Channel Splitting Network for Single MR Image Super-Resolution

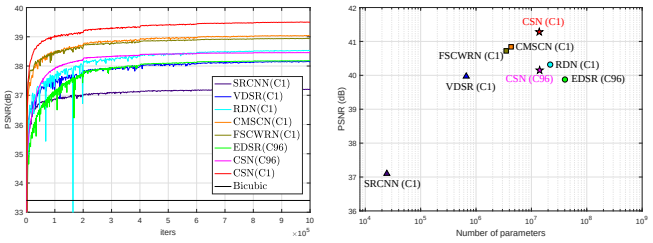
Xiaole Zhao, Yulun Zhang, Tao Zhang, and Xueming Zou

***Abstract*—High resolution magnetic resonance (MR) imaging is desirable in many clinical applications due to its contribution to more accurate subsequent analyses and early clinical diagnoses. Single image super-resolution (SISR) is an effective and cost efficient alternative technique to improve the spatial resolution of MR images. In the past few years, SISR methods based on deep learning techniques, especially convolutional neural networks (CNNs), have achieved state-of-the-art performance on natural images. However, the information is gradually weakened and training becomes increasingly difficult as the network deepens. The problem is more serious for medical images because lacking *high quality* and *effective* training samples makes deep models prone to underfitting or overfitting. Nevertheless, many current models treat the hierarchical features on different channels equivalently, which is not helpful for the models to deal with the hierarchical features discriminatively and targetedly. To this end, we present a novel channel splitting network (CSN) to ease the representational burden of deep models. The proposed CSN model divides the hierarchical features into two branches, i.e., residual branch and dense branch, with different information transmissions. The residual branch is able to promote feature reuse, while the dense branch is beneficial to the exploration of new features. Besides, we also adopt the merge-and-run mapping to facilitate information integration between different branches. Extensive experiments on various MR images, including proton density (PD), T1 and T2 images, show that the proposed CSN model achieves superior performance over other state-of-the-art SISR methods.**

***Index Terms*—Convolutional neural network, channel splitting, feature fusion, magnetic resonance imaging, super-resolution.**

I. Introduction

SPATIAL resolution is one of the most important imaging parameters for magnetic resonance imaging (MRI). In many clinical applications and research work, high resolution (HR) MRI is usually preferred because it can provide more significant structure and texture details with a smaller voxel size [1], thus promoting accurate subsequent analysis and early diagnosis. However, it is limited by several factors, e.g., hardware device, imaging time, desired signal-to-noise ratio (SNR) and body motion etc, and increasing spatial resolution of magnetic resonance (MR) images typically reduces image SNR and/or increases imaging time [2].



(a) Validation curves (b) Testing results

Fig. 1. The performance comparison between several SISR models on proton density (PD) volumes of IXI dataset for SR×2. (a) The validation results on 6 PD volumes (576 2D slices). (b) The test results vs. the number of model parameters on 70 PD volumes (6720 2D slices, Bicubic: 35.04 dB). The symbols △, □, ☆ and ○ represent models with less than 1M, 10M, 20M and more than 20M parameters respectively. C1 indicates that the training sample is a single slice, and C96 indicates that the model treats 96 slices of a 3D volume as 96 channels.

Image super-resolution (SR) is a typical ill-posed inverse problem in computer vision community, which mainly aims at inferring a HR image from one or more low resolution (LR) images. It is a well-studied problem in both natural image (NI) and MR image processing. High resolution means that the pixel density of an image is higher than its LR counterpart. Thus, HR images can offer more details that may be critical in various applications such as medical imaging [3], [4], aerial spectral imaging [5] and remote sensing imaging [6], [7] and security and surveillance [8], where high frequency details are very important and greatly desired. Up to now, many SR methods have been studied and proposed. Early methods include: (i) interpolation methods, e.g., bicubic, Lanczos-σ [9]; (ii) modeling and reconstruction methods, e.g., iterative back projection (IBP) [10], projection onto convex set (POCS) [11] etc.; (iii) traditional shallow learning methods, e.g., example learning [12], [13], dictionary learning [14], [15] etc. The performance of these methods is inherently limited because the additional information available for solving this ill-posed inverse problem is also very limited, e.g., the interpolation methods make use of the basic smoothing priori by implicitly assuming that the image signal is continuous and bandwidth limited, and traditional machine learning-based methods can learn insufficient information due to the limited representa-

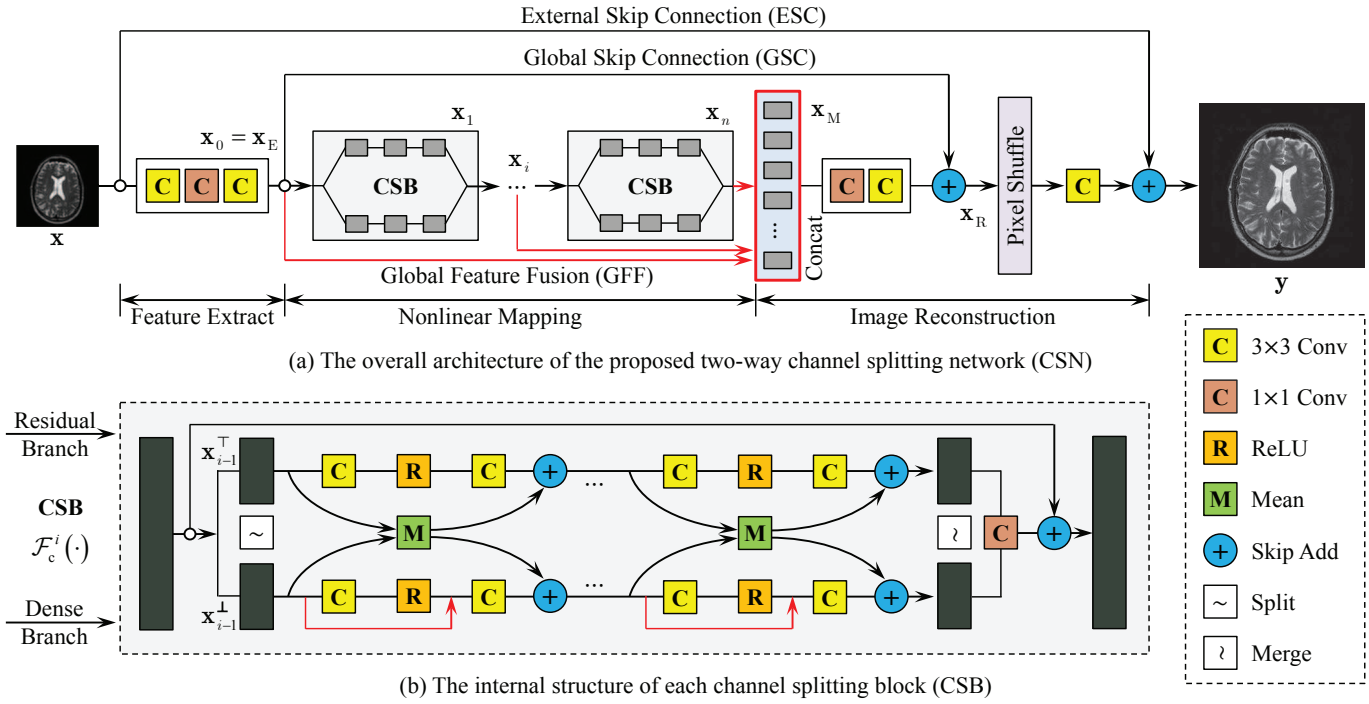
Manuscript received October 12, 2018; revised March 17, 2019; accepted May 30, 2019. Date of publication XX XX, 2019; date of current version XX XX, 2019. This work is supported in part by Sichuan Science and Technology Program under grant 2019YJ0181, and in part by National Key Research and Development Program of China under grants 2016YFC0100800 and 2016YFC0100802. *(Corresponding author: Tao Zhang.)*

X. Zhao is with the School of Life Science and Technology, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 611731, China (e-mail: zxlation@foxmail.com).

T. Zhang and X. Zou are with the High Field Magnetic Resonance Brain Imaging Laboratory of Sichuan and Key Laboratory for NeuroInformation of Ministry of Education, Chengdu, Sichuan 611731, China; They are also with the School of Life Science and Technology, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 611731, China (e-mail: taozhangjin@gmail.com; mark.zou@alltechmed.com).

Y. Zhang is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA (e-mail: yulun100@gmail.com).

Digital Object Identifier XXXX



(a) The overall architecture of the proposed two-way channel splitting network (CSN)

(b) The internal structure of each channel splitting block (CSB)

Fig. 2. The diagram of the proposed CSN model. (a) The overall structure consists of three parts: feature extraction $\mathcal{F}_{\mathrm{E}}(\cdot)$, nonlinear mapping $\mathcal{F}_{\mathrm{M}}(\cdot)$ and image reconstruction $\mathcal{F}_{\mathrm{F}}(\cdot) + \mathcal{F}_{\mathrm{R}}(\cdot)$. (b) Channel splitting block (CSB). The intermediate feature maps within a CSB are split into two branches along the channel direction. One is built as a residual-like structure (residual branch, top), and the other is built as a dense-like structure (dense branch, bottom). Red arrows in GFF [32] and dense branch indicate dense connection (channel concatenation).

tional capacity of these shallow models.

In recent years, various advanced SR methods have emerged with the rapid development of deep learning techniques [16] and greatly promoted the best state of SR performance. Super-resolution convolutional neural network (SRCNN) [17] and fast super-resolution convolutional neural network (FSRCNN) [18] are two pioneering contributions that utilize convolutional neural networks (CNNs) [19] to solve SR tasks. The further improvement based on these pioneering work mainly focused on increasing model depth or sharing model parameters at the beginning, for example, deeply recursive convolutional network (DRCN) [20], deep recursive residual network (DRRN) [21], super-resolution using very deep convolutional networks (VDSR) [22] and memory network (MemNet) [23] etc. These methods, however, are mainly aimed at the SR task of natural images, not specially at medical images.

The medical image processing community has noticed these advances and some medical image SR methods based on deep learning have also appeared [24], [25], [26], [27], [28], [29]. For MR images, training deep models with a large amount of parameters and extremely deep structures is usually more difficult because *high-quality* and *effective* training samples are relatively scarce and unavailable. It is worth noting that *the challenge is not the availability of training data itself, but the acquisition or the quality of relevant annotations/labeling for these data* [30]. Therefore, some SR models that aim at NI are likely to fail when directly trained with MR images despite sufficient training samples, e.g., Fig.1 displays the peak signal-to-noise ratio (PSNR) performance of several recent single image super-resolution (SISR) models and the proposed channel splitting network (CSN) on proton density (PD) images of IXI dataset (http://brain-development.org/ixi-dataset/) for SR×2, where enhanced deep super-resolution network (EDSR) [31] and residual dense network (RDN) [32] are advanced models on NI. But it is failed to train the EDSR model (the same configuration as [31]) with 48000 2D PD images. This problem of training failure caused by the degradation of sample quality will get worse as the network depth (or width) and the number of model parameters increase.

