



SR-USRN: learning image super-resolution with unified structure and reverse network

Jiayu Ji¹ · Xuanyin Wang¹

Received: 13 January 2022 / Revised: 27 April 2022 / Accepted: 2 July 2022 / Published online: 13 August 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Using high-resolution image as reference (Ref) to recover a low-resolution (LR) image with similar texture can get the lost texture details and achieve more promising super-resolution (SR) results. Nowadays, existing reference-based image super-resolution approaches use a texture transformer network to add texture features to SR network. However, they neglect the importance of the structural consistency of texture transformer network and SR network. We propose a novel image super-resolution algorithm with unified structure and reverse network (SR-USRN), which uses the same network to transform texture and process image SR. SR-USRN consists of three steps, including training SR main network (SR-MainNet) without Ref, training reverse network (ReverseNet) to recover the features in SR-MainNet by Ref image and combining SR-MainNet and ReverseNet to train final SR-USRN. We use Ref image and LR image together to train SR-MainNet in first step and share the parameters in the process of SR and texture transformation. This design makes best use of the Ref images and the same structure of network makes texture transformer know what SR network really needs. The ReverseNet is trained to transform the Ref image to the corresponding features in SR-MainNet. Extensive experiments demonstrate that SR-USRN achieves significant improvements over state-of-the-art approaches on both quantitative and qualitative evaluations.

Keywords Reference-based super-resolution · Share parameters · Reverse network · Texture transformer

1 Introduction

Image SR is a traditional but popular research in low-level vision tasks. Image super-resolution (SR) can be divided into single image super-resolution (SISR), and reference-based image super-resolution (RefSR). In recent years, research on SISR has a great progress. Convolutional neural networks (CNNs) [1] has greatly advanced the SOTA of SISR. However, due to the ill-posed nature of SISR problems, some essential features of the images have been lost. If the texture or content of the images does not occur in training dataset, the effect of the network will decrease. From this point of view, RefSR has a better theoretical basis. RefSR enables the algorithm to transfer similar features from reference images to

SR images. Therefore, these algorithms have better generalization ability.

Compared with SISR, there are few studies on RefSR. The main idea of those researches is to utilize the high-resolution (HR) textures from given Ref image to produce visually pleasing results. However, these approaches have limitations in some aspects. Zhang et al. [3] adopt a feature space defined by a pre-trained classification model to search and transfer textures between the LR and Ref image. As a follow-up work, Yang et al. [4] proposes a learnable texture extractor and uses a hard-attention module and a soft-attention module to transfer and fuse texture features. Nevertheless, these approaches only use Ref image to extract texture features, which does not take full advantage of the important training data. These methods train a texture extractor completely different from the SR network in structure. The network utilization is not high and the texture extractor does not directly give SR network what it needs. To address these problems, this paper proposes a novel image super-resolution network which integrates texture extraction and super-resolution. Specifically, several innovative designs of our network are carried out to break out the limitations of TTSR [4] and SRNTT [3]. The

✉ Xuanyin Wang
xywang@zju.edu.cn

Jiayu Ji
11625039@zju.edu.cn

¹ State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, No. 38 Zheda Road, Hangzhou 310027, Zhejiang, China

overall improvement and innovation enable our method to make full use of Ref image which achieves a better visual result compared with SOTA approaches. The main contributions of this paper are as follows.

To the best of our knowledge, we are one of the first to use same architecture of network to accomplish super resolution and texture transfer. More specifically, the same architecture increases the amount of training data implicitly. We train a reverse network for reference images to generate corresponding texture features. The texture features generated by the reverse network are used to carry out texture transfer. This design makes HR features provide the SR network with what it really needs.

2 Related work

In this section, we review previous works of SISR and RefSR which are the most relevant to our work.

2.1 Single image super-resolution

SISR has been studied for a long time. With the development of deep learning, SISR has a great improvement over traditional non-learning-based methods. These SISR methods can be divided into several groups according to the most distinctive features in their model designs such as linear networks, residual networks, attention-based networks, etc. For more details, Anwar et al. [5] and Wang et al. [6] can be referred.

