Pattern Recognition 102 (2020) 107169

Contents lists available at ScienceDirect

Pattern Recognition

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

Blind single image super-resolution with a mixture of deep networks

Yifan Wang, Lijun Wang, Hongyu Wang*, Peihua Li, Huchuan Lu

School of Information and Communication Engineering, Dalian University of Technology, Dalian, 116024, China

ARTICLE INFO

Article history:
Received 13 February 2019
Revised 28 November 2019
Accepted 15 December 2019
Available online 4 February 2020

Keywords:
Blind super-resolution
Mixture of networks
Blur kernels
Lower bound
Latent variables

ABSTRACT

Existing deep neural network based image super-resolution (SR) methods are mostly designed for non-blind cases, where the blur kernel used to generate the low-resolution (LR) images is assumed to be known and fixed. However, this assumption does not hold in many real scenarios. Motivated by the observation that SR of LR images generated by different blur kernels are essentially different but also correlated, we propose a mixture model of deep networks, which is capable of clustering SR tasks of different blur kernels into a set of groups. Each group is composed of correlated SR tasks with similar blur kernels and can be effectively handled by a combination of specific networks in the mixture model. To achieve automatic SR tasks clustering and network selection, we model the blur kernel with a latent variable, which is inferred from the input image by an encoder network. Since the ground-truth of the latent variable is unknown in the training stage, we initialize the encoder network by pre-training it on the blur kernel classification task to avoid trivial solutions. To jointly train the mixture model and the encoder network, we further derive a lower bound of the likelihood function, which circumvents the intractability in direct maximum likelihood estimation. Extensive evaluations are performed on benchmark data sets and validate the effectiveness of the proposed method.

© 2019 Published by Elsevier Ltd.

1. Introduction

Single image super-resolution (SR) aims at restoring the high-resolution (HR) image from its corresponding low-resolution (LR) counterpart. The formation of the LR image $\boldsymbol{L}$ can be modeled by the composition of the image blurring and down-sampling of its original HR version $\boldsymbol{H}$:

$$\boldsymbol{L} = S(\boldsymbol{B} * \boldsymbol{H}) + \boldsymbol{n}, \tag{1}$$

where $S(\cdot)$ denotes the down-sampling operator; $\boldsymbol{B}$ represents the blur kernel convolved on the HR image; $\boldsymbol{n}$ is additive noise.

Most recent methods perform image SR by learning mapping relationships from the LR input to the HR target in a data-driven manner. A number of machine learning techniques, such as sparse coding [1,2], neighbor embedding [3,4], and deep neural networks (DNNs) [5,6], have been intensively investigated for this purpose. In particular, the DNN-based methods have significantly improved the state-of-the-art performance of image SR.

One issue of the above methods, regardless of whether shallow or deep models, is that they perform image SR by assuming a known and prefixed blur kernel in (1), *i.e.*, non-blind SR, which typically does not hold in real scenarios. As suggested by Efrat et al. [7], Riegler et al. [8], SR of LR images generated by different blur kernels are essentially different tasks. Consequently, the performance could significantly degrade when blur kernels used in training and testing are inconsistent. Besides, training a single model for a wide range of different blur kernels may give rise to sub-optimal results (See Fig. 1 for an example). Therefore, handling different blur kernels entails training multiple models, which is inefficient and impractical [8]. More critically, since the true blur kernels are often unknown in real applications, there is no clear criteria for choosing the appropriate model, leading to inferior performance.

As opposed to the above non-blind SR models, blind SR methods seek more accurate inference by simultaneously estimating both blur kernel and HR image, providing potential solutions to the above issues. Unfortunately, since blind SR is an inherently more challenging task, the research is relatively limited in the literature. Some conventional blind SR methods [9,10] estimate the blur kernel and HR image through iterative optimization, which is computationally expensive and suffers from a high risk of converging to local minima. Other blind SR methods [11–13] rely on hand-crafted features and heuristic image priors to estimate the blur kernel, which are only suitable for specific cases, *e.g.*, images with repeated structures [11], but have very limited generalization ability. The strong expressive power of DNNs has been largely unexplored for blind image SR.

In light of the above observations, we propose a new blind image SR method via a mixture of deep networks, which decomposes

* Corresponding author.
E-mail address: whyu@dlut.edu.cn (H. Wang).

https://doi.org/10.1016/j.patcog.2019.107169
0031-3203/© 2019 Published by Elsevier Ltd.

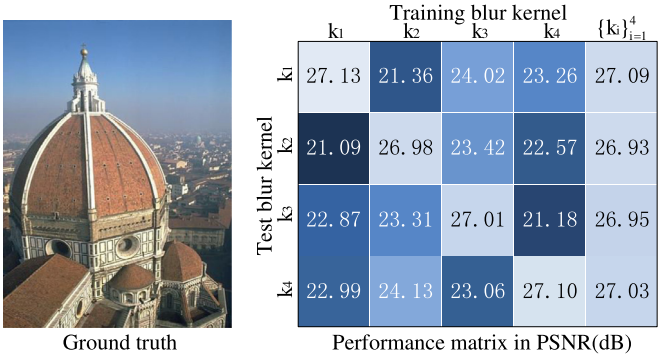


Fig. 1. Performance of 5 networks for × 2 SR. All networks have the same architecture but are trained on different blur kernels and evaluated for all blur kernels. Left: ground truth HR images. Right: performance matrix in terms of PSNR, with each column indicating the performance of a network on all blur kernels. Brighter colors represent higher performance. Networks trained on a specific kernel fail to generalize to unseen kernels. The network trained on all blur kernels (last column) mostly provides sub-optimal results.

the blind SR tasks into more fine-grained groups according to their blur kernels. SR tasks in each group are highly correlated with similar blur kernels and specifically handled by certain networks in the mixture. To automatically identify the unknown blur kernels at test time, we encode their information into a latent variable by an encoder network conditioned on the LR image. The inferred latent variable can then be used to aggregate the outputs of networks in the mixture to perform SR.

The mixture of networks as well as the encoder network are jointly trained under the maximum likelihood estimation (MLE) framework. However, unlike training a single network, direct optimization of such mixture model is intractable and causes computational instability. To circumvent this issue, we derive and maximize an evidence lower bound of the likelihood function. Experiments show that optimization of the lower bound with respect to network parameters performs well in practice. Another concern about the network training is that the ground truth of the latent variable is unknown, which may lead to a trivial solution of the optimization problem, *e.g.*, equally weighting all networks for different blur kernels. We attack this issue by developing a pre-training method of the encoder network for blur kernel classification. Such pre-training strategy can not only provide prior distribution of the latent variable, but also appropriately initialize the encoder in the parameter space, rendering a more effective optimization process in the subsequent joint training.

Our method is able to simultaneously handle multiple blur kernels, delivers more superior performance without knowing any prior knowledge about blur kernels, and thus can better scale up in real applications than non-blind SR methods. Compared with existing blind SR algorithms, our method enjoys the strong representative power of DNNs and does not rely on heuristic priors. Moreover, the blur kernels can be inferred via one forward propagation, which is more efficient than iterative estimations adopted in [9,10].

The contributions of this work are threefold. Firstly, we propose a new paradigm for blind image SR using mixture of deep networks, where a latent variable inferred by the encoder network is used to model the blur kernel and to aggregate the predictions of networks in the mixture. Secondly, a pre-training approach for the encoder network on the blur kernel prediction task is designed to discover the prior distribution of the latent variable and properly initialize the encoder network. Thirdly, we address the intractability in the direct optimization of the mixture model by deriving a lower bound of the likelihood function, which provides supervision for joint training of the whole network. The proposed method performs favorably against state-of-the-art methods. Extensive evaluations are also performed to study the contribution of each component of the proposed method to the final performance.

2. Related Work

Learning based non-blind SR methods have witnessed dramatic growth with various methods being developed. For instance, Yang et al. [1] trains the LR and HR dictionaries jointly with the constraint that LR patches and their HR counterparts share the same sparse representation. The neighbor embedding based algorithms [3,4] are highly reliant on the assumption that LR and HR patches lie on low-dimensional nonlinear manifolds with locally similar geometry. Based on the regression trees algorithm, work [14] builds on linear multivariate regression models using leaf nodes. More recently, significant progress has been achieved [15–17] thanks to the advances in DNNs. The seminal work [5] proposes a super-resolution convolutional neural network (SRCNN) to learn the mapping function from LR to HR images. Two parallel works [18,19] combine the strengths of conventional sparse coding and DNNs to achieve efficient training and good restoration quality. In [15], a 20-layer convolution network is proposed to learn the residual between the HR and LR image and achieves state-of-the-art performance. Work [20] learns to upsample the LR spatial size in the last layer of networks, which further improve the SR performance and speed of Dong et al. [5].