Thus, in the context of MR image super-resolution based on deep learning techniques, the dilemma has become more apparent: *on the one hand, models with shallow structures and fewer parameters are easy to train, but their SR performance is usually unsatisfactory; on the other hand, models with deeper structures and more parameters are promising to improve SR performance, but it is more difficult for them to be fully trained with MR images*. An effective way to alleviate the difficulty of model training is residual learning, which is initially proposed for image recognition [33], [34]. It has been widely proved to be helpful for feature reuse and model convergence, thus making it possible to build extremely deep models. However, residual learning strategy alone is still insufficient to train the model with a very deep structure and a very large number of parameters in case of MR images, e.g., EDSR [31] with about 43M parameters is a typical residual network, but the original configuration can hardly be well trained with 2D MR images in our settings. The problem of training failure can be addressed by concatenating multiple 2D MR images into a single multi-channel training sample at the expense of performance, e.g., we train the original EDSR [31] model by taking all 96 slices

of a 3D volume as 96 channels of a single training sample, as shown in Fig.1 (marked as EDSR (C96)).

In this paper, we improve the above dilemma by introducing a *deep channel splitting network (CSN)* framework. It assumes that the hierarchical features of deep models have certain clustering properties, and explicitly discriminating them is beneficial to ease the representational burden of deep models and further improve the SR performance. Therefore, instead of transferring the feature maps of the previous layer completely to the next layer, we split the feature maps into two different parts (branches) with different information transmissions. Each branch can be structured differently, e.g., in this work, we use propagation mechanisms similar to residual network (ResNet) [33], [34] and dense network (DenseNet) [35] (or RDN [32]) on each branch. Besides, the merge-and-run (MAR) mapping [36], [37] is also applied to facilitate the information integration of different branches. Thus, our model has two notable characteristics: *(1) channel splitting discriminatorily limits the hierarchical features into different clusters and reduces the representational redundancy of the model by curtailing the internal connections; (2) the merge-and-run mapping can promote information sharing and integration between the hierarchical features and therefore help to improve the information flow through the entire network*.

To make full use of the hierarchical features, we also adopt the global feature fusion (GFF) technique proposed by [32], as shown in Fig.2(a). Moreover, multilevel residual mechanism and constant scaling technique [31], [38] are also applied to our models to further stabilize the model training. To verify the effectiveness of the proposed model, a set of standard datasets for the task of single MR image SR is generated from the IXI dataset, which includes three types of MR images (i.e., PD, T1 and T2, each of which can exhibit different contrasts for the same image content.) and two kinds of degeneration (bicubic downsampling and k-space truncation). The quantitative and qualitative experiments on the datasets display the superiority of the proposed model over other advanced methods.

The rest of this paper is organized as follows. In section II, we present some previous contributions related the present work. The proposed method and the experimental results are detailed in section III and section IV, respectively. Section V gives some discussion and future work. Finally, we conclude the whole work in section VI.

II. Related Work

A. Super-Resolution with Deep Learning

Although the work using artificial neural networks (ANNs) to solve SR problems has emerged as early as 2006 [39], the pioneering work with deep learning techniques in the modern sense is SRCNN [17]. Subsequently, some advanced methods based on the increase of network depth and parameter sharing are proposed. Kim *et al.* [22] increased network depth by stacking multiple conv layers and the usage of global residual learning (GRL), and firstly introduced recursive learning trick in a deep network for parameter sharing [20]. Another network introduced by Tai *et al.* [21] has utilized recursive blocks to reuse parameters. Motivated by the fact that human thoughts have persistency, a deep persistent memory network (MemNet) which consists of the so-called memory block, has also been proposed by the same author [23]. To improve information flow and capture more sufficient knowledge for reconstructing the high frequency details, Hu *et al.* [37] proposed a cascaded multi-scale cross network (CMSCN) in which a sequence of subnetworks (Fig.3(b)) is cascaded to infer HR features in a coarse-to-fine manner. To some extent, these methods promote the design of new structures for image generation tasks.

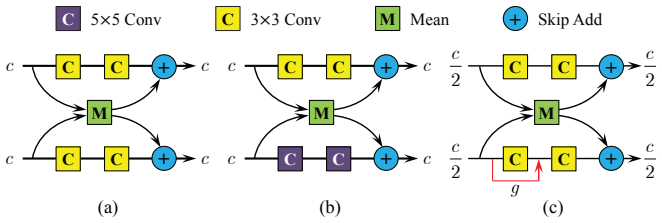


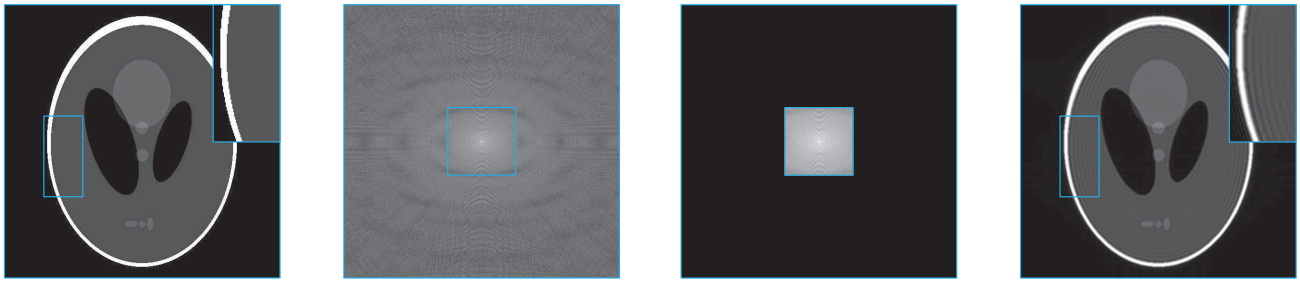
Fig. 3. The merge-and-run stage mappings in a building block. (a) Original mapping [36]. (b) Multi-scale cross (MSC) mapping [37]. (c) The proposed mapping. c is the channel number of feature maps, and g is the growth of a dense connection [35]. BN and ReLU are omitted for simplification, and the red arrow indicates the dense connection.

There is a common feature among the above methods: they use the bicubic-interpolated image of the original LR image as input to their models. This preprocessing is convenient for keeping the size of the output image consistent with the target HR image. However, it places the nonlinear inference of the network in HR image space, resulting in great computation and memory consumption. There exist two solutions for this issue currently: deconvolution or transpose convolution [18], and the efficient sub-pixel convolutional neural network (ESPCNN) [40]. Both of them can effectively solve the above problem by shifting the HR reference to the LR image space. Benefiting from the nonlinear mapping within the LR image space, these methods are capable of increasing the scale of deep models and thus boost the SR performance greatly, e.g., EDSR/MDSR [31]. Further, Zhang *et al.* [32] combined the idea of residual learning [33], [34] with densely connected DenseNet [35] and proposed a novel residual dense network (RDN) to fully utilize the hierarchical features of deep models.

Contrary to the pursuit of high performance, some methods aim to improve the tradeoff between SR performance and time efficiency to improve the practicality of the model, e.g., [40], [52], [58]. Overall, although these methods favor relatively fast inference, they are still lightweight and small-scale so that their representational capacity is limited to some extent.

B. MR Image Super-Resolution

The early application of SR techniques to medical images mainly focuses on multi-frame image super-resolution (MFSR) tasks. For example, IBP [10] was adopted to generate a new image with increased spatial resolution from several spatially shifted, single-shot and diffusion-weighted brain MR images [41]. Greenspan *et al.* [42] and Shilling *et al.* [43] employed IBP and POCS [11] to produce a 3D MR volume with isotropic resolution from several 2D slices, respectively. These methods are usually accompanied with specific data acquisitions (e.g.,



(a) HR image (b) Fourier transform (c) k-space truncation (d) Inverse recovery

Fig. 4. The illustration of k-space truncation degradation (TD) for SR×4. Different from bicubic downsampling, the generated LR images are sometimes accompanied by Gibbs-ringing artifacts. The result in (d) is generated by zero-padding in k-space and inverse Fourier transform (IFT) for display purpose.

rotation, scaling and translation) to simulate the generation of LR images. However, recovering a HR image from multiple degraded LR images usually needs to calibrate and fuse these LR images, which is a very challenging task in itself.

To avoid the difficulty of calibration and fusion between multiple LR images, Rousseau [44] first proposed to enhance MR image resolution with single image SR techniques. In this method, the extra information was introduced into the reconstruction process by referring to another HR image. A similar method was proposed by Manjón *et al.* [45]. They also used a HR image as reference but with a different strategy to produce HR images. These methods introduce very limited extra information because they learn knowledge from only one external HR image. SR methods based on conventional machine learning, e.g., sparse representation [46], [47] and compressive sensing [48], are also applied to medical images subsequently. Recently, more advanced SR methods based on deep learning [16] have also been applied to MR image SR tasks [24], [25], [26], [27], [49]. However, these methods simply use deep learning techniques to deal with the SR tasks of MR images without considering the differences between natural images and medical images. Contrarily, the proposed CSN model aims at dealing with the hierarchical features discriminatively and reducing the representational burden of the model to adapt the degradation of MR training samples.

C. Multi-Stream Networks

Multi-stream networks are widely adopted in the image SR community to boost the model performance by assembling the information from different streams (paths). Wang *et al.* [50] explored an end-to-end CNN architecture by jointly training both deep and shallow CNN networks, where the shallow one stabilizes model training and the deep one ensures an accurate HR reconstruction. Ren *et al.* [51] proposed a context-wise network fusion approach to integrate the outputs of individual networks by extra convolutional layers. Yamanaka *et al.* [52] combined skip connection layers and parallelized CNNs into a single CNN architecture. CMSCN [37] is another multi-stream structure, in which complementary information under different receptive fields is integrated by the merge-and-run mechanism [36] (Fig.3). There are also multi-stream structures for medical image SR tasks, e.g., Oktay *et al.* [49] developed a multi-input cardiac image SR network, which is capable of assembling information from different viewing planes to improve the SR performance. These methods are fundamentally different from the proposed CSN model, in that they form the multi-stream structure by the reuse of the preceding features, while our CSN network construct the multi-stream structure by splitting the preceding features into different branches.

