Pattern Recognition 68 (2017) 191–198

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Ensemble based deep networks for image super-resolution

Lingfeng Wang*, Zehao Huang, Yongchao Gong, Chunhong Pan

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

ARTICLE INFO

Article history:
Received 22 July 2016
Revised 6 January 2017
Accepted 22 February 2017
Available online 2 March 2017

Keywords:
Super-resolution
Ensemble
Sparse prior
Deep networks

ABSTRACT

There have been significant advances in deep learning based single-image super-resolution (SISR) recently. With the advantage of deep neural networks, deep learning based methods can learn the mapping from low-resolution (LR) space to high-resolution (HR) space in an end-to-end manner. However, most of them only use a single model to generate HR result. This brings two drawbacks: (1) the risk of getting stuck in local optima and (2) the limited representational ability of single model when handling various input LR images. To overcome these problems, we novelly suggest a general way through introducing the idea of ensemble into SR task. Furthermore, instead of simple averaging, we propose a back-projection method to determine the weights of different models adaptively. In this paper, we focus on sparse coding network and propose ensemble based sparse coding network (ESCN). Through the combination of multiple models, our ESCN can generate more robust reconstructed results and achieve state-of-the-art performance.

1. Introduction

As one of the classical problems in computer vision, single image super-resolution (SR) aims at is to reconstruct a visually pleasing high-resolution (HR) image from a given low-resolution (LR) input. Since multiple HR patches could correspond to the same LR patch, SR is a highly ill-posed problem. To address this problem, many SR methods have been studied in recent years, such as interpolation-based methods [1], reconstruction-based methods [2–4], and learning-based methods [5–12].

Among them, learning-based methods are widely used to model a mapping from LR to HR patches by learning prior knowledge from external LR-HR image pairs. Sparse coding based method proposed by Yang et al. [5] is a classic learning-based method and there are many extensions [6–8]. Lately, inspired by the great success achieved by deep learning [13], convolution neural networks (CNNs) have been used for image SR and showed large improvements in accuracy [9–12]. Dong et al. [12] proposed super-resolution convolution neural network (SRCNN) and demonstrated that the mapping from LR to HR patches could be learned in an end-to-end manner by a CNN. However, the prior knowledge in SRCNN is learned all from training data and people's domain expertise for SR problem is largely ignored. In order to investigate whether domain expertise can be used to design better deep network architectures, Wang et al. [9,10] proposed sparse coding based network (SCN) for image SR. Based on the learned iterative shrinkage and thresholding algorithm (LISTA) [14], they successful added sparse prior into deep neural networks. Combining the domain expertise of sparse coding and the end-to-end manner of deep learning, their SCN achieved notable improvement over SRCNN with a smaller model.

However, these end-to-end SR methods learn only one LR-HR mapping function. They can not generate flexible HR results because it is hard to handle various input LR images with only one single model. To address this problem, we novelly introduce the idea of ensemble into SR problem and suggest a general method to improve the performance of generic learning-based SR methods. Although ensembles of neural networks are known to generate more accurate predictions compared to single networks and ensemble strategy has been widely applied in image classicization task [13], it has not been investigated in SR problem. To demonstrate the effectiveness of ensemble strategy, we focus on the state-of-the-art SR method SCN and propose an ensemble based sparse coding network (ESCN). In our ESCN, multiple models trained by the same training data but different initializations are used to construct an ensemble. Instead of simple averaging, we propose a back-projection method to determine the weights of different models adaptively. With the help of multiple models ensemble strategy, our ESCN can learn a more robust mapping function to generate flexible HR results. In our experiments, we show the effectiveness of our ESCN on different datasets, where we present state-of-the-art results. By parallel execution of every model, there is not much time cost in our method. Furthermore, we also

* Corresponding author.
E-mail addresses: lfwang@nlpr.ia.ac.cn (L. Wang), zehaohuang18@gmail.com (Z. Huang), yongchao.gong@nlpr.ia.ac.cn (Y. Gong), chpan@nlpr.ia.ac.cn (C. Pan).

http://dx.doi.org/10.1016/j.patcog.2017.02.027
0031-3203/

investigate the generalization ability of ensemble strategy in SR task by apply ensemble into another state-of-the-art SR method. Overall, the contributions of this work are mainly in three aspects:

- We suggest a general way to improve the performance of learning-based SR methods by ensembles. We focus on SCN and propose a state-of-the-art method named ensemble based sparse coding network (ESCN).
- Instead of simple averaging, we propose a back-projection method to solve the weights of different models and show improvements.
- To handle the problem of cumbersome training problem in our ESCN, we suggest a way to speed up the training of SCN by adding batch normalization [15].

In the following section, we will first review some related work about several image SR methods and ensemble strategy. In Section 3 we will present our ensemble based sparse coding network (ESCN). Section 4 shows our implementation details and experiments, where we compare the performance of our ESCN to other state-of-the-art methods. Finally we conclude this paper in Section 5.

2. Related work

2.1. Sparse coding based image SR

Sparse coding based SR [5] is a classical SR method which reconstructed HR patch from a learned dictionary and its corresponding sparse representation. This process can be formulate as

$$\boldsymbol{x} = \boldsymbol{D_x}\boldsymbol{\alpha}, \quad s.t. \quad \boldsymbol{\alpha} = \arg\min_{\boldsymbol{z}} ||\boldsymbol{y} - \boldsymbol{D_y}\boldsymbol{z}||_2^2 + \lambda||\boldsymbol{z}||_1, \tag{1}$$

where $\boldsymbol{y}$ is the LR input patch, $\boldsymbol{x}$ is the HR output patch, $\boldsymbol{\alpha}$ is the sparse representation, $\boldsymbol{D_x}$ and $\boldsymbol{D_y}$ are leaned HR and LR dictionary respectively. $||\cdot||_1$ denotes the ℓ_1-norm and λ is a regularization coefficient which balances the sparsity of $\boldsymbol{\alpha}$.