Super-resolution CNN (SRCNN) [7] proposed by Dong et al. firstly adopts deep learning into SISR by using a three-layer CNN to represent the mapping function. To speed up the SR process, Dong et al. [8] replace the interpolated LR image with the original LR image and adopt deconvolution at the very last layer to enlarge the feature map. VDSR [9], DnCNN [10], etc. proposed lately are linear networks like SRCNN. Enhanced deep residual network (EDSR) [12] modifies the ResNet architecture [13] to work with the SR task. Deeply-recursive convolutional network (DRCN) [15] utilizes recursive learning to solve SR problem. Its motivation is to progressively break down the harder SR problem into a set of simpler ones, which are easy to solve.

To solve the SR problem, a lot of related technologies are carried out. Residual channel attention network (RCAN) [17] designs a channel attention mechanism for each local residual block. To improve perceptual quality of the images, Justin et al. [19] introduces perceptual loss into SR tasks. SR generative adversarial network (SRGAN) [20] adopts generative adversarial networks (GANs) [22] and introduce adversarial loss to increase the SR result. Guo et al. [23] propose a novel dual regression scheme for paired and unpaired data, which carries out a new solution to SR problem.

2.2 Reference-based image super-resolution

Compared to SISR, there are few studies on RefSR which can obtain more accurate details from the Ref image. The early work is to use image aligning or patch matching. Image aligning methods, such as [2,28], must have a good aligning quality between LR and Ref image. The time-consuming aligning approaches are adverse to real applications. Vivek et al. [25] and Zheng et al. [26] used patch match to search proper reference information earlier. Recent years, SR by neural texture transfer (SRNTT) [3] applied patch matching between VGG [27] features of the LR and Ref image to swap similar texture features. In SRNTT, VGG features are untrainable and SRNTT feeds all the swapped features equally into the main network. As a follow-up work of SRNTT, texture transformer network for image SR (TTSR) [4] uses a learnable texture extractor and applies the attention mechanism to the texture features confusion. Due to the different network structure, the features from texture extractor cannot meet the requirements of the main network. These methods only consider Ref image as reference and do not make the most of the reference images. To address these problems, we use the same network structure as main network and texture extractor. Moreover, in the train process of texture extractor which shares the parameters in main network, we make the better use of Ref image.

3 Approach

In this section, we introduce the proposed SR-USRN. Our method is trained in three steps. The process will be discussed in Sects. 3.1, 3.2 and 3.3.

3.1 SR-MainNet

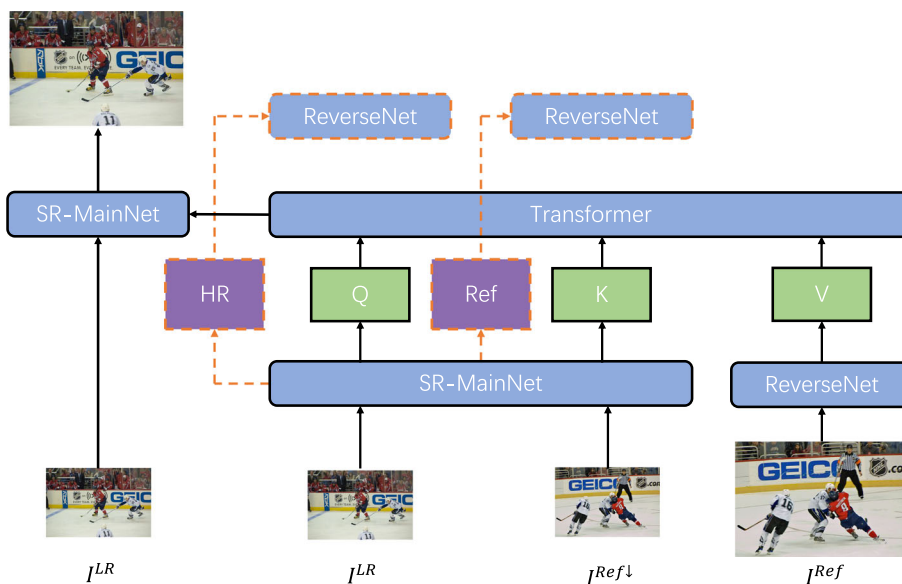
As shown in Fig. 1, the SR-MainNet is the main component of the whole method. Here, we apply the same network to the SR-MainNet and texture extractor. In Fig. 1, I^{LR} , I^{Ref} , I^{HR} and $I^{Ref\downarrow}$ represent the input LR image, the reference image, ground truth of the input image and $4\times$ bicubic-downsampled reference image. In this step, we train two parallel networks. I^{LR} and $I^{Ref\downarrow}$ are fed to SR-MainNet as follows:

$$I^{SR} = \text{SR-MainNet}(I^{LR}) \quad (1)$$

$$I^{RefSR} = \text{SR-MainNet}(I^{Ref\downarrow}) \quad (2)$$

where I^{SR} and I^{RefSR} denote the predicted HR image I^{HR} and Ref image I^{Ref} , respectively. The overall loss function consists of three components in this step. The overall loss can be interpreted as:

Fig. 1 The proposed novel RefSR structure. The orange dotted line represents that it is calculated only during training process. SR-MainNet and ReverseNet in different positions share the parameters, respectively. The Q, K and V are the texture features extracted from a LR image, a down-sampled Ref image and an original Ref image, respectively



$$\mathcal{L}_{\text{overall}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{per}}\mathcal{L}_{\text{per}} \tag{3}$$

The reconstruction loss is essential:

$$\mathcal{L}_{\text{rec}} = \frac{1}{CHW} \|I^{\text{HR}} - I^{\text{SR}}\|_1 + \frac{1}{CHW} \|I^{\text{Ref}} - I^{\text{RefSR}}\|_1 \tag{4}$$

where (C, H, W) is the size of the HR and Ref. To ensure the shape performance and easy convergence, we utilize L_1 loss but not L_2 loss. Our method has two parallel paths, thus the loss has two parts. The following loss functions also have two parts. The adversarial loss is adopted to increase the naturalness of the SR image. As discussed in [4], we adopt WGAN-GP [28] as well. This loss can be interpreted as:

$$\begin{aligned} \mathcal{L}_D &= \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] \\ &\quad + \lambda \mathbb{E}_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} \left[(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2 \right] \\ \mathcal{L}_G &= - \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] \end{aligned} \tag{5}$$

where \mathbb{P}_r and \mathbb{P}_g are the model distribution and real distribution respectively. The aim of adversarial loss is to make the model distribution approximate to the real distribution. For more details, you can refer [28].

This paper applies perceptual loss [3,20,29] to improve visual quality of the SR and RefSR image. Perceptual loss is used to reduce the difference in feature space between the predicted image and the target image. This loss can be interpreted as:

$$\begin{aligned} \mathcal{L}_{\text{per}} &= \frac{1}{C_i H_i W_i} \left\| \phi_i^{\text{vgg}}(I^{\text{SR}}) - \phi_i^{\text{vgg}}(I^{\text{HR}}) \right\|_2^2 \\ &\quad + \frac{1}{C_i H_i W_i} \left\| \phi_i^{\text{vgg}}(I^{\text{RefSR}}) - \phi_i^{\text{vgg}}(I^{\text{Ref}}) \right\|_2^2 \end{aligned} \tag{6}$$

where $\phi_i^{\text{vgg}}(\cdot)$ denotes the i -th layer’s feature map of VGG19, and (C_i, H_i, W_i) represents the shape of the feature map at that layer.

In this step, the preliminary SR-MainNet is trained without reference. HR and Ref image are treated equally. This step serves two purposes. First, we get a SR-MainNet without reference image. Second, the SR-MainNet works as a texture extractor as well, providing the features of the LR and Ref↓ image.

3.2 ReverseNet

Step 1 provides the features of the LR and Ref↓ image. To realize the texture transfer like [4], we should get the features of Ref image. In step 2, the ReverseNet is trained to get the Ref features. The structure of ReverseNet is shown in Fig. 2. This paper chooses VGG19 as the backbone of the ReverseNet. More discussion about the backbone of the ReverseNet is in Sect. 4.3. Then three levels features of Ref images are generated by VGG19, and fed into several convolution layers, i.e. the interface module in Fig. 2. The output of the ReverseNet is the features of Ref image, which has the same form as the features of LR and Ref↓ image. To train the ReverseNet, the output of the SR-MainNet as the input is fed into the ReverseNet. The training process is shown as follows:

$$F^{\text{SR}} = \text{ReverseNet}(I^{\text{SR}}) \tag{7}$$

$$F^{\text{RefSR}} = \text{ReverseNet}(I^{\text{RefSR}}) \tag{8}$$

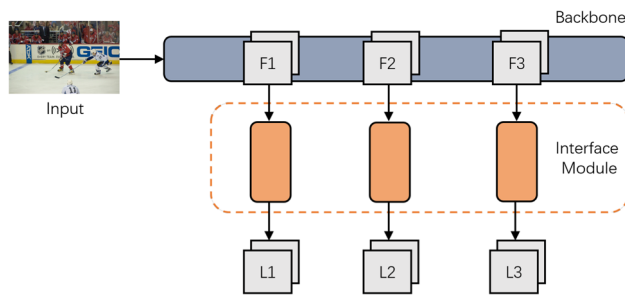


Fig. 2 Reverse Net structure. Blue box is the backbone of the reversenet. The Interface Module includes several convolution layers to process the feature maps from backbone. F1, F2, F3 are three sizes of feature maps captured from backbone. L1, L2, L3 are three levels of texture features which will be fed to SR-MainNet

where F^{SR} and F^{RefSR} are the output of the ReverseNet corresponding to different inputs. F^{SR} and F^{RefSR} each have three levels. Our goal is to make Ref images generate the same type of feature map as LR and Ref \downarrow image. To constrain F^{SR} and F^{RefSR} , the features captured from SR-MainNet are considered as ground truth.

$$F^{\text{LR}} \stackrel{c}{=} \text{SR-MainNet}(I^{\text{LR}}) \quad (9)$$

$$F^{\text{Ref}\downarrow} \stackrel{c}{=} \text{SR-MainNet}(I^{\text{Ref}\downarrow}) \quad (10)$$

where $\stackrel{c}{=}$ denotes F^{LR} and $F^{\text{Ref}\downarrow}$ are the feature maps captured from the SR-MainNet but not the final output. In this step, the parameters of SR-MainNet are fixed. F^{LR} and F^{SR} have the same size in each level. $F^{\text{Ref}\downarrow}$ and F^{RefSR} have the same size in each level as well. The loss function is like this:

$$\mathcal{L}_{\text{rev}} = \frac{1}{CHW} \|F^{\text{LR}} - F^{\text{SR}}\|_1 + \frac{1}{CHW} \|F^{\text{Ref}\downarrow} - F^{\text{RefSR}}\|_1 \quad (11)$$

where (C, H, W) is the size of the feature maps of SR and RefSR. This loss function enables ReverseNet to recover SR-MainNet's middle feature maps from its output. Thus when Ref image is fed into the ReverseNet, the feature output will have the same form as the features of LR and Ref \downarrow image.

3.3 Combine

We train the SR-MainNet without Ref and ReverseNet in last two steps. In this step, two networks are combined to finetune the final SR-MainNet with Ref. The ReverseNet trained in step 2 takes Ref image I^{Ref} as input and outputs V (value) features. The Q (query) and K (key), i.e. F^{LR} and $F^{\text{Ref}\downarrow}$, are caught from the SR-MainNet with the input I^{LR} and $I^{\text{Ref}\downarrow}$.

$$Q \stackrel{c}{=} \text{SR-MainNet}(I^{\text{LR}}) \quad (12)$$

$$K \stackrel{c}{=} \text{SR-MainNet}(I^{\text{Ref}\downarrow}) \quad (13)$$

$$V = \text{ReverseNet}(I^{\text{Ref}}) \quad (14)$$

where $\stackrel{c}{=}$ denotes Q and K are the feature maps captured from the SR-MainNet but not the final output. As shown in Fig. 2, the SR-MainNet and ReverseNet generate the features Q , K , V respectively. Then, Q , K , V are fed into texture transformer module. In the texture transformer, this paper applies the similar transform strategy as [4]. But we do not use the hard-and-soft attention module.

To embed the relevance between the LR and Ref image, the similarity between Q and K is calculated. Q and K are unfolded into patches, denoted as q_i ($i \in [1, H_{\text{LR}} \times W_{\text{LR}}]$) and k_j ($j \in [1, H_{\text{Ref}} \times W_{\text{Ref}}]$). Then normalized inner product is used to calculate the relevance $r_{i,j}$ between each q_i and k_j :

$$r_{i,j} = \left\langle \frac{q_i}{\|q_i\|}, \frac{k_j}{\|k_j\|} \right\rangle \quad (15)$$

Then we construct a swapped feature map M from the relevance. The i -th element m_i ($i \in [1, H_{\text{LR}} \times W_{\text{LR}}]$) is calculated from the relevance $r_{i,j}$:

$$m_i = \arg \max_j r_{i,j} \quad (16)$$

This paper considers M as a index map, which represents the most relevant position in the Ref image to the each position in the LR image. Then we get the transformed HR features T from Ref image. The index selection operation is applied to the unfolded patches of V using the map M as the index:

$$t_i = v_{m_i} \quad (17)$$

where t_i denotes the value of T in the i -th position, which is selected from the m_i -position of V . After the transformation process, the SR-MainNet gets the transformed features T as reference features. The operation can be represented as:

$$F_{\text{out}} = F + \text{Conv}(\text{Concat}(F, T)) \quad (18)$$

where F_{out} indicates the synthesized output features. Conv and Concat represent a convolutional layer and concatenation operation, respectively. It can be seen from the above formula that the equation still holds without T .

$$F_{\text{out}} = F \quad (19)$$

The SR-MainNet without reference is the case without T . Thus the parameters of SR-MainNet trained before can be

Table 1 PSNR/SSIM comparison among different SR methods on four different datasets

Method	CUFED5	Sun80	Urban100
SRCNN [7]	25.33/0.745	28.26/0.781	24.41/0.738
MDSR [33]	25.93/0.777	28.52/0.792	25.51/0.783
RDN [34]	25.95/0.769	29.63/0.806	25.38/0.768
RCAN [35]	26.06/0.769	29.86/0.810	25.42/0.768
SRGAN [20]	24.40/0.702	26.76/0.725	24.07/0.729
ENet [36]	24.24/0.695	26.24/0.702	23.63/0.711
DRN [37]	26.03/0.772	29.84/0.811	25.42/0.770
CrossNet [2]	25.48/0.764	28.52/0.793	25.11/0.764
SRNTT- <i>rec</i> [3]	26.24/0.784	28.54/0.793	25.50/0.783
SRNTT [3]	25.61/0.764	27.59/0.756	25.09/0.774
TTSR- <i>rec</i> [4]	27.09/0.804	30.02/0.814	25.87/0.784
TTSR [4]	25.53/0.765	28.59/0.774	24.62/0.747
SR-USRN- <i>rec</i>	26.60/0.786	30.02/ 0.815	25.84/0.781
SR-USRN	25.66/0.767	29.11/0.793	25.09/0.763

Methods are grouped by SISR methods (top) and RefSR methods (down). Bold numbers denote the highest scores while Italic numbers denote the second highest scores

reused and finetuned here. Moreover, this paper uses SR-MainNet as the texture extractor, which makes distribution of the features similar. This will improve the confusion quality of the Ref and LR image.

In the training process, this step has the same loss function as Sect. 3.1.

3.4 Implementation details

For the SR-MainNet, we adopt the same network used in TTSR [4] except for the soft-and-hard attention module. In order to simplify the calculation process, this paper just uses the hard-attention module to generate the transform map. We use PyTorch to implement the model on an NVIDIA 1080Ti GPU. Throughout the training, we augment the training images by randomly horizontally and vertically flipping followed by randomly rotating 90, 180, and 270. We apply the Adam [30] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ as the optimizer. The learning rate is set to $1e-4$ and the batch-size is 6 images. In the training stage, the size of the input LR image is 40×40 , and the size of Ref image and the output SR image is 160×160 . The weight coefficients for \mathcal{L}_{rec} , \mathcal{L}_{adv} and \mathcal{L}_{per} are 1, $1e-3$ and $1e-2$, respectively in Sects. 3.1 and 3.3. SR-MainNet without reference in Sect. 3.1 is trained 30 epochs. In Sect. 3.2, the ReverseNet is trained 15 epochs. Finally, SR-MainNet and ReverseNet are combined to train another 50 epochs in Sect. 3.3. In the testing stage, there is no size requirement for the LR image and Ref image, except to be cropped to a multiple of 4.

4 Experiments

4.1 Datasets and metrics

Following the same setting as [3,4], we train and test our model on the recently proposed RefSR dataset, CUFED5¹ [3]. To evaluate the generalization capacity of the trained model on CUFED5, we test it on Sun80 [31] and Urban100 [32].² To evaluate the performance of the methods, we calculate PSNR and SSIM on Y channel of YCbCr space as the quantitative criteria.

4.2 Evaluation

To evaluate the effectiveness of SR-USRN, our model is compared with other SOTA SISR and RefSR methods as shown in Table 1. These experiments are carried out with a $4 \times$ scaling factor between LR and HR images.

Following the setting in SRNTT [3] and TTSR [4], this paper trains all the methods on CUFED5 training set, and tests on CUFED5 testing set, Sun80 and Urban100 datasets. For SR methods, although adversarial loss and perceptual loss can improve the visual quality of SR images, it will lead to the loss of PSNR and SSIM scores. Therefore, for fair comparison on PSNR and SSIM, we train another version of our model which is optimized only on reconstruction loss named SR-USRN-*rec*. SRNTT-*rec* and TTSR-*rec* indicate SRNTT and TTSR optimized only on reconstruction loss as well.

The detailed evaluation results are shown in Table 1. As shown in Table 1, SR-USRN-*rec* gets the second highest scores in all three datasets. When trained with adversarial loss and perceptual loss, our method performs better than SRNTT and TTSR, and achieves the highest scores on CUFED5 and Sun80. On Urban100, SR-USRN and SRNTT have similar performance. In Table 1, our method achieves SOTA SR performance whether with or without adversarial loss.

We also compare the number of network parameters and inference time with other RefSR methods in Table 2. The inference time is calculated on input of a 128×128 LR image and a 512×512 Ref image. The time consumption is acceptable compared to the performance increase. Our approach has a slight increase in the number of parameters and inference time over TTSR, which is acceptable.

To compare the differences in visual quality, we show the performance of different methods in Fig. 3. As shown in Fig. 3, our method achieves the best visual quality, compared with existing SISR and RefSR. SR-USRN can recover tex-

¹ CUFED5 is available at <https://zzutk.github.io/SRNTT-Project-Page/>.

² Sun80 and Urban100 are available at <https://github.com/jbhuang0604/SelfExSR>.

Table 2 Number of parameters and inference time of different algorithms

Algorithm	Parameters (M)	Inference time (ms)
CrossNet	35.18	125.6
TTSR	6.73	269.2
SR-USRN	9.23	354.0

ture better than SRNTT and avoid some unpleasant texture transformation in TTSR.

4.3 Ablation study

4.3.1 the use of parallel structure and ReverseNet

SR-USRN mainly contains two networks: the ReverseNet and parallel SR-MainNet. Ablation results are shown in Table 3. In order to verify the effectiveness of the two parts, this section only uses one part to train two versions.

The version without SR-MainNet means that we do not use the parallel structure and only use the original input images without reference images to train SR-MainNet and ReverseNet. The version without ReverseNet means that the second step is skipped and ReverseNet initialized by VGG19 is finetuned directly in the third step. As shown in Table 3, the network with both two parts achieves the highest score. It proves the indispensable of the two parts.

4.3.2 the use of ReverseNet

The use of the ReverseNet is to transform the Ref image to the corresponding features in the SR-MainNet. The ability of the ReverseNet to recover the MainNet features is important to the texture transformer. Here, this section tries to find out the impact of the structure of ReverseNet on test results. Several typical networks are taken into consideration. These networks include VGG19, ResNet18 and ResNet34. As shown in Fig. 2, this paper catches part of VGG19, ResNet18 or ResNet34 as the backbone of ReverseNet, and several con-



Fig. 3 Visual comparison among different SR methods on CUFED5 testing set (top one example), Urban100 (the second and third examples), Sun80 (the fourth example)

Table 3 Ablation study on the use of parallel structure and ReverseNet

Method	SR-MainNet	ReverseNet	PSNR/SSIM
Base + ReverseNet		✓	25.372/0.7609
Base + MainNet	✓		25.561/0.7656
Base + MainNet + ReverseNet	✓	✓	25.646/0.7657

Table 4 Ablation study on different structure of ReverseNet

ReverseNet	PSNR	SSIM
VGG19	25.646	0.7657
ResNet18	25.536	0.7656
ResNet34	25.352	0.7601

volution layers follow three-level features of main structure. The second, fourth, eighth ReLU layers of VGG19 are used for feature swapping. In ResNet18 and ResNet34, the first layer is a convolution layer with 7×7 kernel-size and 2 stride and the third layer is a maxpool layer with 2 stride. To avoid reducing the image size too early, this paper changes the convolution layer to a layer with 3×3 kernel-size and 1 stride, and deletes the first maxpool layer. As shown in Table 4, we can see that VGG19 gets the best result. ResNet34 is deeper than ResNet18 and ResNet18 is deeper than VGG19, while the order of PSNR/SSIM scores is opposite. Because of the limitation of data volume, the ReverseNet with deep structure can not recover the features in SR-MainNet, resulting in the decline of the results. In the future work, we will pay more attention to the design of the ReverseNet.

4.3.3 Effect of reference similarity

As discussed in [3,4], similarity between LR and Ref images is a key factor to the performance of RefSR methods. This paper investigates the performance of the SOTA RefSR and our proposed method at different reference levels. Table 5 lists the results of several algorithms at five levels of references. Comparing the scores of PSNR/SSIM in Table 5, it is obvious that higher relevant level achieves better scores in each method. Our method also meets this phenomenon, which means that our method also has the ability to distinguish the relevance of image content and texture.

4.3.4 Ability to find right texture

Our method uses SR-MainNet to transform texture from Ref image to LR image. In Fig. 4, we uses the transform map to transform the original image directly. With the reduce of similarity, the quality of the transformed image is declined. It proves that SR-MainNet can express the features correctly. In Fig. 5, it is obvious that only one girl in the right image. Three boxes with different colors mark the corresponding

index positions in two images. Three boxes in the faces of different persons map to the same face of the girl in right. It proves that the SR-MainNet can find similar texture even if the content of the image is different.

4.3.5 Generalization performance of SR-USRN

To prove the generalization performance of SR-USRN, several images captured in daily life are processed with super-resolution. These images are taken by iPhone 11. In Fig. 6, images c and f have a more regular and realistic texture than b and e. This experiment shows good generalization performance of SR-USRN.

Table 5 Ablation study on reference images of different similarity

Level	CrossNet	SRNTT-rec	TTSR-rec	SR-USRN
L1	25.48/0.764	26.15/0.781	26.99/0.800	26.582/0.7852
L2	25.48/0.764	26.04/0.776	26.74/0.791	26.535/0.7833
L3	25.47/0.763	25.98/0.775	26.64/0.788	26.516/0.7829
L4	25.46/0.763	25.95/0.774	26.58/0.787	26.499/0.7824
LR	25.46/0.763	25.91/0.776	26.43/0.782	26.462/0.7812

From L1 to L4, the similarity is decreasing. LR means using LR image as Ref image



Fig. 4 Using reference images to recover ground truth. Images with different similarities are shown in the first row. The corresponding transformed images and the ground truth are shown in the second row.

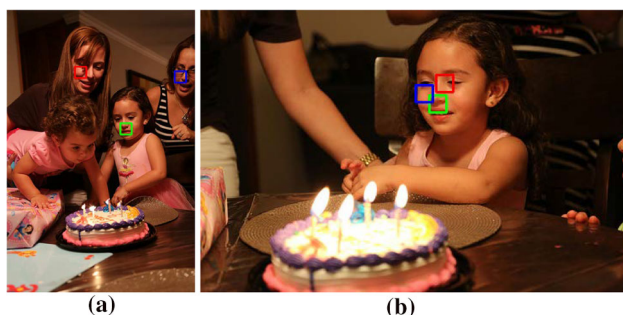


Fig. 5 Corresponding positions in LR and reference images. Image **a** is the LR image and **b** is the reference image. The box with the same color in two images is the corresponding position in transform map

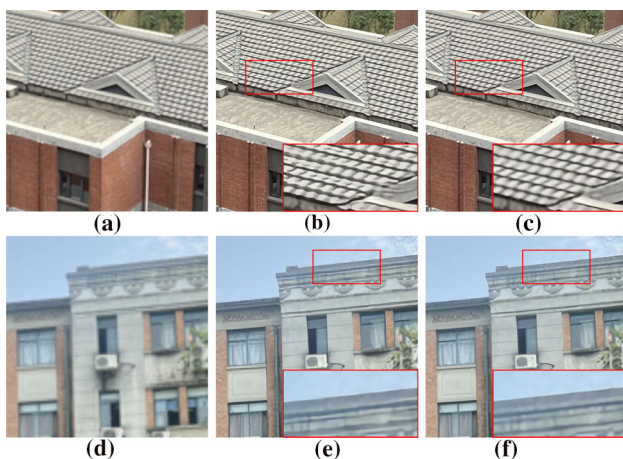


Fig. 6 Image **a** and **d** are original images. Image **b** and **e** are processed by DRN. Image **c** and **f** are processed by SR-USRN

5 Conclusion

In this paper, we propose SR-USRN which uses the same network structure to realize texture transform and super-resolution. SR-USRN consists of three steps, including training SR-MainNet without reference, training ReverseNet to recover the features in SR-MainNet by Ref image and combining SR-MainNet and ReverseNet to train final RefSR Network. This paper uses Ref image to train SR-MainNet in first step and shares the parameters in SR and texture transformation. This design makes full use of the Ref image and the same structure of network makes texture transformer know what SR network really needs. Extensive experiments demonstrate that SR-USRN achieves significant improvements over SOTA approaches on both quantitative and qualitative evaluations. We find that the ReverseNet structure affects the extraction of Ref image features and the final SR results. In the future work, we will refer to the SR-MainNet structure to design a more efficient ReverseNet.

Acknowledgements This project is supported by National Natural Science Foundation of China (Grant No. 52075483).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution (2014)
- Zheng, H., Ji, M., Wang, H., Liu, Y., Fang, L.: Crossnet: An end-to-end reference-based super resolution network using cross-scale warping (2018)
- Zhang, Z., Wang, Z., Lin, Z., Qi, H.: Image super-resolution by neural texture transfer. arXiv (2019)
- Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: CVPR (2020)
- Anwar, S., Khan, S., Barnes, N.: A deep journey into super-resolution: a survey. *ACM Comput. Surv.* **53**, 1–34 (2019)
- Wang, Z., Chen, J., Hoi, S.C.H.: Deep learning for image super-resolution: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **99**, 1 (2020)
- Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**, 295–307 (2016)
- Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network (2016)
- Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
- Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2016)
- Shi, W., Caballero, J., Huszár, F., Totz, J., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network (2016)
- Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution (2017)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2016)
- Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network (2018)
- Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1637–1645 (2016)
- Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: a persistent memory network for image restoration. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4549–4557 (2017)
- Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks (2018)
- Ren, H., El-Khany, M., Lee, J.: Image super resolution based on fusing multiple convolution neural networks. In: Computer Vision and Pattern Recognition Workshops (2017)
- Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution (2016)
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z.A.: Photo-realistic single image super-resolution using a generative adversarial network (2016)

21. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: Esrgan: Enhanced super-resolution generative adversarial networks (2018)
22. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014)
23. Guo, Y., Chen, J., Wang, J., Chen, Q., Cao, J., Deng, Z., Xu, Y., Tan, M.: Closed-loop matters: dual regression networks for single image super-resolution. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5406–5415 (2020)
24. Yue, H., Sun, X., Member, S., Yang, J.: Landmark image super-resolution by retrieving web images. *IEEE Trans. Image Process.* **22**(12), 4865–4878 (2013)
25. Boominathan, V., Mitra, K., Veeraraghavan, A.: Improving resolution and depth-of-field of light field cameras using a hybrid imaging system. In: 2014 IEEE International Conference on Computational Photography (ICCP), pp. 1–10 (2014)
26. Zheng, H., Ji, M., Han, L., Xu, Z., Wang, H., Liu, Y., Fang, L.: Learning cross-scale correspondence and patch-based synthesis for reference-based super-resolution. *01* (2017)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
28. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of Wasserstein GANS (2017)
29. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (2016)
30. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. *Comput. Sci.* (2014)
31. Sun, L., Hays, J.: Super-resolution from internet-scale scene matching. In: 2012 IEEE International Conference on Computational Photography, ICCP (2012)
32. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars (2015)
33. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2017)
34. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR (2018)
35. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV (2018)
36. Sajjadi, M.S.M., Schlkopf, B., Hirsch, M.: Enhancenet: single image super-resolution through automated texture synthesis (2016)
37. Zhang, W., Liu, Y., Dong, C., Qiao, Y.: Ranksgan: generative adversarial networks with ranker for image super-resolution (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.