Cogn Comput (2018) 10:165–178
https://doi.org/10.1007/s12559-017-9512-2

Clustering-Oriented Multiple Convolutional Neural Networks for Single Image Super-Resolution

Peng Ren[1] · Wenjian Sun[1] · Chunbo Luo[2] · Amir Hussain[3]

Received: 10 January 2017 / Accepted: 21 September 2017 / Published online: 4 October 2017

Abstract In contrast to the human visual system (HVS) that applies different processing schemes to visual information of different textural categories, most existing deep learning models for image super-resolution tend to exploit an indiscriminate scheme for processing one whole image. Inspired by the human cognitive mechanism, we propose a multiple convolutional neural network framework trained based on different textural clusters of image local patches. To this end, we commence by grouping patches into K clusters via K-means, which enables each cluster center to encode image priors of a certain texture category. We then train K convolutional neural networks for super-resolution based on the K clusters of patches separately, such that the multiple convolutional neural networks comprehensively capture the patch textural variability. Furthermore, each convolutional neural network characterizes one specific texture category and is used for restoring patches belonging to the cluster. In this way, the texture variation within a whole image is characterized by assigning local patches to their closest cluster centers, and the super-resolution of each local patch is conducted via the convolutional neural network trained by its cluster. Our proposed framework not only exploits the deep learning capability of convolutional neural networks but also adapts them to depict texture diversities for super-resolution. Experimental super-resolution evaluations on benchmark image datasets validate that our framework achieves state-of-the-art performance in terms of peak signal-to-noise ratio and structural similarity. Our multiple convolutional neural network framework provides an enhanced image super-resolution strategy over existing single-mode deep learning models.

Keywords Clustering · Convolutional neural networks · Single image super-resolution

✉ Peng Ren
pengren@upc.edu.cn

[1] College of Information and Control Engineering, China University of Petroleum (East China), Qingdao, China

[2] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, UK

[3] Division of Computing Science and Maths, School of Natural Sciences, University of Stirling, Stirling, UK

Introduction

In the literature of visual information processing, the problem of single image super-resolution has been extensively investigated for the purpose of restoring a high-resolution (HR) image from one single low-resolution (LR) image. This problem is ill-posed because it requires improving image resolution by generating details which are not captured by an LR image. Traditional techniques for lifting single image resolution is to apply various interpolation techniques to the LR image and thus generate new pixels between exiting pixels in terms of neighborhood averaging etc. Though being sufficiently efficient and able to increase image resolution at arbitrary scales, interpolation schemes exhibit limited capability of restoring image details. The reason for this shortcoming is that brute-force interpolations tend to ignore the prior knowledge regarding intrinsic textural characteristics and variabilities arising from images. Machine learning strategies, on the other hand, can provide a solution to single image super-resolution in terms of training a model to characterize textural details based on generic images. Various state-of-the-art machine learning strategies have been applied to single image super-resolution for learning prior image details from generic images. For example, Yang et al. [1, 2] introduced sparse representations, which learn coupled dictionaries from high and low resolution image patch pairs, to single image super-resolution. To overcome the shortcoming of bicubic interpolation in non-smooth region restoration, Yang et al. [3] divided the image feature space into a large set of subspaces and used sufficient training samples to fit linear regression functions for each subspace. However, linear regression functions exhibit limited representational power for high-frequency priors in image subspaces. To overcome this disadvantage, Timofte et al. [4] proposed anchored neighbor regression (ANR), which characterizes exemplar neighborhoods using ridge regression and then utilizes these neighborhoods to compute projections from LR patches to HR patches. In their subsequent work, Timofte et al. [5] proposed adjusted ANR (A+) based on ANR and simple linear regression functions. All these sparse coding strategies tend to learn image priors based on patches cropped from images rather than the whole images themselves, and thus result in a mapping function which associates a low-resolution patch with its corresponding high resolution patch. Additionally, different types of image priors have been comprehensively investigated for training an effective image restoration model. In this regard, Yang et al. [6] classified single image super-resolution strategies into four categories, i.e., (a) prediction models, (b) edge-based methods [7], (c) image statistical methods [8–10], and (d) patch-based (example based) methods [11–15]. Furthermore, Yang et al. [6] also observed that, in most cases, the patch-based methods achieved the best performance over alternative models. One common feature of these methods is that visual details, which are key to improving image restoration quality, are comprehensively learned and characterized in the training procedure. It is thus in both theoretical and practical terms that machine learning schemes turn out to be an effective approach for single image super-resolution.

Furthermore, deep learning schemes have recently gained significant success in broad areas such as image classification [16–18], object recognition [19–21], object detection [22], etc. The deep learning strategies arguably resemble the human brain mechanism in terms of hierarchical structures and neural connections. Key to the effectiveness of deep learning strategies is their capability to precisely characterize features with multi-layer representations. Additionally, most deep learning methods possess an end-to-end advantage which enables integrated procedures for learning. The merits of deep learning have enabled its broad exploitation for solving challenging problems in various research areas including image restoration. In the literature, Jain et al. [23] are among the first to develop convolutional neural networks (CNNs) for improving natural image quality in terms of denoising. Burger et al. [24] developed a different deep learning method by utilizing fully-connected multi-layer perceptron (MLP) neural networks to remove noises from images. Eigen et al. [25] proposed a variety of CNNs to remove dirts or rain patterns. These denoising problems, though concerned with restoring high-quality images, and thus highly related to image super-resolution, are in fact aimed at noise and disturbance elimination based on deep learning. Cui et al. [26] are among the pioneering group of researchers who address the specific image super-resolution problem in terms of deep learning. They proposed a so-called deep network cascade (DNC), which cascades multiple stacked collaborative local auto-encoders for image super-resolution. The auto-encoder and self-similarity search process in each layer of the cascade requires independent optimization, which does not render the DNC as an attractive end-to-end method. Dong et al. [27, 28] proposed a deep learning algorithm termed super resolution using convolutional neural networks (SRCNN). They exploit deep CNNs for learning an end-to-end mapping function between low and high-resolution images. Further, recently developed super-resolution deep learning models include the very deep convolutional networks (VDSR) [29] and deeply-recursive convolutional network (DRCN) [30]. By exploiting the representational power of deep models, SRCNN, VDSR, and DRCN achieve state-of-the-art performance and are considered as the most effective single image super resolution algorithms to date. On the other hand, Yang et al. [6] observed that in most cases, image priors played a more important role than super resolution algorithms themselves, in restoring high-quality images. Furthermore, they demonstrated that high-frequency details are one key factor contributing to accurate restoration, implying that patches with different types of textural characteristics make different contributions to restoring images. However, most deep learning-based super resolution methods indiscriminately conduct common processing procedures on different categories of patches. Specifically, they feed various patches indiscriminately into a deep model for training, and in turn use the trained deep model indiscriminately in restoring various test patches for super resolution. Existing deep learning methods tend to ignore the variability of textural characteristics and are thus incapable of exhaustively exploiting the representational power of deep models.

To address the shortcoming of existing deep learning methods for disregarding textural variability across different types of patches, we propose and develop a single image super resolution framework termed clustering-oriented multiple convolutional neural networks. Our idea is highly inspired by the cognitive mechanism of the human visual system (HVS), which applies different processing schemes to visual information of different textural categories [31, 32]. For example, the human visual system tends to pay more attention to the rich textured parts than smooth parts. The cognitive modes for processing different categories of visual parts are thus different in the human brain and this multi-mode mechanism proves an effective approach to human visual processing. Motivated by this mechanism, we proposed to train multiple convolutional neural networks based on image patches from different textural categories, separately. We first conduct a clustering analysis of image local patches and group them into different categories. We then train multiple CNNs to learn image priors based on exemplar patches from different categories. Specifically, we train a convolutional neural network for super-resolution based on each cluster of patches, and thus, obtain the same total number of trained CNNs as the number of clusters. Each CNN captures certain image priors within a texture category, and multiple CNNs are potentially discriminative between patches of different categories. Therefore, our framework generalizes the capability of CNNs for characterizing the variability of image priors. Experimental evaluation of our approach demonstrates its state-of-the-art super-resolution performance on benchmark image datasets.

The rest of this paper is organized as follows: "A Review of Super-Resolution Convolutional Neural Network" section presents a review of the related state-of-the-art super-resolution convolutional neural network. This is followed by a description of our proposed framework based on clustering-oriented multiple convolutional neural networks in "Clustering-Oriented Multiple Convolutional Neural Networks" section. Comparative experimental results are presented in "Experimental Evaluations" section. Finally, some concluding remarks and future work recommendations are given in "Conclusions and Future Work" section.

A Review of Super-Resolution Convolutional Neural Network

The recently proposed, state-of-the-art super-resolution convolutional neural network (SRCNN) [27, 28] is most closely related to our proposed framework, and is thus briefly reviewed in this section. The SRCNN has a three-layer structure as illustrated in Fig. 1. The first layer convolves one whole low-resolution image with local filters for generating low-resolution feature maps. The second convolution layer operates as a nonlinear mapping of low-resolution feature maps to high resolution feature maps. The final convolution layer integrates the resulting high-resolution features, and reconstructs a high-resolution image.

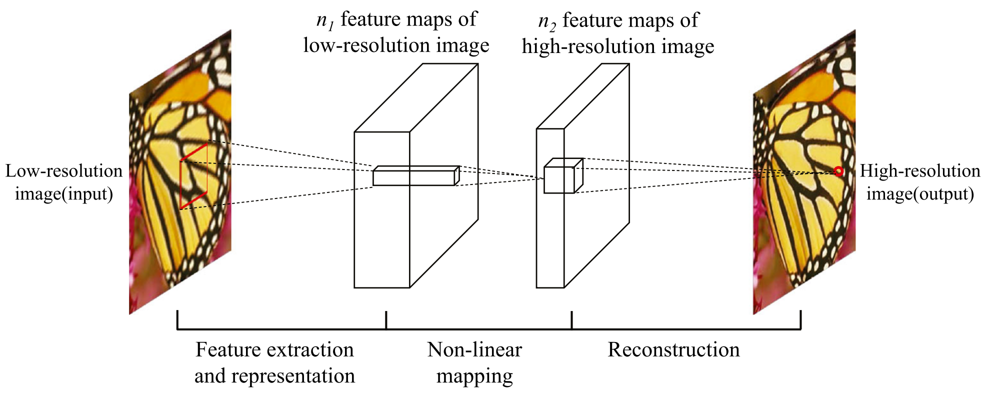
In contrast to the SRCNN, which trains a single convolutional neural network over whole images, we train multiple convolutional neural networks based on local patches cropped from whole images. Furthermore, we train each CNN based on patches grouped in a cluster, representing an associated category of texture features. As a result, our proposed framework characterizes both local features and also learns texture variation for super-resolution. Our framework is described in the next section.

Clustering-Oriented Multiple Convolutional Neural Networks

This section presents our single image super-resolution framework based on clustering-oriented multiple convolutional neural networks. In the training stage, K convolutional neural networks are trained on the basis of K image patch clustering centers separately. In the inference stage, each patch from a low resolution image is restored by the convolutional neural network associated with the clustering center of the patch. All restored patches thus obtained form one whole high-resolution image.

Reforming Training Data

In contrast to the super-resolution convolutional neural network [27] which feeds whole images to train one CNN, we feed image patches into CNNs for training. To this end, we reform training data through cropping local patches from


n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
Low-resolution image(input)
High-resolution image(output)
Feature extraction and representation
Non-linear mapping
Reconstruction
Fig. 1 The diagram for SRCNN

whole high-resolution images, blurring and downscaling patches, and computing architectural residuals. An outline of these procedures for reforming training data is depicted in Fig. 2 and described below.

We crop high-resolution (HR) patches from each high-resolution image in the training dataset. The HR patches are required to be cropped out in an overlapping manner with equal intervals such that the whole image is fully covered. For one HR patch P_H, we first blur it through a Gaussian convolution and then downsample it:

$$P_R = (P_H * G) \downarrow s \tag{1}$$

where G, $*$, and $\downarrow s$ denote the Gaussian kernel, the convolution operation, and the downsampling operation, respectively. The Gaussian convolution reduces high-frequency details and the downsampling operation further downgrades the patch resolution. The reformed patch P_R thus obtained is used to mimic a patch cropped from the low resolution image that corresponds to the high-resolution image producing P_H. The reformed patches form the input data for training multiple CNNs.

On the other hand, following most single image super resolution strategies, we exploit the differences between one HR and its corresponding LR patch as targets for learning. Specifically, in order to generate the target P_T associated with P_R for training a CNN, we first upsample the reformed patch P_R via bicubic interpolation, making it the same size as the HR patch P_H, and then subtract the upscaled patch from P_H as follows:

$$P_T = P_H - (P_R \uparrow s) \tag{2}$$

where $\uparrow s$ denotes the upsampling operation. The target patch P_T characterizes the difference between the HR patch P_H and the reformed (and also LR) patch P_R. As P_H is believed to have more high-frequency details than P_R, P_T tends to capture the architectural high frequency leftovers in the patch. P_T is thus referred to as architectural residual and also acts as the learning target.

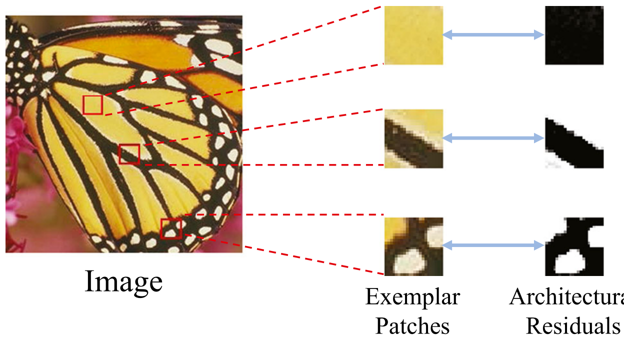


Fig. 3 Several exemplar patches along with their architectural residuals

The tuple $\{P_R, P_T\}$ forms an $\{input, target\}$ training data pair. All such tuples generated from HR training images result in a reformed dataset for training CNNs in our work.

Clustering Reformed Patches

In "Reforming Training Data" section, we transform HR images into $\{reformed\ patch, architectural\ residual\}$ pairs as the reformed training data. We observe that different patches exhibit various types of signatures in architectural residuals. This observation implies that different patches encode different amounts of high frequency visual details. Figure 3 illustrates several exemplar patches along with their architectural residuals extracted from one image. It is clear that they exhibit totally distinct characteristics in architectural residuals, revealing that they contain different amounts of high-frequency ingredients.

As Yang et al. [1] pointed out, images with richer high-frequency details play an important role in characterizing image priors, leading to our belief that patches from different texture categories represent various characteristics for super-resolution. In order to capture the texture variability over different patch texture categories, we perform clustering analysis on reformed patches and gain K cluster centers of patches, as illustrated in Fig. 4.

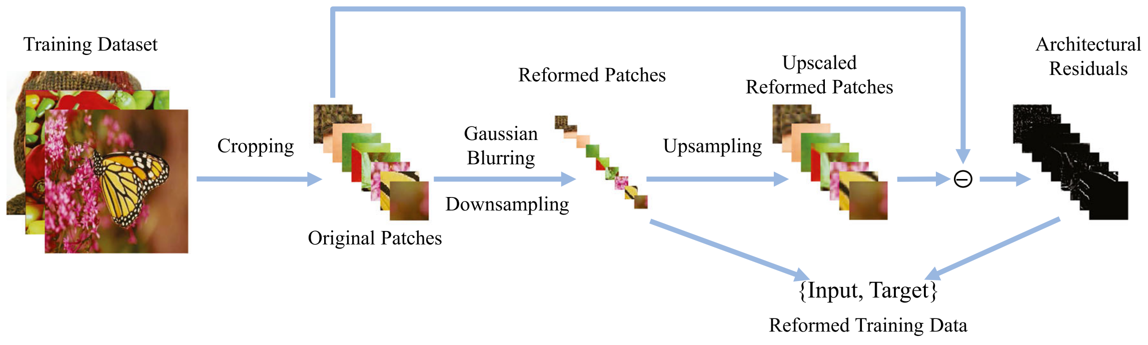


Fig. 2 Reforming training data

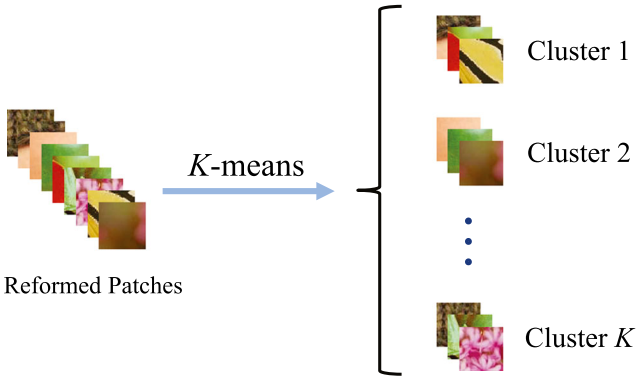


Fig. 4 Clustering the reformed patches

Assume that there are totally M_k patches, i.e., $P_{R_1}^{(k)}$, $\cdots$, $P_{R_{M^{(k)}}}^{(k)}$, in the kth cluster, then the cluster center is

$$P_C^{(k)} = \frac{1}{M^{(k)}} \sum_{m=1}^{M^{(k)}} P_{R_m}^{(k)}. \tag{3}$$

We attach the superscript (k) to the notations (e.g. $P_C^{(k)}$, $M^{(k)}$ and $P_{R_m}^{(k)}$) which are associated with the kth cluster. Each cluster (center) characterizes a certain category of image textural priors which are distinct from one another. To enable an efficient implementation, we exploit K-means for clustering the reformed patches in our framework, but alternative clustering strategies [33] can be used as a substitute.

Following the resultant K clusters, the reformed training dataset is divided into K reformed subsets of patches, which are separately used for training multiple CNNs, as described in the next subsection.

Training Multiple Convolutional Neural Networks

As observed in "Clustering Reformed Patches" section, a range of textural variations exist over patches from different categories. In the context of CNN-based learning, this observation further implies that training one single CNN indiscriminately for various images (e.g., [27]) potentially limits the capability of CNN in fully characterizing textural variability for super-resolution. To address this single CNN limitation, we propose to train multiple CNNs for different textural characteristics, by training K CNNs based on the K clusters of reformed patches separately.

Specifically, we employ a three-layer structure to construct a convolutional neural network for every reformed subset of patches (i.e. each cluster of patches). The flow diagram for training the kth convolutional neural network is depicted in Fig. 5.

The kth reformed subset of patches for training the kth CNN consists of all $\{reformed\ patch, architectural\ residual\}$ pairs with respect to the kth cluster. One reformed patch $P_R^{(k)}$ and its associated architectural residual $P_T^{(k)}$ form the input and target for training the CNN, respectively. The reformed patch $P_R^{(k)}$ is obtained through blurred and downsampled from a HR patch and is thus smaller in size than the architectural residual $P_T^{(k)}$. We upscale the $P_R^{(k)}$ to the same size as $P_T^{(k)}$ in terms of bicubic interpolation. This operation is just a size modification and nevertheless does not provide any high frequency details to the upscaled patch $P_U^{(k)} = P_R^{(k)} \uparrow s$, since no image priors are employed in the upsampling.

The first layer operation for training the CNN is formulated as follows:

$$F_1(P_U^{(k)}) = Relu(W_1^{(k)} * P_U^{(k)} + B_1^{(k)}) \tag{4}$$

where $W_1^{(k)}$ and $B_1^{(k)}$ represent the trainable kernels and biases respectively. The size of $W_1^{(k)}$ is $n_1 \times c \times k_1 \times k_1$, where n_1 is the number of kernels, c and k_1 are the numbers of image channels and the kernel size, respectively. $W_1^{(k)}$ contains n_1 kernels which are used for convolving an input patch, and the size of each kernel is $c \times k_1 \times k_1$. $B_1^{(k)}$ is of dimension $n_1 \times 1$ with each element appending one kernel. $Relu(x) = max(0, x)$ denotes a rectifier activation function [34] as illustrated in Fig. 6.

In terms of the same notational system, the second layer operation is represented as follows:

$$F_2(P_U^{(k)}) = Relu(W_2^{(k)} * F_1(P_U^{(k)}) + B_2^{(k)}) \tag{5}$$

where $F_1(P_U^{(k)})$ is the first layer output, $W_2^{(k)}$ and $B_2^{(k)}$ represent the trainable kernels and biases in the second layer, respectively. The dimensions of $W_2^{(k)}$ and $B_2^{(k)}$ are $n_1 \times k_2 \times k_2 \times n_2$ and $n_2 \times 1$, respectively.

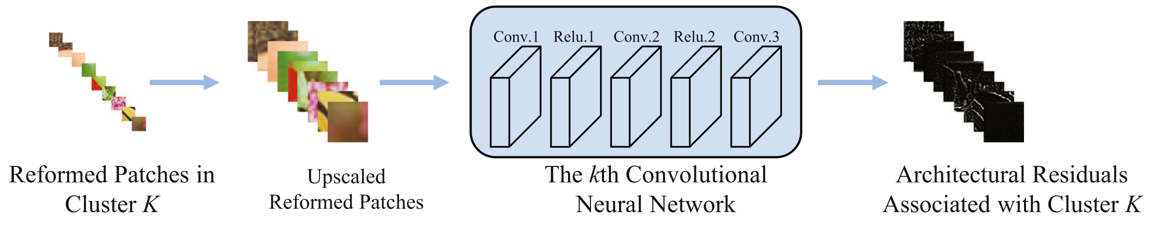

Conv.1
Relu.1
Conv.2
Relu.2
Conv.3
Reformed Patches in Cluster K
Upscaled Reformed Patches
The kth Convolutional Neural Network
Architectural Residuals Associated with Cluster K

Fig. 5 Training the kth convolutional neural network

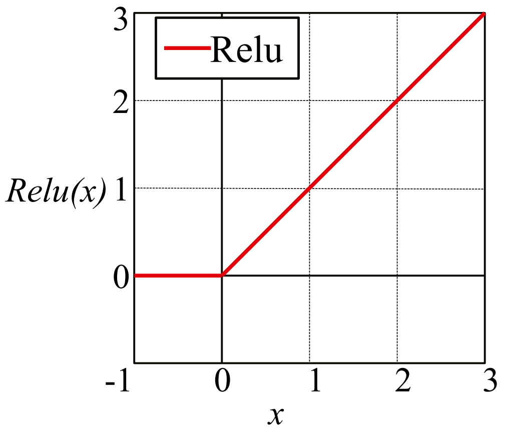


Fig. 6 Rectified linear unit, Relu

In the third layer, we do not operate the *Relu* function and have:

$$F_3(P_U^{(k)}) = W_3^{(k)} * F_2(P_U^{(k)}) + B_3^{(k)} \tag{6}$$

where $F_2(P_U^{(k)})$ is the second layer output, $W_3^{(k)}$ and $B_3^{(k)}$ represent the trainable kernels and biases in the third layer, respectively. The dimensions of $W_3^{(k)}$ and $B_3^{(k)}$ are $n_2 \times k_3 \times k_3 \times c$ and $c \times 1$, respectively. $F_3(P_U^{(k)})$ is third layer yet final output that will be exploited in comparison against the target $P_T^{(k)}$ for training the CNN.

Unlike traditional CNN structures that typically consist of a sequence of convolution and pooling operation pairs, we do not perform pooling after convolution. Most existing CNNs tend to pool the convolution layer outputs to obtain condensed features for the purpose of achieving accurate pattern recognition. In contrast to the feature condensation scenario, super-resolution tasks require enhanced visual details to be added to the original low resolution image. Therefore, we do not apply pooling operation in our CNN construction in order to avoid reducing visual details.

