IET Image Processing

Research Article

ISSN 1751-9659
Received on 17th August 2018
Revised 26th October 2018
Accepted on 3rd December 2018
E-First on 5th March 2019
doi: 10.1049/iet-ipr.2018.6057
www.ietdl.org

Deep residual refining based pseudo-multi-frame network for effective single image super-resolution

Kangfu Mei[1], Aiwen Jiang[1] ✉, Juncheng Li[2], Bo Liu[3], Jihua Ye[1], Mingwen Wang[1]

[1]School of Computer and Information Engineering, Jiangxi Normal University, Nanchang, People's Republic of China
[2]School of Computer Science and Software Engineering, East China Normal University, Shanghai, People's Republic of China
[3]College of Computer Science and Software Engineering, Auburn University, Auburn, USA
✉ *E-mail: jiangaiwen@jxnu.edu.cn*

Abstract: Single image super-resolution (SISR) has gained great attraction and progress in recent years. Since the SISR is an ill-posed inverse problem, most researchers are concentrated on making efforts to learn effective and reasonable mapping functions from low-resolution observation to its potential high-resolution (HR) counterpart. In this study, the authors have proposed a deep residual refining based pseudo-multi-frame network for efficient SISR. A channel-wise attention mechanism is employed for residual refinement. It can ease residual learning process through explicitly modelling non-linear dependencies between channels by using global information embedding. Multiple potential HRs from different deconvolutional layers are further artificially learned, and then adaptively fused into final desired HR image. The authors call this strategy as pseudo-multi-frame SR. It could make full use of available redundant information possessed in hierarchical layers. They have evaluated the proposed network on several popular benchmark datasets. The experimental results have shown that the two highlights proposed can consistently boost final performance. The proposed network can outperform most of the state-of-the-art methods with acceptable less parameters.

1 Introduction

Super-resolution (SR) is to estimate visual pleasing high-resolution (HR) image/video from its low resolution (LR) observations. It is widely used in many practical applications where high-frequency details are desired, such as video surveillance, medical imaging, remote imaging, HDTV and so on. Since a low-resolution image could be degraded from more than one high-resolution case, the SR is, therefore, a typical ill-posed inverse problem. Researchers on SR in these years were to make efforts on learning effective and reasonable mapping functions from LR to potential HR images. Due to the powerful learning ability of deep neural networks, deep learning (DL) based methods have recently gained much attractions and achieved great progress compared with conventional none-DL methods [1, 2] on SR.

In this paper, we considered the construction of SR networks from three main aspects: the computational efficiencies of network, the strategies of residual learning and the information available for fusion.

The computational efficiencies of network: SRCNN [3] was the first DL-based work on image SR. It employed a lightweight structure with three convolution layers to learn an end-to-end mapping between LR and HR. FSRCNN [4] was an accelerating version of SRCNN. One of its highlights was that it directly applied network on original LR image, instead of bicubic interpolating original LR before network input. At the end of network, it introduced a deconvolution layer for HR mapping.

SRCNN and FSRCNN represented two general strategies on how to deal with input LR images before applying them into network. The pre-processing step of bicubic interpolation inevitably increased the computation burden, over-smoothed and blurred original LR image, which might result important details lost. Similarly, ESPCN [5] introduced an efficient sub-pixel convolution layer to effectively replace the handcrafted bicubic filter. Feature mappings are directly extracted from original LR image. As a result, the computational complexity was reduced. Therefore, from the experience of previous work, applying network directly on original LR before sub-pixel convolutional layer to desired HR is more computational efficient.

The strategies of residual learning: VDSR [6] was one of the pioneers that proposed residual learning-based strategies for the single image super-resolution (SISR) problem. Its core idea was based on the observation that LR and HR images shared similar low-frequency information. Therefore, it was reasonable to explicitly modelling the differences, which were also referred to high-frequency details. Residual-based strategies were proved to be more suitable for solving SR problems and have now become the commonly accepted network's configurations.

Explicitly modelling dynamic, non-linear dependencies between channels by using global information could ease learning process [7]. Therefore, we considered introducing a channel-wise attention mechanism into local residual learning block. The introduced mechanism could simplify the designation of residual-based learning strategy while improving final performance. We would demonstrate the effectiveness in our ablation experiments.

The information available for fusion: Most of methods on SR can be categorised into two main streams: SISR and multiple image SR (MISR). The SISR concerns on estimating HR from a single LR image of the same scene, under the assumption that the original imaging setup is not available. In MISR, the input usually consists of more than one LR image, under the assumption that each one is a degraded version of an underlying HR scene. In terms of available redundant information possessed, recovering a HR image from multiple LR images is to some extent easier than from only single LR image.

It is not difficult to find that most of SISR methods are limited that HR could only be restored from only one input, either in the case of function learning from the interpolated LR to HR, or in the case of deconvolutoinal learning from original LR to final HR at the end part of network. They were concentrated on an deterministic learning in one-to-one mode.

However, SR is an ill-posed problem and the relationship between LR and HR is in many-to-many mode. That means, one LR could be degraded from different HRs, and one HR could also

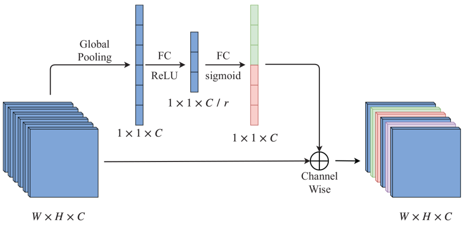

Fig. 1 *Basic structure of SE building block*

generate different LRs depending on different degradations. Motivated by the fact, we proposed to artificially create multiple potential HRs from different deconvolutional layers according to different level LR information in network, and then adaptively fuse these potential HRs into final desired HR output. As these intermediate HRs are not restored from different real LRs, respectively, we take this idea as a kind of *pseudo-multiple-frame SR strategy*.

To verify our considerations on how to construct an effective SISR network, we have proposed a compositional network that contains our concerns in this paper. We have performed two ablation experiments and comparisons with several state-of-the-art methods on public benchmark datasets. As we are not completely committed to performance excellence, in our experiments, we avoid using highly tricky training strategies to enhance performance. Nonetheless, the experiment results still demonstrate that our proposed strategies are effective and the proposed compositional network could achieve top performance with reasonable parameters scale and depth.

The contributions of this paper are two-fold:

- A *residual refine block via channel-wise attention mechanism* is proposed for SR. It explicitly models the dynamic, non-linear dependencies between channels by using global information. As a result, the residual learning is eased and the SR performance is boosted as well. Moreover, the residual refine based strategy is general, which can easily embedded into any residual learning based SISR model.
- A *pseudo-multiple-frame strategy* is proposed to augment redundant information for effectively solving SISR problem. It is motivated by the many-to-many relationship between LRs and HRs. Different from most of current SISR methods in which residual blocks were stacked in a chain way, the *pseudo-multiple-frame SR strategy* can adaptively make full use of hierarchical information and has been demonstrated that it can outperform most of state-of-the-art methods with reasonable parameters size under the same experiment conditions.

2 Related work

Our proposed network falls into the category of deep learning based SISR methods. Therefore, in this paper, we mainly discuss the most related deep learning based work recently proposed on SISR.

2.1 Residual-based SISR

Due to the proximity from original image to ill-posed recovered image, such as LR to HR, most values of residual images are likely to be small even zero. The learning process from input to output can be speed up through residuals. Moreover, skip connections between input and output have also been demonstrated to be able to avoid gradient vanishing/exploding. Therefore, residual-based learning has been playing great importance on ill-posed image restoring problems.

SRGAN(SRResNet) [8] had proposed a generative adversarial network for SR. The proposed perceptual loss consisted of an adversarial loss and a content related loss. It benefited from restoring more photo-realistic HR image. However, the original architecture of ResNet [9] was directly employed without much modification in SRResNet.

EDSR [10] was an improved version of SRResNet, and won the NTIRE2017 Challenge on Single Image Super-Resolution [11]. As the authors claimed, the significant performance improvement of their model was due to optimisation by removing unnecessary modules in conventional residual networks and their performance was further improved by expanding the model size. Therefore, it could be said that the high performance of EDSR was in cost of training an extremely large scale and very wide network.

DRCN [12] was the first to employ recursive neural network on SR solutions. Global residual learning and recursive supervision was included in the training architecture of network.

Different from the chains structure of recursive layers in DRCN, the recursive layers in DRRN [13] were in multi-path mode, in which both global and local residuals were learned.

Similar recursive idea was also supported in MemNet [14]. It introduced a memory block consisted of a recursive unit and a gate unit. Dense skip connections between the current recursive unit and outputs from previous memory blocks were employed. The concatenations were input into gate unit to maintain persistent ‘memory’.

LapSRN [15] progressively reconstructed sub-band residuals of HR images at multiple pyramid levels. Instead of the commonly used L2 loss, Charbonnier loss was adopted to better handle outliers and improve the performance.

SRDenseNet [16] employed densely connected convolutional networks [17] as its basic block, and utilised dense skip connections to combine features from different level blocks to provide rich information for final SR reconstruction.

2.2 Squeeze-and-excitation networks

SENet [7] was the winner of image classification task in ILSVRC 2017. A ‘squeeze-and-excitation’ (SE) block was introduced to improve the representational power of a network by explicitly modelling the interdependencies between the channels of its convolutional features. Feature recalibrate was applied to selectively emphasise informative features and suppress less useful ones.

The basic structure of SE building block is illustrated in Fig. 1.

The SE block embeds global information into each descriptor through *squeeze* operation. A gating mechanism is parameterised in *excitation* stage to learn non-linear interaction between each channels. It contains two fully connected (FC) layers. One FC is for dimensionality reduction with ratio r and capturing channel-wise dependencies of features. The other performs sample-specific activations to govern the excitation of each channel based on the learned channel dependence. The advantage of SE block is that it introduces an effective channel-wise attention mechanism and can be stacked at any place needed in network architecture.

2.3 Dense connection block

Different from ResNets [9], the feature maps in DenseNet [17] were concatenated rather than directly summed. DenseNet adopted a simple connectivity pattern that each layer in the network took in additional inputs from all preceding layers and passed on its output feature maps to all subsequent layers. The authors claimed that this connectivity pattern could ensure maximum information flow between layers in feed-forward nature.

The basic structure of dense connection block consists of a composite sub-block followed by a transition layer, as shown in Fig. 2. The composite sub-block consists of three consecutive operations: batch normalisation (BN), a rectified linear unit (ReLU) and a 3×3 convolution (Conv). The transition layer consists of a 1×1 convolutional layer followed by a 2×2 average pooling layer. Due to the compelling experimental advantages of DenseNet, SRDenseNet [16] had introduced the dense block for SR, and achieved state-of-the-art performance.

3 Residual refine based pseudo-multiple-frame network for effective SISR

In this section, we will describe our proposed residual refine based pseudo-multi-frame network in detail.

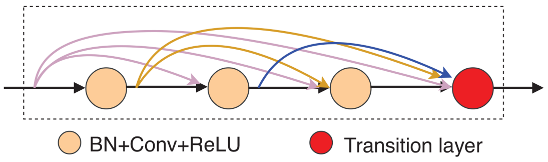

Fig. 2 *Basic structure of dense block in DenseNet*

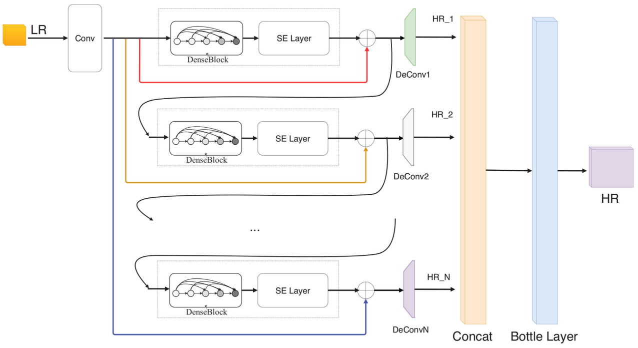

Fig. 3 *Architecture of our proposed residual refine based pseudo-multi-frame network*

The architecture of our proposed network is as shown in Fig. 3. It mainly consists of two parts: *residual refine based dense blocks (RRDs)* and *pseudo-multi-frame information fusion networks*. $I_{\rm LR}$ and $I_{\rm HR}$ are the input and the output of the network.

As suggested in VDSR [6], when dealing with SR problem, surrounding pixels were useful to correctly infer central pixel. With larger receptive field, a SR model could utilise more contextual information to better learn correspondences from LR to HR. Therefore, in our proposed network, a convolutional layer $\text{Conv}_{\rm LR}$ is first applied on original LR image for aggregating informative features. The size of the convolution filters is $3 \times 3 \times 32$, and the step size is 1. As a result, a feature map $F_{\rm LR}$ is extracted with the same spatial dimensions as the input $I_{\rm LR}$.

For the convenience of description, we denote the mapping function of the kth residual refine based dense block (RRD) as $y = g_k(x)$. Supposing we have N residual refine dense blocks, then the output of each RRD is formulated in recursive as follows:

$$\begin{aligned} & r_0 = 0,\ r_1 = g_1(F_{\rm LR} + r_0),\ r_2 = g_2(F_{\rm LR} + r_1), \\ & \cdots \\ & r_N = g_N(F_{\rm LR} + r_{N-1}) \end{aligned} \tag{1}$$

where $\{r_k\}_{k=1,\dots,N}$ represents the output of refined residuals from the kth RRD. r_0 is the pre-defined initial residual. Since the residuals at each level $r_k = g_k(F_{\rm LR} + r_{k-1}),\ k = \{1, \dots, N\}$ are hierarchically learned from its previous residuals, we call them as *refined residuals*.

We take the signals $\tilde{F}^k_{\rm LR} = F_{LR} + r_k,\ k = \{0, 1, \dots, N\}$ as a kind of intermediate ‘degraded’ LRs. Then we learn respective deconvolution layer $\text{DeConv}_k()$ for it to HR image. As a result, we artificially create multiple recovered potential ‘HR’ images by $\tilde{I}^k_{\rm HR} = \text{DeConv}_k\left(\tilde{F}^k_{\rm LR}\right)$. These intermediate ‘HR’ images possess redundant information from hierarchical level features. In the final, we adaptively fuse them to obtain the desired HR output, by using a learned fusion function $f_{\rm Pseudo}$ as formulated in the following equation:

$$I_{\rm HR} = f_{\rm Pseudo}\left(\left[\tilde{I}^1_{\rm HR}, \tilde{I}^2_{\rm HR}, \dots, \tilde{I}^k_{\rm HR}, \dots, \tilde{I}^N_{\rm HR}\right]\right) \tag{2}$$

3.1 Residual refine based dense block

In this section, we present the details about our proposed residual refine block. It consists of a dense block followed by a SE block. The structure is as shown in Fig. 4.

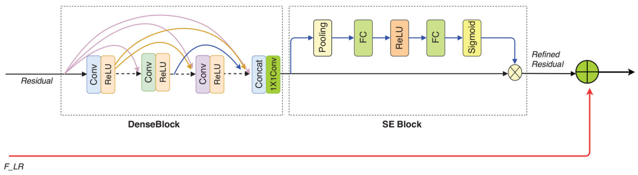

Fig. 4 *Structure details of residual refine block*

As BN layers would increase computational complexity and pooling operations potentially discard pixel-level information, in this paper, following the experiences of previous SR work [16], we remove both the BN layer and the pooling layer out of the dense sub-block in our residual refine block.

Dense skip connections are implemented among Conv + ReLU layers. All outputs $\{F_d\}_{d=1}^{D}$ of Conv + ReLU layers together with the block's input are concatenated on channel dimension, and then transferred into a 1×1 convolutional transition layer. We formulate the process as $U = \sigma(W_{\rm trans}, [F_{\rm LR}, F_1, \dots, F_D])$, where the $W_{\rm trans}$ is the convolutional weights for transition layer.

As we know, increasing the depth of dense block could potentially achieve better performance, such as the dense block used in SRDenseNet has eight Conv + ReLU layers. However, by considering the efficiency and memory usage, in this paper, we set the number of Conv + ReLU layers in our dense sub-block part as $D = 4$, and the growth rate G of the dense block as $G = 32$. That means the convolutional kernel at dth level is in size of $3 \times 3 \times (32 * d) \times 32,\ d = \{1, 2, \dots, D\}$.

Our residual refine blocks take role of learning different level residuals in the proposed network. In order to further boost the discriminative ability of learned residuals, a SE block is applied after dense block U. The SE block intrinsically introduces dynamical channel-wise attentions on its input. The resulted recalibrate mappings are hierarchically learned as refined residuals for level k.

Specifically, at *squeeze* stage, a statistic $z \in \mathfrak{R}^C$ is generated by shrinking feature map $U \in \mathfrak{R}^{W \times H \times C}$ on spatial dimensions $W \times H$. The cth element of z is calculated by (3). The z could be interpreted as a collection of local descriptors, whose statistics are expressive for the whole image

$$z_c = F_{\rm squeeze}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{3}$$

Then, at *excitation* stage, two succeed FC layers are employed to learn the channel-wise attention coefficients $s \in \mathfrak{R}^{1 \times C}$, just as (4) does

$$s = F_{\rm ex}(z, W_{\rm ex}) = \text{sigmod}(W_2 * \text{ReLU}(W_1 * z)) \tag{4}$$

The final output of the SE block is obtained by scaling feature map U with the activations s, as shown in the following equation:

$$\tilde{x}_c = s_c * u_c \tag{5}$$

3.2 Pseudo-multi-frame information fusion network

In this section, we present the details of pseudo-multiple-frame fusion strategy for restoring HR. The information fusion network consists of several parallel deconvolution layers followed by one concatenation layer and one 1×1 convolutional bottle layer, as shown in Fig. 3.

Sub-pixel convolution network is employed as the deconvolution layer DeConv_k. The sub-pixel convolutional network consists of 1×1 convolutional layer followed by a periodic shuffling layer. We formulate the process as $\tilde{I}^k_{\rm HR} = \Omega\left(W_{bk} * \tilde{F}^k_{\rm LR}\right)$, where W_{bk} represents the deconvolutional weights for the potential HR image at the kth level. Ω is a periodic shuffling operator that rearranges the elements of a $H \times W \times C \cdot r^2$ input tensor to an output tensor of $rH \times rW \times C$ size. r is the scale

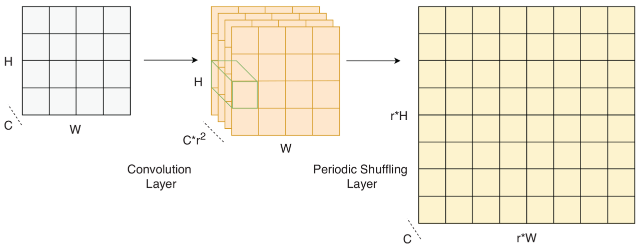

Fig. 5 *Sub-pixel network for learning HR mapping*

factor of HR compared to the size of LR. The architecture of the deconvolution network is illustrated in Fig. 5.

After N intermediate HR mappings are artificially generated in parallel, they are adaptively fused into the final HR. The process is formulated as $I_{\rm HR} = f_{\rm bottle}\left(W_B, \left[\tilde{I}^1_{\rm HR}, \tilde{I}^2_{\rm HR}, \dots, \tilde{I}^N_{\rm HR}\right]\right)$, where W_B represents network weights. Specifically, N pseudo-HRs are first concatenated on channel dimension, followed by a bottle layer $f_{\rm bottle}$. The bottle layer is a 1×1 convolution layer, transforming mapping from $rW \times rH \times NC$ to $rW \times rH \times C$.

As the these intermediate HRs are learned from pseudo-LRs of different level degradations, they enrich much redundant information for final HR. Through bottle layer, each pixel information on the resulted HR becomes learnable, which can be adaptively learned from pixel information from multiple intermediate frames. Therefore, we name this strategy as *pseudo-multiple-frame based SISR*.

Increasing the number of pseudo-frames would enhance final performance, while increase depth of network. Therefore, in this paper, to balance the effectiveness and efficiency, we set the number of pseudo-frames N empirically to be $N = 16$. The experiments show that network with these parameter settings can sufficiently achieve satisfactory performance.

4 Experiments

In this section, we evaluate the effectiveness of our proposed network on solving SISR problem. The related source code and pre-trained models have been distributed on public at source-code https://github.com/MKFMIKU/RPMNet.

4.1 Datasets

On SR issue, in the past, different learning-based methods were trained on different training datasets. Typically, the 91 images from Wright *et al.* [18] were first used in classic methods, such as A+ [2]. The 291 images consisting of 91 images from Wright *et al.* [18] and 200 images from Berkeley Segmentation Dataset [19] were widely used in some popular SR methods, such as DRCN [12], DRRN [13], VDSR [6] and MemNet [14]. The large-scale ImageNet dataset was also often used for training deep SR models. For example, in SRResNet [8], 350 thousand images were randomly selected for training. Similarly, in ESPCN [5] and SRDenseNet [16], 5000 images were randomly selected from ImageNet for training.

DIV2K [11] dataset is a newly distributed image set for SR challenge. It consists of 800 high-definition high-resolution images for training model. The concurrent state-of-the-art methods like EDSR [10] and our *RPMNet* were all trained on the DIV2K training set.

Data augmentation were applied on all available training images, no matter whatever dataset the above-mentioned methods used. The main differences among them were the different augmentation tricks they used. From our experience, the final performance is in some extent affected by the quality of data augmentation.

During test, five popular benchmark datasets: *Set5* [20], *Set14* [21], *BSDS100* [19], *Urban100* [22], *Manga109* [23] are commonly used. The *Set5* and *Set14*, respectively, contain 5 and 14 images with rich textures. The *BSDS100* consists of 100 natural images from test set of the Berkeley segmentation dataset. The *Urban100* contains 100 challenge urban images with details in different frequency bands. The *Manga109* contains 109 images of Japanese manga. For the wide varieties in visual content, these dataset are widely used for SR evaluation.

4.2 Experiment details

Data augmentation: We convert images into YCbCr colour space. Then all operations in our experiment are performed on the luminance component Y.

In order to set up training pairs and testing pairs, following [15], we augment the training data in three ways: (i) scaling: downscale images on three scales $\{1, 0.7, 0.5\}$; (ii) rotation: randomly rotate image by $\{0, (\pi/2), \pi, (3\pi/2)\}$; (iii) flipping: flip images in three mirror cases $\{\text{original}, \text{horizontal}, \text{vertical}\}$. As a result, 36 variants are generated from original image.

After data argumentation, we use sliding window to extract image patches in size of $(32 * r) \times (32 * r)$. The stride step is set $32 * r/2$ pixels. r is the scaling factor. We apply bicubic downsampling to simulate LR patches by using the Matlab function imresize with the option '*bicubic*' on original patches. In our experiment, for each scaling case, the LR training patches is constantly set in size of 32×32 pixels. The corresponding original patches are taken as HR ground-truths during training.

Training details: The network in our proposed *RPMNet* model is very straightforward. Four structure parameters were used to measure our network. They are the channels C of the initial convolutional layer, the number of convolution layers D in the dense part of the RRD, the growth rate G of the RRD and the number of pseudo-frames N for fusion. In this experiment, as previously described, these parameters are empirically set to be $C = 32, D = 4, G = 32, N = 16$. We denote the resulted network structure as $C32\_D4\_G32\_N16$.

The proposed model is trained through minimising the loss between the reconstructed images and the corresponding HR ground-truths. Given a set of restored HR images $\{I_m\}$ and their corresponding ground truth $\{Y_m\}$, we use mean squared error as the loss function, as shown in the following equation:

$$\text{Loss} = \frac{1}{M}\sum_{m=1}^{M} \| I_m - Y_m \|^2 \tag{6}$$

where M is the number of training samples. The loss is minimised by using stochastic gradient descent with the standard back-propagation.

ADAM [24] with learning rate $lr = 0.0001$ is employed as the optimiser. The batch size is $M = 64$. To avoid different tricks for the learning rate decay, the learning rate in our experiment is kept constant during training process. We implement the network in PyTorch without special weight initialisation method or other training tricks.

Three scaling factors $r = \{\times 2, \times 4, \times 8\}$ are experimented, respectively. We train a specific model for each scaling case. It roughly took one day to train a model to converge by using a NVIDIA Titan Xp GPU.

Evaluation: Two commonly-used image quality metrics: PSNR (peak-signal-to-noise ratio) and SSIM (structural similarity index measurement) [25] are employed to evaluate the performance, as defined in the following equation:

$$\begin{aligned} \text{PSNR}(x, y) &= 10\log\left(\frac{255^2}{\text{MSE}(x, y)}\right) \\ \text{SSIM}(x, y) &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \end{aligned} \tag{7}$$

where $\text{MSE}(x, y) = (1/MN - 1)\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij} - y_{ij}\right)^2$ is the mean-squared error. x and y are two images in size of $M \times N$, μ is mean of image's pixel values and σ is the standard variance. C_1 and C_2 are commonly set as $C_1 = (0.01 * 255)^2$ and $C_2 = (0.03 * 255)^2$.

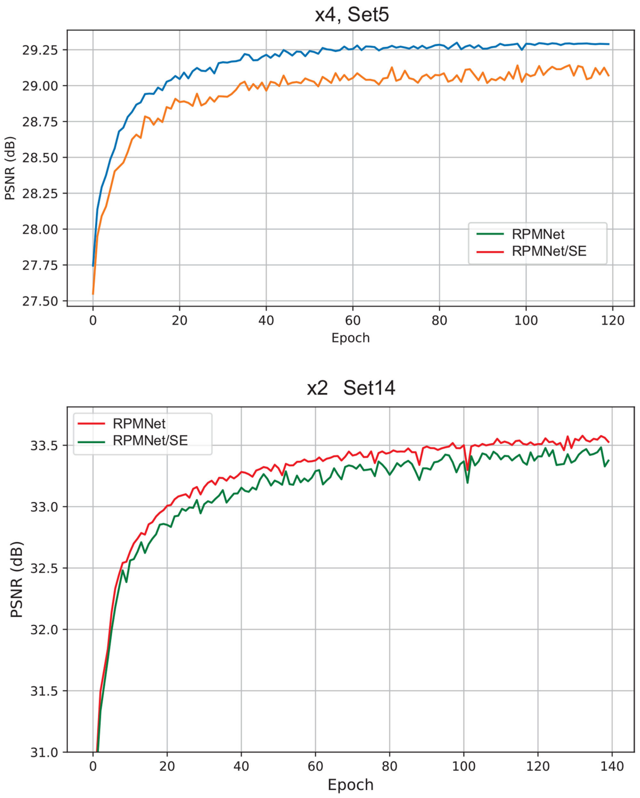

Fig. 6 *Ablation experiment 1: testing curves of RPMNet on PSNR performance with/without SE in RRD. 'RPMNet/SE' represents network without SE block in RRD*

4.3 Results and analysis

4.3.1 Three ablation experiments: One of the ablation experiments is to analyse the impact of the residual refine based strategy on SR. We remove the SE block out of RRD, and the resulted network is denoted as 'RPMNet/SE'.

As shown in Fig. 6, the introduced dynamical channel-wise attentions in residual refine block can consistently improve image restoration performance. Under the same conditions, without applying SE block onto the dense block U, the resulted performance decreased averagely about 0.2 dB when tested at $\times 4$ scale rate on benchmark dataset *Set5* [20] and at $\times 2$ scale rate on benchmark dataset *Set14* [21]. On the other tested datasets, the conclusion is similar.

We further design an ablation experiment to verify the effectiveness of pseudo-multi-frame based strategy. We construct an ablation network as shown in Fig. 7. It shares similar structure with most of popular published SISR models, which only employ one deconvolution layer at the end part. We take this strategy as *single frame* case.

Compared with our full model *RPMNet*, the network in Fig. 7 has differences on information fusion part. It employs skip connections and direct concatenations for deconvolution layer. The deconvolution layer is also a sub-pixel convolution network. The convolutional kernel is $1 \times 1 \times (C * N) \times (r^2)$, where $C = 32,\ N = 16$ and r is the scaling factors.

We argue that *single frame* case cannot take full advantage of hierarchical information that intermediate levels possess. The performance comparisons between *pseudo-multi-frame* case and the *single frame* case are shown in Fig. 8. They are both trained and tested under the same experiment conditions.

From the experimental comparisons in Fig. 8, the pseudo-multi-frame strategy consistently improves the SISR performance over the single frame case on public benchmark datasets. It should be emphasised that the improvements are obtained under the same experiment settings, which can consolidate the advantage of the pseudo-multi-frame strategy. More important is that the pseudo-multi-frame strategy can be easily borrowed for reference by any SISR network with hierarchical information learning based blocks.

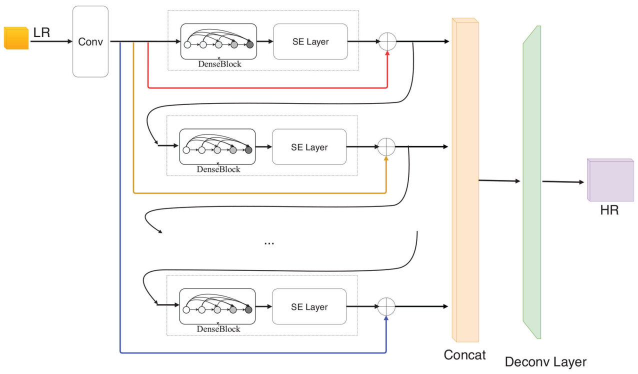

Fig. 7 *Ablation experiment 2: network in single frame case*

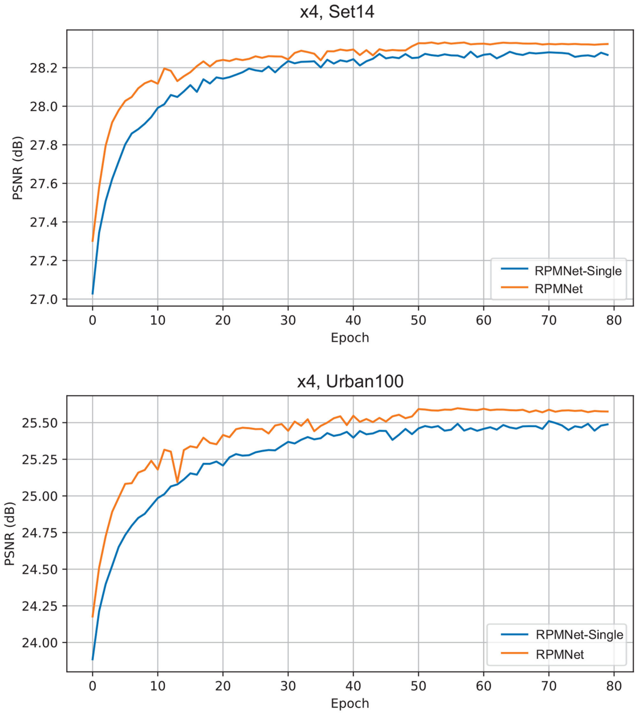

Fig. 8 *Ablation experiment 2: testing curves of RPMNet on PSNR performance with pseudo-multiple frame or not. The 'RPMNet-Single' represents the network as shown in Fig. 7*

In addition to verify RPMNet can suit different components, we further perform an ablation experiment, in which we compare models in the same RMPNet architecture but with different block types. We name our default RPMNet with Denseblock as RPMNet$_{\text{Denseblock}}$. If replace Denseblock in Fig. 3 by Resblock from ResNet [9], we call this varied version as RPMNet$_{\text{Resblock}}$. The performance comparisons tested on different datasets are shown in Table 1. The tested performance curve on dataset Urban100 is shown in Fig. 9.

The experiment results in Table 1 and Fig. 9 demonstrate that using Denseblock in RPMNet is consistently better than using Resblock. Therefore, in the later experiments, we use RPMNet$_{\text{Denseblock}}$ in default.

4.3.2 Comparisons with state-of-the-art methods: We compare our method with several popular state-of-the-art methods including the A+ [2], SelfExSR [26], SRCNN [3], ESPCN [5], VDSR [6], DRCN [12], MemNet [14], LapSRN [15], DRRN [13], SRDenseNet [16] and EDSR [10].

Since most of the state-of-the-art SR models are very sensitive to the subtle network architectural changes, some models are difficult to reach the level of the original paper for the lack of the network configurations. Even the same model could achieve different levels of performance by using different training tricks.

Table 1 Performance comparisons between models in the same RPMNet architecture but with different blocks: Denseblock and Resblock, at × 4 scale

Algorithm	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSDS100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
$\text{RPMNet}_{\text{Resblock}}$	31.23/0.8764	28.11/0.7645	27.24/0.7209	25.17/0.7602	28.90/0.8807
$\text{RPMNet}_{\text{Denseblock}}$	**31.70/0.8856**	**28.27/0.7702**	**27.40/0.7259**	**25.57/0.7761**	**29.64/0.8950**

The performance is evaluated by using average PSNR/SSIM.

The bold values mean better performance.

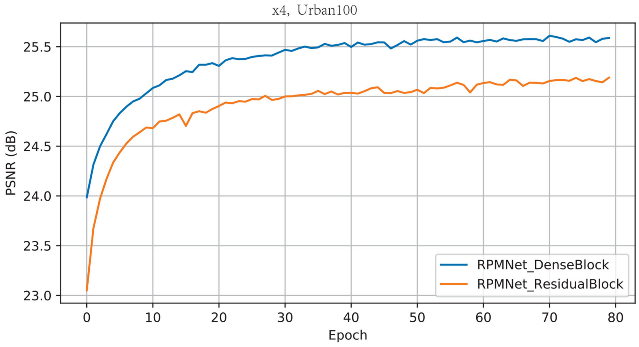

Fig. 9 *Ablation experiment 3: testing curves on PSNR performance for RPMNet variants with Resblock or Denseblock*

Experimenting with different settings makes the results incomparable and unfair if compared. Therefore, to verify the reproduction of published models, and for fair comparisons as well, we retrain most of the compared models by using our augmented training data and experiment settings, except the MemNet [14], EDSR [10] and SRDenseNet [16]. We adopt *early-stop* strategy. If the retrained model reaches its reported test performance on benchmark datasets and model tends to converge, we stop training and use the resulted model as reference for comparison. The experiment results are shown in Table 2.

It should be pointed out that, the EDSR [10] is the winner of 2017 NTIRE Challenge on Single Image Super-Resolution. Though EDSR is also originally trained on the same DIV2K dataset, we cannot reproduce its reported results by using our training settings. Therefore, we directly cite their originally reported performance for reference. The training on MemNet and SRDenseNet are in the similar case. We owe the performance gap to different data augmentation strategies and training tricks that they have used, such as weight initialisation, gradient truncation, data normalisation and so on.

Not surprisingly, by using DIV2K as training dataset, most of the retrained performances are more or less improved than their original reported results. This means that many reported improvements on performance may not be due to the change of the model architecture, but the use of different training data or some unknown training tricks. Fortunately, we are not completely committed to performance excellence in this paper. We mainly want to verify the validity of our proposed residual refining based strategy and pseudo-multi-frame based strategy.

The proposed *RPMNet* is a compositional network based on the two highlighted strategies. It shares some simplified layers with SRDenseNet [16]. With the aim to evaluate the effectiveness of the two highlights, we do not pursue performance excellence. Therefore, the depth and parameters' size of network are balanced with training efficiency. In Fig. 10, the comparisons on parameters' size among some representative models show that our network scale is in mainstream with acceptable parameter size.

From Table 2 and Fig. 10, we can easily find that, on public benchmark datasets, our model could achieve top performance and outperform most of currently popular state-of-the-art methods with approximately equal size of parameters. Most of these methods are retrained under the same experiment settings as our RMPNets. Therefore, factors such as training tricks and data augmentation are excluded.

Though the EDSR [10] reported better results, it is worth noting that EDSR [10] is a deep and wide network which contains a large number of convolutional layers and a huge amount of parameters, about 30 times larger than our model's. This means its training requires more memory, more space and more training data. In contrast, our model is much smaller than EDSR, which makes our model easier to be reproduced and promoted. Compared with EDSR, our model could achieve approaching performance with much smaller parameter size.

In order to have more intuitive comparisons, we give some visual examples in Figs. 11–13. From these results, we can also easily find that our RPMNet could correctly restore texture details better, especially when dealing with large scale such as × 4 and × 8. We owe the advantages to hierarchical information learned in our pseudo-multi-frame fusion process.

4.3.3 Future work: Deep learning paradigm is often criticised for their huge number of tunable parameters. Tensor-based strategy is one of the promising alternatives. Some recent work on tensor analysis such as [27] significantly reduces the number of weight parameters required to be trained by utilising rank-1 canonical decomposition property. This encourages us to extend it as a potential work for our future model compressing in SR.

5 Conclusion

In this paper, we have proposed an effective residual refine based pseudo-multi-frame strategy for SISR. The RRD consists of a modified denseblock followed by a SE layer. Channel-wise attention mechanism maintained by RRD could dynamically model non-linear dependencies between channels, which could recalibrate residual information and boost the restoring performance. The pseudo-multi-frame based fusion strategy draws inspiration from classic multi-frame SR. The ablation results demonstrate that it could make better use of redundant information possessed by hierarchical residual refine blocks. The experiments on public benchmark datasets have shown that the proposed *RPMNet* outperforms most of the state-of-the-art SR methods with reasonably acceptable less parameters.

6 Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant nos. 61365002, 61462042 and 61462045.

Table 2 Performance comparisons among different state-of-the-art methods

Algorithm	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSDS100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
BiCubic	×2	33.69/0.9284	30.34/0.8675	29.57/0.8434	26.88/0.8438	30.82/0.9332
A+ [2]	×2	36.60/0.9542	32.42/0.9059	31.24/0.8870	29.25/0.8955	35.37/0.9663
SelfExSR [26]	×2	36.60/0.9537	32.46/0.9051	31.20/0.8863	29.55/0.8983	35.82/0.9671
SRCNN [3]	×2	36.71/0.9536	32.32/0.9052	31.36/0.8880	29.54/0.8962	35.74/0.9661
ESPCN [5]	×2	37.00/0.9559	32.75/0.9098	31.51/0.8939	29.87/0.9065	36.21/0.9694
VDSR [6]	×2	37.53/0.9583	33.05/0.9107	31.92/0.8965	30.79/0.9157	37.22/0.9729
DRCN [12]	×2	37.63/0.9584	33.06/0.9108	31.85/0.8947	30.76/0.9147	37.63/0.9723
LapSRN [15]	×2	37.52/0.9581	33.08/0.9109	31.80/0.8949	30.41/0.9112	37.27/**0.9855**
DRRN_B1U25 [13]	×2	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	—
MemNet [14]	×2	*37.78/0.9597*	*33.28/0.9142*	*32.08/0.8978*	*31.31/0.9195*	—
EDSR [10]	×2	*38.11/0.960*	*33.92/0.920*	*32.32/0.901*	—	—
RPMNet (ours)	×2	**37.86/0.9603**	**33.78/0.9147**	**32.11/0.9018**	**31.70/0.9282**	**38.26**/0.9754
BiCubic	×4	28.43/0.8022	26.10/0.6936	25.97/0.6517	23.14/0.6599	24.91/0.7826
A+ [2]	×4	30.33/0.8565	27.44/0.7450	26.83/0.6999	24.34/0.7211	27.03/0.8439
SelfExSR [26]	×4	30.34/0.8593	27.55/0.7511	26.84/0.7032	24.83/0.7403	27.83/0.8598
SRCNN [3]	×4	30.50/0.8573	27.62/0.7453	26.91/0.6994	24.53/0.7236	27.66/0.8505
ESPCN [5]	×4	30.66/0.8646	27.71/0.7562	26.98/0.7124	24.60/0.7360	27.70/0.8560
VDSR [6]	×4	31.36/0.8796	28.11/0.7624	27.29/0.7167	25.18/0.7543	28.83/0.8809
DRCN [12]	×4	31.56/0.8810	28.15/0.7627	27.24/0.7150	25.15/0.7530	28.98/0.8816
LapSRN [15]	×4	31.54/0.8811	28.19/0.7635	27.32/0.7162	25.21/0.7564	29.09/0.8845
DRRN_B1U25 [13]	×4	31.68/**0.8888**	28.21/**0.7720**	27.38/**0.7284**	25.44/0.7638	—
MemNet [14]	×4	*31.74/0.8893*	*28.26/0.7723*	*27.40/0.7281*	*25.50/0.7630*	—
SRDenseNet [16]	×4	*32.02/0.8934*	*28.50/0.7782*	*27.53/0.7337*	*26.05/0.7819*	—
EDSR [10]	×4	*32.46/0.8968*	*28.80/0.7876*	*27.71/0.7420*	—	—
RPMNet (ours)	×4	**31.70**/0.8856	**28.27**/0.7702	**27.40**/0.7259	**25.57/0.7761**	**29.64/0.8950**
BiCubic	×8	24.40/0.6045	23.19/0.5110	23.67/0.4808	20.74/0.4841	21.46/0.6138
A+ [2]	×8	25.53/0.6548	23.99/0.5535	24.21/0.5156	21.37/0.5193	22.39/0.6454
SelfExSR [26]	×8	25.49/0.6733	24.02/0.5650	24.19/0.5146	21.81/0.5536	22.99/0.6907
SRCNN [3]	×8	25.34/0.6471	23.86/0.5443	24.14/0.5043	21.29/0.5133	22.46/0.6606
ESPCN [5]	×8	25.75/0.6738	24.21/0.5109	24.37/0.5277	21.59/0.5420	22.83/0.6715
VDSR [6]	×8	25.73/0.6743	23.20/0.5110	24.34/0.5169	21.48/0.5289	22.73/0.6688
DRCN [12]	×8	25.93/0.6743	24.25/0.5510	24.49/0.5168	21.71/0.5289	23.20/0.6686
LapSRN [15]	×8	26.15/**0.7028**	24.45/0.5792	24.54/0.5293	21.81/0.5555	23.39/0.7068
RPMNet (ours)	×8	**26.24**/0.7014	**24.50/0.5883**	**24.64/0.5416**	**22.02/0.5762**	**23.75/0.7215**

Average PSNR/SSIMs for different scale factors. In lack of network configurations and data augmentation details, the performance the MemNet [14], EDSR [10] and SRDenseNet [16] are directly cited from their original papers for hard reproduction in our experiment settings, which are given in italics. All other methods are retrained and tested on our training data. Under the same experiment conditions, the best performance is denoted in bold.

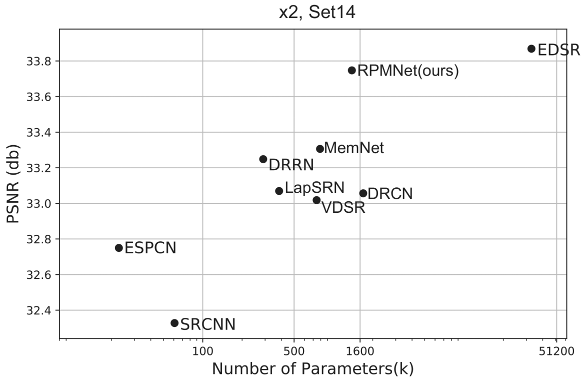

Fig. 10 *Comparisons on parameters size of different models*

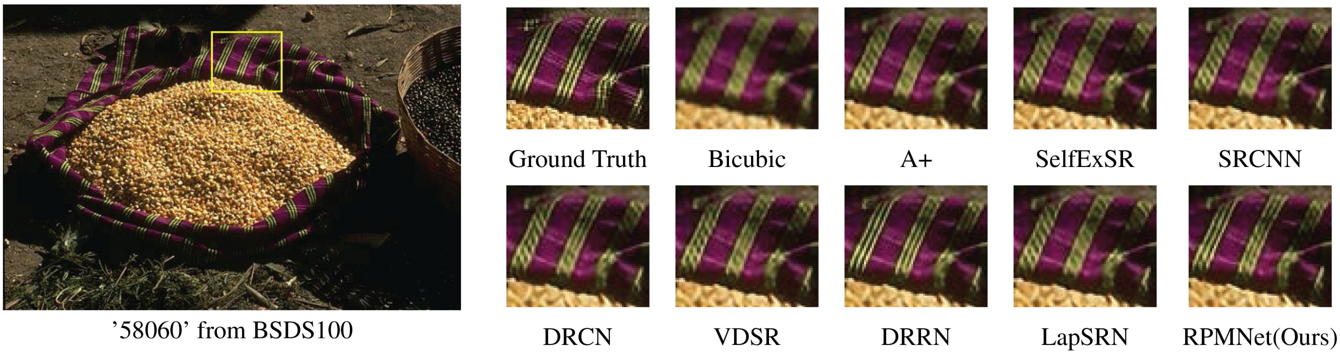

Fig. 11 *Visual comparisons with several state-of-the-art methods on scale rate of* × 2

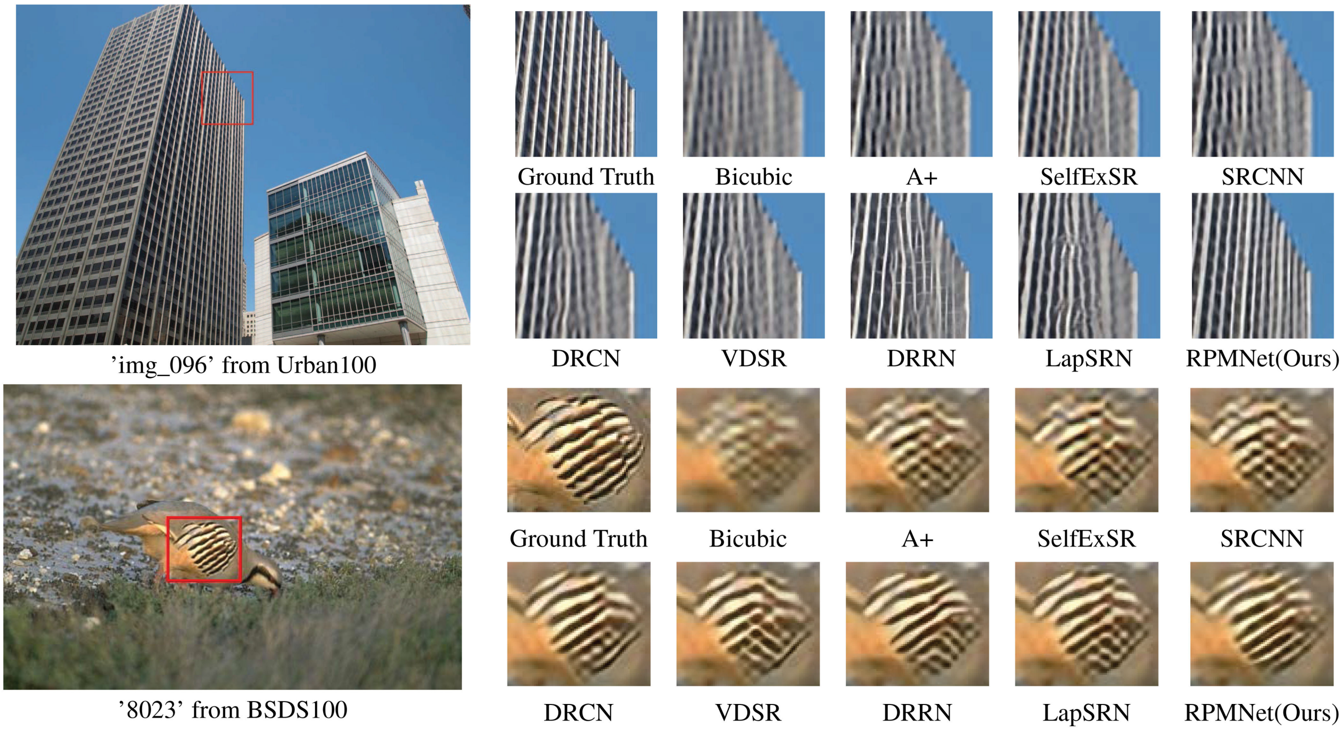

Fig. 12 *Visual comparisons with several state-of-the-art methods on scale rate of* × 4

