the elimination of interpolation during the training phase. Our method directly learns the end-to-end mapping between LR and HR images. The advantage of our approach is that once the network is trained for a magnification factor of 2, the learned parameters can be used to obtain SR for higher magnification factors also. We demonstrate the effectiveness of the proposed approach by conducting experiments using the images of heritage monuments as well as natural scene. Our results are compared with the standard interpolation technique and existing learning-based approaches. Visual and quantitative comparisons confirm the effectiveness of the proposed method.

Prathmesh R. Madhu
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India,
e-mail: prmadhu@daiict.ac.in

Manjunath V. Joshi
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India,
e-mail: mv_joshi@daiict.ac.in

1 Introduction

Heritage and archaeological sites across the world are a major source of information which acquaint us with our social history delineating the advancement of mankind. They are invaluable assets of cultural heritage. Such places serve as an excellent attraction for tourists which indirectly impacts the gross domestic product. This is one of the major reasons for government agencies globally for taking keen interest towards preserving these sites. Over a period of time, a number of natural calamities such as weathering, man-made hazards like pollution etc., have sabotaged these heritage sites. Fearing from any further damages to these sites by the tourists, access to many heritage sites is now restricted. One such example is the *mandapa with musical pillars* in Vithala temple at Hampi in India, where the visitors are not allowed to touch and experience the melodious sound from the musical pillars. A way to preserve the heritage sites is to physically renovate them. However, it requires a prolific amount of historical information in order to renovate them. It also poses a danger to the undamaged monuments and may fail to mimic the skill-full historical work.

One may overcome the above problems by using restoration in digital domain An image can be acquired by sampling a continuous scene using a camera and the restoration can be carried out by using in-painting and super-resolution methods. In this work, we restrict our degradation in spatial domain only and propose a super-resolution approach for spatial resolution enhancement. When the sampling rate of the acquired scene is less than the Nyquist rate which is often the case with mobile cameras used by the tourists, it results in aliasing effect leading to distortion in the captured image. In addition to this, sensor point spread function and the motion of the camera introduces the blur that degrades the quality of the image being captured. Image super-resolution is an algorithmic approach for increasing the spatial resolution of an image from one or more LR observations. It aims at providing details finer than the sampling grid of a given imaging device by increasing the number of pixels per unit area in an image [18]. Alternatively, it aims at restoring the high-frequency components and removal of degradation which arise during the image acquisition process. Hence, a super-resolved image can be considered close to the true image captured using an HR camera.

SR methods can be broadly divided into three categories. They correspond to motion-based, motion-free and single-frame SR techniques. Motion-based techniques use the relative motion between the observed LR images as a cue to estimate an HR image. On the contrary, motion-free techniques use cues such as blur, zoom and defocus. Single frame SR methods aim at reconstructing the HR image using a single degraded LR observation. However, in this case, a database of training images is required. Survey of different SR techniques is presented in the work by Park et al. [17].

The resolution enhancement of an image dates back to 1984 by Tsai and Huang [22], where they demonstrate the reconstruction of single enhanced resolution image using several down-sampled noisy versions of it. One may find various approaches implemented in frequency domain in [2, 14, 19]. Irani and Peleg presented iterative back-projection based method where a super-resolved image is estimated by computing the error between observed LR images and corresponding simulated LR images formed using current SR estimate. A unified approach of super-resolution and demosaicing using the bilateral regularization was proposed by Farsiu et al. [6]. They use L_1 norm minimization to make their method robust to data and modeling errors. Baker and Kanade [1], Lin and Shum [16] present the limitations of classical multi-image SR approaches. These limitations paved the way for example-based SR approach to achieve higher magnification factors. The first example based approach was proposed by Freeman et al. [9], wherein they create an over-complete dictionary consisting of LR-HR patch-pairs constructed using a large number of LR-HR training images. The dictionary generated is then used to restore the missing high-frequency details. Yang et al. [24] proposed a compressive sensing based approach by using sparsity constraint and over-complete dictionaries. New edge-preserving SR approaches are presented in [7, 8, 20]. Gajjar and Joshi [10] proposed a new wavelet-based learning process using the Inhomogenous Gaussian Markov Random Field (IGMRF) prior.

Recently, researchers have attempted to learn the mapping between the LR and HR images using machine learning techniques. Based on the work of Freeman et al.[9], Kim and Kwon proposed a regression-based approach for single-image SR [13]. Here, the authors use kernel ridge regression to estimate the high-frequency details of the underlying HR image. In [11], a hybrid of reconstruction and learning based technique is implemented. With the efficient training scheme of deep neural networks proposed by Hinton et al. [12], deep learning has been applied in various applications such as classification and recognition in the field of computer vision as well as in speech processing, natural language processing etc. Very recently, deep learning methods have been applied to image restoration. In [3], a multi-layer perceptron (MLP) is used for natural image denoising and image deblurring. In [4], authors have combined the non-local self-similarity search with collaborative auto-encoders to super-resolve the LR test image. Convolutional neural networks (CNN) attempt to learn layered, hierarchical representations of higher dimensional data which are called features. Closely related to our work, CNN is applied for SR in [5], where the authors have proposed a CNN for SR.

In this chapter, we present a method for single image super-resolution based on deep learning. We eliminate the need for interpolation techniques as used in [5] during the training and reconstruction. To do this, we insert zeros in the alternate rows and columns of the LR training images and call them as zero-inserted LR (ZILR) training images. Note that we do not manipulate the training data by interpolation to get LR resized as done in [5]. In effect, we use the given training set of LR-HR pairs only and not the manipulated LR pixels. This is advantageous since the

network directly learns the mapping between true LR and HR image pixels avoiding the use of manipulated (interpolated) pixels. For training the network, we use the ZILR and the corresponding original HR images as our input and the output of the network, respectively. Also, during training phase the backpropagation error is computed between the current pixel intensities of network output at those locations where zeros are inserted at the input and the corresponding intensities of true HR, thus manipulating only the pixel intensities of interest. This avoids manipulating already known HR values and it also reduces the training time. It also results in better learning of weights (parameters) of the network since the weights multiplied with the inserted zeros do not contribute while learning. Using the corresponding sub-images of ZILR and HR images, we first train our proposed CNN to find the direct mapping/transformation between LR and HR images. The trained network is then used to obtain the super-resolved output of test LR image. Another advantage of our approach is that once the network is trained for a magnification factor of 2, we need not train the network for obtaining SR for higher magnification factors. Once trained for a factor of 2, the same weights can be used in obtaining SR up to a factor of 8.