To optimize the trainable parameters $\Phi^{(k)} = \{W_1^{(k)}, B_1^{(k)}; W_2^{(k)}, B_2^{(k)}; W_3^{(k)}, B_3^{(k)}\}$ for the kth convolutional neural network, we measure the loss between the CNN final output $F_3\left(P_U^{(k)}; \Phi^{(k)}\right)$ and the target, i.e. the architectural residual P_T. We utilize mean squared error (MSE) over all samples in the kth reformed subset as the loss function for the kth CNN:

$$L\left(\Phi^{(k)}\right) = \frac{1}{M^{(k)}} \sum_{m=1}^{M^{(k)}} \| F_3\left(P_{U_m}^{(k)}; \Phi^{(k)}\right) - P_{T_m}^{(k)} \|^2 \tag{7}$$

where m denotes cardinality of the kth reformed subset of patches. We employ stochastic gradient descent with the standard back propagation algorithm [35] for minimizing the loss function (7). Specifically, we update the kernel matrices and biases as follows:

$$\begin{aligned} \Delta_{1;j+1}^{(k)} &= \gamma \cdot \Delta_{1;j}^{(k)} + \alpha \cdot \frac{\partial L}{\partial W_{l;j}^{(k)}} \\ \Delta_{2;j+1}^{(k)} &= \gamma \cdot \Delta_{2;j}^{(k)} + \alpha \cdot \frac{\partial L}{\partial B_{l;j}^{(k)}} \\ W_{l;j+1}^{(k)} &= W_{l;j}^{(k)} + \Delta_{1;j+1}^{(k)} \\ B_{l;j+1}^{(k)} &= B_{l;j}^{(k)} + \Delta_{2;j+1}^{(k)} \end{aligned} \tag{8}$$

where α, γ, $l \in \{1, 2, 3\}$, j are the learning rate, momentum, layer index, and iteration index, respectively. $\Delta_1^{(k)}$ and $\Delta_2^{(k)}$ reflect the current velocity for updating $W_l^{(k)}$ and $B_l^{(k)}$, respectively.

We train all the K CNNs based on K reformed training subsets separately, according to the procedures described in this subsection.

Super-Resolution

In this subsection, we describe how to perform super-resolution on a low resolution image based on K trained CNNs. Figure 7 illustrates the diagram of the proposed super-resolution scheme.

Given a low-resolution image, we first divide it into overlapping patches and then upsample every patch in terms of bicubic interpolation. The upscaled LR patches are required to have the same size as the architectural residuals. For an upscaled LR patch P_I, we first assign it to one of the K clusters obtained in "Clustering Reformed Patches" section by seeking a minimum Euclidean distance between P_I and the K cluster centers:

$$k^* = \arg\min_{k=1}^{K} \| P_I - P_c^{(k)} \|^2. \tag{9}$$

Based on Eq. 9, the patch P_I is assigned to the cluster k^*. This implies that P_I has textural resemblance with patches in the cluster k^*. As the k^*th CNN was trained specifically to capture the image priors reflected by patches in the cluster k^*, we use the k^*th CNN for performing super-resolution on P_I. We feed P_I into the k^*th trained CNN with the optimized parameter set $\Phi^{(k^*)}$ and obtain the super-resolution patch P_O as follows:

$$P_O = P_I + F_3\left(P_I^{(k^*)}; \Phi^{(k^*)}\right). \tag{10}$$

For all upscaled LR patches extracted from the LR image, we carry out the same super-resolution procedures as described in Eqs. 9 and 10. We sort all the restored patches thus obtained in the same order as their corresponding LR patches in the LR image, and form the final super-resolution image. The overlapped regions are computed by averaging the relevant parts of neighboring patches.

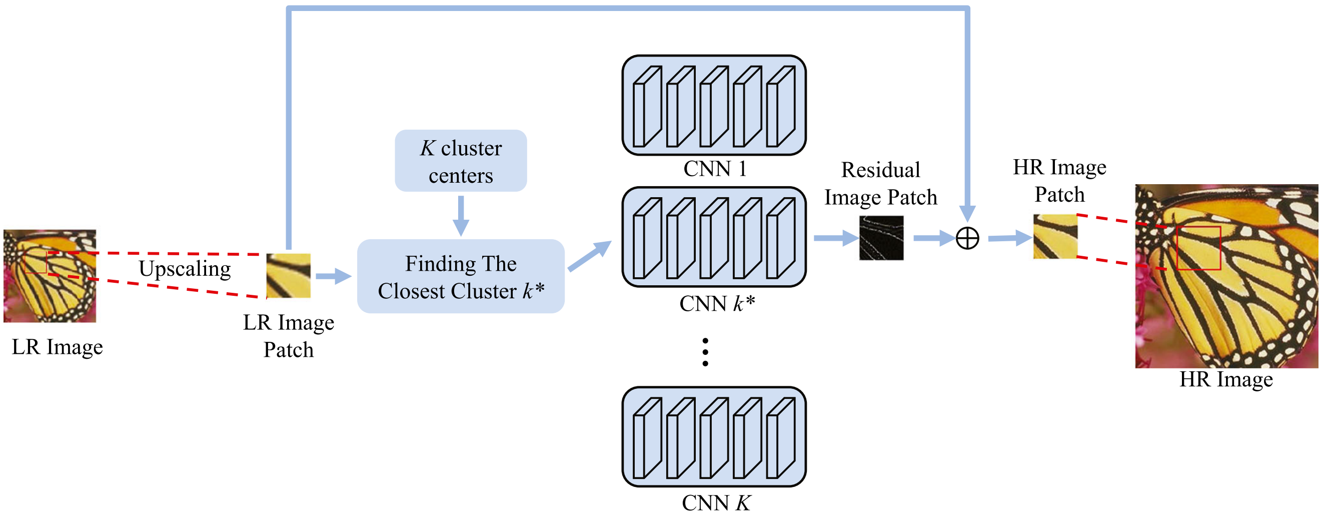


Fig. 7 Diagram of proposed super-resolution scheme based on trained multiple CNNs

Discussions

One major goal of our work is to develop a new strategy for increasing the super-resolution power of SRCNN. We thus construct the individual CNNs in our overall framework following the structure of SRCNN that consists of three layers. The major parameters of SRCNN are kernels and biases. Let k_1, k_2, and k_3 denote the sizes of kernels for the first, second, and third layers, respectively. Let n_1 and n_2 denote the numbers of kernels for the first and second layers, respectively. The number of kernels for the third layer is one. Additionally, a scalar bias is associated with each kernel. Therefore, the total number of parameters for SRCNN is $(k_1^2 + 1)n_1 + (k_2^2 + 1)n_2 + k_3^2 + 1$. Specifically, SRCNN sets the parameter values as follows $k_1 = 9$, $k_2 = 5$, $k_3 = 5$, $n_1 = 128$ and $n_2 = 64$ [28]. In this case, the exact total number of parameters for SRCNN is 12,186. Our framework contains K SRCNN structures and thus has $[(k_1^2 + 1)n_1 + (k_2^2 + 1)n_2 + k_3^2 + 1]K$ parameters. Following the same specific setting as SRCNN, our framework has 36,558 parameters.

It is obvious that the complexity of our model is K times of that of SRCNN, because it adopts K CNNs with each having the same architecture with SRCNN. Though our model complexity is larger than SRCNN, it is acceptable because its complexity follows a linear growth with respect to the number of CNNs and does not incur exponentially increased overheads. On the other hand, in contrast to the deep super-resolution models such as very deep convolutional networks (VDSR) [29] which has 20 weight layers and deeply recursive convolutional network (DRCN) [30] which has 20 convolution layers and 16 recursion layers, our three-layered model is not very deep. However, we improve the super-resolution ability of the CNN model by not making it deeper but expanding it "broader," i.e., training multiple three-layered CNNs based on categorized patch samples.

We compare the convergence rates of SRCNN and our framework, in terms of the training loss (7). The convergence curves with respect to the training epoch based on the 91-image dataset [2, 4] are given in Fig. 8. Though the individual CNNs in our framework and SRCNN share the common network structure, our method converges slightly more efficient than SRCNN especially in the first few training epochs. Unlike SRCNN, which indiscriminately learns super-resolution information from all visual categories, our method encourages each individual CNN to learn super-resolution information from one specific visual category. The coarsely categorized visual information exhibits less textural variability than general visual information, and thus enables efficient training processes for the class-specific CNNs in our framework.

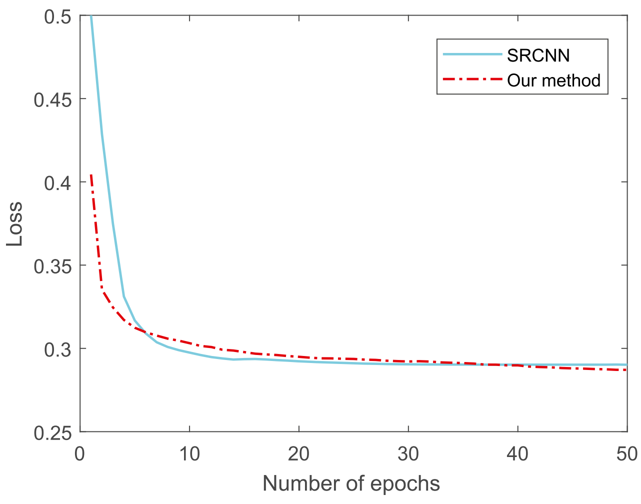


Fig. 8 Training convergence rate

The reasons for the effectiveness of our clustering-oriented multiple-CNN framework are twofold. First, the multiple CNNs are trained based on local patches rather than whole images. This strategy enables our framework to exhibit a more powerful local representational capability compared to a single CNN trained on whole images. Furthermore and more importantly, the multiple CNNs are separately trained based on different types of textures. This strategy enables multiple CNNs to better discriminate between different texture categories and thus enhances the effectiveness of individual CNNs for texture-specific super-resolution.

The proposed clustering-oriented multiple CNN model can be considered as a coarse-to-fine strategy. The K-means clustering procedure coarsely divides all patches into K general textural groups. Subsequently, fine textures within each coarsely categorized textural group are learned by an individual CNN for super-resolution. The number of texture classes should be carefully set in order to not only avoid learning trivial or wrong details from poorly clustered patches leading to similar effects of over-fitting but also enhance the learning capability of CNNs. A larger number of clusters do not necessarily increase the overall super-resolution performance. More specifically, the basic K-means is a straightforward clustering method for coarse textural categorization. If we increase the number of texture classes (i.e. number of clusters) K and try to use K-means to assign patches into finer textural categories, it may incur certain misclassification in the fine texture scales. On the other hand, CNNs have strong representational power and play the role of learning fine textures for super-resolution in our framework. If we set K to be a large value and use the less representational K-means method to characterize fine textures, the functionality of CNNs for learning fine textures would be neutralized. Furthermore, the training of CNNs would be misled by K-means misclassified samples. Therefore, a small class number K is appropriate for the clustering procedure and enables effective performance for the overall framework. Our extensive experiments confirm this observation and reveal that the current method achieves an optimal trade-off between performance and model complexity.

Experimental Evaluations

We conduct experimental evaluation of our proposed framework and carry out empirical comparisons with alternative state-of-the-art methods. We first describe our experimental parameter settings. We then perform a systematic analysis of our framework by varying values of several key factors. We finally carry out empirical comparisons between our method and alternative methods, and demonstrate the advantages of our proposed framework.

Experimental Settings

Based on our observation that the representational power of a convolutional neural network improves with respect to increasing diversity of training data, we carry out flip data augmentation to increase the number and diversity of training samples for each cluster. Specifically, following the experimental set-up used by state-of-the-art methods tested on benchmark datasets in [6, 28], we crop training patches from whole images, and employ the low resolution patch size 33 × 33 and the scale factors 2, 3 and 4. Here, the scale factor refers to the times of size/resolution increased for a low-resolution image. The downsampled patches are upscaled using bicubic interpolation.

For constructing individual convolutional neural networks, we adapt the SRCNN structure [27, 28]. Based on the fact that human vision is more susceptible to luminance, we transform all training RGB images into YCbCr color space and only apply super-resolution to the channel Y (luminance). The parameters c in the first and final layer are thus set to 1, and the CNN simply operates on the channel Y. Further, we utilize bicubic interpolation to enlarge the resolution of chrominance channels. The width of the Gaussian kernel for blurring patches is empirically set to 3 and the standard deviation σ is 1.6.

For the comparative experiments, we use a 91 image dataset [2, 4] and the Berkeley Segmentation Dataset and Benchmark [36] as training datasets. We use datasets $Set5$ [37], $Set14$ [38], and $BSD100$ [36] for testing the super-resolution performance. Finally, we adopt the peak signal-to-noise ratio (PSNR, dB) and structural similarity(SSIM) [39] for evaluating the super-resolution accuracy.

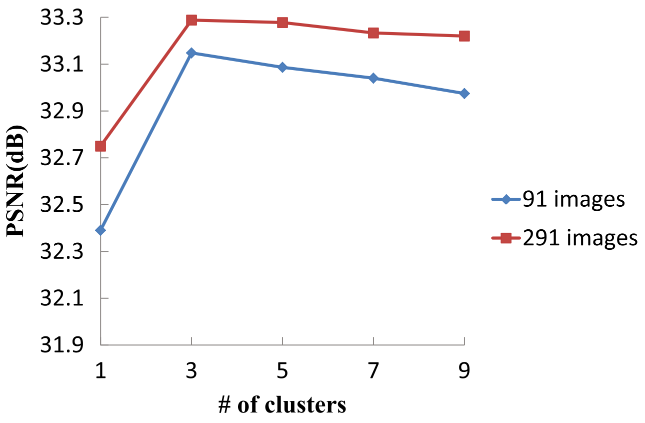


Fig. 9 Super-resolution performance subject to different numbers of training samples and different numbers of clusters

Table 1 Super-resolution performance subject to different kernel numbers and different kernel size

$PSNR(dB)/SSIM$ $n_1 \times n_2$ $k_1 \times k_2 \times k_3$	32×16	64×32	128×64
$9 \times 1 \times 5$	32.79/0.909	32.87/0.91	32.98/0.912
$9 \times 3 \times 5$	32.70/0.908	32.78/0.91	32.86/0.911
$9 \times 5 \times 5$	33.03/0.913	33.08/0.914	33.15/0.914

Systematic Empirical Analysis

In this subsection, we empirically evaluate the effectiveness of our framework under a range of settings. We commence by empirically evaluating it comparative performance with respect to different training sample numbers and cluster numbers. We train our framework using a small training dataset [2, 4] comprising 91 images. In an alternative comparative setup, we also train our framework using the 91 image dataset along with additional 200 images from the Berkeley Segmentation Dataset and Benchmark [36]. In both cases, different numbers of clusters are identified for evaluation. We examine the impact of number of clusters by categorizing the patches into $\{1, 3, 5, 7, 9\}$ clusters in terms of K-means. The $Set5$ dataset is used for testing and the PSNRs obtained are illustrated in Fig. 9.

It is clear from Fig. 9 that the case with larger number of training samples achieves better performance. Specifically, by introducing a larger number of training samples, our proposed method achieves higher PSNR in all cases compared to the case when trained by a smaller number of samples. On the other hand, though CNNs trained based on multiple clusters outperform those based on a single cluster, experiments with three clusters offer the best results over alternative numbers of clusters. This demonstrates that more accurate super-resolution can result from the use of multiple CNNs with a small number of K-means clusters.

We then evaluate the effects of convolutional kernel number and kernel size on super-resolution. Specifically, we conduct tests with respect to different kernel number $n_1 \times n_2$ combinations and different kernel size $k_1 \times k_2 \times k_3$ combinations. Here, n_1 and n_2 are numbers of kernels for the first and second layers, respectively. The number of kernels for the third layer is one. k_1, k_2, and k_3 are the kernel sizes for the first, second, and third layer, respectively. Table 1 shows the performance of our framework, trained based on the 91 image dataset [2, 4] for a scale factor 3 and with 3 clusters, and tested on the dataset $Set5$. We use the peak signal-to-noise ratio (PSNR, dB) and structural similarity (SSIM) as evaluation metrics. It is clear from Table 1 that the alternative settings result in slightly different performances. Specifically, the results show that different kernel numbers and sizes can slightly affect the super-resolution performance, with a generally larger number and size providing slightly better performance.

Table 2 The mean PSNR/SSIM of bicubic interpolation, NE+LLE [40], JOR [41], A+ [5], SRCNN [27, 28], FSRCNN [42], VDSR [29], DRCN [30], and ours on the datasets $Set5$, $Set14$, and $BSD100$ for scale factors 2, 3, and 4

Dataset	Scale factor	Bicubic	NE+LLE	JOR	A+	SRCNN	FSRCNN	VDSR	DRCN	Ours
Set5	2	33.66/0.9299	35.77/0.9490	–	36.54/0.9544	36.66/0.9542	37.00/0.9558	37.53/0.9587	37.63/0.9588	37.65/0.9662
	3	30.39/0.8682	31.84/0.8956	32.55/–	32.59/0.9088	32.75/0.9090	33.16/0.9140	33.66/0.9213	33.82/0.9226	33.29/0.9286
	4	28.42/0.8104	29.61/0.8402	30.19/–	30.28/0.8603	30.49/0.8628	30.71/0.8657	31.35/0.8838	31.53/0.8854	30.79/0.8893
Set14	2	30.23/0.8687	31.76/0.8993	–	32.28/0.9056	32.45/0.9067	32.63/0.9088	33.03/0.9124	33.04/0.9118	33.45/0.9257
	3	27.54/0.7736	28.6/0.8076	29.09/–	29.13/0.8188	29.3/0.8215	29.43/0.8242	29.77/0.8314	29.76/0.8311	29.70/0.8482
	4	26/0.7019	26.81/0.7331	27.26/–	27.32/0.7491	27.5/0.7513	27.59/0.7535	28.01/0.7674	28.02/0.7670	27.73/0.7813
BSD100	2	29.56/0.8431	–	–	31.21/0.8863	31.36/0.8879	31.80/0.9074	31.90/0.8960	31.85/0.8942	32.10/0.9020
	3	27.21/0.7385	27.85/–	28.17/–	28.29/0.7835	28.41/0.7863	28.60/0.8137	28.82/0.7976	28.80/0.7963	28.66/0.7959
	4	25.96/0.6675	26.47/–	26.74/–	26.82/0.7087	26.90/0.7101	26.98/0.7398	27.29/0.7251	27.23/0.7233	27.02/0.7173

Table 3 The PSNR/SSIM values of bicubic interpolation, NE+LLE, SC, A+, SRCNN, and ours on the dataset *Set*5 for scale factors 2, 3, and 4

Set5 images	Scale factor	Bicubic	NE+LLE	SC	A+	SRCNN	Ours
Baby	2	37.07/0.9525	38.33	–	38.52/0.9655	38.54/0.9659	39/0.9718
Bird		36.81/0.9720	40	–	41.12/0.9865	40.91/0.9857	42.48/0.9908
Butterfly		27.43/0.9148	30.38	–	32.01/0.9634	32.75/0.9640	34.46/0.9788
Head		34.86/0.8626	35.63	–	35.77/0.8866	35.72/0.8861	36.09/0.9116
Woman		32.14/0.9478	34.52	–	35.31/0.9697	35.37/0.9689	36.2/0.9780
Average	2	33.66/0.9299	35.77	–	36.55/0.9544	36.66/0.9542	37.65/0.9662
Baby	3	33.91/0.9050	35.06	35.04	35.21	35.25/0.9249	35.25/0.9336
Bird		32.58/0.9246	34.56	34.15	35.54	35.48/0.9539	36.21/0.9660
Butterfly		24.04/0.8186	25.75	26.17	27.24	27.95/0.9056	29.42/0.9399
Head		32.88/0.8013	33.6	33.58	33.77	33.71/0.8275	33.79/0.8603
Woman		28.56/0.8890	30.22	30.25	31.2	31.37/0.9287	31.77/0.9431
Average	3	30.39/0.8682	31.84	31.84	32.59	32.75/0.9090	33.29/0.9286
Baby	4	31.78/0.8583	32.99	–	33.28	33.13/0.8835	33.13/0.8944
Bird		30.18/0.8715	31.72	–	32.54	32.52/0.9095	32.91/0.9300
Butterfly		22.1/0.7326	23.38	–	24.42	25.46/0.8503	26.45/0.8979
Head		31.59/0.7560	32.24	–	32.52	32.44/0.7817	32.34/0.8155
Woman		26.46/0.8310	27.72	–	28.65	28.89/0.8842	29.1/0.9085
Average	4	28.42/0.8104	29.61	–	30.28	30.49/0.8628	30.79/0.8893

Empirical Comparisons with State-of-the-art Methods

In this subsection, we make quantitative and qualitative empirical comparisons between our framework and a number of alternative state-of-the-art methods. The latter include bicubic interpolation, neighbor embedding and local linear embedding method (NE+LLE) [40], sparse coding (SC) [2], jointly optimized regressors (JOR) [41], adjusted anchored neighborhood regression (A+) [5], super-resolution convolutional networks (SRCNN) [27, 28], fast super-resolution convolutional networks (FSRCNN) [42], very deep convolutional networks (VDSR) [29], and deeply recursive convolutional network (DRCN) [30].

Table 2 shows the means of PSNR/SSIM values of alternative methods on datasets $Set5$, $Set14$, and $BSD100$ for scale factors 2, 3, and 4. Here, we use 291 images for

Table 4 The PSNR/SSIM values of bicubic interpolation, NE+LLE, SC, A+, SRCNN, and ours on the dataset *Set*14 for scale 3

Set14 images	Scale factor	Bicubic	NE+LLE	SC	A+	SRCNN	Ours
Baboon	3	23.21/0.5423	23.55	23.33	23.62	23.67/0.6091	23.79/0.6541
Barbara		26.25/0.7554	26.74	26.7	26.47	26.55/0.7850	26.61/0.7993
Bridge		24.4/0.6452	24.98	24.95	25.17	25.24/0.7117	25.49/0.7511
Coastguard		26.55/0.6181	27.07	27.12	27.27	27.36/0.6714	27.41/0.7167
Comic		23.12/0.7039	23.98	24.07	24.38	24.55/0.7887	24.84/0.8335
Face		32.82/0.7995	33.56	33.53	33.76	33.72/0.8272	33.78/0.8554
Flowers		27.23/0.8041	28.38	28.62	29.05	29.26/0.8563	29.66/0.8844
Foreman		31.18/0.8995	33.21	31.05	34.3	33.89/0.9304	36.31/0.9534
Lenna		31.68/0.8597	33.01	33.06	33.52	33.67/0.8873	33.76/0.9005
Man		27.01/0.7500	27.87	27.96	28.28	28.42/0.8030	28.59/0.8298
Monarch		29.43/0.9212	30.95	31.35	32.14	32.81/0.9494	33.83/0.9592
Pepper		32.390.8675	33.8	31.73	34.74	34.71/0.8901	35.07/0.9057
ppt3		23.71/0.8786	24.94	25.13	26.09	27.04/0.9413	27.24/0.9556
Zebra		26.63/0.7931	28.31	28.65	28.98	29.29/0.8527	29.48/0.8769
Average	3	27.54/0.7741	28.6	28.38	29.13	29.3/0.8215	29.7/0.8482

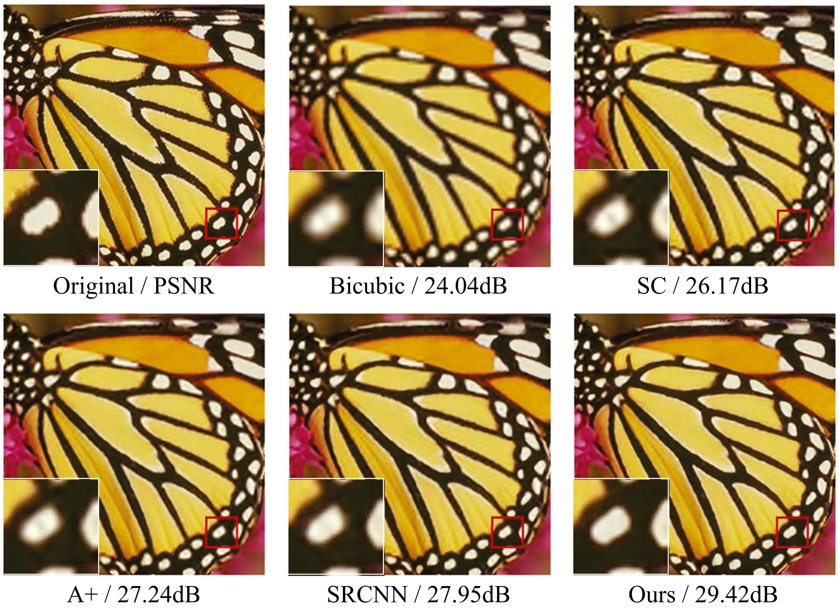


Fig. 10 Super-resolution for the "butterfly" image from *Set*5 with scale factor 3

training our model. We can see that our method achieves comparable performance with VDSR and DRCN and outperforms all the other comparison methods. Here, it is very interesting to compare our method with SRCNN (along with its variant FSRCNN), VDSR, and DRCN since they are all deep learning based models. Our method systematically improves SRCNN (as described in "Clustering-Oriented Multiple Convolutional Neural Networks" section), and the experimental results also validate that our method outperforms SRCNN in terms of both PSNR and SSIM on all the three datasets *Set*5, *Set*14, and *BSD*100. FSRCNN structurally improves SRCNN to a compact hourglass-shape CNN, which exhibits greater representational power than the original structure. Though our multiple CNN framework is composed of original SRCNNs without the structural improvements, it still outperforms FSRCNN in all cases except a few SSIM results on the BSD100 dataset. In this scenario, it is anticipated that our multiple CNN framework composed of FSRCNNs would result in even better performance. Furthermore, we observe that though our method is inferior to VDSR and DRCN for the scale factors 3 and 4 in terms of PSNR on the datasets *Set*5 and *Set*14, its PSNR outperforms VDSR and DRCN for the scale factor 2. On the other hand, our method outperforms VDSR and DRCN for all the three scale factors in terms of SSIM on the datasets *Set*5 and *Set*14. On the bigger dataset *BSD*100,

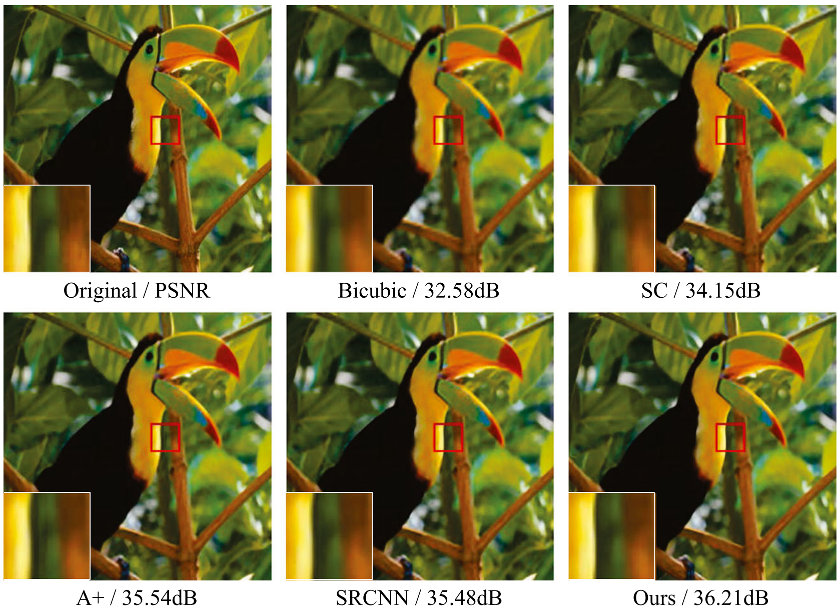


Fig. 11 Super-resolution for the "bird" image from *Set*5 with scale factor 3

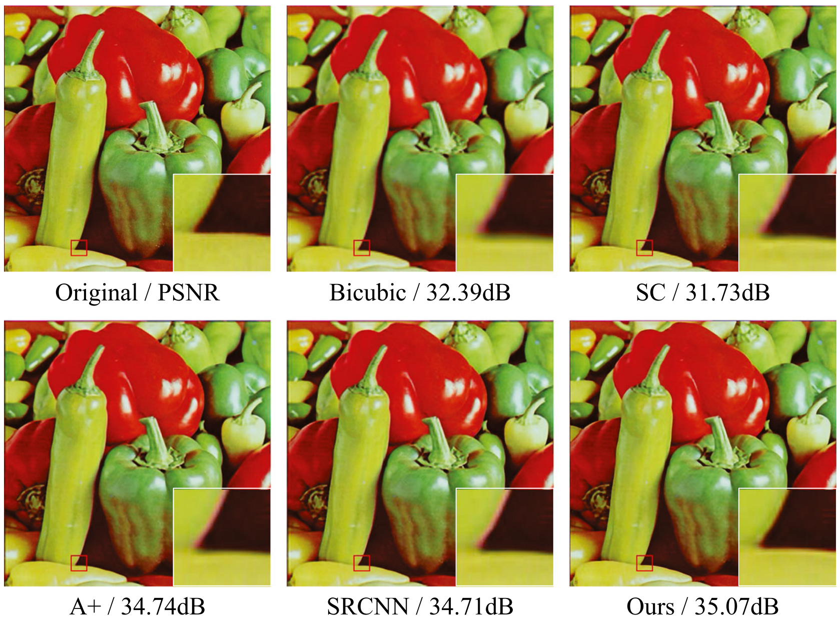


Fig. 12 Super-resolution for the "pepper" image from *Set*14 with scale factor 3

our method, VDSR and DRCN achieve comparable performance with slight differences in qualitative experimental results. Specifically, our method is inferior to VDSR and DRCN for the scale factors 3 and 4 but outperforms them for the scale factor 2 in terms of both PSNR and SSIM. It should be noted that VDSR has 20 weight layers and DRCN has 20 convolution layers and 16 recursion layers while our framework consists of three three-layered CNNs resulting in an overall nine-layered concise structure. This observation reflects that our method has much lower structure complexity than VDSR and DRCN but still achieves comparable performance with them. One reason for the contrastive results is that VDSR or DRCN takes whole images as inputs for training one overall model, and our method uses the categorized local patches to train multiple CNNs. The deep structure of VDSR or DRCNN trained on a basis of whole images is in favor of achieving high PSNR which is measured on a whole image scale. On the other hand, our multiple CNN framework is trained based on not whole images but categorized local patches. Our model is thus more effective for local feature characterization such that it favors small scale super-resolution and enables accurate visual structure preservation. Therefore, our method achieves high SSIM which characterizes detail similarities on small scales.

Table 3 illustrates the PSNR/SSIM values of bicubic interpolation, NE+LLE, SC, A+, SRCNN and ours on *Set*5 images respectively for scales 2, 3, and 4. Table 4 illustrates the PSNR/SSIM values of bicubic interpolation, NE+LLE, SC, A+, SRCNN and our proposed approach on *Set*14 images for scale 3. The slash signs – in the tables indicate that experimental results are not available from the referenced papers. It is clear that our framework outperforms the alternative state-of-the-art methods listed in Tables 3 and 4.