While impressive performance has been reported, these methods proceed based on a restrictive setup where the blur kernel is known and fixed in both training and testing phases. However, this assumption does not hold in most cases, hindering the usage of non-blind SR methods in real applications. Thereby, the task of blind SR has attracted increasingly more attention from the community. Though the importance of accurately estimating blur kernels in reconstructing HR images has been highlighted by Efrat et al. [7], only a few works have investigated the estimation of the blur kernel in the SR process. Among them, Bégin and Ferrie [9], He et al. [10] assume a parametric Gaussian model for the blur kernel. Michaeli and Irani [11] proposes a nonparametric method to estimate the blur kernel by maximizing the similarity of recurring patches across multi-scales of the LR image. It is further developed in [12], where the convolutional consistency and the predictions of a non-blind SR method are explored to guide the estimation process. In [8], a conditioned regression method is proposed to incorporate the additional kernel information into the non-blind SR methods, which can handle a variety of blur kernels with a single model. However, the blur kernel is still assumed to be known in advance. Zhang et al. [21] proposes a dimensionality stretching strategy, which takes both LR image and its degradation maps (*i.e.*, blur kernel and noise level) as input and learns a single SR network for multiple degradations.

The aforementioned blind SR methods usually address the problem by iterative MAP (maximum a posterior) approaches based on shallow models and heuristic priors. Moreover, the estimation of blur kernel and the reconstruction of HR image are processed separately, which is tedious and inflexible. Recently, the Super-Resolution Challenge [22] introduces a new competition for image SR with unknown downsampling operators. However, most participant methods are not specifically designed for blind SR and perform image SR without explicit blur kernel inference. Though deep learning approaches have been well studied to handle blur kernels in the field of image deblurring [23,24], they mainly concentrate on image deconvolution and are often hard to adapt to the task of single image SR [8]. In comparison, we propose to solve blind SR with a mixture of deep networks, which achieves superior performance and is computationally more efficient.

Ensemble learning has been widely applied in many computer vision tasks such as image classification [25] and object detec-

tion [26]. Some non-blind SR works [27–29] also explore mixture models and yield considerable performance improvement. Among them, a mixture of experts method is introduced in [27] to jointly learn the feature space partition and local regression models. Both [28] and [29] propose to combine multiple sparse coding networks [18] into an ensemble to solve the SR problem. In contrast, we focus on solving blind SR problems with mixture of networks. Besides, the derived lower bound of the likelihood function to address optimization intractability as well as our pre-training method for the encoder network have not been explored in these existing works.

3. Proposed method

3.1. Formulation

Given the LR image $\boldsymbol{L}$ and its HR counterpart $\boldsymbol{H}$, most learning based SR methods maximize the conditional distribution $p(\boldsymbol{H}|\boldsymbol{L};\boldsymbol{\theta})$ to find the optimal model parameter $\boldsymbol{\theta}^{\star}$:

$$\boldsymbol{\theta}^{\star} = \arg\max_{\boldsymbol{\theta}} \log p\big(\boldsymbol{H}|\boldsymbol{L};\boldsymbol{\theta}\big). \tag{2}$$

Gibbs distribution [30] is widely-adopted to model the conditional distribution, *e.g.*, $p(\boldsymbol{H}|\boldsymbol{L};\boldsymbol{\theta}) \propto \exp(-||\boldsymbol{H} - \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L})||^2)$, making the above maximum likelihood equivalent to the following regression problem:

$$\arg\min_{\boldsymbol{\theta}} ||\boldsymbol{H} - \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L})||^2, \tag{3}$$

where $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L})$ denotes a mapping function to predict the HR image from $\boldsymbol{L}$. In non-blind image SR, mapping function $\boldsymbol{F}_{\boldsymbol{\theta}}(.)$ is learned by assuming that the blur kernel for generating LR images is known and fixed.

This paper considers the problem of blind image SR, where the blur kernel is unknown at test time. We model the unknown kernel of the LR image $\boldsymbol{L}$ with a latent variable $z \in \{1, 2, \ldots, M\}$ with M states. Thus, the marginal distribution of $\boldsymbol{H}$ conditioning on $\boldsymbol{L}$ can be computed by:

$$\log p\big(\boldsymbol{H}|\boldsymbol{L};\boldsymbol{\theta}\big) = \log \sum_{z} p(z|\boldsymbol{L})\, p(\boldsymbol{H}|z,\boldsymbol{L};\boldsymbol{\theta}). \tag{4}$$

Note that the sum of terms emerges inside the logarithm, making direct maximization of the log marginal probability intractable [31]. To circumvent this problem, we introduce another parameterized distribution $q(z|\boldsymbol{L};\boldsymbol{\phi})$ to approximate the true conditional distribution $p(z|\boldsymbol{L})$. The marginal distribution can then be derived as follows:

$$\begin{aligned}\log p\big(\boldsymbol{H}|\boldsymbol{L};\boldsymbol{\theta}\big) &= \log \sum_{z} q(z|\boldsymbol{L};\phi)\, \frac{p(\boldsymbol{H},z|\boldsymbol{L};\boldsymbol{\theta})}{q(z|\boldsymbol{L};\boldsymbol{\phi})} \\ &= \log E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\left[\frac{p(\boldsymbol{H},z|\boldsymbol{L};\boldsymbol{\theta})}{q(z|\boldsymbol{L};\boldsymbol{\phi})}\right].\end{aligned} \tag{5}$$

According to Jensen's inequality [31], the following relationships are derived:

$$\begin{aligned}&\log E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\left[\frac{p(\boldsymbol{H},z|\boldsymbol{L};\boldsymbol{\theta})}{q(z|\boldsymbol{L};\boldsymbol{\phi})}\right] \\ &\quad\geq E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}[\log p(\boldsymbol{H},z|\boldsymbol{L};\boldsymbol{\theta})] - E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\big[\log q(z|\boldsymbol{L};\boldsymbol{\phi})\big] \\ &\quad= E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\big[\log p(\boldsymbol{H}|z,\boldsymbol{L};\boldsymbol{\theta})\big] - D_{KL}\big(q(z|\boldsymbol{L};\boldsymbol{\phi})||p(z|\boldsymbol{L})\big),\end{aligned} \tag{6}$$

where $D_{KL}(q||p)$ is the Kullback-Leibler (KL) divergence between q and p. The right hand side (RHS) of the above inequality, denoted as $J(\boldsymbol{\theta},\boldsymbol{\phi})$, is equivalent to the marginal likelihood $p(\boldsymbol{H}|\boldsymbol{L};\boldsymbol{\theta})$ up to a constant and named as the evidence lower bound (ELBO) [32]. It can be maximized to find $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ that provide as tight a bound as possible on the marginal likelihood. To this end, we adopt the following Gibbs distribution to model $p(\boldsymbol{H}|z,\boldsymbol{L};\boldsymbol{\theta})$:

$$p\big(\boldsymbol{H}|z,\boldsymbol{L};\boldsymbol{\theta}\big) \propto exp\left(-\frac{||\boldsymbol{H} - \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)||^2}{\alpha}\right), \tag{7}$$

where $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)$ denotes the mapping function to predict the HR image given the LR image $\boldsymbol{L}$ and variable z, and α denotes the scaling parameter. Maximization of the lower bound can then be rewritten as the following minimization problem:

$$\begin{aligned}\arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} -J(\boldsymbol{\theta},\boldsymbol{\phi}) = \arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} &E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\big[||\boldsymbol{H} - \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)||^2\big] \\ &+ \alpha D_{KL}\big(q(z|\boldsymbol{L};\boldsymbol{\phi})||p(z|\boldsymbol{L})\big),\end{aligned} \tag{8}$$

where the first term in the RHS encourages a small expected reconstruction error, while the second one as a regularization for the parameter $\boldsymbol{\phi}$ forces the approximate posterior $q(z|\boldsymbol{L};\boldsymbol{\phi})$ to be close to the true distribution $p(z|\boldsymbol{L})$. The scale parameter α is used to balance their effects.

Table 1
Architecture details of E-Net. C: convolutional layer; L: Leaky ReLU; B: batch normalization; G: global average pooling; F: fully-connected layer; S: softmax.

Layers	1	2	3	4	5
Type	C+B+L	C+B+L	C+B+L	C+B+L	G+F+S
Output Channels	32	64	128	256	4
Filter size		3×3			–
Filter stride		2			–

3.2. Blind SR with mixture of networks

The lower bound described in (6) provides us with a very general framework, which can be implemented by a variety of learning systems. In this paper, we model the approximate conditional distribution $q(z|\boldsymbol{L};\boldsymbol{\phi})$ and the mapping function $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)$ in the lower bound with deep neural networks to leverage their strong learning capabilities. Specifically, the random variable z modeling the information of blur kernels is predicted by an encoder network, which takes the LR image as input and infers the probability $q(z|\boldsymbol{L};\boldsymbol{\phi})$ of latent variable z. For the mapping function, since the SR tasks with different blur kernels are inherently different, training a single network for all blur kernels may struggle to capture their subtle differences. A more suitable manner to address this problem is to adopt a divide-and-conquer strategy, *i.e.*, decomposing the SR tasks into groups of similar sub-tasks according to their blur kernels and assigning different groups to different networks. As a consequence, the networks are specifically learned for certain similar tasks, making network training easier and inference more accurate. Based on this observation, we adopt a mixture of deep networks to implement the mapping function $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)$. Fig. 2 illustrates the overall architecture of our model. The details about the architecture design and network inference are presented in the following.