$\boldsymbol{D_x}$ and $\boldsymbol{D_y}$ are jointly learned for LR and HR image patches, with the goal of sharing the same sparse representations for LR patches as their corresponding HR patches [16].

2.2. Deep learning based image SR

Recently there have been significant advances in single image SR based on deep learning techniques. SRCNN introduced by Dong et al. [6,17] and SCN proposed by Wang et al. [9,10] are two of the representative methods. Compared to traditional learning-based SR methods, SRCNN directly learns to map patches from LR to HR images and achieves a large improvements in accuracy. Lately, in order to investigate whether domain expertise is helpful in deep network, Wang *et al.* proposed SCN. Based on LISTA [14], a fast approximation method of sparse coding, they encoded the sparse representation prior into deep network. Combining the domain expertise of sparse prior and the merits of deep learning, SCN yields better SR performance than SRCNN with efficient training and small model size.

2.3. Ensemble strategy

Ensemble is a simple way to improve the performance of supervised learning algorithms. An ensemble of classifiers is a set of classifiers whose individual predications are combined by weighted averaging or unweighted voting [18]. The ensemble is often much more accurate than the individual classifiers that make them up. Theoretical, Dietterich [18] explains the advantage of ensemble by three reasons: statistical reason, computational reason and representational reason. In practice, ensemble is widely used in many machine learning algorithms, including bagging [19], boosting [20], and random forests [21]. With the development of deep learning technique, the idea of ensemble is also widely applied in deep networks [13]. However, there is still no investigation about the application of ensemble in SR problem.

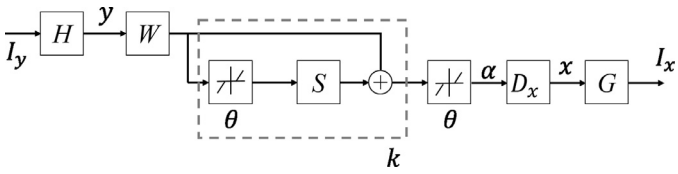


Fig. 1. The structure of SCN model with a patch extraction layer **H**, a LISTA sub-network for sparse coding (with k recurrent stages denoted by the dashed box), a HR patch recovery layer $\mathbf{D_x}$ and a patch combination layer **G**.

3. Proposed method

In this section, we will introduce our ensemble based SR method. The main idea is that the ensemble of classifiers can generate more accurate predications than the individual classifiers that make it up [18], and it is the same in regression problem. In learning-based SR methods, every single model can be regarded as a regressor, which predicts output HR image. The most simplest ensemble of regressors is a set of them whose individual predications are combined by averaging or voting.

However, the way of simple averaging does not consider about the difference between multiple leaned models and their corresponding reconstructed results. Better predictions should have high weights in ensemble. But in test or deployment stage of SR, we do not have true HR images to evaluate the performance of these results. To address this problem, we suggest a back-projection method to calculate the accuracy of reconstructed results through mapping them from HR space into LR space. Then we can combine multiple output HR images with different weights to construct an ensemble.

For obtaining competitive reconstructed results, we focus on the state-of-the-art SR method SCN, and propose an ensemble based sparse coding network (ESCN).

3.1. Sparse coding based network for image SR

Before introducing our ESCN, we firstly review the work of SCN [9] in this part. Based on LISTA [14], SCN successful added sparse prior into deep neural networks. Comparing to conventional sparse coding based SR method [5], SCN learns an end-to-end mapping and optimizes all the layer parameters jointly. Fig. 1 shows the network structure of SCN model. It consists of three parts: patch extraction/feature representation, LISTA network for sparse coding, and reconstruction. Each of these parts will be described in the following.

- **Patch extraction/feature representation**: This part is used to extract the feature representations of input image. The input bicubic-upscaled LR image $\boldsymbol{I_y}$ goes through the patch extraction and feature representation layer $\boldsymbol{H}$ then is transformed into corresponding feature representations. Layer $\boldsymbol{H}$ contains 100 filters of spatial size 9×9 so that the size of input patch is 9×9 and the dimension of feature representation $\boldsymbol{y}$ is 100. In order to reduce the number of parameters, layer $\boldsymbol{H}$ is implemented as the combination of two layers: the first layer has 4 trainable filters and the second layer contains 25 fixed filters to shift the 4 trainable filters to 25 fixed positions.
- **LISTA network for sparse coding**: In this part, LISTA network [14] is used to solve the sparse representation of LR patch $\boldsymbol{y}$.