III. Proposed Method

A. Overall Network Architecture

The overall structure of the proposed CSN model is outlined in Fig.2. Similar to other deep models for image SR, it consists mainly of 3 parts: feature extraction, nonlinear mapping and image recovery. Firstly, the feature extraction network (FEN) is employed to express the input image $\mathbf{x}$ as a set of shallow features. These shallow features are then transmitted to the nonlinear mapping network (NMN), which contains a series of stacked channel splitting blocks (CSB). Subsequently, the hierarchical features from all CSBs are concatenated together to produce the final output of the NMN. This operation is also called global feature fusion (GFF) [32]. Finally, the collected deep hierarchical features are fed into the image reconstruction network (IRN) to generate the final HR predication $\mathbf{y}$ of the entire CSN model.

1) Feature Extraction: The FEN contains two 3×3 conv layers with a 1×1 conv layer in the middle. Denote $\mathcal{F}_{\mathrm{E}}(\cdot)$ as the corresponding mapping function, then the shallow features $\mathbf{x}_{\mathrm{E}}$ extracted by the FEN can be represented as:

$$\mathbf{x}_{\mathrm{E}} = \mathcal{F}_{\mathrm{E}}(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is the original LR input. The 1×1 conv layer indicates a point-to-point linear transformation of the features extracted by the first 3×3 conv layer. This 1×1 conv layer is considered to be helpful to further improve the robustness of the extracted features because the features on different channels also contain spatial information in the context of image SR.

2) Nonlinear Mapping: The entire NMN net is denoted as $\mathcal{F}_{\mathrm{M}}(\cdot)$. Therefore, the output of the NMN is given by $\mathcal{F}_{\mathrm{M}}(\mathbf{x}_{\mathrm{E}})$. Supposing we have n CSBs in the entire network and $\mathbf{x}_0 = \mathbf{x}_E$ is the input of the first CSB, then the output $\mathbf{x}_i$ of the i-th CSB can be obtained by:

$$\mathbf{x}_i = \mathcal{F}_{\mathrm{c}}^i(\mathbf{x}_{i-1}), \quad i = 1,\ 2,\ \ldots,\ n, \tag{2}$$

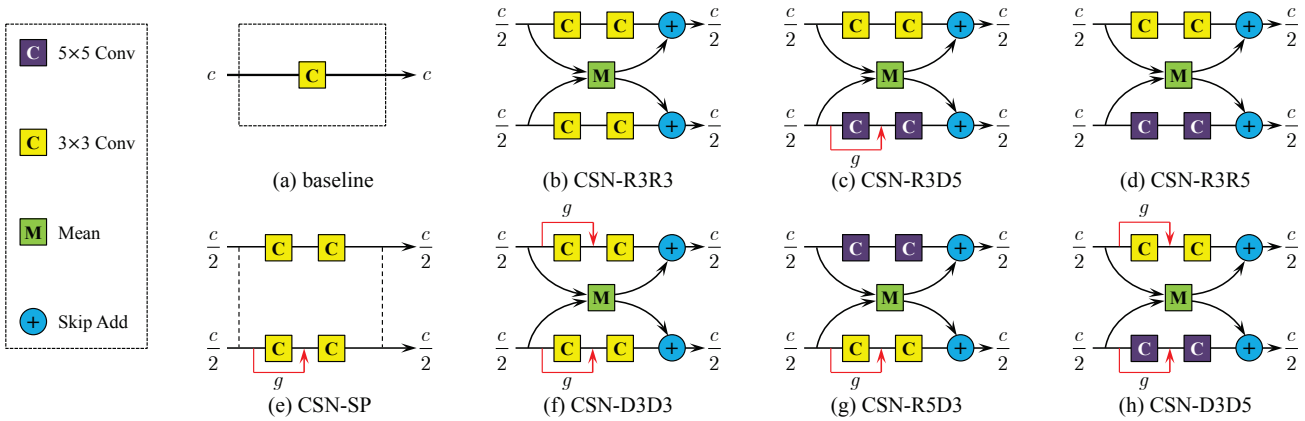


Fig. 5. Several stage mapping structures for comparison. The nonlinear function ReLU between two conv layers is omitted for simplification, and red arrows represent dense connection. Note Fig.3(a) is different from CSN-R3R3 and there is a ReLU layer after the conv layer in (a). Please refer to Fig.2(b) for the overall structure of a CSB and the detailed structure of CSN-R3D3 stage mapping.

where the function $\mathcal{F}_{\mathrm{c}}^{i}(\cdot)$ corresponds to the operations of the i-th CSB. More details about $\mathcal{F}_{\mathrm{c}}^{i}(\cdot)$ will be presented in section III-B. Therefore, the output of the last CSB can be iteratively formulated as follow:

$$\mathbf{x}_n = \mathcal{F}_{\mathrm{c}}^{n}(\mathbf{x}_{n-1}) = \mathcal{F}_{\mathrm{c}}^{n}(\mathcal{F}_{\mathrm{c}}^{n-1}(\cdots(\mathcal{F}_{\mathrm{c}}^{1}(\mathbf{x}_0))\cdots)). \quad (3)$$

The output tensor of the i-th CSB $\mathbf{x}_i$ is produced by a series of operations (e.g., convolution, ReLU and constant scaling etc.) within the block, so it is viewed as a set of local feature maps [32]. These local feature maps constitute the final output of our nonlinear mapping network. It should be noted that the output of the preceding CSB is directly used as the input of the next CSB. This is similar to the so-called continuous memory (CM) mechanism [32] and contributes to the information propagation in the network [34].

3) Image Reconstruction: This phase includes two related parts: global fusion of the local features $\mathbf{x}_i$ and the restoration of HR images based on the fused features. To fuse the local features, the outputs of all CSBs are first concatenated into a single tensor (red rectangle in Fig.2(a)):

$$\mathbf{x}_{\mathrm{M}} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n], \quad (4)$$

where $[\ldots]$ implies the concatenation. Next, the global features are extracted by fusing all local features from all the preceding channel splitting blocks. This is completed by a 1×1 conv followed by a 3×3 conv ($\mathcal{F}_{\mathrm{F}}(\cdot)$ in Fig.2). Finally, the global residual learning (GRL) is used to stabilize the training of the model [31], [32], which is simply implemented via a global skip connection (GSC):

$$\mathbf{x}_{\mathrm{R}} = \mathcal{F}_{\mathrm{F}}(\mathbf{x}_{\mathrm{M}}) + \mathbf{x}_0 = \mathcal{F}_{\mathrm{F}}([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n]) + \mathbf{x}_0, \quad (5)$$

where $\mathbf{x}_{\mathrm{R}}$ is the fused features that will be used to recover the HR image $\mathbf{y}$. As for HR image restoration, it is mainly made up of a pixel shuffle layer followed by a 3×3 convolutional layer, and an external residual learning (ERL). Formally, it can be represented as:

$$\mathbf{y} = \mathcal{F}_{\mathrm{R}}(\mathbf{x}_{\mathrm{R}}) + \mathbf{x}, \quad (6)$$

where $\mathcal{F}_{\mathrm{R}}(\cdot)$ is the function corresponding to the pixel shuffle layer and the following 3×3 convolutional layer, and $\mathbf{x}$ is the original LR input to the model. Note that the pixel shuffle layer is implemented by ESPCNN [40] in the way of [31].

B. Channel Splitting Block

The CMSCN network explored by [37] is a multi-stream structure that integrates the complementary information under different receptive fields. Moreover, it has been proved that residual learning enables feature reuse and dense learning enables new features exploration, both of which are important for learning good representations [53]. Inspired by this, we present a two-way channel splitting block (CSB) to incorporate different information with different propagation mechanisms. As shown in Fig.2(b), the distinctive features of the proposed CSB are *channel splitting and merging*, and *fusion of residual learning and dense learning*. Besides, local residual learning (LRL) is also applied to further improve the information propagation. It has been shown that LRL is helpful to stabilize the training process and improve the representational capacity of the model, resulting better SR performance [32].

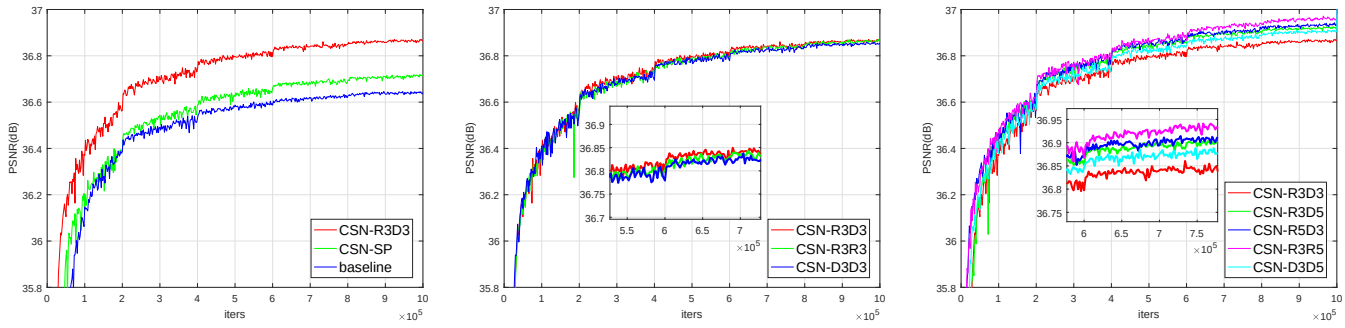
1) Channel Splitting and Merging: As for the i-th CSB, the input tensor $\mathbf{x}_{i-1}$ is first equally split into two tensors along the channel direction, $\mathbf{x}_{i-1}^{\perp}$ and $\mathbf{x}_{i-1}^{\top}$, which are the inputs of the lower (dense) branch and the upper (residual) branch respectively. It could be formally expressed as:

$$\mathbf{x}_{i-1}^{\perp},\ \mathbf{x}_{i-1}^{\top} = \mathcal{S}_{\mathrm{c}}(\mathbf{x}_{i-1}), \qquad i = 1,\ 2,\ \ldots,\ n, \quad (7)$$

where $\mathcal{S}_{\mathrm{c}}(\cdot)$ is the channel splitting function. It can be viewed as a unary operator that splits the input tensors into two parts along the channel direction. Through this channel splitting operation, we can apply different information transmission mechanisms on each branch. For example, we adopt residual learning and dense learning on the upper and lower branches respectively. Correspondingly, there exists a channel merging operation, $\mathcal{M}_{\mathrm{c}}(\cdot)$, at the end of each CSB:

$$\mathbf{x}_{i-1} = \mathcal{M}_{\mathrm{c}}(\mathbf{x}_{i-1}^{\perp},\ \mathbf{x}_{i-1}^{\top}), \quad i = 1,\ 2,\ \ldots,\ n. \quad (8)$$

It should be noted that $\mathcal{M}_{\mathrm{c}}(\cdot)$ is a bivariate function, while the $[\ldots]$ in (4) and (5) is a multivariate operator. The channel splitting and merging operations are expected to artificially



(a) (b) (c)

Fig. 6. The performance comparison between different structures shown in Fig.5 on $\mathcal{V}$(T1, TD) for SR×2 (bicubic = 31.72dB). (a) Channel splitting and the merge-and-run mapping. (b) Different branch structures. (c) Different kernel sizes. The baseline stage mapping is a single convolutional layer followed by a ReLU operation (Fig.5(a)), which emphasizes the impact of channel splitting.

interfere with the information flow within the network, which helps the model to process the hierarchy features with different properties in a targeted way. In addition, it is also an effective manner to maintain the scale of model parameters and increase the depth of the network.