The outline of the chapter is as follows. Section 2 describes deep convolutional neural networks (CNNs) and learning of the CNN parameters. In section 3, general description of the proposed method is presented. Implementation details are explained in section 4, while the experimental results and the performance of the proposed approach are dealt in section 5. Some concluding remarks are drawn in section 6.

2 Convolutional Neural Networks

This section is a mini-tutorial on Convolutional neural networks in 2D. Readers may skip this section if they have a fair amount of knowledge in this field.

Convolutional neural networks (CNNs) are very useful in modeling a complex relationship between the input and the output data. They can be used in an unsupervised way to learn the features or in a supervised way for classification purpose. These networks are designed using three basic constituents: local receptive fields, shared weights, and pooling. Much of the earlier works on CNN was concentrated on classification. For this, the corresponding network architecture consists of three kinds of layers namely convolutional, pooling and the fully-connected layer, similar to multi-layer perceptron (MLP). Every layer (except the pooling layer) is associated with the parameters or weights that correspond to a set of learnable filters. During the forward pass, we convolve each filter across the input volume, which results in a set of 2-dimensional (2D) activation maps corresponding to that filter. Intuitively, the network learns filters that activate when there is a specific feature at different spatial positions in the input. A set of filter weights connect each neuron to a local region of the input volume making the connectivity as sparse. The spatial extent of this connectivity is called as the receptive field of a neuron. The filter weights are shared for the entire input image. Due to this weight-sharing property of CNN, there is a dramatic reduction in the number of weights to be learned when compared to conventional multilayer perceptron (MLP) [15]. Pooling layer is used for reducing the dimension and has a great significance while solving classification problem. In our work, since we are working on a reconstruction problem, we do not use the pooling layer. When a CNN has more than one hidden layer, then it is referred to as deep CNN (DCNN). These layers are added to learn hierarchical features between the input and the output.

The advantages of using CNN when compared to other neural networks are as follows:

- Sharing of weights i.e., reduced number of network parameters to be learned.
- Self-learning of filters by the network that eliminates the need for prior knowledge.
- Input data is 2D, preserving the structure and spatial dependencies.
- Less memory requirement since the same filter is applied to the entire input image.

As seen in Figure 1, a 3 hidden layered deep CNN used for super-resolution comprises of only convolutional layers. Layers such as pooling and fully connected layer are also used in the model while solving classification problem.

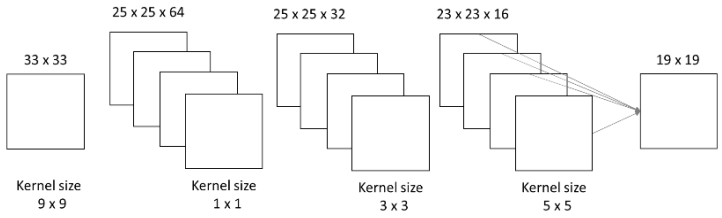


Fig. 1 General block diagram of CNN used for super-resolution.

2.1 Training a CNN

In order to learn the parameters, one has to train the network by using a large amount of data (here images). The three steps involved in training a CNN correspond to forward propagation, back-propagation and updating weights. In what follows, we discuss each of these steps briefly.

2.1.1 Forward Propagation

Given an image of size N x N which is followed by our convolutional layer, if we use an m x m sized filter(s) denoted as ω, then the output of the convolutional layer will be of size $(N-m+1)$ x $(N-m+1)$. To get the convolved output at location (i, j) of an ℓ^{th} layer i.e., $x^{\ell}(i, j)$, we need to sum up the contributions from previous layer cells i.e.,

$$x^{\ell}(i,j) = \sum_{c=0}^{m-1} \sum_{d=0}^{m-1} \omega(c,d) y^{\ell-1}(i+c, j+d). \tag{1}$$

Eq. (1) represents convolving the filter and the receptive field at (i, j). Here, $y^{\ell-1}(i+c, j+d)$ represents the convolved output of $(\ell-1)^{th}$ layer at location $(i+c, j+d)$. Carrying out the convolution over the entire image results in a convolved image or convolution layer map. Use of a number of filters gives us a number of images in the convolution layer. A non-linearity applied on $x^{\ell}(i, j)$ for all i, j results in $y^{\ell}(i, j) = \sigma(x^{\ell}(i, j))$.

Here, σ represents a non-linear function. Different kinds of non-linear functions can be used. Since our work involves reconstruction, we use the rectified linear unit (ReLU) which thresholds negative values to zero and it can be mathematically represented as ReLU(x) = $max(0, x)$. It may be mentioned here that a sigmoid function is used while solving a classification problem.

2.1.2 Backward Propagation for computing error derivatives with respect to (w.r.t) weights (parameters)

Let us consider an error function E that corresponds to squared error between input (x) and the current output (y). Note that we consider the error corresponding to zero inserted locations only while computing E. In order to perform a backward pass, we need to know the partial derivatives of error with respect to each neuron output $\left(\frac{\partial E}{\partial y^{\ell}(i,j)}\right)$. To do this, we first find the gradient of E with respect to (w.r.t) each weight for l^{th} layer which is computed using the chain rule as,

$$\begin{aligned}\frac{\partial E}{\partial \omega(c,d)} &= \sum_{i=0}^{N-m}\sum_{j=0}^{N-m}\frac{\partial E}{\partial x^{\ell}(i,j)}\frac{\partial x^{\ell}(i,j)}{\partial \omega(c,d)} \\ &= \sum_{i=0}^{N-m}\sum_{j=0}^{N-m}\frac{\partial E}{\partial x^{\ell}(i,j)} y^{\ell-1}(i+c,j+d).\end{aligned} \tag{2}$$

Note that, because of the weight sharing property, $\omega(c,d)$ occurs in the sum of all $x^{\ell}(i,j)$ expressions. Also, from the forward propagation, we know that $\frac{\partial x^{\ell}(i,j)}{\partial \omega(c,d)} = y^{\ell-1}(i+c,j+d)$. Hence to compute the gradient, we need the values of $\frac{\partial E}{\partial x^{\ell}(i,j)}$, which are called deltas (errors). Once again, using the chain rule, one can calculate these deltas as

$$\begin{aligned}\frac{\partial E}{\partial x^{\ell}(i,j)} &= \frac{\partial E}{\partial y^{\ell}(i,j)}\frac{\partial y^{\ell}(i,j)}{\partial x^{\ell}(i,j)} \\ &= \frac{\partial E}{\partial y^{\ell}(i,j)}\frac{\partial}{\partial x^{\ell}(i,j)}\left(\sigma(x^{\ell}(i,j))\right) \\ &= \frac{\partial E}{\partial y^{\ell}(i,j)}\sigma'(x^{\ell}(i,j)).\end{aligned} \tag{3}$$

Note that σ' represents the derivative of σ.