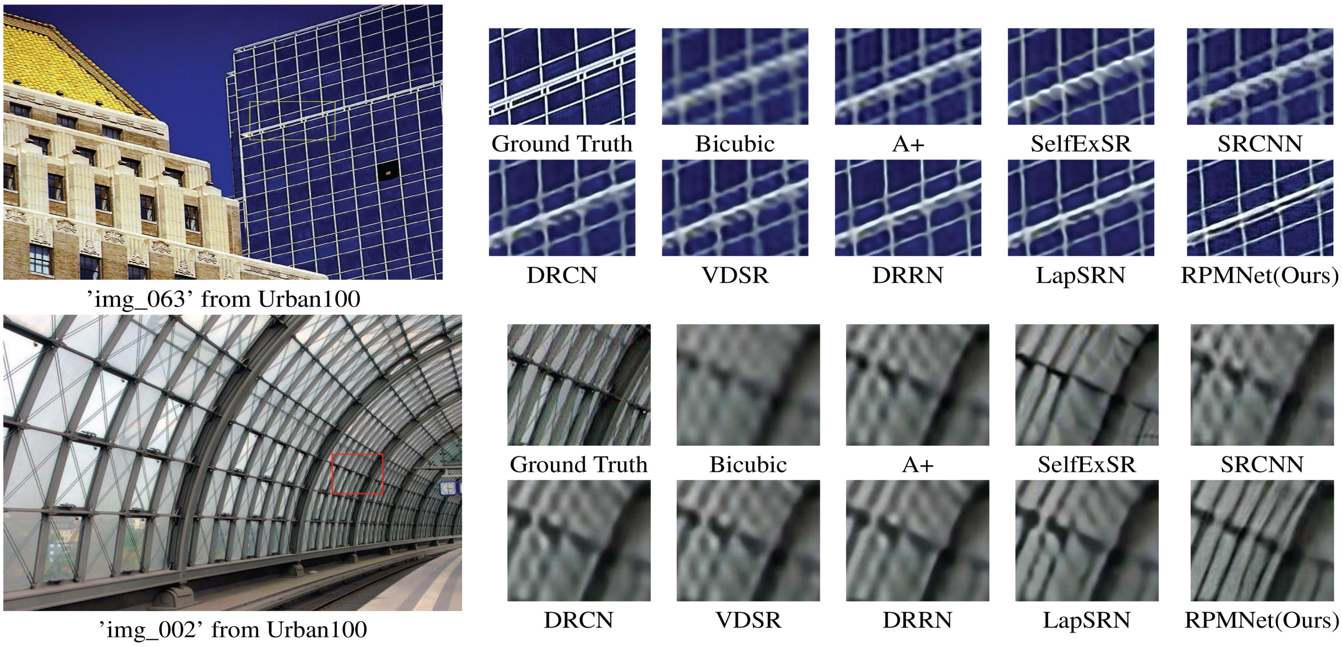

Fig. 13 *Visual comparisons with several state-of-the-art methods on scale rate of* × 8

7 References

[1] Gao, X., Zhang, K., Li, X.: 'Image super-resolution with sparse neighbor embedding', *IEEE Trans. Image Process.*, 2012, **21**, (7), pp. 3194–3205

[2] Radu, T., Vincent, D.S., Luc, V.G.: 'A+: adjusted anchored neighborhood regression for fast super-resolution'. Proc. of Asian Conf. on Computer Vision, Singapore, 2014

[3] Chao, D., Chen Change, L., Kaiming, H., *et al.*: 'Learning a deep convolutional network for image super-resolution'. Proc. of the European Conf. on Computer Vision, Zurich, Switzerlandw, 2014

[4] Chao, D., Chen Change, L., Xiaoou, T.: 'Accelerating the super-resolution convolutional neural network'. Proc. of the European Conf. on Computer Vision, Amsterdam, Netherlands, 2016

[5] Wenzhe, S., Jose, C., Ferenc, H., *et al.*: 'Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016

[6] Kim, J., Kwon Lee, J., Mu Lee, K.: 'Accurate image super-resolution using very deep convolutional networks'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016

[7] Jie, H., Li, S., Gang, S.: 'Squeeze-and-excitation networks'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018

[8] Christian, L., Lucas, T., Ferenc, H., *et al.*: 'Photo-realistic single image super-resolution using a generative adversarial network'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, USA, 2016

[9] Kaiming, H., Xiangyu, Z., Shaoqing, R., *et al.*: 'Deep residual learning for image recognition'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016

[10] Bee, L., Sanghyun, S., Heewon, K., *et al.*: 'Enhanced deep residual networks for single image super-resolution'. The IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Honolulu, USA, 2017

[11] Agustsson, E., Timofte, R.: 'Ntire 2017 challenge on single image super-resolution: dataset and study'. The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, USA, 2017

[12] Jiwon, K., Kwon Lee, J., Mu Lee, K.: 'Deeply-recursive convolutional network for image super-resolution'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016

[13] Ying, T., Jian, Y., Xiaoming, L.: 'Image super-resolution via deep recursive residual network'. The IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, USA, 2017

[14] Ying, T., Jian, Y., Xiaoming, L., *et al.*: 'Memnet: a persistent memory network for image restoration'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Venice, Italy, 2017

[15] Weisheng, L., Jiabin, H., Narendra, A., *et al.*: 'Deep laplacian pyramid networks for fast and accurate super-resolution'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, USA, 2017

[16] Tong, T., Gen, L., Xiejie, L., *et al.*: 'Image super-resolution using dense skip connections'. IEEE Int. Conf. on Computer Vision, Venice, Italy, 2017

[17] Gao Huang, L.M., Liu, Z., Weinberger, K.Q.: 'Densely connected convolutional networks'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, USA, 2017

[18] Wright, J., Ma, Y., Huang, T., *et al.*: 'Image super-resolution as sparse representation of raw image patches'. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Anchorage, USA, 2008

[19] Arbelaez, P., Maire, M., Fowlkes, C., *et al.*: 'Contour detection and hierarchical image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (5), pp. 898–916

[20] Bevilacqua, M., Roumy, A., Guillemot, C.: 'Low-complexity single-image super-resolution based on nonnegative neighbor embedding'. Proc. of the British Machine Vision Conf., Surrey, UK, 2012

[21] Zeyde, R., Elad, M., Protter, M.: 'On single image scale-up using sparse-representations'. International Conference on Curves and Surfaces, Avignon, France, June 24-30, 2010

[22] Jia Bin, H., Abhishek, S., Narendra, A.: 'Single image super-resolution from transformed self-exemplars'. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Boston, USA, 2015

[23] Matsui, Y., Ito, K., Aramaki, Y., *et al.*: 'Sketch-based manga retrieval using manga109 dataset', *Multimedia Tools Appl.*, 2017, **76**, (20), pp. 21811–21838

[24] Kingma, D., Ba, J.: 'Adam: a method for stochastic optimization'. Proc. of the 3rd Int. Conf. for Learning Representations, San Diego, USA, 2015

[25] Wang, Z., Bovik, A.C., Sheikh, H.R., *et al.*: 'Image quality assessment: from error visibility to structural similarity', *IEEE Trans. Image Process.*, 2004, **13**, (4), pp. 600–612

[26] Saxe, A.M., Mcclelland, J.L., Ganguli, S.: 'Exact solutions to the nonlinear dynamics of learning in deep linear neural networks', *IEEE Trans. Geosci. Remote Sens.*, 2018, **56**, (12), pp. 6884–6898

[27] Makantasis, K., Doulamis, A.D., Doulamis, N.D., *et al.*: 'Tensor-based classification models for hyperspectral data analysis', *IEEE Trans. Geosci. Remote Sens.*, 2018, **99**, pp. 1–15