2) Feature Reuse and New Feature Exploration: A CSB module assembles two residual-like branches in parallel with a merge-and-run mapping [36] but each branch has different structures. Suppose the i-th CSB contains m stage mappings and each branch in a stage includes two convolutional layers with a ReLU operation in the middle (Fig.2(b)). Denote $\mathcal{H}^{\perp}_{i,j}(\cdot)$ and $\mathcal{H}^{\top}_{i,j}(\cdot)$ as the transition functions of the lower and the upper residual branches in the j-th stage mapping respectively. Then the transition function of the j-th stage mapping can be represented in matrix form as below:

$$\begin{bmatrix}\mathbf{x}^{\top}_{i-1,j}\\ \mathbf{x}^{\perp}_{i-1,j}\end{bmatrix} = \begin{bmatrix}\mathcal{H}^{\top}_{i,j}(\mathbf{x}^{\top}_{i-1,j-1})\\ \mathcal{H}^{\perp}_{i,j}(\mathbf{x}^{\perp}_{i-1,j-1})\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{x}^{\top}_{i-1,j-1}\\ \mathbf{x}^{\perp}_{i-1,j-1}\end{bmatrix}, \quad (9)$$

where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$ are the index of the i-th CSB and the j-th stage mapping in this CSB. $\mathbf{x}^{\perp}_{i-1,j-1}$ and $\mathbf{x}^{\top}_{i-1,j-1}$ ($\mathbf{x}^{\perp}_{i-1,j}$ and $\mathbf{x}^{\top}_{i-1,j}$) are the inputs (outputs) of the j-th stage mapping in the i-th CSB, and $\mathbf{x}^{\perp}_{i-1,0} = \mathbf{x}^{\perp}_{i-1}$ and $\mathbf{x}^{\top}_{i-1,0} = \mathbf{x}^{\top}_{i-1}$ are the input tensors of the lower and upper branches respectively. $\mathbf{I}$ denotes identity matrix. Therefore, the coefficient matrix

$$\mathbf{C} = \frac{1}{2}\begin{bmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & \mathbf{I}\end{bmatrix}, \quad (10)$$

is an idempotent transformation matrix of the merge-and-run mapping [36], [37]. The idempotent property can promote the information flow across the different modules and encourage gradient back-propagation during model training, similar to identity mapping [34]. It is worth noting that the upper branch is a residual-like structure similar to EDSR [31] and the lower branch is a simplified dense-like structure similar to DenseNet [35] or RDB [32], which uses only one skip dense connection to explore new features. Through merge-and-run mapping, we can effectively integrate the superiority of feature reuse and new feature exploration provided by the residual branch and the dense branch.

3) Local Residual Learning (LRL): The feature maps from these two branches, $\mathbf{x}^{\perp}_{m}$ and $\mathbf{x}^{\top}_{m}$, are merged together after m stages of merge-and-run mappings. Next, a local residual learning (LRL) [32] is also introduced in the CSB to further improve the information flow. The output of this CSB module is thus given by:

$$\mathbf{x}_i = \mathcal{L}(\mathcal{M}_{\mathrm{c}}(\mathbf{x}^{\perp}_{i-1,m},\ \mathbf{x}^{\top}_{i-1,m})) + \mathbf{x}_{i-1}, \quad (11)$$

where $\mathcal{L}(\cdot)$ corresponds to a 1×1 convolutional operation at the end of the CSB, as shown in Fig.2(b). Unlike [32], the local residual features are derived from our CSB module, instead of the densely connected block [35].

C. Multilevel Residual Mechanism

Normally, the LR images and the corresponding HR images share same information to a large extent, which indicates that a large part of the topological structure of their high-dimensional manifolds are similar to each other. Therefore, it is beneficial to explicitly allow the model to learn the residual between the *original* LR input and the HR output [22]. However, because LR and HR images have different sizes, the residual between them cannot be directly obtained. We adopt a bicubic interpolated version of the LR image to match the size of the HR image, and use it to approximate the residual between the original LR image and the HR image. This is implemented by simply adding the interpolated image to the output of the last convolutional layer of the entire network, which we term as external skip connection (ESC) (Fig.2(a)). Thus, (6) should be rewritten as:

$$\mathbf{y} = \mathcal{F}_{\mathrm{R}}(\mathbf{x}_{\mathrm{R}}) + \hat{\mathbf{x}}, \quad (12)$$

where $\hat{\mathbf{x}}$ is the interpolated version of the original LR input $\mathbf{x}$. Although we use bicubic interpolation here, one can use any other interpolation algorithm (e.g., nearest neighbour, bilinear and B-Spline etc.).

Combined with GSC and LRL, the whole network shows a characteristic of multilevel residual learning. Our experience shows that this can further stabilize the training process, and even help to slightly improve the model performance. Because the degeneration of MR training samples causes model training more unstable, it is especially helpful for the task of single MR image super-resolution.

TABLE I
THE DETAILED CONFIGURATION OF THE PROPOSED CSN MODEL. ONLY ONE SINGLE STAGE MAPPING IS SHOWN IN NMN DUE TO THE EXACTLY SAME STRUCTURE OF EACH CSB AND EACH STAGE MAPPING. ALL CONV LAYERS ARE PADDED TO KEEP THE SIZE OF FEATURE MAPS UNCHANGED.

Config	FEN (three conv layers)			NMN (stage mapping $\times$ m = CSB, CSB $\times$ n)					IRN (Upscaling depends on the scaling factor r)				
	General convolution			Residual branch		Dense branch		Merge	Feature Fusion		Upscale ($r = 2\|3, 4$)		Recover
Layer	conv1	conv2	conv3	conv1	conv2	conv1	conv2	conv1	conv1	conv2	conv1	conv1,2	conv1
Filter	256	256	256	128	128	64	128	256	256	256	$256{\cdot}r^2$	$256{\cdot}2^2$	1|96
Kernel	3×3	1×1	3×3	3×3	3×3	3×3	3×3	1×1	1×1	3×3	3×3	3×3	3×3
Act	/	/	/	ReLU	/	ReLU	/	/	/	/	/	/	/

TABLE II
THE TESTING PERFORMANCE OF DIFFERENT STRUCTURES SHOWN IN FIG.5 ON $\mathcal{T}$(T1, TD) FOR SR×2. THE MAXIMAL VALUES OF EACH COLUMN ARE IN RED, AND THE SECOND ONES ARE IN BLUE.

Network Configuration	Network Parameters	Network Depth	PSNR (dB)	SSIM
Bicubic	/	/	33.38	0.9460
baseline	13643521	27	38.37	0.9803
CSN-SP	13646593	43	38.46	0.9807
CSN-R3D3	13646593	43	38.62	0.9813
CSN-R3R3	13647614	43	38.61	0.9813
CSN-D3D3	13645566	43	38.59	0.9812
CSN-R3D5	22035198	43	38.67	0.9815
CSN-R5D3	22035198	43	38.68	0.9816
CSN-R3R5	22036225	43	38.70	0.9817
CSN-D3D5	22034177	43	38.64	0.9814

TABLE III
TEST RESULTS OF THE MODELS WITH DIFFERENT ESC APPROXIMATIONS ON $\mathcal{T}$(PD, BD). THE MAXIMAL PSNR AND SSIM VALUES OF EACH ROW ARE IN RED, AND THE SECOND ONES ARE IN BLUE.

scale	ESC-None	ESC-NN	ESC-Bilinear	ESC-Bicubic
×2	41.20/0.9893	41.19/0.9893	41.18/0.9893	41.28/0.9895
×3	35.80/0.9688	35.79/0.9688	35.82/0.9689	35.87/0.9693
×4	33.32/0.9478	33.31/0.9482	33.34/0.9483	33.40/0.9486

D. Training Objective and Network Depth

Our model is a typical end-to-end mapping from LR images to HR images. The estimation of model parameters is achieved by minimizing the loss between the reconstructed HR images and the ground truth HR images. Given a training dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\},\ i = 1, 2, \ldots, |\mathcal{D}|$, where $|\mathcal{D}|$ is the total number of training samples, we use l_1 loss for model training:

$$L(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} ||\mathbf{y}^{(i)} - \mathcal{F}_{\text{CSN}}(\mathbf{x}^{(i)}; \boldsymbol{\theta})||_1, \tag{13}$$

where $\boldsymbol{\theta}$ indicates the set of model parameters, and $\mathcal{F}_{\text{CSN}}(\cdot)$ is the mapping function of the entire CSN model. $\mathbf{y}^{(i)}$ is the HR target corresponding to the LR input $\mathbf{x}^{(i)}$. Despite that minimizing l_2 loss is generally preferred since it maximizes the peak signal to noise ratio (PSNR), l_1 loss provides better convergence for model training [31]. This is especially helpful in case of the degradation of training samples.

The depth of a deep network is usually defined as the longest path from the input to the output. Thus, the depth of the overall CSN model is given by:

$$D = n(2m + 1) + s + 6, \tag{14}$$

where n is the number of CSBs in the entire network and m is the number of stage mappings in each CSB. s represents the depth of the pixel shuffle layer. Note that s depends on the scaling factor [31], i.e., $s = 1$ for SR×2 and SR×3, and $s = 2$ for SR×4.