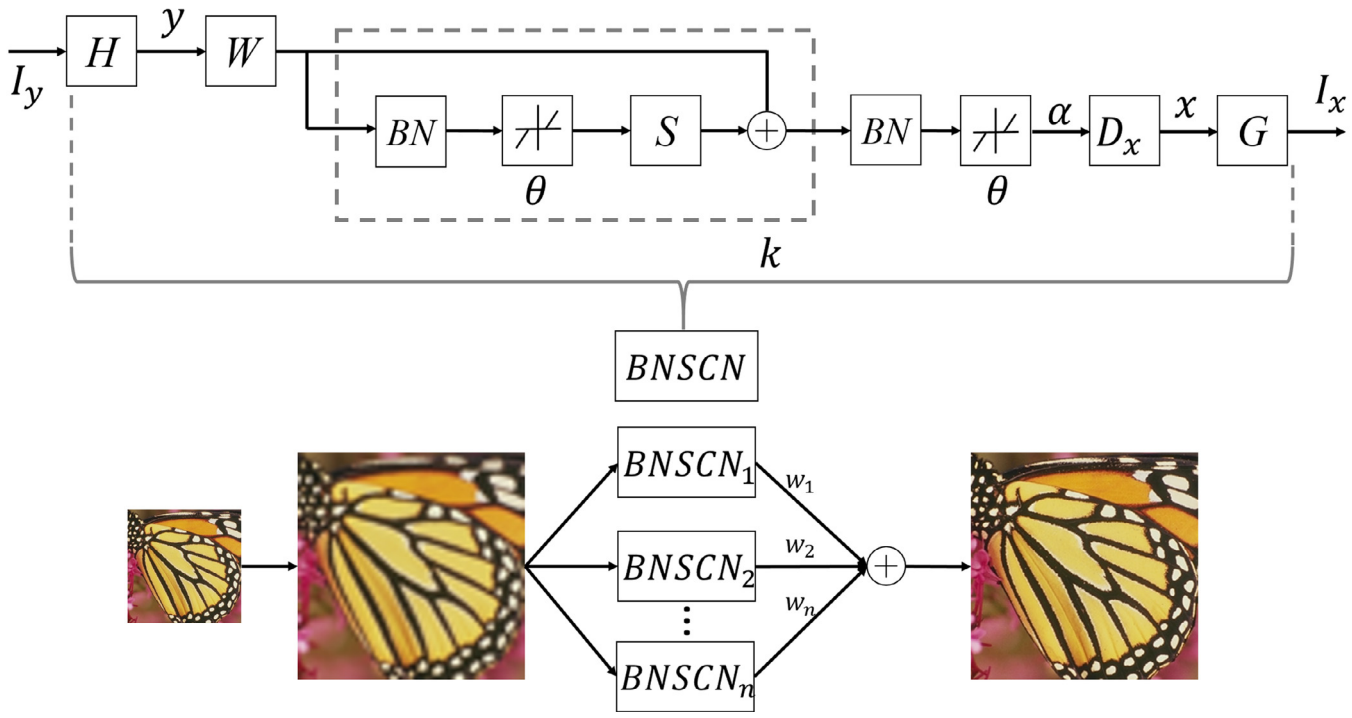


Fig. 2. Top: The structure of SCN model with batch normalization (BNSCN). Bottom: Our ESCN structure with n BNSCN models ensemble. The final HR image is generate by the combination of n output HR results with different weights.

With a finite number k of recurrent stages, it can approximate the sparse representation $\boldsymbol{\alpha} \in \mathbb{R}^{100}$ of input patch $\boldsymbol{y}$ efficiently. Each stage of LISTA contains two linear layers parameterized by $\boldsymbol{W}$ and $\boldsymbol{S}$, and a nonlinear neuron layer with activation function $h_{\boldsymbol{\theta}}$. The process of solving $\boldsymbol{\alpha}$ can be formulated as

$$\boldsymbol{\alpha}_{k+1} = h_{\theta}(\boldsymbol{W}\boldsymbol{y} + \boldsymbol{S}\boldsymbol{\alpha}_k), \tag{2}$$

where $h_{\boldsymbol{\theta}}$ is defined as

$$[h_{\boldsymbol{\theta}}(\boldsymbol{a})]_i = \text{sign}(a_i)(|a_i| - \theta_i)_+. \tag{3}$$

All the parameters including weights $\boldsymbol{W}$, $\boldsymbol{S}$ and thresholds $\boldsymbol{\theta}$ are learned during training.

- **Reconstruction**: This part consists of two layers: HR dictionary $\boldsymbol{D_x}$ and patch combination layer $\boldsymbol{G}$. HR patch $\boldsymbol{x}$ is reconstructed by the production of the solved sparse representation $\boldsymbol{\alpha}$ and the HR dictionary $\boldsymbol{D_x}$. After getting all HR patches, the final layer $\boldsymbol{G}$ recover all of them into a HR image $\boldsymbol{I_x}$. Similar to the implementation of layer $\boldsymbol{H}$, $\boldsymbol{G}$ is also split into two layers: a fixed layer which aligns pixels in overlapping patches and a trainable layer which combines overlapping pixels with different weights.

3.2. Ensemble based sparse coding network

In deep learning based SR methods, most of them use one model to generate HR images. This brings two drawbacks: (1) since deep models solved by stochastic gradient descent (SGD) may get stuck in local optima, using a single model is not accurate enough. (2) in SR task, there are various input LR images in application and it is hard for only one single model to handle all these input LR images and gets robust reconstructed results. To overcome the first drawback, we introduce the idea of ensemble into SR task and develop ensemble based sparse coding network (ESCN). For the second, we propose a back-projection method to enhance the adaptability of our ensemble.