Hence, it is simple to calculate the deltas $\frac{\partial E}{\partial x^{\ell}(i,j)}$ at the current layer by just using the derivative of the activation function $\sigma'(x)$ as we already know errors at current layer $\frac{\partial E}{\partial y^{\ell}(i,j)}$. Now the error needs to be backpropagated, where we again use the chain rule as

$$\begin{aligned}\frac{\partial E}{\partial y^{\ell-1}(i,j)} &= \sum_{c=0}^{m-1}\sum_{d=0}^{m-1}\frac{\partial E}{\partial x^{\ell}(i-c)(j-d)}\frac{\partial x^{\ell}(i-c)(j-d)}{\partial y^{\ell-1}(i,j)} \\ &= \sum_{c=0}^{m-1}\sum_{d=0}^{m-1}\frac{\partial E}{\partial x^{\ell}(i-c)(j-d)}\omega(c,d).\end{aligned} \tag{4}$$

Note that we have used $\frac{\partial x^{\ell}(i-c)(j-d)}{\partial y^{\ell-1}(i,j)} = \omega(c,d)$ in Eq. (4) which can be obtained from equation (1).

2.1.3 Updating the weights

In order to choose the optimum network parameters, we minimize the error function E. Stochastic gradient descent (SGD) method is an approximation of the gradient descent optimization method for minimizing an error function. SGD performs the update of weights and uses the gradient of the error function w.r.t weights of a single or a few training examples which are calculated using the backpropagation discussed in section 2.1.2. Hence, the update equation for SGD is given as

$$w := w - \alpha \nabla_w E(w; x^{(i)}, y^{(i)}), \tag{5}$$

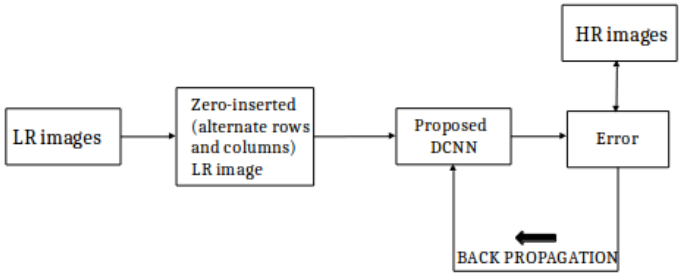
where w and $x^{(i)}, y^{(i)}$ represent the parameter vector and i^{th} training example, respectively. Here the term $\nabla_w E(w; x^{(i)}, y^{(i)})$ represents the gradient of the error function with respect to the network parameters, i.e. weights and α is the learning rate. The pseudo code for SGD is presented in Algorithm 1.

Algorithm 1: Pseudocode for SGD

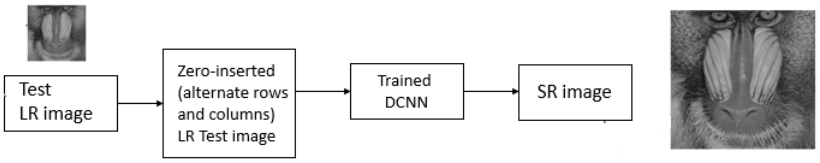
```
for every iteration: do
    Randomly shuffle the input data;
    for every example in the data: do
        compute gradients with respect to weights using backpropagation;
        update weights as weights := weights - learning rate*gradient with respect to
          weights.
    end
end
```

3 Proposed Approach

The proposed technique of deep learning based super-resolution is illustrated by the block diagram shown in Figure 2. Given an LR test image and LR and HR training images, we first learn the mapping in terms of network parameters using a deep convolutional neural network (DCNN) for a magnification factor of 2. The only pre-processing we perform is the insertion of zeros into alternate rows and columns of the LR in order to enlarge it to the desired size. This is in contrast to the bicubic interpolation performed as a preprocessing step during the training phase in the method proposed by Dong et al. [5]. Inserting zeros avoids the creation of unwanted input data which is generated by the interpolation operation. In figure 2(a), we show the block schematic of the training and the testing phase is shown in figure 2(b). The expanded view of the proposed DCNN is shown in figure 3.



(a)



(b)

Fig. 2 Block Schematic of the proposed approach for image super-resolution. (a) Training phase; (b) Testing phase.