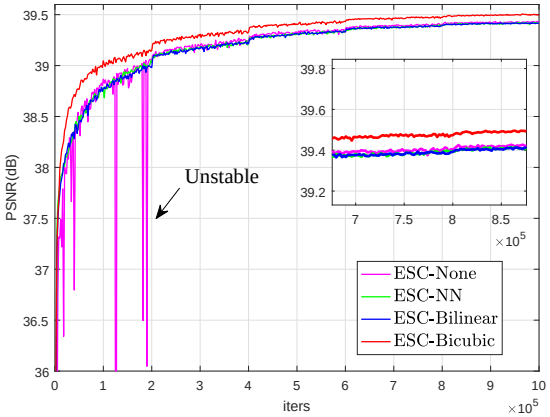


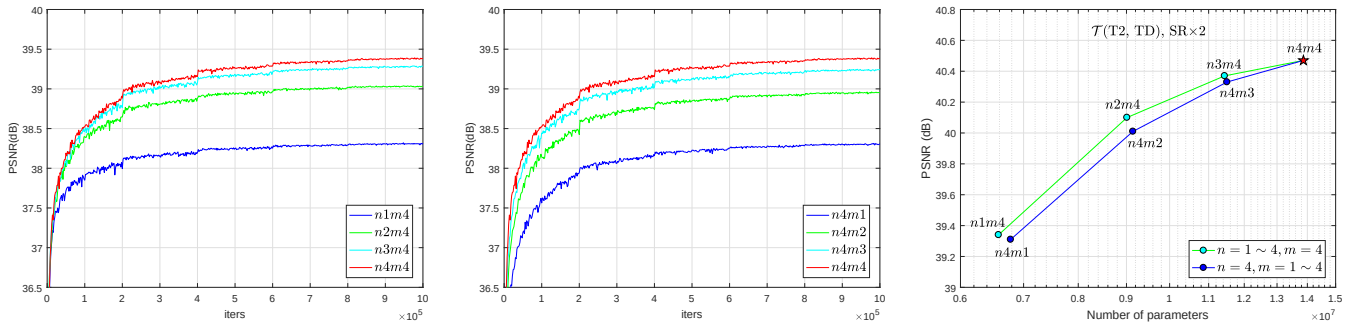
Fig. 7. The impact of different ESC approximations on model training and performance. The comparison is carried out on $\mathcal{D}$(PD, BD) for SR×2. Note that ESC-None shows obvious instability.

IV. EXPERIMENTAL RESULTS

In this section, we first introduce the generation of training examples and the implementation details. Then we investigate the impact of different configurations of CSB and the whole network on SR performance. Next, our CSN model is compared with several typical SISR methods under two common image degradations: bicubic downsampling (BD) and k-space truncation (TD). We use PSNR and structural similarity index metric (SSIM) [54] as the metrics of quantitative evaluation.

A. Dataset and Sample Generation

The IXI dataset is used to construct our SR datasets, and it contains three types of MR images: 581 T1 volumes, 578 T2 volumes and 578 PD volumes. Firstly, we take the intersection of these three subsets, resulting in 576 3D volumes for each type of MR images. These 3D volumes are then clipped to the size of 240×240×96 (height×width×depth) to fit 3 scaling factors (×2, ×3 and ×4). In this work, we only focus on the in-plane SR of 2D MR slices. Therefore, each 3D MR volume contains 96 training samples with a single channel. The LR images are generated according to bicubic downsampling and k-space truncation. As for truncation degradation, the HR images are first converted into k-space by discrete Fourier transform (DFT) and then truncated along both height and width directions (Fig.4). We randomly selected 500 volumes for training ($\mathcal{D}$), 70 volumes for testing ($\mathcal{T}$) and the remaining



(a) Impact of building blocks (n) (b) Impact of stage mappings (m) (c) Comparison on $\mathcal{T}(\text{T2, TD})$

Fig. 8. The performance comparison between the models with different number of stage mappings and building blocks. (a) and (b): The validation performance of the models on $\mathcal{V}(\text{T2, TD})$ (SR×2, Bicubic: 31.92dB). (c) The testing performance of all compared models on $\mathcal{T}(\text{T2, TD})$ (SR×2, Bicubic: 33.06dB).

for quick validation ($\mathcal{V}$). We employ the convention: *dataset name (MR image type, image degradation model)*, to indicate a specific sub dataset for convenience. For instance, $\mathcal{D}(\text{T2, BD})$ represents the T2 training dataset with bicubic degradation and $\mathcal{T}(\text{PD, TD})$ represents the PD testing dataset with k-space truncation degradation. The processed datasets are available at: https://pan.baidu.com/s/1Ak3GiJk5H1Pdn3igzElb7w (**kn3d**).

At present, the model only targets at the task of single 2D MR image super-resolution. Thus, we have $500 \times 96 = 48000$ training examples in a single training dataset. The generated datasets can be conveniently applied to develop 3D algorithms as each dimension is clipped to the common multiple of 2, 3 and 4, which will be a part of our future work.

B. Implementation Details

The configuration of the model is shown in Fig.2 with $n = m = 4$. The size of minibatch and the number of feature maps are set to 16 and 256 respectively. For the dense branch within a CSB, the growth (g in Fig.3 and Fig.5) is set to 64. If not specified, the kernel size follows the annotation of Fig.2.

We train the models by using image patches of size 24×24 randomly extracted from LR slices with the corresponding HR patches. Data augmentation is simply implemented by random horizontal flips and 90° rotations, as [31] and [32]. All models are implemented (or reimplemented) in TensorFlow 1.7.0 and trained on a NVIDIA GeForce GTX 1080 Ti GPU for one million iterations. We adopt Xavier initialization [56] for all model parameters and Adam optimizer [55] to minimize the loss by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Learning rate is initialized as 10^{-4} for all layers and halved at every 2×10^5 iterations i.e., piecewise constant decay.

C. Model Analysis

In this section, we study several components of the proposed model, including the structure of stage mapping, multilevel residual learning, global feature fusion and building block utilization. The structure of the entire network and the building block refers to Fig.2.

1) Channel Splitting Block: The proposed stage mapping can be configured in several ways, thus equipping different CSB modules. For comparison, we have studied the structure of different stage mappings from the following aspects:

- If without channel splitting, four convolutional layers in a stage mapping correspond to a single convolutional layer with nearly the same number of model parameters. It is a reference structure of a stage mapping and we take it as the baseline, as shown in Fig.5(a).
- To investigate the role of the MAR mapping, we remove it from the proposed CSB and obtain the structure shown in Fig.5(e). We term it as CSN-SP, where S means splitting and P means plain.
- We also design two stage mappings shown in Fig.5(b) and Fig.5(f) to check the effect of different branch structures on the performance of the model. They are referred as CSN-R3R3 and CSN-D3D3 respectively, where R and D represent residual branch and dense branch, and the numbers indicate the kernel size.
- To study the impact of different kernel sizes, four other structures are designed. They are termed as CSN-R3D5 (Fig.5(c)), CSN-R5D3 (Fig.5(g)), CSN-R3R5 (Fig.5(d)) and CSN-D3D5 (Fig.5(h)), respectively.

The proposed stage mapping structure (as shown in Fig.2(b) and Fig.3(c)) is marked as CSN-R3D3. Both m and n are set to 4 for all experiments in this section.

The performance of the compared stage mapping structures on $\mathcal{V}(\text{T1, TD})$ for SR×2 is shown in Fig.6. It can be seen from Fig.6(a) that both channel splitting and the MAR mapping can significantly improve the model performance. According to Fig.6(b), the performance of CSN-R3D3 is slightly better than that of CSN-R3R3 and CSN-D3D3. However, it is noteworthy that *the model parameters R3R3>R3D3>D3D3, and the depth of these networks is the same*, implying that mixing different branch structures is indeed helpful to boost the performance of the model, although slightly. It is observed from Fig.6(c) that the model performance R3R5>R5D3>R3D5>D3D5>R3D3. However, the parameters of the first four structures are about 1.6 times that of CSN-R3D3, causing a worse tradeoff between model performance and model scale. We can assume that their performance improvement on CSN-R3D3 is mainly due to the increase of model parameters. In addition, we can find that the residual branch favors better performance. The conclusions are further verified by the testing results shown in Table II.

2) External Skip Connection: To investigate the impact of external skip connections (ESC), we build three other models

TABLE IV
QUANTITATIVE COMPARISON BETWEEN DIFFERENT METHODS ON 6 TEST DATASETS (2 IMAGE DEGRADATIONS AND 3 MR IMAGE TYPES). THE MAXIMAL PSNR (DB) AND SSIM VALUES OF EACH COMPARISON GROUP ARE MARKED IN RED, AND THE SECOND ONES ARE MARKED IN BLUE (PSNR / SSIM).