Consider a single LR image $\boldsymbol{y}$ and the original HR image $\boldsymbol{X}$. The bicubic interpolation of $\boldsymbol{y}$ is denoted as $\boldsymbol{Y}$. Our goal is to recover $\boldsymbol{X}$ from $\boldsymbol{Y}$ by a mapping function $F(\cdot)$. In ensemble strategy, multiple mapping functions are combined to form a more accurate and robust mapping function. In deep learning based SR, the learned model can be regarded as a mapping function. Therefore, we use a set of learned models $M_i(\cdot)$ to make up an ensemble. We can get different reconstructed results $M_i(\boldsymbol{Y})$ from $M_i(\cdot)$. Consequently, the final mapping function of ensemble $F(\boldsymbol{Y})$ can be formulated as the combination of all the results with different weights w_i:

$$F(\boldsymbol{Y}) = \sum_{i=1}^{n} w_i M_i(\boldsymbol{Y}), \tag{4}$$

where n denotes the number of combined models. Each mapping function is weighted by w_i according to its accuracy. However, we can not evaluate the accuracy of $M_i(\boldsymbol{Y})$ since the true HR image $\boldsymbol{X}$ is unavailable during test or deployment stage. To address this problem, we propose back-projection method to translate $M_i(\boldsymbol{Y})$ from HR space to LR space and evaluate its accuracy by comparing with the input LR image $\boldsymbol{y}$. In the following, we will describe our schemes of learning $M_i(\boldsymbol{Y})$ by training basic SCN model, and solving w_i by our back-projection method.

3.3. Multiple models training

Since ESCN is constructed by a set of SCN models, we train basic SCN n times with different initializations to get n SCN models. However, the training of SCN in [9] is time consuming. This problem is more serious in our ESCN since we need to train basic model several times. For speeding up the training of SCN, we introduce batch normalization (BN) [15] into SCN and named it BNSCN. As shown in the top part of Fig. 2, two batch normalization layer are added before the non-linear activation layers respectively. The bottom part of Fig. 2 shows the architecture of our ESCN.

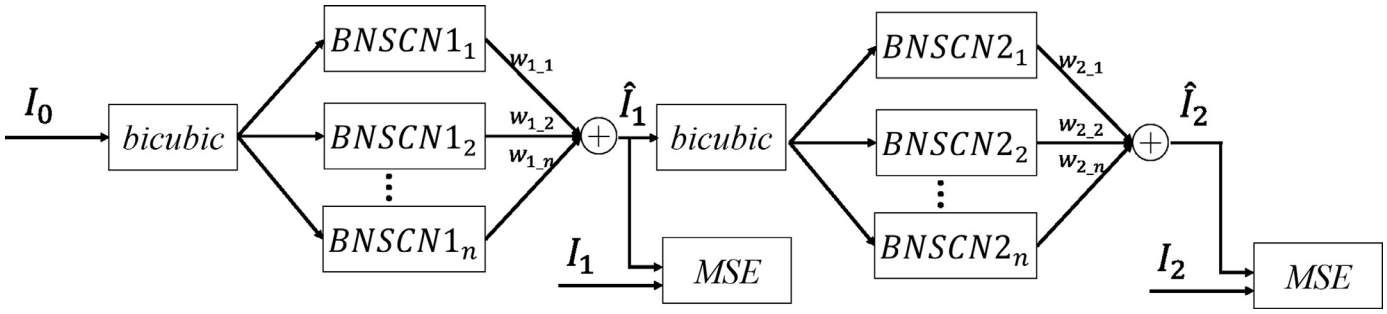


Fig. 3. The cascade structure of ESCN for scalable SISR.

3.4. Solving weights

In order to solve the weights of different models, we propose a back-projection method. The main idea of our back-projection method is the accuracy of reconstructed results can be evaluated by comparing the differences between the downsampling version of $F(\boldsymbol{Y})$ (the input LR image $\boldsymbol{y}$) and $M_i(\boldsymbol{Y})$. With this idea, (4) can be rewritten as:

$$\boldsymbol{x} = \sum_{i=1}^{n} w_i M_i(\boldsymbol{Y}) \downarrow, \tag{5}$$

where $\downarrow$ denotes downsampling operation. Since $\boldsymbol{x}$ and $M_i(\boldsymbol{Y})\downarrow$ are already known, w_i can be solved by

$$w_i = \arg\min_{w_i} ||\boldsymbol{x} - \sum_{i=1}^{n} w_i M_i(\boldsymbol{Y}) \downarrow ||_2^2,$$
$$s.t. \quad \sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0. \tag{6}$$

This is a linear programming problem and it can be solved by simplex algorithm. For example, when $n = 2$, w_1 and w_2 can be solved by linear search. After getting $M_i(\boldsymbol{Y})$ and w_i, the final HR result can be easily reconstructed by $F(\boldsymbol{Y}) = \sum_{i=1}^{n} w_i M_i(\boldsymbol{Y})$.

3.5. Cascade based ESCN for scalable SISR

As the magnification factor increases, the performance of SR becomes worse. To handle this problem, cascade is a good solution. Timofte et al. [22] used cascade to generate HR images from coarse to fine. Wang et al. [9] used the cascade of small factor model to obtain high factor results. In our experiment, we also observed that a cascade of BNSCNs trained for factor 2 can generate better SR results than a single BNSCN trained for a larger factor, such as 3 or 4. Thus, we cascade our ESCN to generate scalable SR results. Fig. 3 shows our cascaded ESCN. We use one ESCN to generate factor 2 results and the cascade of two ESCN to generate results of factor 4. Then we can generate results of factor 3 by applied downsampling operation to factor 4 results. The cascade structure of ESCN seems very complicated, but it is easy to find that the multiple models in each ESCN can be handled in parallel. The cost of solving different weights by back-projection is negligible. Therefore, the time cost of our cascade based ESCN is almost the same as the cascade of SCN in [9].

4. Experiments

4.1. Datasets

Training dataset. In order to compare with SCN [9] fairly, we train our basic BNSCN models using 91 images from Yang et al. [5]. Data augmentation is adopted by adding following distortions:

- Flip: horizontal-flip and vertical-flip.
- Rotation: affine transform with 90, 180 and 270°.
- Translation: random x-y shifts between $[-4, 4]$ pixels.
- Zoom: random scaling factors between [1/1.2, 1.2].