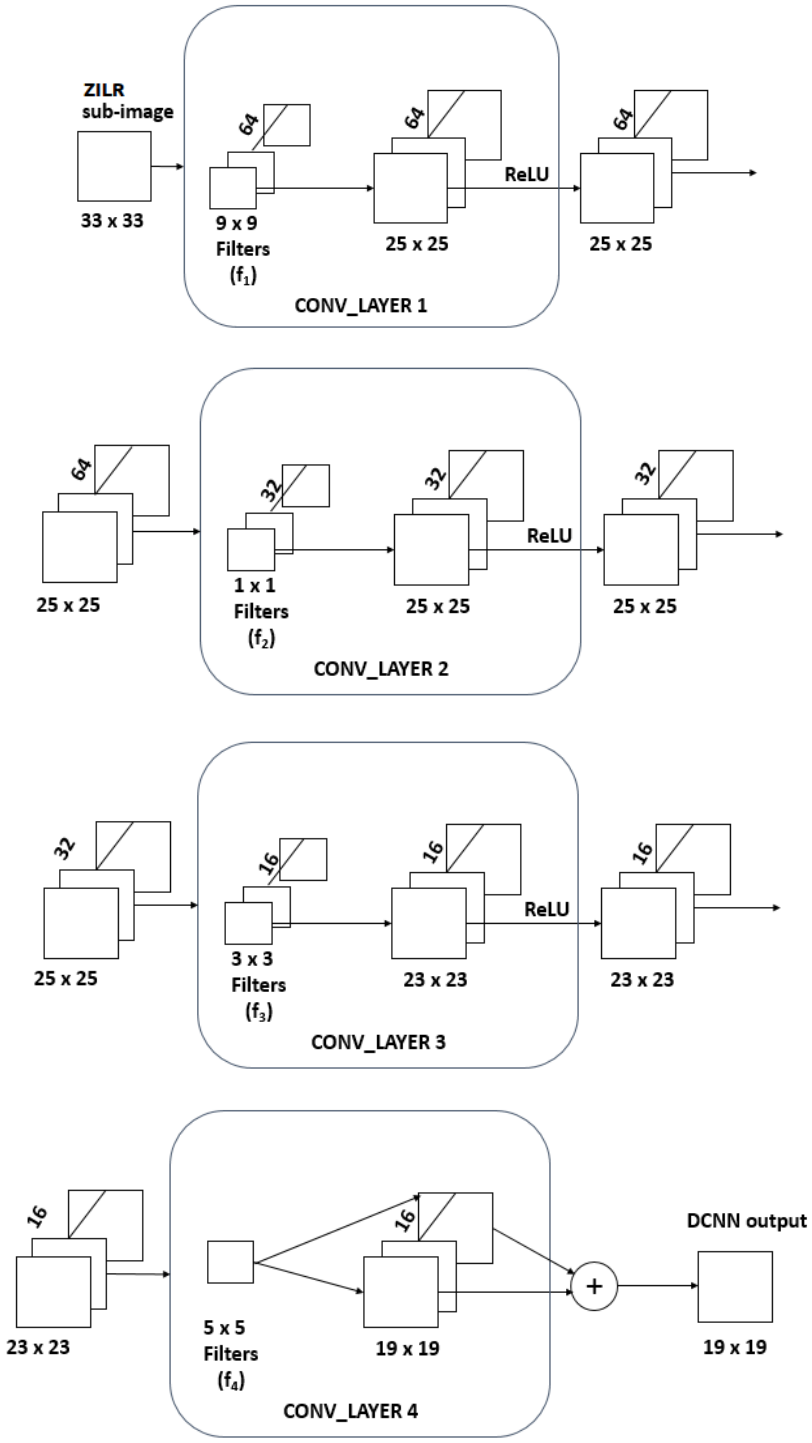


Fig. 3 Expanded view of the proposed DCNN architecture. Note that there are 4 convolutional layers. The output of each convolution layer is given as input to the next convolution layer.

Note that in our approach, we propose a standard upsampling method by inserting zeros in alternate rows and columns of the LR images. This concept can be considered as an equivalent of de-noising autoencoder proposed by Vincent et al. [23], in which the image is corrupted with zeros. A de-noising auto-encoder tries to encode the input mapping between the zero inserted input and the true output. It attempts to undo the effect of a corruption process by capturing the statistical dependencies between the corrupted input pixels. However, we model this as a super-resolution problem, where the missing values can be replaced by zeros and the DCNN learns the mapping between neighbors of ZILR pixels and the corresponding pixels in HR image. We train our proposed DCNN using sub-images (patches) of ZILR and HR images, details of which can be found in section 4. The error at the output is calculated by the square of the difference between the network output and the cropped version of the HR sub-image corresponding to the output patch size where the values corresponding to ZILR only are considered in both the images. Weights are updated using standard error back-propagation and stochastic gradient descent method as discussed in section 2.1. Once we obtain the learned network parameters, we use these in forward-pass to obtain the SR for the test image.

4 Implementation Details

For a fair comparison with the traditional state-of-art methods in example-based learning, we use the same training sets i.e. LR and HR images as used in [21]. There are 91 color images of different sizes. The Set5 consisting of 5 images is used to evaluate the performance for magnification factors of 2, 4 and 8.

As seen from Figure 3, our input consists of the sub-images of size 33 x 33, and size of our filters f_1, f_2, f_3, f_4 are 9, 1, 3, 5, respectively. For all the 91 HR images, we first down-sample them by removing alternate rows and columns i.e. under-sampling by a factor of 2 to generate the LR training images. We then insert zeros in alternate rows and columns of these LR training images and call them ZILR training images. All our experiments are conducted by simulating the LR images from the available 91 HR images. One may also use LR-HR pairs acquired by a real camera.

In order to train the DCNN, we prepare 33 x 33 pixel sub-images from zero-inserted LR training images and their corresponding HR images. Note that the sub-images do not represent overlapping patches which require post-processing as done in [5]. With 91 training images, we get approximately 35,000 sub-images. These sub-images are obtained from the training images with a stride[1] of 11. We consider the luminance channel in our experiments and hence a single channel is used from the color images, as seen in Figure 3. To remove border effects, we do not use padding during training and hence our network results in a smaller output i.e. 19 x 19. Due to this, the squared error is calculated by computing the difference between central 19 x 19 crop of the ground truth (HR) sub-image of the corresponding LR input and the network output. The filter weights are initialized randomly using the samples from the Gaussian distribution with the zero mean and standard deviation of 0.001.

The training phase was implemented using CAFFE in GPU mode. Algorithm 2 gives the implementation details for training phase. The following notations are used in Algorithm 2:

- W_i is the weight matrix comprising of weights connecting the i^{th} layer with the previous layer. Here, i = 1, 2, 3, 4.
- b_i is the bias vector for i^{th} layer.
- **W** is a matrix consisting of W_1, W_2, W_3, W_4
- **b** is a matrix consisting of b_1, b_2, b_3, b_4

[1] stride - It is an arbitrary shift we use to obtain the next subimage within the image

Algorithm 2: Training of proposed DCNN for a magnification factor of 2.

Input: LR images , Corresponding true HR images.
Output: Trained weights and biases $(W_1, W_2, W_3, W_4, b_1, b_2, b_3, b_4)$.

1. Given the HR images create ZILR images to make LR and HR of the same size.
2. Extract sub-images of size 33 x 33 from the LR and its corresponding HR images. Let $x^{(i)}$ and $y^{(i)}$ denote i^{th} input and output sub-images to the network.
3. Initialize a 4 layered CNN with weights and biases.
4. Calculate the hidden layer activations and final network output using the equation (1).
5. Compute the error function E at the output

$$E = J(\mathbf{W}, \mathbf{b}) = \frac{1}{m} \sum_{i=1}^{m} (||h_{\mathbf{W},\mathbf{b}}(x^{(i)}) - y^{(i)}||^2)$$

   where $h_{W,b}(x^{(i)})$ is the output of the network for $x^{(i)}$ and m is the total number of training sub-images.
6. Optimize the error function by updating the weights using back-propagation and stochastic gradient descent in order to obtain the trained weights and biases.

5 Experimental results

In this section, we discuss the efficacy of the proposed technique in the context of super-resolving an LR image using the learned deep convolutional neural network. We present the results on the luminance channel for the magnification factors of 2, 4 and 8. As already discussed in section 4, the training data is the same in all experiments. The learning rate α during training was set to 0.0001. Given the test LR image and the already learned weights, we implement the forward pass in order to get SR for the test image.