3.2.1. Latent variable inference with encoder network

The encoder network, namely E-Net, is a convolutional neural network (CNN) with 5 trainable layers, which takes the LR image $\boldsymbol{L}$ as input and predicts the probability $q(z|\boldsymbol{L};\boldsymbol{\phi})$ for the latent variable. Table 1 provides the detailed network architecture. The first 4 layers are convolutional layers with a spatial stride of 2×2, each followed by a batch normalization and a Leaky ReLU. The feature maps produced by the last convolutional layer are aggregated into a fixed-length feature vector through a global average pooling layer. To alleviate over-fitting, we also apply dropout operation to the feature vector during training. A fully-connected layer then takes the aggregated feature to produce an M-dimensional vector

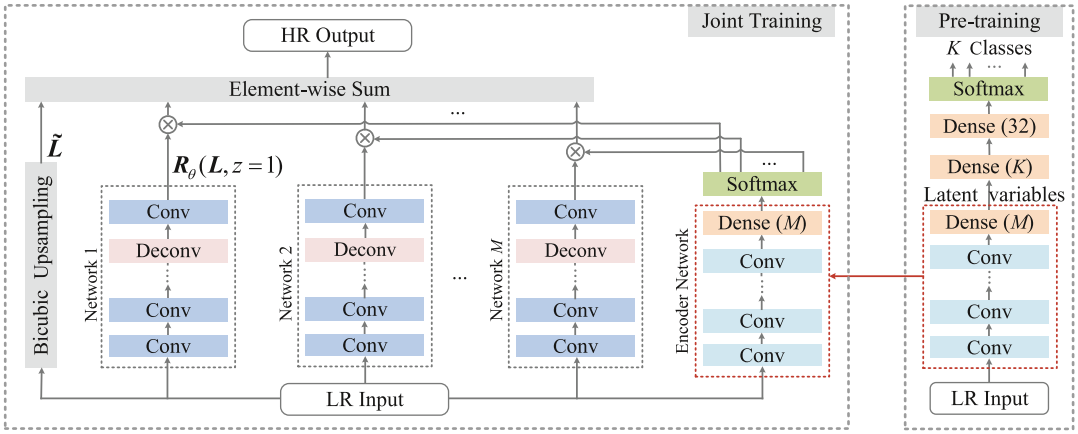


Fig. 2. The architecture of the proposed method for blind single image SR.

Table 2
Architecture details of the network in the mixture. C: convolutional layer; R: ReLU; D: deconvolutional layer; u: upscaling factor.

Part	Feature extraction	Mapping	Reconstruction	
Layers	1	2-10	11	12
Type	C+R	C+R	D+R	C
Output Channel	64	64	8	1
Filter size	3 × 3	3 × 3	$(u+2)\times(u+2)$	3 × 3
Filter stride	1	1	u	1

$\boldsymbol{s}=[s_1,\ldots,s_M]$ corresponding to confidence scores of all possible states of the latent variable. Finally, the score vector is normalized through a Softmax layer into the posterior probability $\boldsymbol{q}=[q_1,\ldots,q_M]$, where $q_i=\exp(s_i)/\sum_{j=1}^{M}\exp(s_j)$ represents the posterior probability $q(z=i|\boldsymbol{L},\boldsymbol{\phi})$.

The latent variable inference can be interpreted as the categorization of input LR images into M different clusters according to their corresponding blur kernels. LR images belonging to the same cluster are likely to be generated by similar blur kernels. Therefore, their SR processes are highly correlated and can be performed by specific networks. The latent variable z then serves as the cluster indicator to select suitable networks for image SR.

3.2.2. Mapping function based on mixture of networks

The mapping function $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)$ is implemented by a mixture of M networks, with each corresponding to a specific state of the latent variable. All networks in the mixture share the same network architecture with independently learned network parameters. Many CNN-based SR models [5,15,18] can work as the network in the mixture. As detailed in Table 2, the network architecture is designed by following [20], which can be divided into three parts. The first part is a convolutional layer used for feature extraction. The second part performs non-linear mapping using 9 convolutional layers with skip connections between consecutive layers. The last part reconstructs HR image content with a deconvolutional layer and a convolutional layer. ReLUs are adopted as non-linear units following each layer except for the last one. Besides, we explore the residual learning idea to facilitate more accurate HR restoration. For the zth network, it takes the LR image $\boldsymbol{L}$ as input and is trained to reconstruct the residual $\boldsymbol{H}-\tilde{\boldsymbol{L}}$ between the HR image $\boldsymbol{H}$ and bicubic-interpolated LR image $\tilde{\boldsymbol{L}}$. The final prediction of the zth network is then computed as $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)=\tilde{\boldsymbol{L}}+\boldsymbol{R}(\boldsymbol{L},z)$, with $\boldsymbol{R}(\boldsymbol{L},z)$ denoting the reconstructed residual.

In an ideal case, each network in the mixture is specifically trained for a group of SR tasks with similar blur kernels. At test time, each network performs non-blind image SR by assuming that the input LR image is generated by the blur kernels for which it is trained. Next, we describe how to aggregate the predictions of all the networks to infer the final HR image.

3.2.3. HR image inference

Given an input LR image $\boldsymbol{L}$ at test time, we obtain the posterior probability $q(z|\boldsymbol{L};\boldsymbol{\phi})$ inferred by E-Net and a set of HR predictions $\{\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)|z=1,2,\ldots,M\}$, where the zth prediction is made by the zth network in the mixture. The final HR image is inferred by maximum a posterior estimation. Similar to the training stage, we use the lower bound derived in (6) to approximate the posterior. The optimization problem to infer the HR image can be written as follows:

$$\begin{aligned}\boldsymbol{H}^{\star} &= \arg\max_{\hat{\boldsymbol{H}}}\ p(\hat{\boldsymbol{H}}|\boldsymbol{L};\boldsymbol{\theta})\\ &= \arg\max_{\hat{\boldsymbol{H}}}\ E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\left[\log p(\hat{\boldsymbol{H}}|z,\boldsymbol{L};\boldsymbol{\theta})\right]-D_{KL}\left(q(z|\boldsymbol{L};\boldsymbol{\phi})||p(z|\boldsymbol{L})\right)\\ &= \arg\min_{\hat{\boldsymbol{H}}}\ E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}\left[\|\hat{\boldsymbol{H}}-\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)\|^2\right].\end{aligned}\tag{9}$$

The KL-divergence in the lower bound is ignored since it does not depend on the HR image. By setting the derivative of (9) *w.r.t.* $\hat{\boldsymbol{H}}$ to zero, we obtain the closed-form solution as

$$\boldsymbol{H}^{\star}=E_{q(z|\boldsymbol{L};\boldsymbol{\phi})}[\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)]=\sum_{z=1}^{M}q_z\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)=\sum_{z=1}^{M}q_z\boldsymbol{R}_{\boldsymbol{\theta}}(\boldsymbol{L},z)+\tilde{\boldsymbol{L}},\tag{10}$$

where $\tilde{\boldsymbol{L}}$ denotes the bicubic-interpolated LR image; $\boldsymbol{R}(\boldsymbol{L},z)$ indicates the predicted residual; the posterior probability q_z inferred by E-Net serves as the weight to combine predictions of all the networks in the mixture.

3.3. Network training

Thus far, we have implemented the approximate posterior probability $q(z|\boldsymbol{L};\boldsymbol{\phi})$ and the mapping function $\boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L},z)$ in the objective function (8) using E-Net and mixture of networks, respectively. In order to train the networks using the objective function, we still need to figure out the true posterior probability $p(z|\boldsymbol{L})$ in the KL-divergence term that acting as a regularization on E-Net. Instead of simply assuming a uniform distribution for the latent variable z, we propose a pre-training method for E-Net, which can not only initialize E-Net to render a more effective joint training stage, but also provide reasonable estimations for the true posterior $p(z|\boldsymbol{L})$.

3.3.1. Pre-training E-Net for blur kernel classification

E-Net aims to capture the structural information of blur kernels used to generate the input LR images. In order to properly initialize E-Net, we propose to pre-train E-Net for the task of LR image

classification according to their corresponding blur kernels. Given a set of N HR images $\{\boldsymbol{H}_n\}_{n=1}^{N}$, and a set of K different blur kernels $\{\boldsymbol{B}_k\}_{k=1}^{K}$, we generate LR images for each HR image using all the blur kernels via (1). Consequently, we obtain a set of training triplets $\{\boldsymbol{H}_i, \boldsymbol{L}_i, k_i\}_{i=1}^{N\times K}$, where $\boldsymbol{L}_i$ is down-sampled from $\boldsymbol{H}_i$ using the k_ith blur kernel. Note that the last fully-connected layer of E-Net produces an M-dimensional score vector $\boldsymbol{s}$, corresponding to the M states of the latent variable. Since the number of blur kernels K is often significantly larger than M, we modify E-Net by adding an additional classification module at the end, consisting of two fully-connected layers followed by a Softmax layer (See the right panel in Fig. 2). The classification module takes the score vector $\boldsymbol{s}$ as input and predicts the probability $p(k|\boldsymbol{L};\boldsymbol{\phi})$ for each blur kernel conditioning on the input LR image $\boldsymbol{L}$. E-Net as well as the classification module can be pre-trained in an end-to-end manner for blur kernel identification using the following cross-entropy loss:

$$\arg\min_{\boldsymbol{\phi}} -\frac{1}{N\times K}\sum_{i=1}^{N\times K}\log p\big(k_i|\boldsymbol{L}_i;\boldsymbol{\phi}\big). \tag{11}$$

After training convergence, the additional classification module can be discarded. The pre-trained E-Net captures important knowledge for discrimination of the blur kernels used to generate the input LR images, serving as a good initial point for the joint training stage. In the following, we explore these knowledge to regularize the joint training.