After data augmentation, we get 1638 images for training.

Testing dataset. For a fair comparison, we use Set5 [23], Set14 [24] and BSD100 [25] which are often used in other SR works for testing. The original images are downsampled by bicubic interpolation to generate LR-HR image pairs for evaluation.

4.2. Implementation details

Basic BNSCN model. We use the same parameters as SCN in each basic BNSCN model except adding two batch normalization (BN) layers. The input LR patch size $s_y = 9$, LR feature dimension $m_y = 100$, dictionary size $n = 128$, output HR patch size $s_x = 5$, and patch aggregation filter size $s_g = 5$. The number of recurrent stages k is set to 1 following original SCN. During training, the ground truth images $\{\boldsymbol{X}_i\}$ are prepared as 56×56 HR patches randomly cropped from the training images. To obtain corresponding LR samples $\{\boldsymbol{Y}_i\}$, we first sub-sample HR patches by the upscaling factor, and upscale them by the same factor via bicubic interpolation. Following [12], all the convolution layers in basic BNSCN have no padding, and the network produces a smaller output 44×44. Therefore, we evaluate the mean squared error loss only by the difference between the central 44×44 crop of $\{\boldsymbol{X}_i\}$ and the network output.

Training details. We use SGD with a mini-batch size of 64. Momentum and weight decay parameters are set to 0.9 and 0.0001, respectively. We initialize the weights of each model as in [9]. All basic BNSCN models are trained over 80 epochs (5560 iterations with batch size 64). The learning rate starts from 0.1 and then decreased by a factor of 10 every 20 epochs. Gradient clipping is used in our experiment to avoid gradient explosion. In our implementation, all the training is implemented using the Caffe package [26] on a GTX 970 GPU. Fig. 4 shows the PSNR curve of our BNSCN during training. With the help of batch normalization, every BNSCN model can achieve 37.0 dB at only about 5×10^5 back-propagations, which is 120 times faster than the training in [9].

Testing details. In testing, we only process the luminance channel with our super-resolution method. To keep the size of output image the same as original image, we extend the boundary of input bicubic-upscaled image by reflection, following [9]. We also shave the image border in the same way as Dong et al. [12] for objective evaluations to ensure fair comparison. In our experiment, we use 4 models for ensemble.

4.3. Comparison with state of the arts

We compare our ESCN with the state-of-the-art SR methods on all the images in testing datasets for upscaling factors 2, 3 and

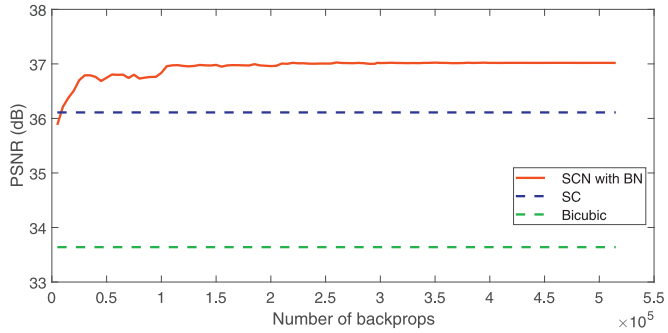


Fig. 4. The PSNR change for x2 SR on Set5 during training using our SCN with batch normalization method (BNSCN). The horizontal dash lines show the results of sparse coding (SC) and bicubic interpolation. We add batch normalization into SCN model and accelerate the training speed significantly (achieve 37.0 dB at about 5 $\times$ 10^5 number of backprops).

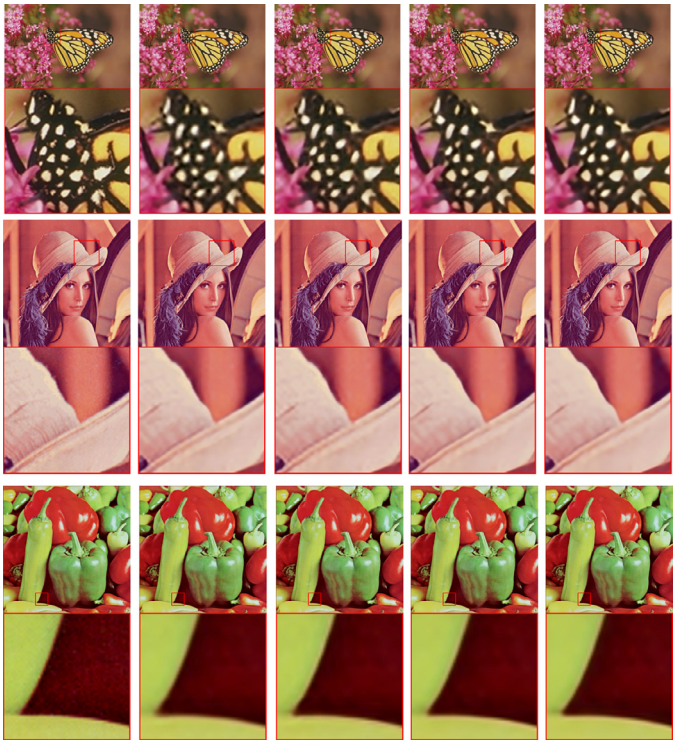


Fig. 5. SR results given by Ground Truth (first col), A+ [8] (second col), SRCNN [17] (third col), CSCN [9] (fourth col) and ESCN(Ours) (fifth col). Images from top to bottom: the "monarch" image upscaled by $\times$3; the "lenna" image upscaled by $\times$3; the "pepper" image upscaled by $\times$3.