To start with, we consider an experiment for a magnification factor of 2 on face image which is shown in figure 4. In order to obtain an LR image, we downsample the given test image by removing alternate rows and columns. Figure 4(a) shows an LR face image of a woman. The upsampled image for a factor of 2 with inserted zeros in alternate rows and columns is shown in figure 4(b). The bicubically interpolated image is shown in figure 4(c). Figures 4(d, e) display the super-resolved images by Kim and Kwon's [13] and Dong et al.'s [5] approaches, respectively. Super-resolved image using our approach is displayed in figure 4(f), and the ground truth i.e., true HR image (available test image itself in this case) is shown in figure 4(g). In comparison to the image shown in figure 4(c), the super-resolved image in figure 4(f) shows clearer details. The features within the image such as texture in the cap, eyelashes, eyes, mouth etc. are hazy in the bicubic interpolated image shown in figure 4(c) while they are well preserved in figure 4(f). One can see that the SR features in the image of the proposed approach are comparable to that of the ground truth displayed in figure 4(g). The details are better preserved in the proposed approach when compared to figure 4(c) and the images shown in figures 4(d, e).

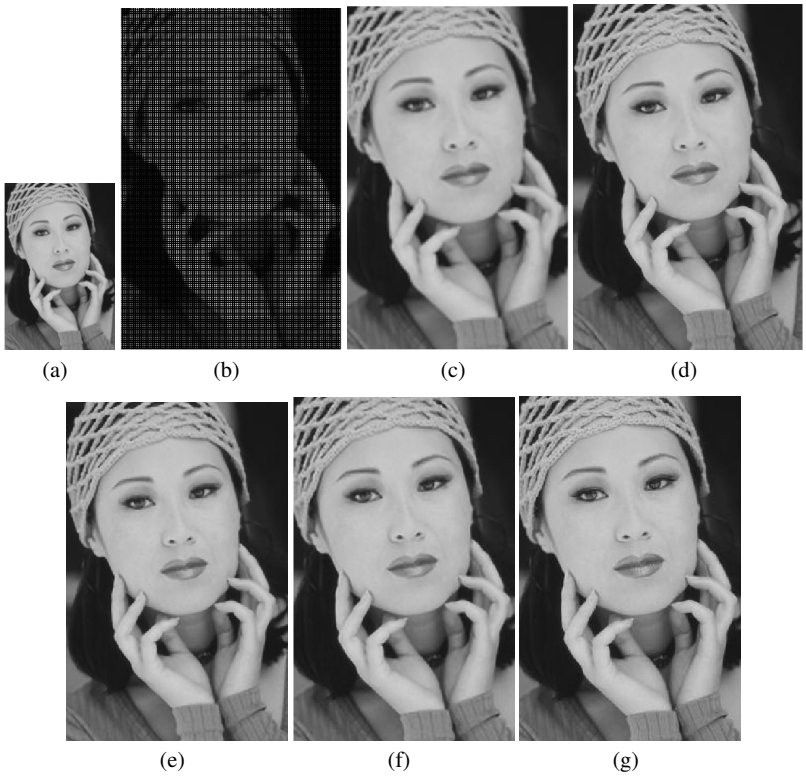


Fig. 4 SR Results for 2X : Woman. (a) low-resolution; (b) ZILR; (c) Bicubically interpolated image; (d) Super-resolved image using Kim and Kwon [13] approach; (e) Super-resolved image using SRCNN [5] approach; (f) Super-resolved image using proposed method; (g) Ground truth image.

To test our algorithm for images having high-frequency components, we now consider an image of a bird sitting on a tree branch. One may note that learning here is more challenging since this image has a significant number of edge features and the network has to learn the edge details. The low-resolution image is displayed in figure 5(a), while our SR result is shown in figure 5(f). We can observe that the beak, eyes and the branches of the tree in the proposed approach look sharper than the bicubic interpolated image. It can also be seen from figure 5(f) that the texture near the beak and branches of the tree are better preserved when compared to the image in figure 5(d) and are similar to the image shown in figure 5(e).

Next, we consider two test images from historical monument **Hampi temple** with reference to the cultural heritage. Figure 6(a) shows a low-resolution image of a bull, named Nandi which is said as Lord Shiva's chariot, taken at Hampi temple, while our SR result is shown in figure 6(f). Observe the finer details in comparison

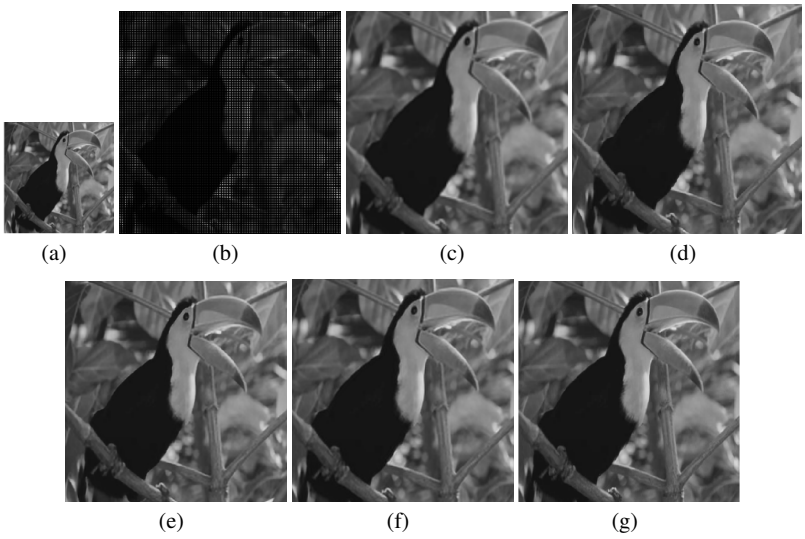


Fig. 5 SR Results for 2X : Bird. (a) low-resolution; (b) ZILR; (c) Bicubically interpolated image; (d) Super-resolved image using Kim and Kwon [13] approach; (e) Super-resolved image using SRCNN [5] approach; (f) Super-resolved image using proposed method; (g) Ground truth image.

to bicubic interpolation shown in figure 6(c). Our SR result looks similar to the image of figure 6(f). In figure 7 we show another result where the LR image which is shown in figure 7(a) corresponds to that of an engraved piece of art-work on one of the musical pillars at Hampi temple. Here, we see that the finer texture of the stone and the edge transitions are nicely captured in the image of the proposed method shown figure 7(f)

Table 1 PSNR (dB) comparison of our proposed method with bicubic interpolation (BI), Kim et al. [13] and SRCNN [5]. Here BI and MF represent bicubic interpolation and magnification factor respectively.