3.3.2. Joint training E-Net and mixture of networks for SR

In the joint training stage, the mixture of networks is randomly initialized, while E-Net is initialized using the pre-trained network parameters. We feed each LR image $\boldsymbol{L}$ in the training set into the pre-trained E-Net to obtain the score vector $\boldsymbol{s} = [s_1, \ldots, s_M]$ and the normalized posterior probabilities $\{\hat{q}(z|\boldsymbol{L}, \hat{\boldsymbol{\phi}})|z = 1, \ldots, M\}$, where $\hat{\boldsymbol{\phi}}$ denotes the pre-trained parameters of E-Net and $\hat{q}(z|\boldsymbol{L}, \hat{\boldsymbol{\phi}}) = \frac{\exp(s_z)}{\sum_{j=1}^{M}\exp(s_j)}$. Through pre-training, the score vector $\boldsymbol{s}$ has already learned to encode all the information to infer the blur kernel of the input LR image, and reveals the geometric structure in the distribution of the latent variable z. Therefore, we use the corresponding normalized posterior probability $\hat{q}(z|\boldsymbol{L}, \hat{\boldsymbol{\phi}})$ as an initial guess of the true probability $p(z|\boldsymbol{L}_i)$. The mixture of networks as well as E-Net can then be jointly trained for blind image SR by minimizing the objective function (8) which is rewritten as:

$$\begin{aligned}\arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} &\sum_{i=1}^{N\times K} E_{q(z|\boldsymbol{L}_i;\boldsymbol{\phi})}\Big[||\boldsymbol{H}_i - \boldsymbol{F}_{\boldsymbol{\theta}}(\boldsymbol{L}_i, z)||^2\Big] \\ &+ \lambda D_{KL}\Big(q(z|\boldsymbol{L}_i;\boldsymbol{\phi})||\hat{q}(z|\boldsymbol{L}_i;\hat{\boldsymbol{\phi}})\Big).\end{aligned} \tag{12}$$

The first term in (12) enforces the mixture of networks to accurately restore the HR images; while the second term constrains the posterior probability $q(z|\boldsymbol{L}_i;\boldsymbol{\phi})$ predicted by E-Net to be close to that inferred by the pre-trained network. The objective function (12) is reminiscent of the distillation training technique [33], which aims to transfer knowledge from a pre-trained ensemble model to a single tiny model by penalizing the disagreement between their predictions. Ours bears a similar idea but is derived from a different view (*i.e.*, the ELBO of the likelihood function) in a principled manner.

3.3.3. Implementation details

For parameter initialization, weights in convolutional and fully-connected layers are randomly initialize using technique described in [34]. The deconvolutional filters are initialized from a zero-mean Gaussian distribution with standard deviation of 0.01. In the pre-training stage, E-Net with its additional classification module is trained by about 60 epochs. The learning rate is initialized to $1e-2$ and is decreased by a factor of 10 every 20 epochs. In the joint training stage, the initial learning rates of the mixture model and E-Net are set to $1e-2$ and $1e-4$, respectively, and are decreased by a factor of 10 when the validation loss is stabilized. The dropout rate is set to 0.5 for E-Net during all the training procedure. The parameter λ in Eq. (12) is set to 0.05 through cross-validation. All objective functions are optimized using stochastic gradient descent with a batch size of 64, and a momentum of 0.9. We also use the gradient clipping [15] with a clipping range of $[-0.6, 0.6]$ to avoid gradient exploding problem. Code and models can be found at https://github.com/yifanw90/BlindSR.

4. Experiment

4.1. Set up

We adopt 58 blur kernels $\{\boldsymbol{B}_k\}_{k=1}^{58}$ following [8], which are 2D Gaussian kernels of size 11×11 pixels, with zero mean and various covariances. The eigenvalues of the covariances range from 0.75 to 3 with a step of 0.75, while the orientation of eigenvectors lie in the range from 0 to π with a step of $\pi/8$. We randomly sample half of them as "seen blur kernels" for training, and the rest half ones serve as "unseen blur kernels" to test the generalization ability of the trained models.

Our training data consists of the widely-adopted 91 images proposed by [1] and the dataset introduced in [20]. Data augmentation including random rotation and flipping is applied to reduce overfitting. Our experiments are performed on three upscaling factors, *i.e.*, 2, 3, and 4. Given an upscaling factor $u \in \{2, 3, 4\}$ and a training HR image, we generate the corresponding LR image by first blurring the HR image with each of the 29 seen kernels and then downsampling the blurred HR image by the scale factor u. We further densely crop LR-HR image patches from the LR-HR image pairs with the size of $w_l \times w_l$ and $w_h \times w_h$, respectively, where w_l is set to 24 pixels and $w_h = u \cdot w_l$. In the training process, the cropped LR and HR image patches serve as the inputs and ground truths of the network, respectively.

We evaluate the performance on five public data sets: Set5 [35], Set14 [2], BSDS200 [36], Urban100 [37], and Manga109 [38]. Two groups of LR images for testing are generated with the 29 seen and 29 unseen kernels, respectively, using the same method described above. We adopt PSNR and SSIM metrics for quantitative evaluation. The super-resolution and evaluation operations are only performed on the luminance channel in YCbCr color space.

4.2. Investigation of different settings

4.2.1. Comparisons of different ensemble sizes

We evaluate the performance by varying the number of networks M in the mixture. Fig. 3 reports the average PSNR of different ensemble size on BSDS200 for upscaling factor 2, where the red and blue lines indicate the performance on images blurred by seen and unseen kernels, respectively. Both of them steadily increase with the increasing of M until saturation at around $M = 6$. It clearly indicates that a mixture of multiple networks is more superior to a single network. However, when the number of networks is sufficient to cover the variations caused by different blur kernels, further improvements are marginal. Considering both effectiveness and efficiency, we choose a mixture of $M = 4$ networks in the subsequent experiments.

4.2.2. Analysis on E-Net and pre-training method

We further design four different variants of the proposed method to justify the contributions of E-Net and the pre-training method to the final performance. In the first variant (named as Averaged Ensemble), E-Net is discarded from the proposed method,

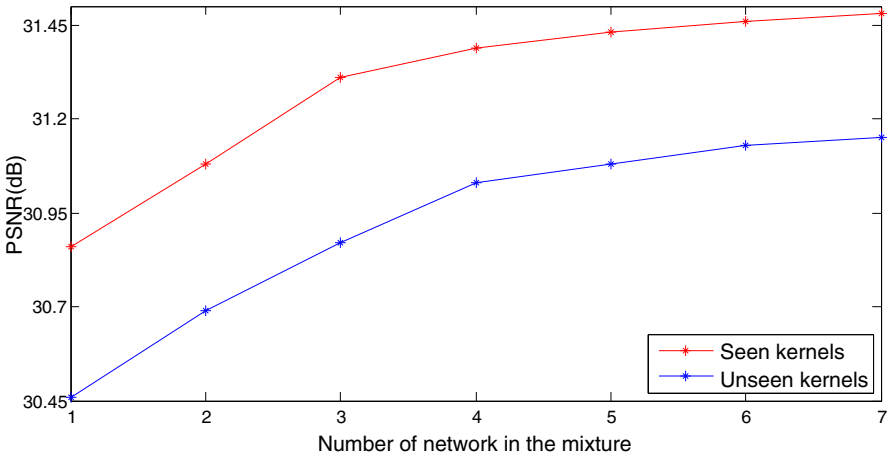


Fig. 3. Average PSNR (dB) with different numbers of networks in the mixture over BSDS200 for upscaling factor 2. The red and blue lines indicate the performance on images blurred by seen kernels and unseen kernels, respectively.