4. Compared methods are A+ [8], SelfEx [27], SRCNN [17] (trained with larger model size and more data) and CSCN [9]. The implementations are all from the publicly available codes provided by the authors.

In Fig. 5, we compare the visual qualities of the SR results generated by our method and other top-performing methods. Our approach produces sharp edges with little artifacts. In Table 1, we provide a summary of quantitative evaluation on several datasets. The proposed ESCN method yields the highest average PSNR in all these datasets.

To point out the improvement of our ESCN is not limited in several images, we compare our method with SRCNN [12], and CSCN [9] in the 5 images of Set5 and the 100 images of BSD100 with different scales. In Table 2, we show the PSNR of Set5 images which are tested on 2, 3 and 4 scale respectively. Our ESCN achieves the highest PSNR in all test cases. For better visualize the improvement of our ESCN in BSD100, we show the curves of improved PSNR and SSIM between our ESCN and other methods in Fig 6. The curves of improved PSNR and SSIM are above 0 in most of the testing images, which means our ESCN performs the best PSNR and SSIM in these three methods.

4.4. Improvement of ensemble strategy

To explore the improvement of ensemble strategy, we compare the performance of ESCN as the ensemble size, n, is varied. Fig 7 and Table 3 show the performance of different ensemble size. Although ensembling more models generally gives better performance, we observe significant improvements when the ensemble size is 2 and 3. Compared to original SCN, our ESCN improves nearly 0.2 dB in PSNR. Compared to our BNSCN ($n = 1$), ensemble strategy can also improves about 0.1 dB. It is worth mentioning that 0.1 dB is still a significant improvement since the time cost of our ESCN is almost the same as SCN.

4.5. Improvement of back-projection method

In this section, we will discuss the advantage of our back-projection method. We compare the performance of our back-projection method and simple averaging on different datasets when the ensemble size is 4 in Table 4. Since the weights solved by our methods are near 0.25, the differences between averaging and back-projection are minimal. However, our method still shows improvements when scales are 3 and 4, which are more important than scale 2 in SR task. Since there are only 5 images in Set5 and the results in Set5 are not very convincing, we focus on the performances in Set14 and BSD100. Our method achieves higher PSNR than averaging, which indicates the weights solved by back-projection are closer to the best combinations.

5. Discussions and conclusions

In order to show the adaptability of ensemble strategy in the field of SR, we introduce our ensemble strategy into a new state-of-the-art deep learning based SR method, very deep network for super-resolution (VDSR) [11]. Different from Kim et al. [11], we train VDSR models uses 91 images from Yang et al.[5]. Data augmentation technique is used to obtain more training images. So the performance of our VDSR is different from Kim et al. [11]. Fig 8 show the improvement of our ensemble based VDSR (EVDSR). As we can see, ensemble strategy improves about 0.1 dB in PSNR. Therefore, our ensemble strategy can be regarded as a universal technique for improving the performance of learning-based SR methods.

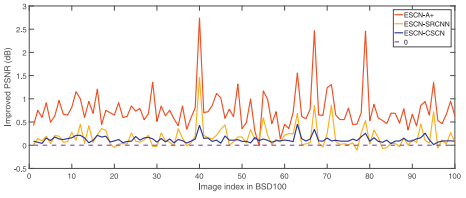
We suggest a simple and general way to improve the performance of learning-based SR methods through introducing the idea of ensemble into SR task. For obtaining competitive reconstructed results, we focus on the state-of-the-art SR method sparse coding network (SCN), and propose an ensemble based sparse coding network (ESCN). Our ESCN combines multiple reconstructed results generated by different basic SCN models and achieves higher performances. Besides, we propose a back-projection method to solve the weights of different models adaptively and show improvements than simple averaging in large scale SR problem. In our experiments, our ensemble strategy improves the performance of original SCN by getting more robust reconstruction results with negligible time cost. Furthermore, we adopt our ensemble strategy into other deep learning based SR method and show the generalization ability of our ensemble strategy.

Table 1
PSNR and SSIM comparison on three test data sets among different SR methods. **Red** iFndicates the best and **blue** indicates the second performance.

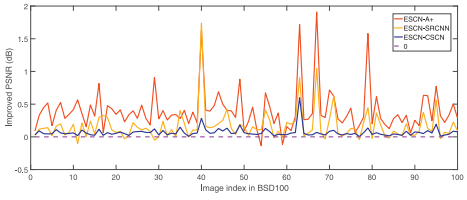
Dataset	Scale	Bicubic PSNR/SSIM	A+ PSNR/SSIM	SRCNN PSNR/SSIM	SelfEx PSNR/SSIM	CSCN PSNR/SSIM	ESCN PSNR/SSIM
Set5	×2	33.66/0.9299	36.57/0.9545	36.66/0.9542	36.48/0.9535	36.93/0.9552	37.14/0.9571
	×3	30.39/0.8682	32.67/0.9093	32.75/0.9090	32.57/0.9089	33.10/0.9144	33.28/0.9173
	×4	28.42/0.8104	30.36/0.8617	30.49/0.8628	30.31/0.8620	30.86/0.8732	31.02/0.8774
Set14	×2	30.23/0.8687	32.47/0.9063	32.45/0.9067	32.22/0.9034	32.56/0.9074	32.67/0.9093
	×3	27.54/0.7736	29.29/0.8203	29.30/0.8215	29.14/0.8197	29.41/0.8231	29.51/0.8264
	×4	26.00/0.7019	27.47/0.7514	27.50/0.7513	27.39/0.7516	27.64/0.7578	27.75/0.7611
BSD100	×2	29.56/0.8431	30.77/0.8756	31.36/0.8879	31.18/0.8855	31.40/0.8884	31.54/0.8909
	×3	27.21/0.8374	28.18/0.7791	28.41/0.7863	28.30/0.7843	28.50/0.7875	28.58/0.7917
	×4	25.96/0.6674	26.74/0.7065	26.90/0.7103	26.85/0.7108	27.03/0.7161	27.11/0.7197

Table 2
The result of PSNR and SSIM on the Set5 dataset with different factors. **Red** indicates the best and **blue** indicates the second performance.