method \ dataset	mode	scale	bicubic downsampling $\mathcal{T}$(:, BD)			k-space truncation $\mathcal{T}$(:, TD)		
			PD	T1	T2	PD	T1	T2
Bicubic [2D]	C1	×2	35.04 / 0.9664	33.80 / 0.9525	33.44 / 0.9589	34.65 / 0.9625	33.38 / 0.9460	33.06 / 0.9541
NLM [57]		×2	37.26 / 0.9773	35.80 / 0.9685	35.58 / 0.9722	36.18 / 0.9707	34.71 / 0.9581	34.56 / 0.9641
SRCNN [17]		×2	38.96 / 0.9836	37.12 / 0.9761	37.32 / 0.9796	38.23 / 0.9802	36.52 / 0.9705	37.04 / 0.9773
VDSR [22]		×2	39.97 / 0.9861	37.67 / 0.9783	38.65 / 0.9836	39.89 / 0.9850	37.58 / 0.9760	38.74 / 0.9823
RDN [32]		×2	40.31 / 0.9870	37.95 / 0.9795	38.75 / 0.9838	40.39 / 0.9862	38.08 / 0.9784	40.02 / 0.9826
CMSCN [37]		×2	40.84 / 0.9883	38.06 / 0.9800	39.54 / 0.9857	41.14 / 0.9882	38.23 / 0.9795	39.63 / 0.9845
FSCWRN [28]		×2	40.72 / 0.9880	37.98 / 0.9797	39.44 / 0.9855	40.91 / 0.9876	38.04 / 0.9786	39.82 / 0.9851
CSN [Ours]		×2	41.28 / 0.9895	38.27 / 0.9810	39.71 / 0.9863	41.77 / 0.9897	38.62 / 0.9813	40.47 / 0.9868
EDSR [31]	C96	×2	39.87 / 0.9857	37.56 / 0.9774	38.28 / 0.9824	39.47 / 0.9837	37.09 / 0.9741	38.11 / 0.9803
CSN [Ours]		×2	40.15 / 0.9865	37.60 / 0.9778	38.53 / 0.9831	39.50 / 0.9839	36.99 / 0.9737	38.20 / 0.9807
Bicubic [2D]	C1	×3	31.20 / 0.9230	30.15 / 0.8900	29.80 / 0.9093	30.88 / 0.9167	29.79 / 0.8793	29.50 / 0.9016
NLM [57]		×3	32.81 / 0.9436	31.74 / 0.9216	31.28 / 0.9330	32.02 / 0.9324	30.83 / 0.9027	30.57 / 0.9197
SRCNN [17]		×3	33.60 / 0.9516	32.17 / 0.9276	32.20 / 0.9440	32.90 / 0.9432	31.72 / 0.9187	31.80 / 0.9381
VDSR [22]		×3	34.66 / 0.9599	32.91 / 0.9378	33.47 / 0.9559	34.27 / 0.9555	32.57 / 0.9304	33.23 / 0.9515
RDN [32]		×3	35.08 / 0.9628	33.31 / 0.9430	33.91 / 0.9591	35.00 / 0.9609	33.33 / 0.9416	33.99 / 0.9576
CMSCN [37]		×3	35.26 / 0.9641	33.25 / 0.9423	34.16 / 0.9613	35.41 / 0.9638	33.18 / 0.9398	34.45 / 0.9611
FSCWRN [28]		×3	35.37 / 0.9653	33.24 / 0.9423	34.27 / 0.9618	35.30 / 0.9636	33.09 / 0.9390	34.34 / 0.9603
CSN [Ours]		×3	35.87 / 0.9693	33.53 / 0.9464	34.64 / 0.9647	36.09 / 0.9697	33.68 / 0.9464	34.95 / 0.9653
EDSR [31]	C96	×3	34.39 / 0.9578	32.76 / 0.9347	33.15 / 0.9528	33.97 / 0.9531	32.27 / 0.9274	32.89 / 0.9482
CSN [Ours]		×3	34.68 / 0.9598	32.83 / 0.9360	33.36 / 0.9547	34.12 / 0.9540	32.25 / 0.9266	33.00 / 0.9490
Bicubic [2D]	C1	×4	29.13 / 0.8799	28.28 / 0.8312	27.86 / 0.8611	28.82 / 0.8713	27.96 / 0.8182	27.60 / 0.8511
NLM [57]		×4	30.27 / 0.9044	29.31 / 0.8655	28.85 / 0.8875	29.27 / 0.8906	28.68 / 0.8439	28.37 / 0.8718
SRCNN [17]		×4	31.10 / 0.9181	29.90 / 0.8796	29.69 / 0.9052	30.52 / 0.9078	29.31 / 0.8616	29.32 / 0.8960
VDSR [22]		×4	32.09 / 0.9311	30.57 / 0.8932	30.79 / 0.9240	31.69 / 0.9244	30.14 / 0.8818	30.51 / 0.9162
RDN [32]		×4	32.73 / 0.9387	31.05 / 0.9042	31.45 / 0.9324	32.64 / 0.9362	31.00 / 0.9018	31.49 / 0.9301
CMSCN [37]		×4	32.53 / 0.9374	30.83 / 0.8997	31.32 / 0.9312	32.23 / 0.9321	30.55 / 0.8920	31.28 / 0.9278
FSCWRN [28]		×4	32.91 / 0.9415	30.96 / 0.9022	31.71 / 0.9359	32.78 / 0.9387	30.79 / 0.8973	31.71 / 0.9334
CSN [Ours]		×4	33.40 / 0.9486	31.23 / 0.9093	32.05 / 0.9413	33.51 / 0.9489	31.27 / 0.9092	32.28 / 0.9421
EDSR [31]	C96	×4	31.80 / 0.9284	30.46 / 0.8902	30.52 / 0.9198	31.44 / 0.9219	30.04 / 0.8803	30.31 / 0.9137
CSN [Ours]		×4	32.19 / 0.9325	30.53 / 0.8915	30.81 / 0.9231	31.72 / 0.9246	30.07 / 0.8794	30.54 / 0.9163

according to our CSN model, two of which use nearest neighbor (NN) and bilinear respectively to approximate the residual between the original LR input $\mathbf{x}$ and the corresponding HR target $\mathbf{y}$, and the other does not use ESC. They are termed as ESC-None, ESC-NN, ESC-Bilinear, and the one we use is termed as ESC-Bicubic. We train these models on $\mathcal{D}$(PD, BD) and the validation performance is plotted in Fig.7. It can be easily observed that ESC-Bicubic perform significantly better than other models. The corresponding results on $\mathcal{T}$(PD, BD) also illustrate this conclusion (Table III).

Another important observation is that the ESC contributes to stable model training, no matter which interpolation method is used. Therefore, the ESC can reduce the possibility of training failure, which is also beneficial in the case of the degradation of training examples.

3) The Number of Stage Mappings and Blocks: It can be seen from (14) that the network depth D is mainly determined by the number of CSBs m and the number of stage mappings n. We examine the impact of these two hyperparameters on the performance of the model. Firstly, we fix m to 4 and change n from 1 to 4. Fig.8(a) displays the evolution curves of PSNR performance on $\mathcal{V}$(T2, TD) for SR×2. It can be seen that the performance is improved gradually with the increased number of building blocks, but at the expense of increased parameters. Next, we fix n to 4 and change m from 1 to 4. The PSNR curves of the models on the same dataset are plotted in Fig.8(b). We observe a similar trend of the curves as m changes. The result is unsurprising because increasing m or n increases the network depth and model parameters.

Finally, we show the final SR performance of all compared models on the corresponding testing dataset $\mathcal{T}$(T2, TD) in Fig.8(c), versus the number of parameters. It is worth noting that the models with t building blocks and 4 stage mappings perform better than the models with 4 building blocks and t stage mappings ($t = 1, 2, 3$), although the former has fewer model parameters. In next experiments, we choose $n = m = 4$ for our CSN model. Therefore, the network depth is 43 for SR×2 and SR×3, and 44 for SR×4.

D. Comparison with Other Methods

To further illustrate the effectiveness of the proposed CSN model, we compare it with several advanced SISR methods quantitatively and qualitatively, including NLM [57], SRCNN [17], VDSR [22], RDN [32], CMSCN [37], FSCWRN [28] and EDSR [31]. These models are retrained on the generated datasets with all image types and scaling factors. As mentioned earlier, the degradation of training samples may lead to training failure of some models, especially for those with extremely deep structure and large number of parameters, e.g., EDSR [31]. To solve the problem, we train the EDSR by taking 96 slices of a 3D volume as 96 channels of a 2D sample. This can effectively avoid the training failure problem but at the cost of accuracy reduction. We attach C1 and C96 to the model name to mark these two cases. For fair comparison, we train both CSN (C1) and CSN (C96).

1) Bicubic Degradation (BD): Bicubic downsampling is a widely used simulation of LR image generation in image SR

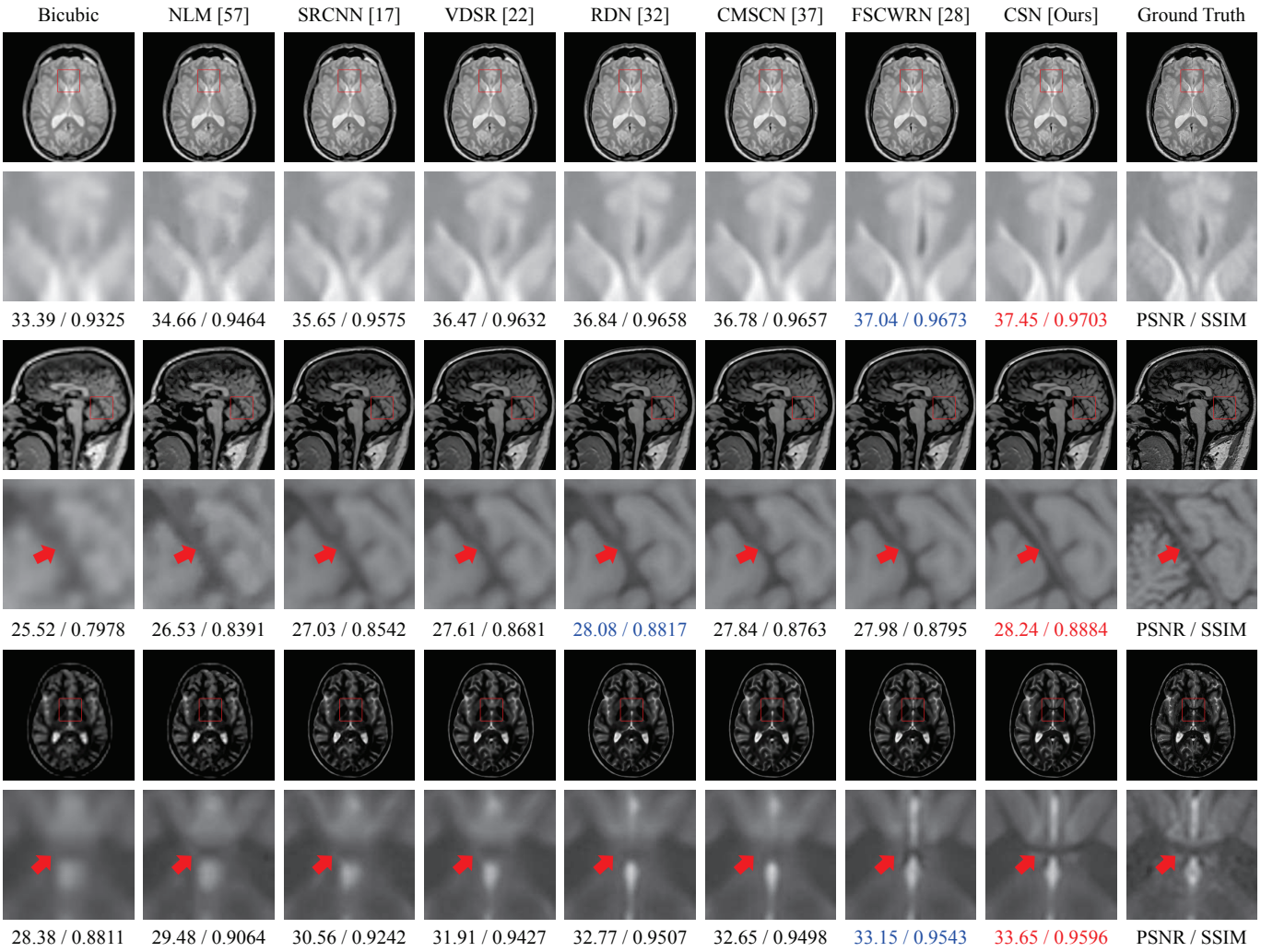


Fig. 9. The visual effect of the compared methods on a PD (top), T1 (middle) and T2 (bottom) image with SR×3, SR×4 and SR×4, respectively. The image degradation is **bicubic downsampling** (C1). The maximal PSNR (dB) and SSIM values for each displayed image are in red, and the second ones are in blue.

settings which simply shrinks HR images to a smaller size with the bicubic kernel. We examine this image degradation in this section first. Columns 4 to 6 of Table IV show that the quantitative results of the compared methods over the testing datasets under this image degradation. Overall, all the deep learning based methods (SRCNN [17], VDSR [22], RDN [32], CMSCN [37], FSCWRN [28], EDSR [31] and our CSN) present great advantages over the traditional methods (Bicubic and NLM [57]). However, the proposed CSN model gives the best SR performance in both C1 and C96 cases although it has fewer parameters and shallower model structures than RDN [32] and EDSR [31].