Test images	MF	BI	[13]	[5]	Proposed
Woman	2	32.29	34.80	**35.02**	34.94
Bird	2	36.81	40.16	40.61	**40.81**
Nandi	2	35.79	36.35	37.62	**37.64**
Art engraved stone at Hampi temple	2	31.64	31.73	32.84	**32.89**
Art-piece on walls of Hampi temple	4	29.5	29.74	30.54	**30.95**
Head	4	31.23	32.40	31.88	**32.88**
Pillars	8	33.03	-	-	34.36

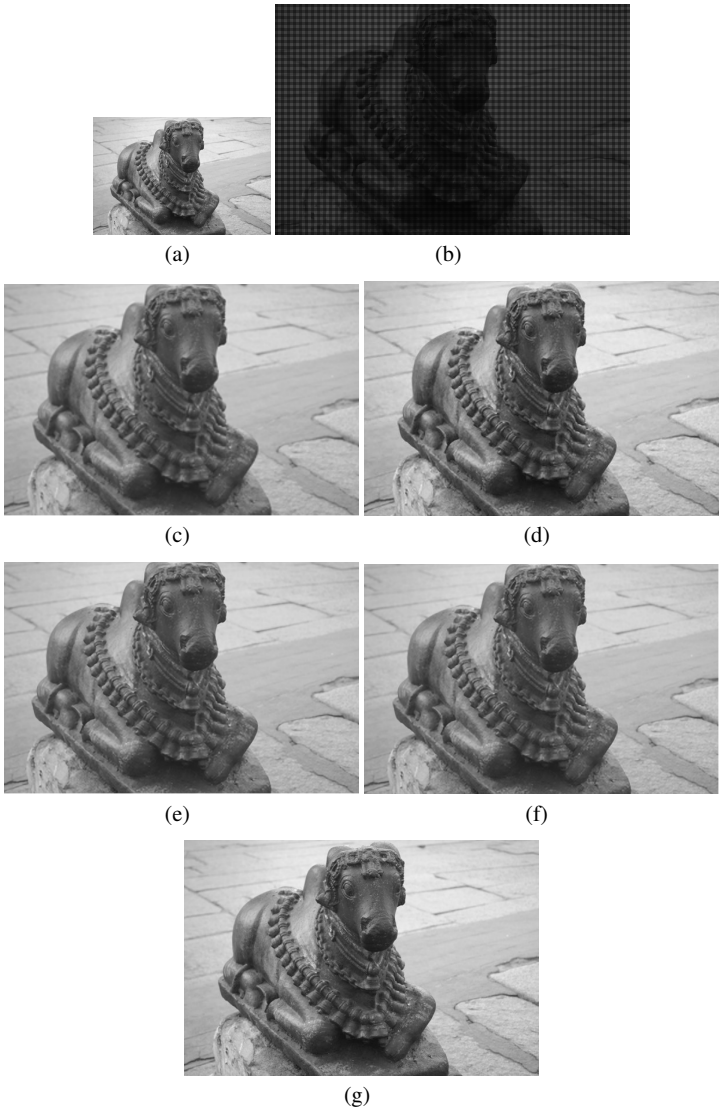


Fig. 6 SR Results for 2X of Nandi captured at Hampi temple. (a) low-resolution; (b) ZILR; (c) Bicubically interpolated image; (d) Super-resolved image using Kim and Kwon [13] approach; (e) Super-resolved image using SRCNN [5] approach; (f) Super-resolved image using proposed method; (g) Ground truth image.

We now discuss results for a magnification factor of 4. In this case, the HR image is downsampled by a factor of 4 to obtain the LR image. The super-resolution is then performed by upsampling the LR in stages up to a factor of 4 using the following

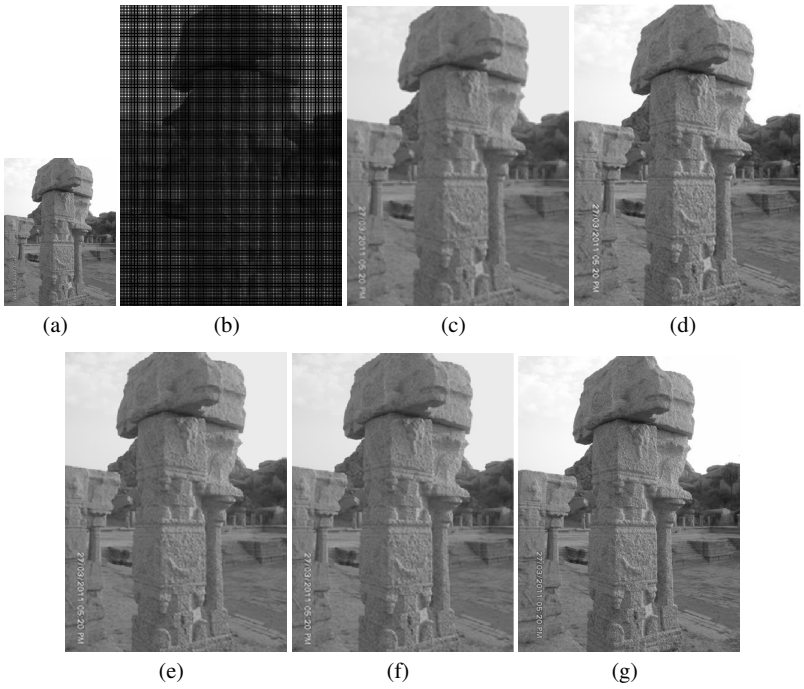


Fig. 7 SR Results for 2X : An art engraved on stone at one of the musical pillars at Hampi. (a) low-resolution; (b) ZILR; (c) Bicubically interpolated image; (d) Super-resolved image using Kim and Kwon [13] approach; (e) Super-resolved image using SRCNN [5] approach; (f) Super-resolved image using proposed method; (g) Ground truth image.

Table 2 SSIM comparison. Here BI and MF represent bicubic interpolation and magnification factor respectively.