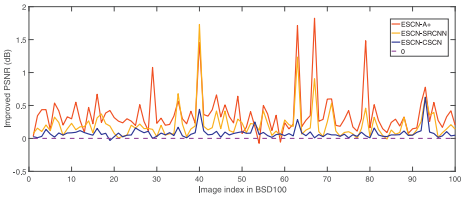
Set5	Scale	Bicubic PSNR/SSIM	A+ PSNR/SSIM	SRCNN PSNR/SSIM	SelfEx PSNR/SSIM	CSCN PSNR/SSIM	ESCN PSNR/SSIM
baby	2	37.07/0.9520	38.49/0.9650	38.54/0.9655	38.42/0.9641	38.42/0.9643	38.63/0.9664
bird	2	36.81/0.9721	41.21/0.9866	40.01/0.9859	41.15/0.9865	41.49/0.9865	41.97/0.9883
butterfly	2	27.43/0.9154	32.05/0.9648	32.75/0.9648	31.79/0.9627	33.41/0.9683	33.44/0.9709
head	2	34.86/0.8623	35.75/0.8866	35.72/0.8850	35.67/0.8851	35.77/0.8869	35.87/0.8888
woman	2	32.14/0.9476	35.35/0.9694	35.37/0.9687	35.34/0.9689	35.55/0.9701	35.77/0.9718
average	2	33.66/0.9299	36.57/0.9545	36.66/0.9542	36.48/0.9535	36.93/0.9552	37.13/0.9571
baby	3	33.91/0.9039	35.17/0.9228	35.25/0.9241	35.16/0.9232	35.24/0.9232	35.35/0.9262
bird	3	32.58/0.9256	35.77/0.9563	35.48/0.9549	35.75/0.9584	35.96/0.9584	36.22/0.9617
butterfly	3	24.04/0.8216	27.36/0.9112	27.95/0.9098	26.86/0.9051	28.59/0.9269	28.87/0.9300
head	3	32.88/0.8003	33.73/0.8271	33.71/0.8272	33.71/0.8267	33.84/0.8291	33.92/0.8319
woman	3	28.56/0.8896	31.31/0.9290	31.37/0.9291	31.38/0.9313	31.87/0.9344	32.01/0.9369
average	3	30.39/0.8682	32.67/0.9093	32.75/0.9090	32.57/0.9089	33.10/0.9144	33.28/0.9173
baby	4	31.78/0.8567	33.24/0.8835	33.13/0.8824	33.11/0.8824	33.22/0.8845	33.39/0.8882
bird	4	30.18/0.8729	32.71/0.9147	32.52/0.9112	32.85/0.9212	32.93/0.9189	33.21/0.9241
butterfly	4	22.10/0.7369	24.63/0.8441	25.46/0.8566	24.01/0.8341	26.15/0.8817	26.18/0.8857
head	4	31.59/0.7536	32.46/0.7819	32.44/0.7801	32.47/0.7819	32.66/0.7874	32.74/0.7900
woman	4	26.46/0.8318	28.76/0.8846	28.89/0.8837	29.11/0.8907	29.34/0.8935	29.57/0.8987
average	4	28.42/0.8104	30.36/0.8617	30.49/0.8628	30.31/0.8620	30.86/0.8732	31.02/0.8774



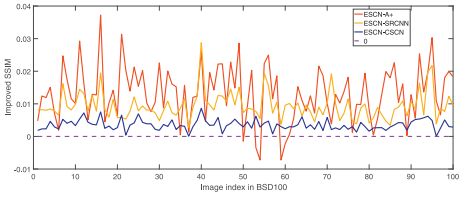
(a) Improved PSNR with factor 2



(b) Improved PSNR with factor 3



(c) Improved PSNR with factor 4



(d) Improved SSIM with factor 4

Fig. 6. Improved PSNR (dB) and SSIM on the BSD100 dataset between ESCN(Ours) and other SR methods: A+, SRCNN and CSCN. The horizontal axis means the index of BSD100 images. The vertical axis means the difference between the PSNR or SSIM of our ESCN and other methods, such as $PSNR_{ESCN}() - PSNR_{A+}()$ or $SSIM_{ESCN}() - SSIM_{A+}()$.

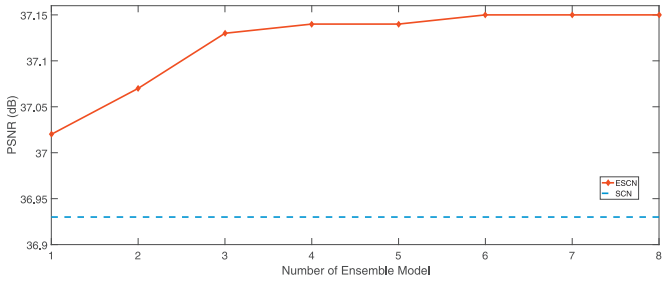


Fig. 7. PSNR for x2 SR on Set5 with different numbers of ensemble models using our ESCN.