Fig.9 displays the visual comparison of these methods under the bicubic degradation. The top row shows the result on a PD image with scaling factor SR×3. It can be observed that our CSN model has successfully restored the black area and it has the most similar shape to the ground truth. However, several other methods, such as Bicubic, NLM [57], SRCNN [17] and even VDSR [22] almost lost the black area. The middle row is the result on a T1 image with SR×4. There is a gray ridge at the position marked by the red arrow, which can hardly be recognized in the results of other methods, but our model gives a more credible indication of the ridge. Similar results can also be observed from the bottom row, which shows the comparison on a T2 image with SR×4. There is a dark ditch at the position marked by the red arrow, and only our CSN model has succeeded in restoring this information.

2) Truncation Degradation (TD): k-space truncation of HR images is a process that simulates the real image acquisition process where a LR image is scanned by reducing acquisition lines in both phase and slice encoding directions. The missing information is therefore in k-space and the degradation pattern of LR images is different from simply shrinking the size of HR images in the image domain by, e.g., bicubic interpolation [25]. Table IV also shows the quantitative comparison between different methods under the truncation degradation. Again, the proposed CSN model presents the best SR performance in both C1 and C96 cases. It is worth noting that the performance of Bicubic, NLM [57], SRCNN [17] and VDSR [22] is slightly worse than that of these methods in case of BD, e.g., SR×2 on T2 images. On the contrary, the performance of other methods in case of TD is better than that in case of BD. This is probably

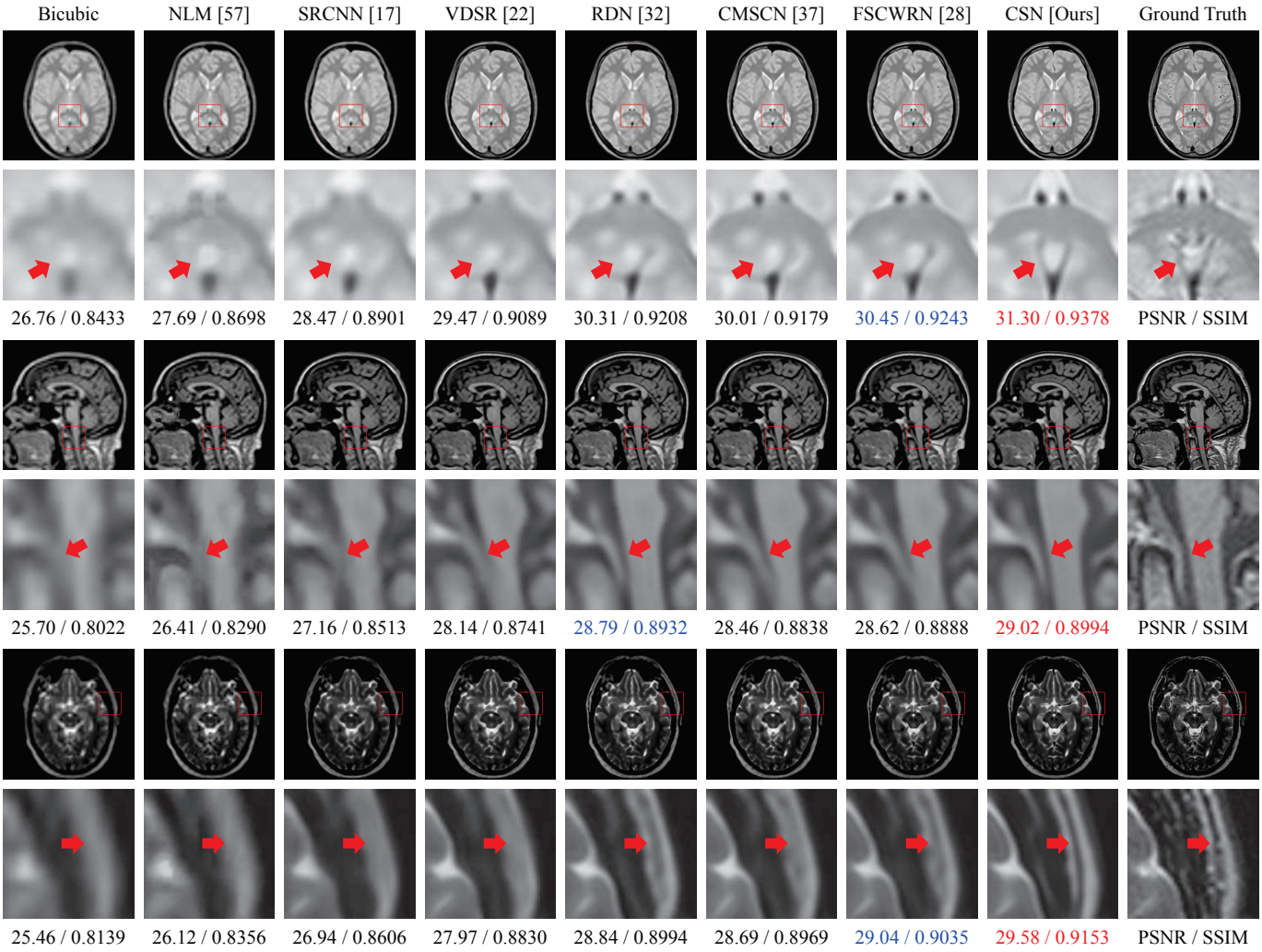
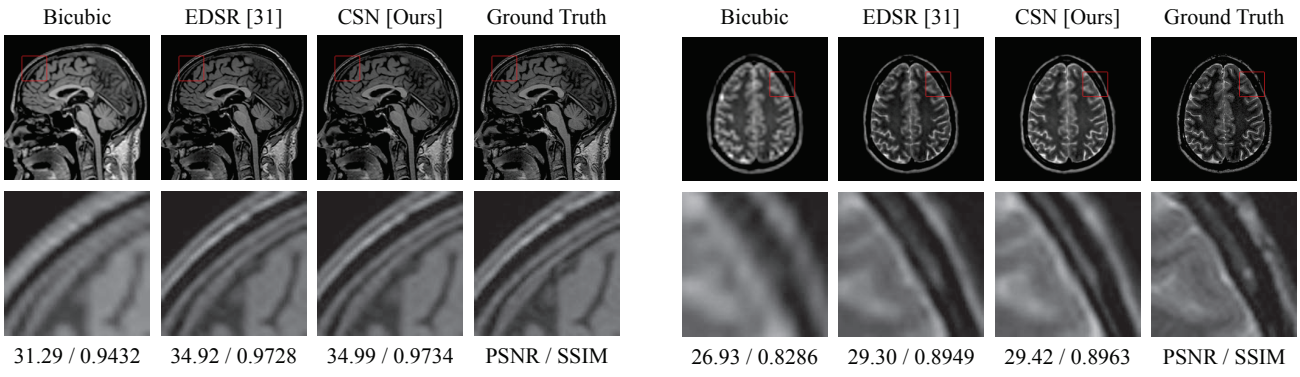


Fig. 10. The visual effect of the compared methods on a PD (top), T1 (middle) and T2 (bottom) image with scaling factor SR×4. The image degradation is ***k*-space truncation** (C1). The maximal PSNR (dB) and SSIM values for each displayed image are in red, and the second ones are in blue.



(a) bicubic downsampling (BD) (b) *k*-space truncation (TD)

Fig. 11. The visual comparison of EDSR [31] and the proposed CSN model in case of C96. (a) a T1 image with scaling factor SR×2. (b) a T2 image with scaling factor SR×4. In this case, each testing example is a 3D volume. For both (a) and (b), a randomly selected slice is used for display purposes.

because TD degrades image quality more seriously than BD and models such as RDN [32], CMSCN [37], FSCWRN [28] and our CSN have better representational capacity than models such as SRCNN [17] and VDSR [22].

Fig.10 presents the visual effects of the compared methods in case of TD and the proposed CSN model presents obvious advantages over other models. For instance, the bottom row is the results on a T2 image with SR×4. Our CSN model is able

to reconstruct the dark contour at the position indicated by the red arrow, which cannot be clearly observed in the results of other models. The top and middle rows show the results on a PD and a T1 image, also highlighting the advantages of the proposed CSN model. Fig.11 shows the visual comparison between the EDSR [31] and our CSN (C96) model in case of C96. Both BD and TD are presented. In this case, our model has less obvious advantages over EDSR [31], but still performs better on the whole.

V. Discussion and Future Work

A. Multiple Branches

Like the original MAR mapping in [36], the stage mapping in our CSB can also be easily extended to multiple branches ($\geqslant$3). The difference is that we branch the network by channel splitting, instead of feature reuse. In extreme cases, it can be extended to c branches with each branch occupying one channel of the input feature. This means that we explicitly differentiate the hierarchical features rather than having the network learn to distinguish between different features. Therefore, when the training samples are degraded and the model is complex, it helps to ease model training.

B. Depth and Width

Branching the network by reusing the entire feature tensor makes the model much wider, like [36], [37]. This significantly increases model parameters when the network depth is the same. Since EDSR [31] is a typical network with very wide structure and causes training failure, while RDN [32] is a deeper but less wide network and can be successfully trained. Therefore, we speculate that the width of the model may also be one of the reasons for training failure in case of training sample degradation. Our work can be regarded as a manner to going deeper with nearly unchanged model width and parameters.

C. Branch Structure

Currently, we only utilize the structures similar to ResNet [34], [33] and DenseNet [35] (or RDN [32]) for different branches. The experimental results show that mixing different branch structures is helpful to improve the performance of the model, but it is not conspicuous. This is probably because the structural difference between the two branches is relatively small. We conjecture that as the structural difference of the branches increases, so does the performance difference. The further investigation will be a part of our future work.

D. 3D Extension

The present work only aims at the task of 2D MR image super-resolution, and the further extension could be in 3D case. However, since many types of medical images are in 3D format, it is intuitively possible to further enhance SR performance if the 3D structural information can be reasonably utilized [3], [24], [25], [27]. A prominent problem in the 3D settings is that the number of parameters will increase dramatically as the network depth increases, leading model training more difficult. Our model can deepen the network without significantly increasing the model width and parameters, which also helps to extend 3D models.