Test images	MF	BI	Proposed SR method
Woman	2	0.87	0.91
Bird	2	0.93	0.94
Art-piece on walls of Hampi temple	4	0.62	0.66
Head	4	0.42	0.46

steps: (a) insert zeros in alternate rows and columns (b) super-resolve the resulting zero inserted image to obtain SR for a magnification factor of 2 (c) insert zeros in alternate rows and columns in the resulting SR image (d) use the same network to super-resolve to obtain super-resolution for a factor of 4. Note that no new model is trained here for a magnification factor of 4, instead, we make use of the already trained model that provides 2X magnification. This is possible since our input consists of patches of an image rather than the entire image itself. Once the network is trained the same parameters can then be used in obtaining the upsampled patches for any magnification factor assuming that the result of the upsampled image is closer to

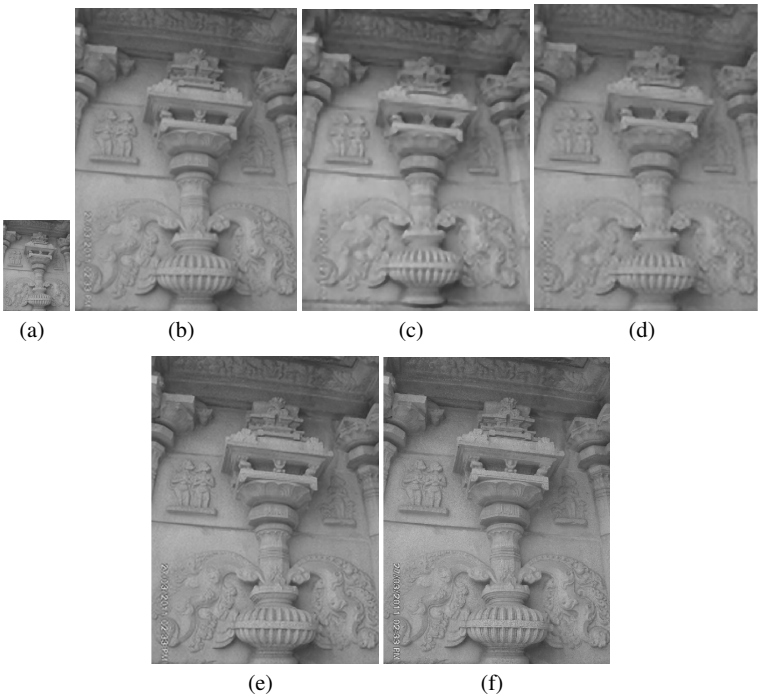


Fig. 8 SR Results for 4X : An art-piece captured at Hampi temple. (a) low-resolution; (b) Bicubic interpolation; (c) Super-resolved image using Kim and Kwon [13] approach; (d) Super-resolved image using SRCNN [5] approach; (e) Super-resolved image using proposed method; (f) Ground truth image.

the ground truth image. Figures 8(a, f) show the LR image of the art piece engraved on the wall of Hampi temple and the ground truth image. Figures 8(b) and 8(e) correspond to the bicubically interpolated image and the super-resolved image by our approach, respectively. Figures 8(c, d) correspond to the SR image using Kim et al.'s [13] and SRCNN [5] approach, respectively. We observe that the high-frequency details have been preserved well in the image shown in figure 8(e) indicating that our method performs better for magnification factor 4 as well even when we do not train the network separately. One more result for a magnification factor of 4 wherein the experiment is conducted on the natural image is shown in figure 9. Here also, we observe sharper details in figure 9(e), whereas one can clearly see a blurring in figures 9(b, c). This validates our claim that learning a mapping between LR and HR images using CNN does help in improving the spatial resolution. Looking at the perceptual quality we observe that our method gives visually similar results as in [5]. However, our method has the advantage over [5] in the following ways: (1) elimination of bicubic interpolation during the training phase, (2) computationally more efficient since a single trained network for a factor of 2 is enough to reconstruct a

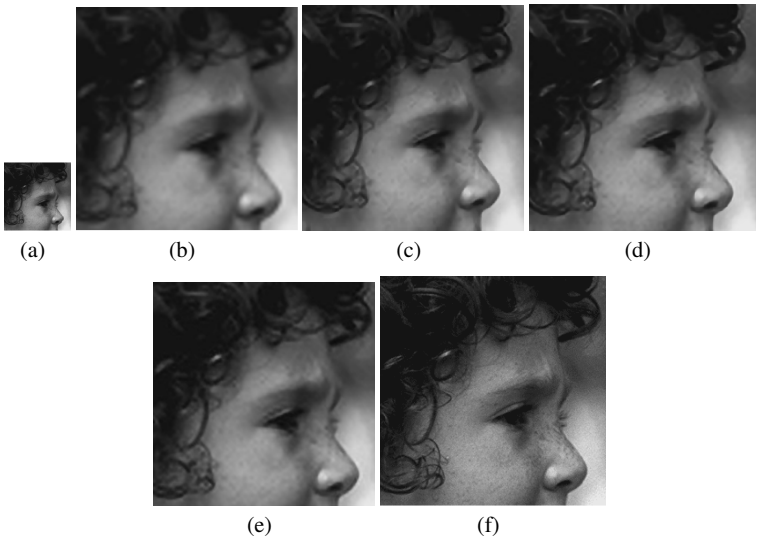


Fig. 9 SR Results for 4X : Head. (a) Low-resolution image; (b) Bicubically interpolated image; (c) Super-resolved image using Kim and Kwon [13] approach; (d) Super-resolved image using Dong et al. [5] approach; (e) Super-resolved image using proposed method; (f) Ground truth image.

super-resolved image for higher magnification factors and (3) error computation for backpropagation is computationally less involved during training.

Finally, we show one result for a large magnification factor of 8 by considering another image from Hampi site. Similar to super-resolving by a factor of 4, here also the LR image is obtained by downsampling the HR image, but by a factor of 8 which is then super-resolved in stages. The LR and the ground truth image are shown in figures 10(a, d), while the SR result using our approach is shown in figure 10(c). Figure 10(b) displays the bicubically interpolated image of the LR in figure 10(a). One can clearly see the difference between bicubic and our approach. The image interpolated bicubically appears blurred whereas the image super-resolved using our approach compares well with the ground truth.