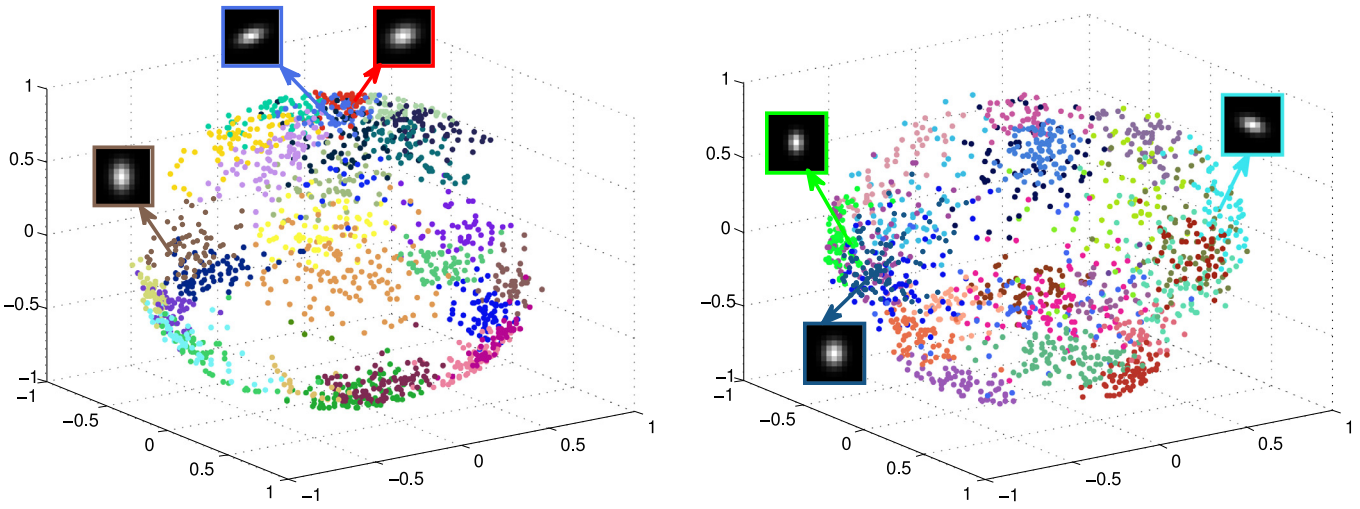
Table 3
Average PSNR (dB) comparisons on three data sets blurred by 29 seen and unseen kernels among different ensemble strategies.

Test kernels	29 seen blur kernels				29 unseen blur kernels			
Method	Scale	Set5	Set14	BSDS200	Scale	Set5	Set14	BSDS200
Averaged Ensemble	× 2	35.65	31.77	31.50	× 2	34.96	31.32	31.02
	× 3	32.46	29.03	28.60	× 3	32.43	28.97	28.57
	× 4	30.30	27.35	27.04	× 4	30.34	27.32	27.01
Without Pre-training	× 2	35.68	31.80	31.47	× 2	35.01	31.30	31.06
	× 3	32.43	29.07	28.61	× 3	32.42	29.03	28.55
	× 4	30.36	27.34	27.11	× 4	30.29	27.33	27.04
Hard Selection	× 2	35.48	31.68	31.39	× 2	34.88	31.19	29.97
	× 3	32.31	28.94	28.49	× 3	32.07	28.60	28.24
	× 4	30.19	27.21	26.98	× 4	30.11	27.13	26.94
Hard Selection with Training	× 2	35.60	31.74	31.49	× 2	34.94	31.34	31.06
	× 3	32.45	29.01	28.55	× 3	32.42	28.99	28.52
	× 4	30.29	27.32	27.05	× 4	30.31	27.35	27.09
Ours	× 2	35.86	31.99	31.63	× 2	35.18	31.52	31.18
	× 3	32.74	29.21	28.78	× 3	32.69	29.17	28.77
	× 4	30.52	27.57	27.22	× 4	30.61	27.63	27.28

and the outputs of different networks in the mixture are simply aggregated through averaging to produce the final prediction. In the second variant (named as Without Pre-training), the proposed network architecture is kept unchanged. However, the pre-training of E-Net is not performed. Both E-Net and the mixture of networks are jointly trained from scratch for blind image SR. The latent variable z is assumed to be uniformly distributed with a constant posterior probability $p(z|\boldsymbol{L}) = \frac{1}{M}$ in (8). In the third variant (named as Hard Selection), We keep the architecture and training strategy unchanged. At test time, we only adopt the prediction of the network that has the maximum posterior probability predicted by E-Net as the final output. This variant is equivalent to the hard selection of the mixture networks. In the forth variant, after pre-training E-Net, we train D-Net and E-Net under the hard selection setting. Since the hard selection operation is not fully differentiable, we propose to optimize the networks by iterating between two-stages. In the first stage, we fix E-Net and train the mixture of networks by minimizing the Mean Squared Error between the ground truth and the network output with the maximum posterior probability. In the second stage, we fix the network mixture and optimize E-Net by minimizing the first term of (12). We name this variant as "Hard Selection with Training" to distinguish it from the third one.

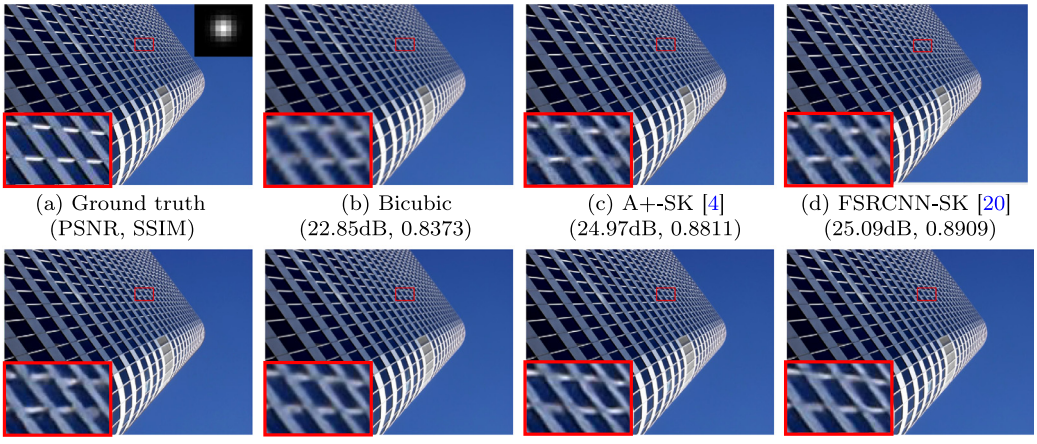
Table 3 reports the performance of the above four settings as well as the proposed method on four data sets blurred by seen and unseen blur kernels. The comparison results on both two kernel groups suggest that even without pre-training, E-Net provides a more elegant and effective solution for aggregating the mixture networks than the averaging based ensemble strategy. The performance gain brought by E-Net should be attributed to the fact that E-Net learns to select the most appropriate networks in the mixture to perform SR according to the blur pattern identified from the input image, ensuring more accurate and specialized restoration of HR details. The proposed pre-training strategy allows E-Net to better encode the blur kernel information and further improves the performance with a considerable margin compared to training E-Net from scratch. Besides, directly using the output of the single network that has the maximum posterior probability delivers degenerated performance. The performance of "Hard Selection with Training" improves the performance of its baseline but is still unsatisfactory especially on the unseen kernels. The proposed method outperforms the hard selection settings by a large margin, confirming that networks in the mixture are highly correlated and collaborative.

For a more comprehensive understanding of E-Net and the pre-training strategy, we perform visualization of E-Net prediction. We randomly sample 100 images from BSDS200, and generate two groups of 2900 LR images with 29 seen and unseen blur kernels, respectively. We use the pre-trained E-Net to predict the posterior probability $\hat{\boldsymbol{q}}^{(i)} = [\hat{q}_1^{(i)}, \hat{q}_2^{(i)}, \hat{q}_3^{(i)}, \hat{q}_4^{(i)}]$ for each LR image $\boldsymbol{L}_i$, with $\hat{q}_z^{(i)}$ denoting $\hat{q}(z|\boldsymbol{L}_i; \hat{\boldsymbol{\phi}})$. Principal components analysis is further adopted to map each 4-dimensional probability vector into a data point in the 3-dimensional space as visualized in Fig. 4.



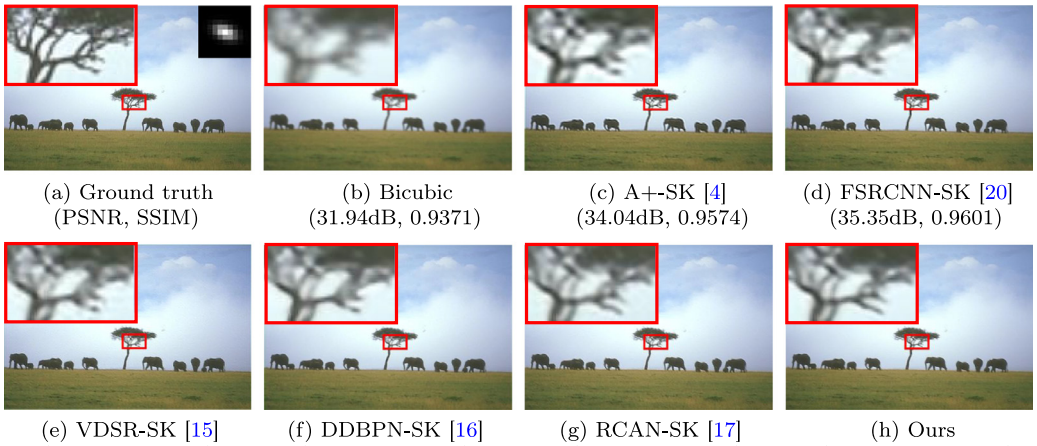
(a) Seen kernels (b) Unseen kernels

Fig. 4. Visualization of the prediction by E-Net. Each data point represents an LR image with color indicating the corresponding blur kernel. Three sampled blur kernels are also depicted. The embedded space is distance-preserving *w.r.t.* the blur kernels.



(a) Ground truth (PSNR, SSIM)
(b) Bicubic (22.85dB, 0.8373)
(c) A+-SK [4] (24.97dB, 0.8811)
(d) FSRCNN-SK [20] (25.09dB, 0.8909)
(e) VDSR-SK [15] (25.34dB, 0.9062)
(f) DDBPN-SK [16] (25.44dB, 0.9103)
(g) RCAN-SK [17] (25.67dB, 0.9126)
(h) Ours (**26.17dB**, **0.9243**)

Fig. 5. SR results on "img-005" from Urban100 with upscaling factor 4. The unseen blur kernel is shown at the top-right corner of (a), and **bold** indicates the best performance.



(a) Ground truth (PSNR, SSIM)
(b) Bicubic (31.94dB, 0.9371)
(c) A+-SK [4] (34.04dB, 0.9574)
(d) FSRCNN-SK [20] (35.35dB, 0.9601)
(e) VDSR-SK [15] (35.49dB, 0.9562)
(f) DDBPN-SK [16] (35.67dB, 0.9606)
(g) RCAN-SK [17] (36.14dB, 0.9615)
(h) Ours (**36.43dB**, **0.9673**)

Fig. 6. SR results on "253036" from BSDS200 with upscaling factor 2. The unseen blur kernel is shown at the top-right corner of (a), and **bold** indicates the best performance.

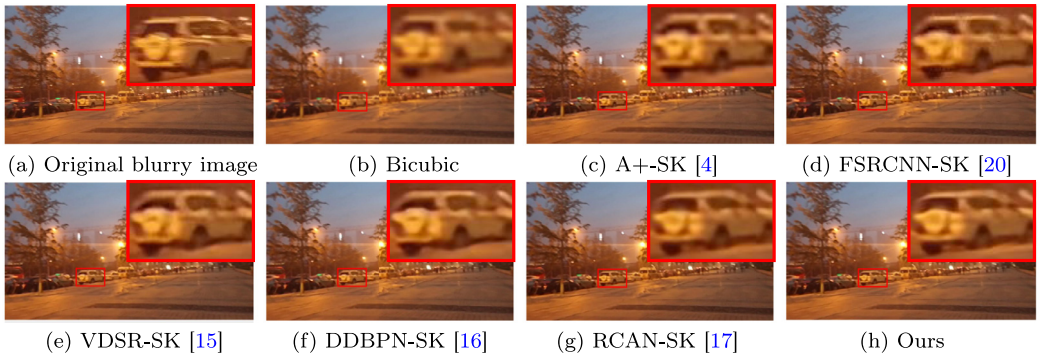


Fig. 7. SR results on real-world motion-blurred images with upscaling factor 4.

Fig. 8. Test images from BSDS200 dataset used for quantitative evaluation of blind SR method. Left to right, top to bottom: (a)-(j). The red box indicates the kernel from the 29 unseen blur kernels, which used to generate LR images.

LR clusters from both seen kernels (*c.f.* Fig. 4(a)) and unseen kernels (*c.f.* Fig. 4(b)) show similar distributions: the LR images are grouped into different clusters according to the blur kernels. In addition, clusters close to one another have similar blur kernels and thus their SR tasks are highly correlated. Meanwhile, clusters that are scattered apart have much different blur kernels. These observations suggest that E-Net can effectively encode the LR image space into a latent embedding which captures the geometric structure of LR image space and are distance-preserving *w.r.t.* the blur kernels. The visualization results justify our motivation for using E-Net to aggregate the predictions of networks in the mixture.

4.3. Comparison with state-of-the-arts

We compare the proposed method with Bicubic interpolation and 8 state-of-the-arts, including 7 non-blind SR methods (A+ [4], SRF [14], SRCNN [5], FSRCNN [20], VDSR [15], DDBPN [16], RCAN [17]) and 2 blind SR methods ([11] and [12]). The detailed comparison protocols are described as below.

For the non-blind SR methods, results on only the bicubic kernel are reported in their original papers. In this paper, we train these methods using 29 seen blur kernels, and name the trained methods with the suffix "SK" (trained on 29 seen blur kernels) to distinguish them from the original ones. We respectively evaluate them on 29 seen kernels and 29 unseen kernels in a blind manner. Namely, SR is performed without knowing the corresponding blur kernel in prior. For fair comparison, all the methods are implemented using the source code provided by the authors and trained using the same hardware settings as ours.

Tables 4 and 5 illustrate the comparison results of our method against the above algorithms in terms of PSNR and SSIM. DDBPN-SK and RCAN-SK deliver favorable performance, the superiority of which may be attributed to the strong learning power of the adopted deep networks. Notwithstanding, the proposed method delivers competitive performance on both seen and unseen blur kernels across all the data sets. We believe the performance of the proposed method could be further improved if we use more sophisticated backbones in the mixture of networks. We also report the runtime of our method with four state-of-the-art methods in Table 6, which indicates that the proposed method is the second fastest among the compared methods.

Figs. 5 to 7 present some qualitative comparisons. Among them, Figs. 5 and 6 are examples tested on the two different unseen blur kernel, respectively. Fig. 7 shows the results on a real-world motion-blurred image captured by a camera. It can be observed that the proposed method can produce perceptually more pleasant HR images under different unseen blur situations.

Besides, we compare our method with two recent state-of-the-art blind SR methods [11] and [12] on ten images from BSDS200 dataset (*c.f.* Fig. 8). Each test image is blurred by the unseen blur kernel shown in the red box in Fig. 8, and downsampled by 3 times. Since their source codes and trained models are not publicly available, we use the results provided by the authors for fair comparison. Table 7 shows the quantitative results on the test images. Fig. 9 shows two examples of qualitative results. The comparison

Table 4
Average PSNR(dB) and SSIM comparisons on five data sets blurred by 29 seen blur kernels. **Bold** indicates the best performance.

Method	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSDS200 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	× 2	28.94/0.8599	26.78/0.7750	27.16/0.7671	23.88/0.7353	25.83/0.8479
A+−SK	× 2	34.53/0.9261	30.93/0.8690	30.56/0.8596	27.90/0.8501	33.59/0.9381
SRF-SK	× 2	34.57/0.9259	30.89/0.8681	30.59/0.8585	27.86/0.8509	33.63/0.9383
SRCNN-SK	× 2	34.65/0.9281	30.99/0.8719	30.87/0.8662	27.98/0.8550	33.70/0.9399
FSRCNN-SK	× 2	34.96/0.9365	31.37/0.8780	30.99/0.8744	28.13/0.8636	33.87/0.9433
VDSR-SK	× 2	35.38/0.9440	31.52/0.8861	31.17/0.8805	28.39/0.8759	34.04/0.9563
DDBPN-SK	× 2	35.59/0.9486	31.78/0.8900	31.46/0.8893	28.65/0.8821	34.38/0.9596
RCAN-SK	× 2	**35.89**/0.9553	31.93/0.9015	**31.65**/0.8993	28.84/0.8906	34.57/0.9625
Ours	× 2	35.86/**0.9558**	**31.99**/**0.9078**	31.63/**0.9036**	**28.91**/**0.8924**	**34.65**/**0.9649**
Bicubic	× 3	29.23/0.8540	26.82/0.7620	27.08/0.7520	23.81/0.7189	25.91/0.8363
A+−SK	× 3	31.53/0.8869	28.36/0.7998	27.83/0.7929	25.15/0.7986	29.41/0.8928
SRF-SK	× 3	31.65/0.8887	28.38/0.8007	27.89/0.7935	25.19/0.7972	29.36/0.8932
SRCNN-SK	× 3	31.64/0.8900	28.41/0.8033	27.92/0.7977	25.26/0.7977	29.44/0.8946
FSRCNN-SK	× 3	31.86/0.8963	28.59/0.8097	28.18/0.8004	25.45/0.7999	29.63/0.8996
VDSR-SK	× 3	32.19/0.9041	28.79/0.8170	28.37/0.8080	25.78/0.8036	29.98/0.9087
RCAN-SK	× 3	32.63/0.9193	29.18/0.8303	28.77/**0.8223**	26.23/0.8140	30.51/0.9200
Ours	× 3	**32.74**/**0.9208**	**29.21**/**0.8355**	**28.78**/0.8217	**26.30**/**0.8189**	**30.58**/**0.9229**
Bicubic	× 4	27.53/0.8074	25.49/0.7062	25.95/0.6996	22.68/0.6599	24.20/0.7821
A+−SK	× 4	29.48/0.8622	26.69/0.7515	26.64/0.7299	23.82/0.7223	26.77/0.8386
SRF-SK	× 4	29.56/0.8610	26.77/0.7510	26.68/0.7308	23.86/0.7214	26.72/0.8388
SRCNN-SK	× 4	29.68/0.8636	26.83/0.7529	26.73/0.7319	23.91/0.7237	26.84/0.8419
FSRCNN-SK	× 4	29.85/0.8675	26.91/0.7578	26.88/0.7355	24.04/0.7281	27.01/0.8461
VDSR-SK	× 4	30.11/0.8721	27.15/0.7619	27.00/0.7402	24.21/0.7342	27.25/0.8543
DDBPN-SK	× 4	30.30/0.8800	27.32/0.7697	27.12/0.7527	24.42/0.7433	27.44/0.8676
RCAN-SK	× 4	30.48/0.8822	27.50/0.7728	**27.25**/0.7550	24.54/0.7507	27.62/0.8712
Ours	× 4	**30.52**/**0.8824**	**27.57**/**0.7749**	27.22/**0.7563**	**24.61**/**0.7528**	**27.69**/**0.8735**

Table 5
Average PSNR(dB) and SSIM comparisons on five data sets blurred by 29 unseen blur kernels. **Bold** indicates the best performance.

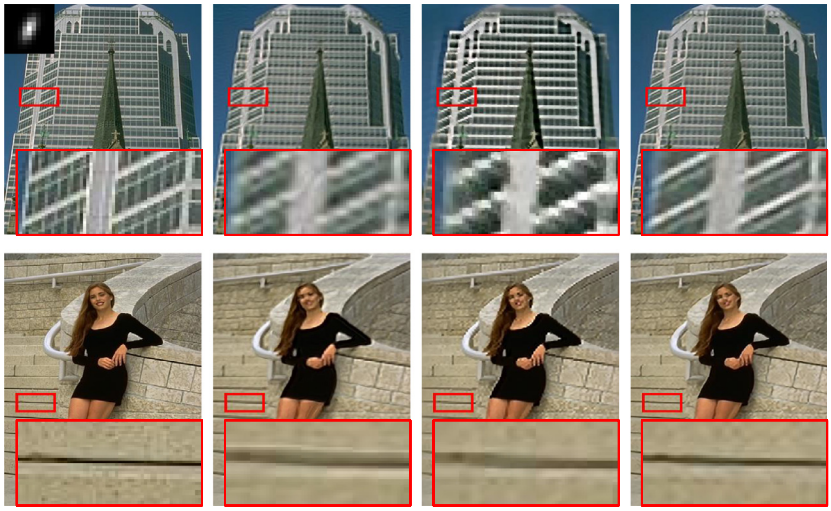
Method	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSDS200 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	× 2	28.85/0.8570	26.69/0.7699	27.08/0.7620	23.80/0.7295	25.71/0.8435
A+−SK	× 2	34.06/0.9172	30.47/0.8560	30.27/0.8405	27.33/0.8488	32.63/0.9295
SRF-SK	× 2	34.11/0.9190	30.45/0.8572	30.33/0.8416	27.35/0.8487	32.70/0.9289
SRCNN-SK	× 2	34.19/0.9216	30.61/0.8590	30.40/0.8433	27.42/0.8496	32.75/0.9318
FSRCNN-SK	× 2	34.35/0.9269	30.79/0.8661	30.51/0.8570	27.68/0.8530	32.90/0.9378
VDSR-SK	× 2	34.68/0.9311	31.01/0.8747	30.70/0.8678	27.89/0.8598	33.15/0.9451
DDBPN-SK	× 2	34.92/0.9424	31.28/0.8800	30.96/0.8800	28.08/0.8677	33.33/0.9503
RCAN-SK	× 2	35.03/0.9451	31.41/0.8849	31.07/0.8891	28.30/0.8752	33.48/0.9532
Our	× 2	**35.18**/**0.9510**	**31.52**/**0.8893**	**31.18**/**0.8934**	**28.42**/**0.8816**	**33.62**/**0.9586**
Bicubic	× 3	29.10/0.8510	26.73/0.7575	27.01/0.7478	23.73/0.7141	25.77/0.8323
A+−SK	× 3	31.46/0.8816	28.15/0.7979	27.65/0.7793	25.05/0.7898	29.18/0.8855
SRF-SK	× 3	31.50/0.8823	28.19/0.7984	27.69/0.7806	25.03/0.7902	29.22/0.8870
SRCNN-SK	× 3	31.55/0.8839	28.27/0.7996	27.75/0.7824	25.19/0.7913	29.30/0.8897
FSRCNN-SK	× 3	31.70/0.8872	28.40/0.8068	27.89/0.7861	25.38/0.7951	29.49/0.8905
VDSR-SK	× 3	31.97/0.8943	28.57/0.8146	28.16/0.7902	25.61/0.8024	29.77/0.9033
RCAN-SK	× 3	32.47/0.9102	28.93/0.8294	28.62/0.8097	26.12/0.8113	30.29/0.9159
Our	× 3	**32.69**/**0.9199**	**29.17**/**0.8332**	**28.77**/**0.8195**	**26.24**/**0.8158**	**30.42**/**0.9208**
Bicubic	× 4	27.50/0.8061	25.48/0.7045	25.96/0.6981	22.68/0.6581	24.18/0.7805
A+-SK	× 4	29.52/0.8599	26.58/0.7504	26.54/0.7283	23.55/0.7150	26.43/0.8311
SRF-SK	× 4	29.49/0.8596	26.57/0.7503	26.50/0.7292	23.53/0.7157	26.48/0.8320
SRCNN-SK	× 4	29.60/0.8607	26.65/0.7515	26.56/0.7397	23.62/0.7166	26.54/0.8341
FSRCNN-SK	× 4	29.73/0.8615	26.78/0.7546	26.69/0.7335	23.81/0.7204	26.77/0.8410
VDSR-SK	× 4	30.06/0.8696	27.02/0.7593	26.87/0.7383	24.03/0.7297	27.08/0.8503
DDBPN-SK	× 4	30.31/0.8784	27.29/0.7688	27.02/0.7497	24.33/0.7421	27.37/0.8656
RCAN-SK	× 4	30.45/0.8800	27.46/0.7711	27.16/0.7526	24.50/0.7488	27.57/0.8698
Our	× 4	**30.61**/**0.8827**	**27.63**/**0.7747**	**27.28**/**0.7562**	**24.67**/**0.7525**	**27.76**/**0.8737**

Table 6
Average runtime(s) comparisons on BSDS200.

Scale	FSRCNN-SK	VDSR-SK	DDBPN-SK	RCAN-SK	Ours
× 2	0.0039	0.078	0.328	0.332	0.026
× 4	0.0017	0.078	0.250	0.429	0.014

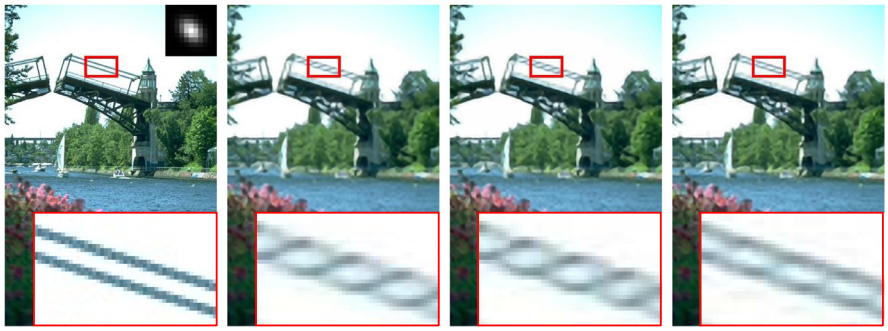
Table 7
PSNR(dB) and SSIM comparisons on test images in Fig. 8. **Bold** indicates the best performance.

Method	Image	a	b	c	d	e	f	g	h	i	j	Mean
[11]	PSNR	25.32	23.29	26.23	27.97	22.28	25.22	23.34	25.01	23.35	28.22	25.02
	SSIM	0.7498	0.7023	0.7181	0.8017	0.5723	0.6376	0.7463	0.8328	0.6482	0.8330	0.7242
[12]	PSNR	26.76	23.89	28.02	29.27	19.60	26.28	24.57	27.58	24.90	29.76	26.06
	SSIM	0.7798	0.7153	0.7544	0.8199	0.5209	0.6649	0.7782	0.9072	0.6991	0.8459	0.7486
Ours	PSNR	**27.37**	**24.38**	**28.69**	**30.02**	**24.03**	**27.02**	**25.29**	**28.50**	**25.38**	**31.27**	**27.19**
	SSIM	**0.8135**	**0.7403**	**0.7863**	**0.8378**	**0.6606**	**0.6942**	**0.8171**	**0.9319**	**0.7153**	**0.8822**	**0.7879**



(a) Ground truth (b) [11] (c) [12] (d) Ours

Fig. 9. SR comparisons with two blind SR methods for upscaling factor 3. The unseen blur kernel is shown at the top-left corner of (a) Ground truth.



(a) Ground truth (PSNR, SSIM) (b) VDSR-SK [15] (23.76dB, 0.7078) (c) RCAN-SK [17] (23.82dB, 0.7122) (d) Ours (23.98dB, 0.7152)

Fig. 10. Failure case on image "22093" from BSDS200 with upscaling factor 4. The unseen blur kernel is shown at the top-right corner of the (a) Ground truth.

between the two blind methods may be biased since the results provided by the authors are rendered by different non-blind SR methods[1] given the estimated kernels. Nonetheless, our improvements upon both methods are significant.

4.4. Limitation

As blind single image SR is a heavily ill-posed problem, some limitations still remain. While images produced by the proposed network look realistic, they can not match the ground truth images exactly. In particular, it is very challenging to reconstruct complex textures and fine details which are largely missing in the LR images for large upscaling factors. Fig. 10 shows a failure case, where the three compared methods fail to predict fine details. However, we believe that our method provides a promising direction for blind SR, and can benefit future research in this area.

5. Conclusion

This paper presents a novel method for blind single image SR via a mixture of deep networks, which decomposes blind SR into groups of sub-tasks according to their blur kernels and learns different networks specifically for different groups. In order to identify various blur kernels, an encoder network (E-Net) is designed to encode the blur information into a latent variable space, which is used to aggregate the predictions of networks in the mixture. A lower bound of the likelihood function is also derived to address the intractability of Maximum Likelihood Estimation. In addition, a pre-training strategy is proposed to provide prior knowledge for E-Net, which results in easier training and more accurate inference for the mixture of networks. Extensive evaluations on bench-

[1] Michaeli and Irani [11] and Shao and Elad [12] conduct blind SR by plugging their estimated kernels into the non-blind SR methods [39] and [40], respectively.

mark data sets validate the effectiveness of the proposed method. We believe that the proposed mixture model and training strategies can also benefit other related tasks, like image deblurring, denoising, *etc.* In the future, we hope to further improve the robustness of our method under complex scenarios through investigating more recent network designs, and collect large-scale blind SR datasets to facilitate more effective network training.

Acknowledgements

We would like to thank Dr. Tomer Michaeli for his kind help in running the blind SR method [11]. We would like to thank Dr. Wen-Ze Shao for his kind help in running the blind SR method [12]. Their help enables us to make comparisons with [11] and [12]. This work was partially supported by National Natural Science Foundation of China (Grant 61906031, 61471082 and 61671103), China Postdoctoral Science Foundation (Grant 2019M661095), and National Postdoctoral Program for Innovative Talent (Grant BX20190055).

References

[1] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[2] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: Proceedings of International Conference on Curves and Surfaces, 2010, pp. 711–730.
[3] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 275–282.
[4] R. Timofte, V. De Smet, L. Van Gool, A+: adjusted anchored neighborhood regression for fast super-resolution, in: Proceedings of Asian Conference on Computer Vision, 2014, pp. 111–126.
[5] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: Proceedings of European Conference on Computer Vision, 2014, pp. 184–199.
[6] Y. Wang, L. Wang, H. Wang, P. Li, Resolution-aware network for image super-resolution, IEEE Trans Circuits Syst. Video Technol. 29 (5) (2019) 1259–1269.
[7] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, A. Levin, Accurate blur models vs. image priors in single image super-resolution, in: Proceedings of IEEE International Conference on Computer Vision, 2013, pp. 2832–2839.
[8] G. Riegler, S. Schulter, M. Rüther, H. Bischof, Conditioned regression models for non-blind single image super-resolution, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 522–530.
[9] I. Bégin, F.P. Ferrie, Blind super-resolution using a learning-based approach, in: Proceedings of International Conference on Pattern Recognition, 2004, pp. 85–89.
[10] Y. He, K.-H. Yap, L. Chen, L.-P. Chau, A soft MAP framework for blind super-resolution image reconstruction, Image Vis. Comput. 27 (4) (2009) 364–373.
[11] T. Michaeli, M. Irani, Nonparametric blind super-resolution, in: Proceedings of IEEE International Conference on Computer Vision, 2013, pp. 945–952.
[12] W. Shao, M. Elad, Simple, accurate, and robust nonparametric blind super-resolution, in: Proceedings of International Conference on Image and Graphics, 2015, pp. 333–348.
[13] X. Zhao, Y. Wu, J. Tian, H. Zhang, Single image super-resolution via blind blurring estimation and dictionary learning, Neurocomputing 212 (2016) 3–11.
[14] S. Schulter, C. Lesistner, H. Bischof, Fast and accutate image upscaling with super-resolution forests, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 184–199.
[15] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[16] M. Haris, G. Shakhnarovich, N. Ukita, Deep back-projection networks for super-resolution, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1664–1673.
[17] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of European Conference on Computer Vision, 2018, pp. 294–310.
[18] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 370–378.
[19] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, L. Zhang, Convolutional sparse coding for image super-resolution, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 1823–1831.
[20] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: Proceedings of European Conference on Computer Vision, 2016, pp. 391–407.
[21] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3262–3271.
[22] R. Timofte, E. Agustsson, L.V. Gool, M.-H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, X. Wang, Y. Tian, K. Yu, NTIRE 2017 challenge on single image super-resolution: methods and results, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1110–1121.
[23] J. Sun, W. Cao, Z. Xu, J. Ponce, Learning a convolutional neural network for non-uniform motion blur removal, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 769–777.
[24] C.J. Schuler, M. Hirsch, S. Harmeling, B. Schölkopf, Learning to deblur, IEEE Trans. Pattern Anal. Mach.Intell. 38 (7) (2016) 1439–1451.
[25] W.-J. Yu, Z.-D. Chen, X. Luo, W. Liu, X.-S. Xu, Delta: a deep dual-stream network for multi-label image classification, Pattern Recognit. 91 (2019) 322–331.
[26] J. Wang, X. Tao, M. Xu, Y. Duan, J. Lu, Hierarchical objectness network for region proposal generation and object detection, Pattern Recognit. 83 (2018) 260–272.
[27] K. Zhang, B. Wang, W. Zuo, H. Zhang, L. Zhang, Joint learning of multiple regressors for single image super-resolution, IEEE Signal Process. Lett. 23 (1) (2016) 102–106.
[28] D. Liu, Z. Wang, N.M. Nasrabadi, T.S. Huang, Learning a mixture of deep networks for single image super-resolution, in: Proceedings of Asian Conference on Computer Vision, 2016, pp. 145–156.
[29] L. Wang, Z. Huang, Y. Gong, C. Pan, Ensemble based deep networks for image super-resolution, Pattern Recognit. 68 (2017) 191–198.
[30] J. Bruna, P. Sprechmann, Y. LeCun, Super-resolution with deep convolutional sufficient statistics, in: Proceedings of International Conference on Learning Representations, 2016.
[31] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer, 2009.
[32] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians, J. Am. Stat. Assoc. 112 (518) (2017) 859–877.
[33] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv:1503.02531. (2015).
[34] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[35] M. Bevilacqua, A. Roumy, C. Guillemot, M.-L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Proceedings of British Machine Vision Conference, 2012, pp. 1–10.
[36] 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 IEEE International Conference on Computer Vision, vol. 2, 2001, pp. 416–423.
[37] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.
[38] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, K. Aizawa, Sketch-based manga retrieval using manga109 dataset, Multimed. Tools Appl. 76 (20) (2017) 21811–21838.
[39] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: Proceedings of IEEE International Conference on Computer Vision, 2009, pp. 349–356.
[40] A. Marquina, S. Osher, Image super-resolution by TV-regularization and bregman iteration, J. Sci. Comput. 37 (3) (2008) 367–382.

Yifan Wang is currently pursuing a Ph.D degree in signal and information processing, Dalian University of Technology. Her research interests includes image super-resolution, image enhancement and deep learning.

Lijun Wang received Ph.D. degree in signal and information processing, Dalian University of Technology, in 2019, where he is currently pursuing a postdoctoral research. His current research interests include visual saliency, object tracking, depth prediction, and deep learning.

Hongyu Wang received the Ph.D. degree in precision instrument and optoelectronics engineering from Tianjin University, in 1997. He is currently a professor in School of Information and Communication Engineering, Dalian University of Technology. His current research interests include high spectral image processing, image enhancement, and video surveillance.

Peihua Li received the Ph.D. degree from Harbin Institute of Technology in 2002. He is currently a professor in School of Information and Communication Engineering, Dalian University of Technology. His current research interests include image classification and search using theoretical and computational methods of information geometry.

Huchuan Lu received the Ph.D. degree in system engineering from the Dalian University of Technology (DUT), in 2008. He is currently a professor in School of Information and Communication Engineering, DUT. His current research interests include computer vision and pattern recognition with a focus on visual tracking, saliency detection, and segmentation. He is a member of the ACM and an Associate Editor of the IEEE Transactions on Cybernetics.