E. Information Sharing

In this paper, we only deal with the SR task for a single type of 2D MR images and a single scaling factor. However, there is evidence that combining the information from different image types and scaling factors is helpful to improve the performance of deep models [22], [31]. The medical images SR framework combined multi-type and multi-scale information is also expected to further improve the SR performance of deep models.

VI. Conclusion

A major problem with using deep models to super-resolve MR images is the lack of *high-quality* and *effective* training samples, which probably leads to performance degradation or even training failure of deep models. In this work, we have presented a novel deep channel splitting network (CSN) for the task of 2D MR image super-resolution, which is primarily made up of a series of cascaded channel splitting blocks (CSBs). The hierarchical features are split into two branches with different information propagations (residual branch and dense branch), which helps the model to discriminate different features explicitly. To integrate branch information, the MAR [36] mapping is also applied to merge the hierarchical features on different branches.

Channel splitting helps to increase the depth of the network and the diversity of processing the hierarchical features. We conjecture that the performance improvement of the proposed model benefits from both two parts and additional performance can be further gained by exploring other branch structures and information fusion strategies. As it improves the dilemma between improving model performance and easing model training to some extent, it also has the potential to deal with other types of medical images, such as CT, ultrasound and PET etc.

References

[1] E. Carmi, S. Liu, N. Alona, A. Fiat, D. Fiat, "Resolution enhancement in MRI," *Magn. Reson. Imag.*, vol. 24, no. 2, pp. 133-154, Feb. 2006.

[2] E. Plenge, D. H. J. Poot, M. Bernsen, *et al*, "Super-resolution methods in MRI: can they improve the trade-off between resolution, signal-to-noise ratio, and acquisition time?," *Magn. Reson. Med.*, vol. 68, no. 6, pp. 1983-1993, Feb. 2012.

[3] J. Hu, X. Wu, J. Zhou, "Single image super resolution of 3D MRI using local regression and intermodality priors," in *Proc. Int. Conf. Digit. Image Process.*, vol.10033, 2016. pp.100334C.

[4] W. Shi, J. Caballero, C. Ledig, X. Zhuang, *et al*., "Cardiac image super-resolution with global correspondence using multi-atlas patchmatch," in *Med. Image. Comput. Comput. Assist. Interv.*, Sep. 2013, pp. 9-16.

[5] A. Rangnekar, N. Mokashi, E. Ientilucci, C. Kanan, M. Hoffman. (2017). "Aerial spectral super-resolution using conditional adversarial networks." [Online]. Available: http://cn.arxiv.org/abs/1712.08690

[6] M. W. Thornton, P. M. Atkinson, D. A. Holland, "Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping," *Int. J. Remote Sens.*, vol. 27, no. 3, pp. 473-491, Feb. 2006.

[7] Z. Pan, J. Yu, H. Huang, S. Hu, A. Zhang, H. Ma, *et al*., "Super-resolution based on compressive sensing and structural self-similarity for remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4864-4876, Jan. 2013.

[8] T. Ahmad and X. M. Li, "An integrated interpolation-based super resolution reconstruction algorithm for video surveillance," *J. Commun.*, vol. 7, no. 6, pp. 464-472, Jun. 2012.

[9] D. Gottlieb and C. W. Shu, "On the gibbs phenomenon and its resolution," *Siam Review*, vol. 39, no. 4, pp. 644-668, Dec. 1997.

[10] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graph. Model. Image Process.*, vol. 53, no. 3, pp. 231-239, May. 1991.

[11] H. Stark and P. Oskoui, "High resolution image recovery from image plane arrays using convex projections," *J. Opt. Soc. Am.*, vol. 6, no. 11, pp. 1715C1726, 1989.

[12] W. T. Freeman, T. R. Jones, E.C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56-65, Aug. 2002.

[13] C. Kim, K. Choi, J. B. Ra, "Example-based super-resolution via structure analysis of patches," *IEEE Signal. Process. Lett.*, vol. 20, no. 4, pp. 407-410, Mar. 2013.

[14] J. Yang, J. Wright, T. Huang, Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp.1-8

[15] J. Yang, Z. Wang, Z. Lin, *et al*., "Coupled Dictionary Training for Image Super-Resolution," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3467-3478, Apr. 2012.

[16] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, May. 2015.

[17] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Analy. Mach. Intell.*, vol. 38, no. 2, pp. 295-307, Feb. 2015.

[18] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp.391-407.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[20] J. Kim, J. K. Lee, K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp.1637-1645.

[21] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2790-2798.

[22] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646C1654.

[23] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: a persistent memory network for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2017, pp. 4549-4557.

[24] C. H. Pham, A. Ducournau, R. Fablet, and F. Rousseau, "Brain MRI super-resolution using deep 3D convolutional networks," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 197-200.

[25] Y. Chen, Y. Xie, Z. Zhou, F. Shi, A. G. Christodoulou, and D. Li. (2018). "Brain MRI super resolution using 3D deep densely connected neural networks." [Online] Available: http://cn.arxiv.org/abs/1801.02728

[26] C. Zhao, A. Carass, B. E. Dewey, and J. L. Prince. (2018). "Self super-resolution for magnetic resonance images using deep networks." [Online]. Available: https://arxiv.org/abs/1802.09431

[27] Y. Chen *et al*. (2018). "Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network." [Online]. Available: https://arxiv.org/abs/1803.01417

[28] J. Shi, Z. Li, S. Ying, C. Wang, Q. Zhang, P. Yan. MR image super-resolution via wide residual networks with fixed skip connection. IEEE Journal of Biomedical and Health Informatics. 2019, Preprint.

[29] J. Shi, Q. Liu, C. Wang, Q. Zhang, S. Ying, H. Xu, "Super-resolution reconstruction of MR image with a novel residual learning network Algorithm," Physics in Medicine and Biology. 2018, vol. 63, no. 8, pp. 085011.

[30] G. Litjens *et al*., "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, no. 9, pp. 60-88, Dec. 2017.

[31] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops. (CVPRW)*, Jul. 2017, pp.1132-1140.

[32] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. (2018). "Residual dense network for image super-resolution." [Online]. Available: https://arxiv.org/abs/1802.08797

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp.770-778.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp.630-645.

[35] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. (2016). "Densely connected convolutional networks." [Online]. Available: https://arxiv.org/abs/1608.06993

[36] L. Zhao, J. Wang, X. Li, Z. Tu, and W. Zeng. (2017). "Deep convolutional neural networks with merge-and-run mappings." [Online]. Available: https://arxiv.org/abs/1611.07718

[37] Y. Hu, X. Gao, J. Li, Y. Huang, and H. Wang. (2018). "Single image super-resolution via cascaded multi-scale cross network." [Online]. Available: https://arxiv.org/abs/1802.08808

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. (2016). "Inception-v4, inception-resnet and the impact of residual connections on learning." [Online]. Available: https://arxiv.org/abs/1602.07261

[39] Y. Huang and Y. Long, "Super-resolution using neural networks based on the optimal recovery theory," *Journal of Computational Electronics*, vol. 5, no. 4, pp. 275-281, Jan. 2006.

[40] W. Shi *et al*. (2016). "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." [Online]. Available: http://cn.arxiv.org/abs/1609.05158v2

[41] S. Peled and Y. Yeshurun, "Superresolution in MRI: application to human white matter fiber tract visualization by diffusion tensor imaging," *Mag. Reson. Med.*, vol. 45, no. 1, pp. 29-35, Jan. 2001.

[42] H. Greenspan, G. Oz, N. Kiryati, and S. Peled, "MRI inter-slice reconstruction using super-resolution," *Magn. Reson. Imag.*, vol. 20, no. 5, pp. 437-446, Jun. 2002.

[43] R. Z. Shilling *et al.*, "A super-resolution framework for 3-D high-resolution and high-contrast imaging using 2-D multislice MRI," *IEEE Trans. Med. Imag.*, vol. 28, no. 5, pp. 633-644, May. 2009.

[44] F. Rousseau, "Brain hallucination," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2008, pp. 497C508.

[45] J. V. Manjón, P. Coupé, A. Buades, D. L. Collins, and M. Robles, "MRI super-resolution using self-similarity and image priors," *Int. J. Biomed. Imaging.*, vol. 2010, Article ID: 425891 (11 pages), Dec. 2010.

[46] A. Rueda, N. Malpica, E. Romero, "Single-image super-resolution of brain MR images using over complete dictionaries," *Med. Image Anal.*, vol. 17, no. 1, pp. 113-132, Jan. 2013.

[47] Y. H. Wang, J. Qiao, J. B. Li, P. Fu, S. C. Chu, and J. F. Roddick, "Sparse representation-based MRI super-resolution reconstruction," *Measurement*, vol. 47, no. 1, pp. 946-953, Jan. 2014.

[48] S. Roohi, J. Zamani, M. Noorhosseini, and M. Rahmati, "Super-resolution MRI images using Compressive Sensing," in *Iranian Conference on Electrical Engineering (ICEE2012)*, May. 2012 pp. 1618-1622.

[49] O. Oktay, *et al*., "Multi-input cardiac image super-resolution using convolutional neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent*, Oct. 2016, pp. 246-254.

[50] Y. Wang, L. Wang, H. Wang, P. Li. (2016). "End-to-end image super-resolution via deep and shallow convolutional networks." [Online]. Available: https://arxiv.org/abs/1607.07680

[51] H. Ren, M. Elkhamy, J. Lee, "Image super resolution based on fusing multiple convolution neural networks," in *Comput. Vis. Pattern Recognit. Workshops.*, Jul. 2017, pp. 1050-1057.

[52] J. Yamanaka, S. Kuwashima, T. Kurita. (2017). "Fast and accurate image super resolution by deep CNN with skip connection and network in network." [Online]. Available: https://arxiv.org/abs/1707.05425

[53] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. (2017). "Dual path networks." [Online]. Available: https://arxiv.org/abs/1707.01629

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Proces.*, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[55] D. P. Kingma and J. L. Ba. (2014). "Adam: a method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980v9

[56] X. Glorot., Y. Bengio.: Understanding the difficulty of training deep feedforward neural networks. in *Proc. AISTATS10*, May 2010, vol. 9, pp. 249C256.

[57] J. V. Manjon, P. A. Coupe, V. Fonov, C. D. Louis, and M. Robles, "Non-local MRI upsampling," *Med. Image Anal.*, vol. 14, no. 6, pp. 784-792, Dec. 2010.

[58] N. Ahn, B. Kang, K. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *ECCV 2018*, Sep. 2018, pp. 256-272.