Finally, we select the images "butterfly," "bird" from *Set*5 and "pepper" from *Set*14 to exhibit super-resolution results of alternative methods in Figs. 10, 11, and 12, respectively. We can see from the visual results that our method obtains the clearest restored high resolution images among alternative comparison methods.

Conclusions and Future Work

Inspired by the cognitive mechanism of the human vision system (HVS), we have presented a novel clustering-oriented multiple-CNN framework for single image super-resolution. Our framework enhances the representational power of CNN for super-resolution from two perspectives. First, allocating local patches into different clusters captures the textural variability within an image. Second, each CNN from the multiple CNNs is trained based on a patch cluster such that it comprehensively learns image priors for one specific texture category. The proposed framework resembles the multi-mode mechanism of the human brain for processing different textural visual parts. It empirically provides more robust super-resolution results than the single CNN strategy (e.g., [27, 28]) and achieves state-of-the-art performance.

To the best of our knowledge, the proposed framework is a pioneering study of training multiple CNNs based on different categories of patches for single image super-resolution. It can be further extended in a number of ways in the future. First, more appropriate clustering methods with various metric choices could be exploited for textural categorizations before multiple CNN based learning. Second, a weighting scheme for different texture categories could be studied since each category makes a potentially different contribution to the super-resolution performance. Third, some ensemble learning strategies can be exploited in our framework for integrating multiple CNNs in a more principled manner. These possible future work directions could further generalize our methodology with more sophisticated learning schemes and potentially enhance its effectiveness.

Funding This study was funded by the National Natural Science Foundation of China (No. 61671481) and Qingdao Applied Fundamental Research (No. 16-5-1-11-jch), the Fundamental Research Funds for Central Universities, the EPSRC ESR project through ADR Funding Feasibility study on a fully deployable resilient flooding predicting, monitoring and response system and the 2016/2017 University of Exeter Outward Mobility Academic Fellowship.

Compliance with Ethical Standards

Conflict of Interests The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Yang J, Wright J, Huang T, Ma Y. Image super-resolution as sparse representation of raw image patches. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2008. p. 1–8.
2. Yang J, Wright J, Huang TS, Ma Y. Image super-resolution via sparse representation. IEEE Trans Image Process. 2010;19(11):2861–73.
3. Yang C-Y, Yang M-H. Fast direct super-resolution by simple functions. In: Proceedings of the IEEE international conference on computer vision. 2013. p. 561–568.
4. Timofte R, Smet VD, Gool LV. Anchored neighborhood regression for fast example-based super-resolution. In: Proceedings of the IEEE international conference on computer vision. 2013. p. 1920–1927.
5. Timofte R, Smet VD, Gool LV. A+: adjusted anchored neighborhood regression for fast super-resolution. In: Proceedings of the Asian conference on computer vision. 2014. p. 111–126.
6. Yang C-Y, Ma C, Yang M-H. Single-image super-resolution: a benchmark. In: Proceedings of the European conference on computer vision. 2014. p. 372–386.
7. Sun J, Xu Z, Shum H-Y. Image super-resolution using gradient profile prior. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2008. p. 1–8.
8. Kim KI, Kwon Y. Single-image super-resolution using sparse regression and natural image prior. IEEE Trans Pattern Anal Mach Intell. 2010;32(6):1127–33.
9. Zhang H, Yang J, Zhang Y, Huang TS. Non-local kernel regression for image and video restoration. In: Proceedings of the European conference on computer vision. 2010. p. 566–579.
10. Chatterjee P, Milanfar P. Clustering-based denoising with locally learned dictionaries. IEEE Trans Image Process. 2009;18(7):1438–51.
11. Wang Q, Tang X, Shum H. Patch based blind image super resolution. In: Proceedings of the IEEE international conference on computer vision. 2005. p. 709–716.
12. Ni KS, Nguyen TQ. Image superresolution using support vector regression. IEEE Trans Image Process. 2007;16(6):1596–1610.
13. Freedman G, Fattal R. Image and video upscaling from local self-examples. ACM Trans Graph (TOG). 2011;30(2):12.
14. Lu X, Yuan Y, Yan P. Image super-resolution via double sparsity regularized manifold learning. IEEE Trans Circ Syst Vid Technol. 2013;23(12):2022–33.
15. Lu X, Yuan Y, Yan P. Alternatively constrained dictionary learning for image superresolution. IEEE Trans Cybern. 2014;44(3):366–77.
16. Wang W, Shen J, Shao L. Deep learning for video saliency detection. 2017 arXiv:1702.00871.
17. Yao X, Han J, Cheng G. Semantic annotation of high-resolution satellite images via weakly supervised learning. IEEE Trans Geosci Remote Sens. 2016;54(6):3660–71.
18. Zhu Q, Du B, Turkbey B, et al. Deeply-supervised CNN for prostate segmentation. In: Proceedings of the international joint conference on neural networks. 2017. p. 178–184.
19. Zhang D, Han J, Li C. Detection of co-salient objects by looking deep and wide. Int J Comput Vis. 2016;120(2):215–32.
20. Spratling MW. A hierarchical predictive coding model of object recognition in natural images. Cogn Comput. 2017;9(2):151–67.
21. Wen G, Hou Z, Li H. Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cogn Comput. 2017;1–14.
22. Zhang F, Du B, Zhang L, et al. Weakly supervised learning based on coupled convolutional neural networks for aircraft detection. IEEE Trans Geosci Remote Sens. 2016;54(9):5553–63.
23. Jain V, Seung S. Natural image denoising with convolutional networks. In: Proceedings of the advances in neural information processing systems. 2009. p. 769–776.
24. Burger HC, Schuler CJ, Harmeling S. Image denoising: can plain neural networks compete with bm3d? In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2012. p. 2392–2399.
25. Eigen D, Krishnan D, Fergus R. Restoring an image taken through a window covered with dirt or rain. In: Proceedings of the IEEE international conference on computer vision. 2013. p. 633–640.
26. Cui Z, Chang H, Shan S, Zhong B, Chen X. Deep network cascade for image super-resolution. In Proceedings of the European conference on computer vision. 2014. p. 49–64.
27. Dong C, Loy CC, He K, Tang X. Learning a deep convolutional network for image super-resolution. In: Proceedings of the European conference on computer vision. 2014. p. 184–199.
28. Dong C, Loy CC, He K, Tang X. Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell. 2016;38(2):295–307.
29. Kim J, Kwon Lee J, Mu Lee K. Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 1646–1654.
30. Kim J, Kwon Lee J, Mu Lee K. Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 1637–1645.

31. Ho-Phuoc T, Guyader N, Guérin-Dugué A. A functional and statistical bottom-up saliency model to reveal the relative contributions of low-level visual guiding factors. Cogn Comput. 2010;2(4):344–59.
32. Kim S, Kwon S, Kweon IS. A perceptual visual feature extraction method achieved by imitating v1 and v4 of the human visual system. Cogn Comput. 2013;5(4):610–28.
33. Shen J, Hao X, Liang Z. Real-time superpixel segmentation by DBSCAN clustering algorithm. IEEE Trans Image Process. 2016;25(12):5933–42.
34. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the international conference on artificial intelligence and statistics. 2011. p. 315–323.
35. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
36. 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 of the IEEE international conference on computer vision. 2001. p. 416–423.
37. Bevilacqua M, Roumy A, Guillemot C, Alberi-Morel ML. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proceedings of the British machine vision conference. 2012. p. 135.1–135.10.
38. Zeyde R, Elad M, Protter M. On single image scale-up using sparse-representations. In: Proceedings of the international conference on curves and surfaces. 2010. p. 711–730.
39. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12.
40. Chang H, Yeung D-Y, Xiong Y. Super-resolution through neighbor embedding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2004. p. 275–282.
41. Dai D, Timofte R, Van Gool Luc L. Jointly optimized regressors for image super-resolution. Comput Graph Forum. 2015;34(2):95–104.
42. Dong C, Loy CC, Tang X. Accelerating the super-resolution convolutional neural network. In Proceedings of the European conference on computer vision. 2016. p. 391–407.