To show the edge over the traditional interpolation techniques and existing learning based methods, we show the quantitative comparison in terms of peak signal to noise ratio (PSNR) in table 1. PSNR is computed as follows. Given a noise-free m x n monochrome image I (true image) and its approximation K (reconstructed image), the mean squared error (MSE) between these two images is defined as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2. \tag{6}$$

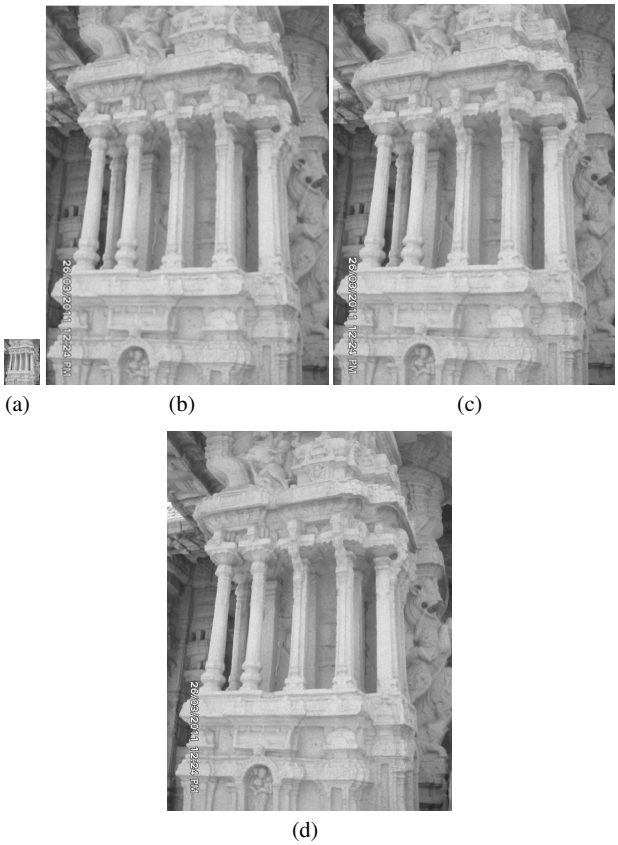


Fig. 10 SR Results for 8X : An image consisting of the musical pillars at Hampi temple. (a) Low-resolution image; (b) Bicubically interpolated image; (c) super-resolved image using proposed method; (d) Ground truth image.

Using equation (6) the peak signal to noise ratio (PSNR) measured in dB is then defined as

$$PSNR = 10 * log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{7}$$

where MAX corresponds to the maximum value of the noise-free image I. We also compare our results with the bicubic interpolation using the structural similarity measure (SSIM) index as shown in table 2. A higher value of SSIM indicates higher similarity of the SR image with the ground truth HR image. From table 1 and 2, we conclude that in addition to the perceptual betterment, we also see an improvement in PSNR and SSIM for the proposed approach indicating the usefulness of deep learning framework for super-resolution.

6 Conclusion

In this chapter, we addressed the problem of super-resolution from image processing and machine learning perspective. We have proposed a deep learning based technique for super-resolution which aims at restoring details of a high-resolution image from the transformation learned by analyzing the spatial relationship during the training phase. We learn this transformation using a deep convolutional neural network. We have considered the limitations of classical multi-image SR and presented an approach showing results for higher magnification factors. The CNN filters and sub-image sizes play a major role in obtaining better results. The results obtained for the grayscale (luminance channel) show perceptual as well as quantifiable improvements over the digital zoom performed using bicubic interpolation.

Acknowledgements The authors would like to thank NVIDIA Corporation for providing the TITAN X GPU for the academic research. They are also thankful to their colleagues Dr. Milind G. Padalkar, Meet H. Soni, Ketul D. Parikh and Surabhi D. Sohoney for sharing their pearls of wisdom with them during the course of this research. The authors are also immensely grateful to the reviewers of the book for their comments on the earlier versions of the manuscript.

References

[1] Baker S, Kanade T (2002) Limits on super-resolution and how to break them. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(9):1167–1183

[2] Bose N, Kim H, Valenzuela H (1993) Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes. In: Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on, IEEE, vol 5, pp 269–272

[3] Burger HC, Schuler CJ, Harmeling S (2012) Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. arXiv preprint arXiv:12111544

[4] Cui Z, Chang H, Shan S, Zhong B, Chen X (2014) Deep network cascade for image super-resolution. In: Computer Vision–ECCV 2014, Springer, pp 49–64

[5] Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: Computer Vision–ECCV 2014, Springer, pp 184–199

[6] Farsiu S, Robinson D, Elad M, Milanfar P (2004) Advances and challenges in super-resolution. International Journal of Imaging Systems and Technology 14(2):47–57

[7] Fattal R (2007) Image upsampling via imposed edge statistics. In: ACM Transactions on Graphics (TOG), ACM, vol 26, p 95

[8] Freedman G, Fattal R (2011) Image and video upscaling from local self-examples. ACM Transactions on Graphics (TOG) 30(2):12

[9] Freeman WT, Jones TR, Pasztor EC (2002) Example-based super-resolution. Computer Graphics and Applications, IEEE 22(2):56–65

[10] Gajjar PP, Joshi MV (2010) New learning based super-resolution: use of dwt and igmrf prior. IEEE Transactions on Image Processing 19(5):1201–1213

[11] Glasner D, Bagon S, Irani M (2009) Super-resolution from a single image. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, pp 349–356

[12] Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural computation 18(7):1527–1554

[13] Kim KI, Kwon Y (2008) Example-based learning for single-image super-resolution. In: Pattern Recognition, Springer, pp 456–465

[14] Kim SP, Su WY (1993) Recursive high-resolution reconstruction of blurred multiframe images. Image Processing, IEEE Transactions on 2(4):534–539

[15] Li FF, Karpathy A, Johnson J (2016) Cs231n: Convolutional neural networks for visual recognition. Accessed: 2016-06-28

[16] Lin Z, Shum HY (2004) Fundamental limits of reconstruction-based super-resolution algorithms under local translation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(1):83–97

[17] Park SC, Park MK, Kang MG (2003) Super-resolution image reconstruction: a technical overview. Signal Processing Magazine, IEEE 20(3):21–36

[18] Protter M, Elad M, Takeda H, Milanfar P (2009) Generalizing the nonlocal-means to super-resolution reconstruction. Image Processing, IEEE Transactions on 18(1):36–51

[19] Rhee S, Kang MG (1999) Discrete cosine transform based regularized high-resolution image reconstruction algorithm. Optical Engineering 38(8):1348–1356

[20] Sun J, Sun J, Xu Z, Shum HY (2008) Image super-resolution using gradient profile prior. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, pp 1–8

[21] Timofte R, De Smet V, Van Gool L (2013) Anchored neighborhood regression for fast example-based super-resolution. In: The IEEE International Conference on Computer Vision (ICCV)

[22] Tsai R, Huang TS (1984) Multiframe image restoration and registration. Advances in computer vision and Image Processing 1(2):317–339

[23] Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning, ACM, pp 1096–1103

[24] Yang J, Wright J, Huang TS, Ma Y (2010) Image super-resolution via sparse representation. Image Processing, IEEE Transactions on 19(11):2861–2873