Table 3
The result of PSNR and SSIM on the Set5, Set14 and BSD100 datasets with different number of ensemble models in ESCN.

DataSets	Scale	n=1 PSNR/SSIM	n=2 PSNR/SSIM	n=3 PSNR/SSIM	n=4 PSNR/SSIM
Set5	X2	37.02/0.9565	37.07/0.9569	37.13/0.9570	37.14/0.9571
	X3	33.25/0.9171	33.27/0.9173	33.28/0.9173	33.28/0.9173
	X4	30.99/0.8767	31.01/0.8771	31.02/0.8771	31.02/0.8774
Set14	X2	32.60/0.9083	32.64/0.9089	32.62/0.9061	32.67/0.9093
	X3	29.50/0.8262	29.51/0.8264	29.51/0.8264	29.51/0.8264
	X4	27.74/0.7607	27.75/0.7610	27.75/0.7611	27.75/0.7611
BSD100	X2	31.47/0.8898	31.51/0.8905	31.52/0.8907	31.54/0.8909
	X3	28.57/0.7914	28.57/0.7916	28.58/0.7916	28.58/0.7917
	X4	27.10/0.7193	27.10/0.7196	27.11/0.7196	27.11/0.7197

Table 4
The result of PSNR on the Set5, Set14 and BSD100 datasets with different ensemble methods when ensemble size is 4. **Red** indicates the best performance.

Scale	Ensemble Method	Set5	Set14	BSD100
×2	Average	37.1358	32.6700	31.5362
	Ours	37.1361	32.6696	31.5359
×3	Average	33.2775	29.5145	28.5769
	Ours	33.2776	29.5147	28.5770
×4	Average	31.0172	27.7514	27.1081
	Ours	31.0170	27.7515	27.1082

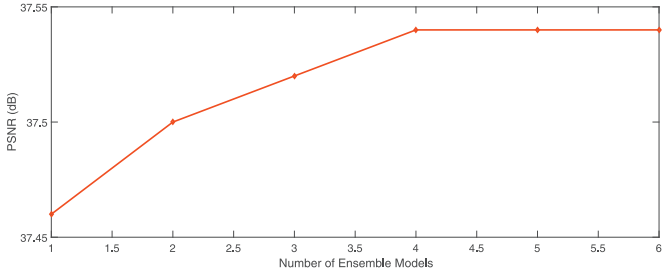


Fig. 8. PSNR for x2 SR on Set5 with different numbers of ensemble models using our EVDSR.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61403376, 91646207 and 61370039).

References

[1] P. Thévenaz, T. Blu, M. Unser, Image interpolation and resampling, in: Handbook of Medical Imaging, Processing and Analysis, 2000, pp. 393–420.
[2] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, IEEE, 2004, p. I.
[3] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2009, pp. 349–356.
[4] M. Protter, M. Elad, H. Takeda, P. Milanfar, Generalizing the nonlocal-means to super-resolution reconstruction, Image Process. IEEE Trans. 18 (1) (2009) 36–51.
[5] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, Image Process. IEEE Trans. 19 (11) (2010) 2861–2873.
[6] W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, Image Process. IEEE Trans. 20 (7) (2011) 1838–1857.
[7] R. Timofte, V. Smet, L. Gool, Anchored neighborhood regression for fast example-based super-resolution, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2013, pp. 1920–1927.
[8] R. Timofte, V. De Smet, L. Van Gool, A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conference on Computer Vision, Springer, 2014, pp. 111–126.
[9] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2015, pp. 370–378.
[10] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, T.S. Huang, Robust single image super-resolution via deep networks with sparse prior, IEEE Trans. Image Process. 25 (7) (2016) 3194–3207.
[11] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral), 2016.
[12] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: European Conference on Computer Vision, Springer, 2014, pp. 184–199.
[13] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[14] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the International Conference on Machine Learning, 2010, pp. 399–406.
[15] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167(2015).
[16] J. Yang, Z. Wang, Z. Lin, S. Cohen, T. Huang, Coupled dictionary training for image super-resolution, Image Process. IEEE Trans. 21 (8) (2012) 3467–3478.
[17] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, Pattern Anal. Mach. Intell. IEEE Trans. 38 (2) (2016) 295–307.
[18] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.
[19] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[20] R.E. Schapire, The boosting approach to machine learning: An overview, in: Nonlinear Estimation and Classification, Springer, 2003, pp. 149–171.
[21] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[22] R. Timofte, R. Rothe, L. Van Gool, Seven ways to improve example-based single image super resolution, arXiv preprint arXiv:1511.02228 (2015).
[23] M. Bevilacqua, A. Roumy, C. Guillemot, M.L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012).
[24] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: Proceedings of the International Conference on Curves and Surfaces, Springer, 2010, pp. 711–730.
[25] D. Martin, C. Fowlkes, D. Tal, J. Malik, 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, 2, IEEE, 2001, pp. 416–423.
[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093 (2014).
[27] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 5197–5206.

Lingfeng Wang received his B.S. Degree in computer science from Wuhan University, Wuhan, China, in 2007. He is currently an associate professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision and image processing.

Zehao Huang received his B.S. Degree in automatic control from Beihang University, Beijing, China, in 2015. He is currently a research intern with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision and image processing.

Yongchao Gong received his B.S. degree in automation from the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui, China, in 2012. He is currently pursuing the Ph.D. degree at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include pattern recognition, machine learning and image processing.

Chunhong Pan received his B.S. Degree in automatic control from Tsinghua University, Beijing, China, in 1987, his M.S. Degree from Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, China, in 1990, and his Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, in 2000. He is currently a professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing.