# 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 facilitates accurate subsequent analysis and early diagnosis. However, it is limited by several factors such as

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 is with the High Field Magnetic Resonance Brain Imaging Laboratory of Sichuan and Key Laboratory for Neuro Information of Ministry of Education, Chengdu, Sichuan 611731, China; He is 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).

X. Zou is with the School of Life Science and Technology, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 611731, China (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).

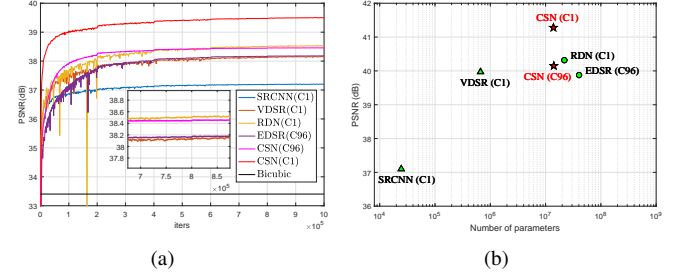Manuscript received October XX, 2018; revised November XX, 2018.



Fig. 1. The performance comparison between several typical 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 testing 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, 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.

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 SNR and/or increases imaging time [2].

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-$\sigma$ [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 learning based methods can learn insufficient information due to the limited representational ability of these shallow models.

In recent years, various advanced SR methods have emerged with the rapid development of deep learning techniques [16]
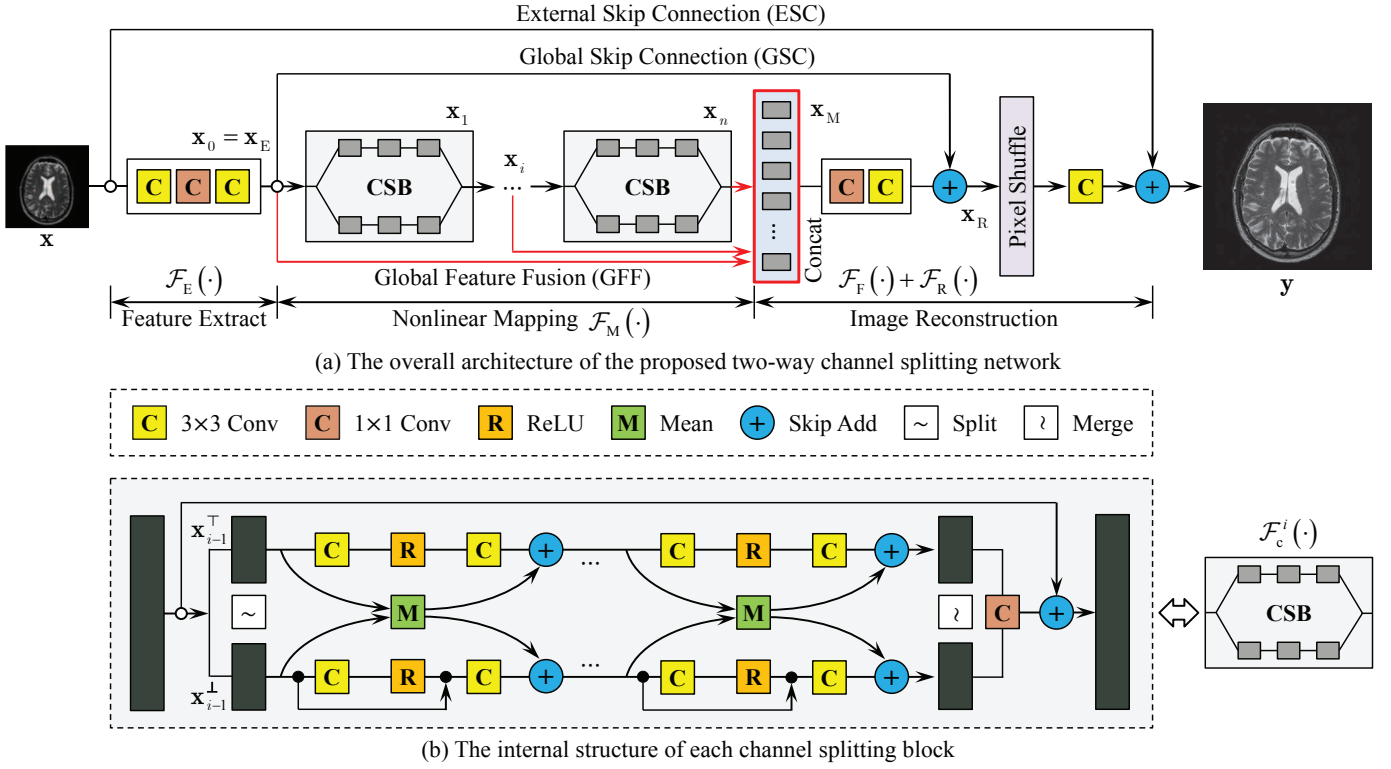
Fig. 2. The architecture of the proposed CSN model and the channel splitting block CSB. (a) The overall structure of the proposed CSN model, which consists of three typical parts: feature extraction, nonlinear mapping and image reconstruction. Notice that GFF [30] is applied to fully utilize hierarchical features, while ESC and GSC are mainly applied to stabilize model training. (b) The channels of the intermediate feature maps within a CSB are split into two branches. One is built as a residual structure (residual branch), and the other is built as dense network (dense branch).

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]. 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* [28]. 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 SR (SISR) models and the proposed channel splitting network (CSN) on the proton density (PD) volumes of IXI

dataset (http://brain-development.org/ixi-dataset/) for SR$\times$2, where enhanced deep super-resolution network (EDSR) [29] and residual dense network (RDN) [30] are advanced models on NI. But it is failed to train the EDSR model (the same configuration as [29]) with 48000 2D PD images. This problem of training failure caused by the degradation of sample quality will get worse as the depth (or width) of the network 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 [31], [32]. 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 [29] 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 [29] 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) [31], [32] and dense network (DenseNet) [33] (or RDN [30]) on each branch. Besides, the merge-and-run (MAR) mapping [34], [35] 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*.

In order to make full use of the hierarchical features of the model, we adopt the global feature fusion (GFF) technology [30] (Fig.2(a)). Moreover, multilevel residual mechanism and constant scaling technique [29], [36] 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 are generated from the IXI dataset, which include three types of MR images (PD, T1 and T2) and two image degenerations (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 [37], the pioneering work with deep learning techniques in the modern sense is SRCNN [17]. Subsequently, some advanced methods based on increasing network depth and parameter sharing are proposed. Kim *et al.* [22] increased the network depth by stacking more convolutional layers with global residual learning (GRL) and firstly introduced recursive learning in a deep network for parameter sharing [20]. Another work which 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 to capture sufficient knowledge for reconstructing the high frequency details, Hu *et al.* [35] 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.

There is a common feature among the above methods: they use the interpolated image of the original LR image as the input of their models. This preprocessing is convenient for keeping the size of the output image consistent with the target HR image. However, it places the inference of the network in the HR image space, resulting in great computation and memory consumption. There exist two solutions for this issue currently: deconvolution or transpose convolution [18] and efficient sub-pixel convolutional neural network (ESPCNN) [38]. 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 the models and thus boost the SR performance greatly, e.g., EDSR/MDSR [29]. Further, Zhang *et al.* [30] combined the idea of residual learning [31], [32] with densely connected DenseNet [33] and proposed a novel residual dense network (RDN) to fully utilize the hierarchical features of deep models.

### B. MR Image Super Resolution

The early application of SR methods to medical images mainly focuses on the task of multi-frame image super resolution (MFSR). For example, IBP [10] was adopted to generate a new MR image with increased spatial resolution from several spatially shifted, single-shot and diffusion-weighted brain MR images [39]. Greenspan *et al.* [40] and Shilling *et al.* [41] employed IBP and POCS [11] to produce a 3D MR volume with isotropic resolution from several 2D slices, respectively. These methods are usually accompanied by specific acquisition strategies (e.g., rotation, scaling and translation) to simulate the LR image generation. 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 [42] 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.* [43]. 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 [44], [45] and compressive sensing [46], are also applied to medical images subsequently. Recently, more advanced methods based on deep learning [16] have also been applied to MR image SR
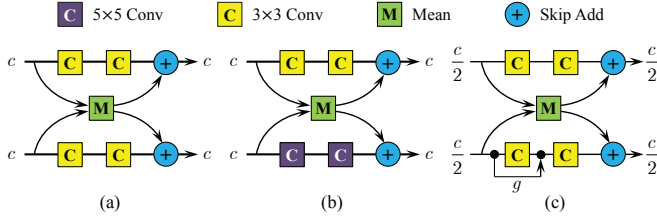
Fig. 3. The merge-and-run stage mappings in a building block. (a) Original mapping [34]. (b) Multi-scale cross (MSC) mapping [35]. (c) The proposed mapping. $c$ is the channel number of feature maps, and $g$ is the growth of a dense connection [33]. BN and ReLU are omitted for simplification.

tasks [24], [25], [26], [27], [47]. These methods, however, 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 by the image SR community to boost the SR performance by assembling the information from different streams (paths). Wang *et al.* [48] explored an end-to-end CNN architecture by jointly training both deep and shallow CNNs, where the shallow network stabilizes model training and the deep one ensures accurate HR reconstruction. Ren *et al.* [49] proposed a context-wise network fusion approach to integrate the outputs of individual networks by extra convolutional layers. Yamanaka *et al.* [50] combined skip connection layers and parallelized CNNs into a single CNN architecture. CMSCN [35] is another multi-stream structure, in which complementary information under different receptive fields is integrated by the merge-and-run mechanism [34] (Fig.3). There are also multi-stream structures for medical image SR tasks, e.g., Oktay *et al.* [47] 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 in 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 architecture of the proposed CSN model is outlined in Fig.2. Like other typical deep SR models, it consists mainly of three parts: feature extraction, nonlinear mapping and image reconstruction. The feature extraction sub network (FEN) is first applied to express the input image $\mathbf{x}$ as a set of shallow features. These shallow features are then transmitted to the nonlinear mapping sub network (NMN), which is composed of a series of stacked channel splitting blocks (CSB). Subsequently, the hierarchical features outputted by each CSB are concatenated together to produce the final output of the

NMN. This operation is also called global feature fusion (GFF) [30]. Finally, the collected hierarchical features are fed into the image reconstruction sub network (IRN) to generate the final HR predication $\mathbf{y}$ of the entire model.

*1) Feature Extraction:* The FEN contains two $3\times3$ conv layers with a $1\times1$ conv layer in the middle. Denote $\mathcal{F}_E(\cdot)$ as the corresponding function of FEN, then the shallow features $\mathbf{x}_E$ extracted by FEN can be represented as:

$$\mathbf{x}_E = \mathcal{F}_E(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is the original LR input. The $1\times1$ conv layer is the point-to-point linear transformation of the feature maps of the first $3\times3$ conv layer, which is used to enhance the robustness of the extracted features.

*2) Nonlinear Mapping:* The NMN is denoted as $\mathcal{F}_M(\cdot)$, i.e., the output of the NMN is given by $\mathcal{F}_M(\mathbf{x}_E)$. Supposing we have $n$ channel splitting blocks in the entire network, the output $\mathbf{x}_i$ of the $i$-th CSB can be obtained by:

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

where the function $\mathcal{F}_c^i(\cdot)$ corresponds to the operations of the $i$-th CSB. More details about $\mathcal{F}_c^i(\cdot)$ will be presented in section III-B. $\mathbf{x}_0 = \mathbf{x}_E$ is the input of the first CSB (or the whole nonlinear mapping sub network). Therefore, the output of the last CSB can be iteratively formulated as follow:

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

The output of the $i$-th CSB $\mathbf{x}_i$ is produced by a series of operations (e.g., convolutional layer, ReLU and constant scaling etc.) within the block, so it is viewed as a set of local features [30]. These local features constitute the final output of our nonlinear mapping sub 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 [30] and contributes to the information propagation in the network [32].

*3) Image Reconstruction:* This phase includes two related parts: global fusion of the local features $\mathbf{x}_i$ and 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):

$$\mathbf{x}_M = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n], \tag{4}$$

where $[\ldots]$ implies the concatenation. Next, the global features are extracted by fusing all local features from all the preceding channel split blocks. This is completed by a composite function $\mathcal{F}_F(\cdot)$ corresponding to a $1\times1$ convolutional layer followed by a $3\times3$ convolutional layer. Finally, the global residual learning (GRL) is utilized to stabilize the training process [29], [30], which is simply implemented via a global skip connection (GSC):

$$\mathbf{x}_R = \mathcal{F}_F(\mathbf{x}_M) + \mathbf{x}_0 = \mathcal{F}_F([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n]) + \mathbf{x}_0, \tag{5}$$

where $\mathbf{x}_R$ 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 $3\times3$ convolutional
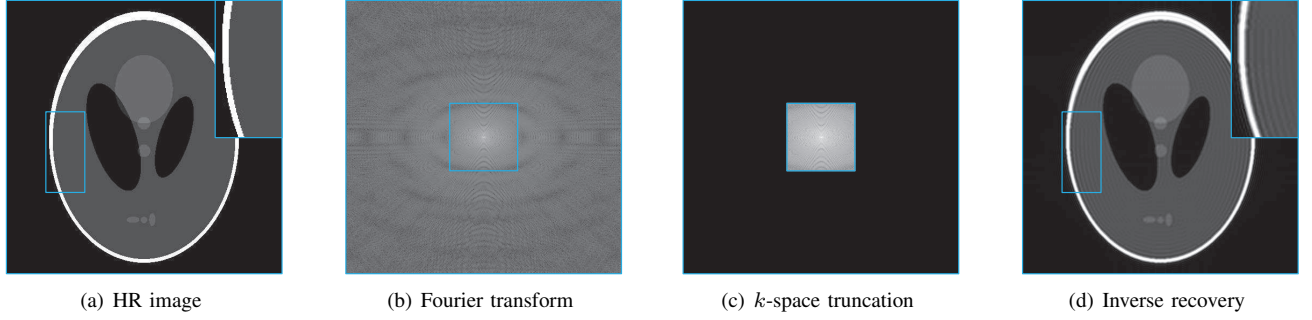
Fig. 4. The image degradation of $k$-space truncation for SR×4. Different from bicubic downsampling, the generated LR images are sometimes accompanied by Gibbs-ringing artifacts. The result in (d) is generated by zero-padding in $k$-space and inverse Fourier transform (IFT) for display purpose.

layer, and an external residual learning (ERL). Formally, it can be represented as:

$$\mathbf{y} = \mathcal{F}_R(\mathbf{x}_R) + \mathbf{x}, \tag{6}$$

where $\mathcal{F}_R(\cdot)$ is the function corresponding to the pixel shuffle layer and the following $3 \times 3$ convolutional layer, and $\mathbf{x}$ is the original LR input to the model. Note that the pixel shuffle layer is conducted by ESPCNN [38] in the way of [29].

### B. Channel Splitting Block

The CMSCN model explored by [35] 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 [51]. 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 flow. It has been shown that LRL is helpful to stabilize the training process and improve the representation ability of the model, resulting better SR performance [30].

*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:

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

where $\mathcal{S}_c(\cdot)$ 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, $\mathcal{M}_c(\cdot)$, at the end of each CSB:

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

It should be noted that $\mathcal{M}_c(\cdot)$ is a bivariate function, while $[\ldots]$ in (4) and (5) is a multivariate operator. The channel

splitting and merging operations are expected to artificially 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 way 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 [34] 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 $\mathcal{H}_{i,j}^{\perp}(\cdot)$ and $\mathcal{H}_{i,j}^{\top}(\cdot)$ 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:

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

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

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

is a idempotent transformation matrix of the merge-and-run mapping [34], [35]. The idempotent property can promote the information flow across the different modules and encourage gradient back-propagation during model training, similar to identity mapping [32]. It is worth noting that the upper branch is a residual-like structure similar to EDSR [29] and the lower branch is a simplified dense-like structure similar to DenseNet [33] or RDB [30], 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, $\mathbf{x}_m^{\perp}$ and $\mathbf{x}_m^{\top}$, are merged together after $m$ stages of merge-and-run mappings. Next, a local residual learning (LRL) [30] is introduced in the CSB to further

improve the information flow. The output of this CSB module is given by:

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

where $\mathcal{L}(\cdot)$ corresponds to a $1 \times 1$ convolutional operation at the end of the CSB, as shown in Fig.2(b). Unlike [30], the local residual features are derived from our CSB module, instead of the densely connected block [33].

### 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 the size of the LR/HR images is different, 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:

$$\mathbf{y} = \mathcal{F}_R(\mathbf{x}_R) + \hat{\mathbf{x}}, \qquad (12)$$

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 show that this can further stabilize the training process, and even help to slightly improve the model performance. Because the degeneration of MR training samples makes the model training more unstable, it is especially helpful for the task of single MR image super resolution.

### 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:

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

where $\boldsymbol{\theta}$ indicates the set of model parameters, and $\mathcal{F}_{CSN}(\cdot)$ is the mapping function of the entire CSN model and $\mathbf{y}^{(i)}$ is the HR target corresponding to LR input $\mathbf{x}^{(i)}$. Despite that minimizing $l_2$ loss is generally preferred since it maximizes peak signal to noise ratio (PSNR), $l_1$ loss provides better convergence for model training [29]. This is especially helpful in case of the degradation of training samples.
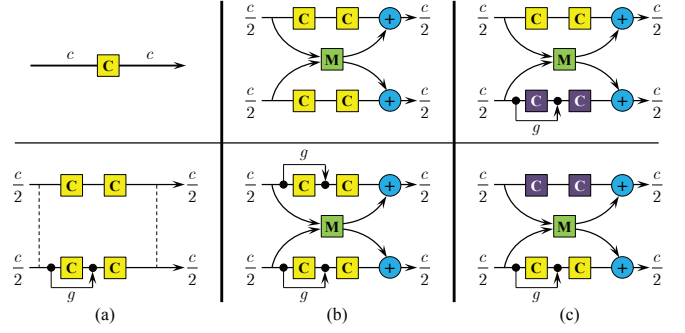


Fig. 5. Several structures for stage mapping. (a) The baseline structure and CSN-SP. (b) R3R3 and D3D3. (c) R3D5 and R5D3. Note Fig.3(a) is different from R3R3. For simplification, ReLU between the two convolutional layers is omitted, please refer to Fig.2(b) for the overall structure of a CSB.

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:

$$D = n(2m + 1) + s + 6, \qquad (14)$$

where $s$ represents the depth of the pixel shuffle layer. Note that $s$ depends on the scaling factor [29], i.e., $s = 1$ for SR$\times 2$ and SR$\times 3$, and $s = 2$ for SR$\times 4$.

## 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) [52] as the metrics of quantitative evaluation.

### A. Data Generation

The IXI dataset is used to construct our SR datasets, which 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 $240 \times 240 \times 96$ (height$\times$width$\times$depth) to fit 3 scaling factors ($\times 2$, $\times 3$ and $\times 4$). In this work, we only focus on the in-plane SR of 2D MR slices. Therefore, each 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 for quick validation ($\mathcal{V}$). For convenience, we use dataset (modality, degradation) to indicate a specific dataset, e.g., $\mathcal{D}(T2, BD)$ represents the T2 training dataset with bicubic degradation and $\mathcal{T}(PD, TD)$ represents the PD testing dataset with $k$-space truncation degradation.
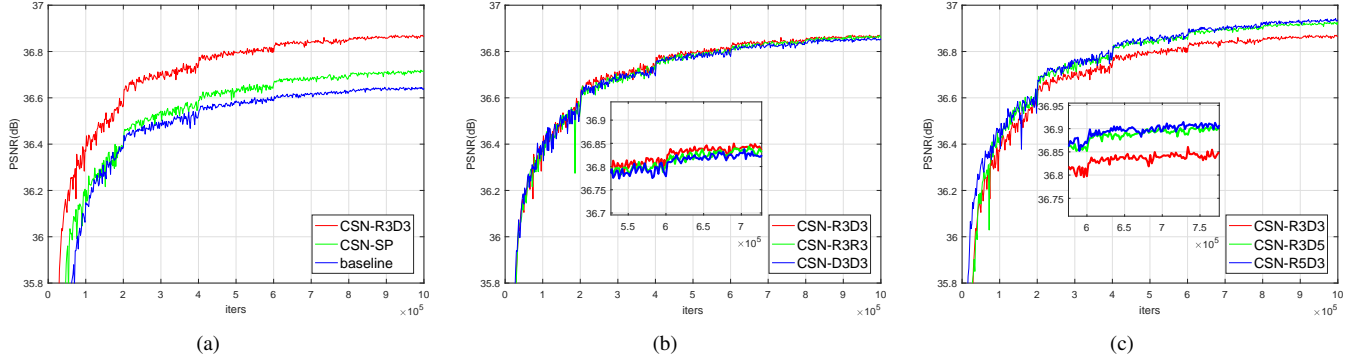
Fig. 6. The performance comparison between the structures shown in Fig.5 on $\mathcal{V}(T1, TD)$ for SR×2 (bicubic = 31.72dB). (a) Channel splitting and merge-and-run mapping. (b) Different branch structures. (c) Different kernel sizes. The baseline stage mapping is a single convolutional layer followed by a ReLU operation, which emphasizes the impact of channel splitting.

TABLE I
THE PERFORMANCE OF THE MODELS WITH DIFFERENT STAGE MAPPINGS ON $\mathcal{T}(T1, TD)$ (SR×2). THE MAXIMAL VALUES OF EACH COLUMN ARE IN **BOLD**, AND THE SECOND ONES ARE UNDERLINED.

| Network Configuration | Network Parameters | Network Depth | PSNR (dB) | SSIM |
|---|---|---|---|---|
| Bicubic | - | - | 33.38 | 0.9460 |
| baseline | 13862496 | 27 | 38.37 | 0.9803 |
| CSN-SP | 13865568 | 43 | 38.46 | 0.9807 |
| CSN-R3D3 | 13865568 | 43 | 38.62 | 0.9813 |
| CSN-R3R3 | 13866592 | 43 | 38.61 | 0.9813 |
| CSN-D3D3 | 13864544 | 43 | 38.59 | 0.9812 |
| CSN-R3D5 | **22254176** | 43 | 38.67 | 0.9815 |
| CSN-R5D3 | **22254176** | 43 | **38.68** | **0.9816** |

TABLE II
THE QUANTITATIVE RESULTS OF THE MODELS WITH DIFFERENT ESC APPROXIMATIONS ON $\mathcal{T}(PD, BD)$. THE MAXIMAL VALUES OF EACH ROW ARE IN **BOLD**, AND THE SECOND ONES ARE UNDERLINED.

| 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** |



Fig. 7. The impact of different ESC approximation 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.

model parameters and Adam optimizer [53] 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 $2 \times 10^5$ iterations (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 stage mapping from the following aspects:

* If without channel splitting, four convolutional layers in a stage mapping corresponds to a single convolutional layer with almost the same number of parameters (the top row in Fig.5(a)), which is the reference structure of a stage mapping. We take it as the baseline.

* To investigate the role of the MAR mapping, we removed it from the proposed CSB, resulting in the structure shown in the bottom row of Fig.5(a). We term it as CSN-SP, where S means splitting and P means plain.

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 (2D slices) in total. The generated dataset can be conveniently applied to develop 3D SR 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 as 64. If not specified, the kernel size follows the annotations of Fig.2.

We train the models by using image patches of size $24 \times 24$ randomly extracted from LR slices with the corresponding HR patches. Data augmentation is simply conducted by random horizontal flips and 90° rotations, as [29] and [30]. 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 use xavier initialization [54] for all

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

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 $\mathcal{V}(T2, TD)$ (SR$\times$2, Bicubic: 31.92dB). (c) The testing performance of all compared models on $\mathcal{T}(T2, TD)$ (SR$\times$2, Bicubic: 33.06dB).

* We also design two stage mappings shown in Fig.5(b) to check the effect of different branch structures, which are termed as R3R3 (top) and D3D3 (bottom). R and D represent residual branch and dense branch, and the number indicates the kernel size.
* Structures shown in Fig.5(c) are used to investigate the impact of different kernel size, which are termed R3D5 (top) and R5D3 (bottom) correspondingly.

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

The performance of the compared stage mapping structures on $\mathcal{V}(T1, TD)$ for SR$\times$2 are shown in Fig.6, where Fig.6(a) shows the impact of channel splitting and MAR mapping on the performance of the model. It can be seen that both of them can significantly improve the SR performance of the model. From Fig.6(b), we can observe the slightly better performance of R3D3 than that of R3R3 and D3D3. It is worth noting that *the model parameters R3R3>R3D3>D3D3 and the depth of the networks is the same*, which implies that mixing different branch structures is indeed helpful to boost the performance, although slightly. Fig.6(c) shows that R3D5 and R5D3 perform obviously better than R3D3. However, the number of model parameters of them is about 1.6 times of R3D3. Besides, despite R3D5 and R5D3 have the same number of parameters but R5D3 performs slightly better than R3D5. The conclusions are further verified by the results on the corresponding testing dataset $\mathcal{T}(T1, TD)$, as shown in Table I.

*2) External Skip Connection:* To verify the impact of the external skip connection (ESC), we build three other models according to our CSN model, two of which used 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 II).

Another important observation is that ESC helps stabilize model training, no matter which interpolation method is used. Thus, 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 Stages and Blocks:* From (14), it can be seen that the network depth $D$ is mainly determined by the number of CSB $m$ 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$\times$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 (computational cost and memory occupation). 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 $D$ and the number of 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 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 ($t = 1, 2, 3$), despite the former has fewer model parameters. We choose $n = m = 4$ for our CSN model, the depth of which thus is 43 (for SR$\times$2 and SR$\times$3) or 44 (SR$\times$4) according to (14).

### D. Comparison with Other Methods

To further illustrate the effectiveness of the proposed CSN model, several state-of-the-art SISR methods, including NLM [55], SRCNN [17], VDSR [22], RDN [30], and EDSR [29] are compared with the proposed CSN model quantitatively and qualitatively. The models are retrained on the generated datasets on all kinds of MR images and for all 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 [29]. To solve this problem, we train EDSR with the entire 3D volumes by taking the 96 slices as 96 channels. 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 indicate if a model is trained slice by slice

TABLE III
QUANTITATIVE COMPARISON BETWEEN DIFFERENT ALGORITHMS ON $\mathcal{T}$ (:, BD). THE MAXIMAL PSNR (dB) AND SSIM VALUES OF EACH ROW ARE IN BOLD, AND THE SECOND ONES ARE UNDERLINED.

|  | scale | Bicubic (2D) | NLM [55] (2D) | SRCNN [17] (C1) | VDSR [22] (C1) | RDN [30] (C1) | CSN (C1) | EDSR [29] (C96) | CSN (C96) |
|---|---|---|---|---|---|---|---|---|---|
| PD | SR×2 | 35.04/0.9664 | 37.26/0.9773 | 37.10/0.9774 | 39.97/0.9861 | <u>40.31/0.9870</u> | **41.28/0.9895** | 39.87/0.9857 | 40.15/0.9865 |
|  | SR×3 | 31.20/0.9230 | 32.81/0.9436 | 33.60/0.9516 | 34.66/0.9599 | <u>35.08/0.9628</u> | **35.87/0.9693** | 34.39/0.9578 | 34.68/0.9598 |
|  | SR×4 | 29.13/0.8799 | 30.27/0.9044 | 31.10/0.9181 | 32.09/0.9311 | <u>32.73/0.9387</u> | **33.40/0.9486** | 31.80/0.9284 | 32.19/0.9325 |
| T1 | SR×2 | 33.80/0.9525 | 35.80/0.9685 | 37.12/0.9761 | 37.67/0.9783 | <u>37.95/0.9795</u> | **38.27/0.9810** | 37.56/0.9774 | 37.60/0.9778 |
|  | SR×3 | 30.15/0.8900 | 31.74/0.9216 | 32.17/0.9276 | 32.91/0.9378 | <u>33.31/0.9430</u> | **33.53/0.9464** | 32.76/0.9347 | 32.83/0.9360 |
|  | SR×4 | 28.28/0.8312 | 29.31/0.8655 | 29.90/0.8796 | 30.57/0.8932 | <u>31.05/0.9042</u> | **31.23/0.9093** | 30.46/0.8902 | 30.53/0.8915 |
| T2 | SR×2 | 33.44/0.9589 | 35.58/0.9722 | 37.32/0.9796 | 38.65/0.9836 | <u>38.75/0.9838</u> | **39.71/0.9863** | 38.28/0.9824 | 38.53/0.9831 |
|  | SR×3 | 29.80/0.9093 | 31.28/0.9330 | 32.20/0.9440 | 33.47/0.9559 | <u>33.91/0.9591</u> | **34.64/0.9647** | 33.15/0.9528 | 33.36/0.9547 |
|  | SR×4 | 27.86/0.8611 | 28.85/0.8875 | 29.69/0.9052 | 30.79/0.9240 | <u>31.45/0.9324</u> | **32.05/0.9413** | 30.52/0.9198 | 30.81/0.9231 |



| Bicubic | NLM | SRCNN-C1 | VDSR-C1 | RDN-C1 | EDSR-C96 | CSN-C96 | CSN-C1 | Ground Truth |
|---|---|---|---|---|---|---|---|---|
| 28.61/0.8725 | 30.13/0.9095 | 30.44/0.9146 | 31.16/0.9266 | <u>31.60/0.9334</u> | 31.00/0.9228 | 31.07/0.9242 | **31.84/0.9377** | PSNR/SSIM |
| 29.98/0.8844 | 31.11/0.9091 | 31.93/0.9227 | 32.75/0.9341 | <u>33.31/0.9421</u> | 32.54/0.9336 | 32.96/0.9378 | **33.87/0.9511** | PSNR/SSIM |
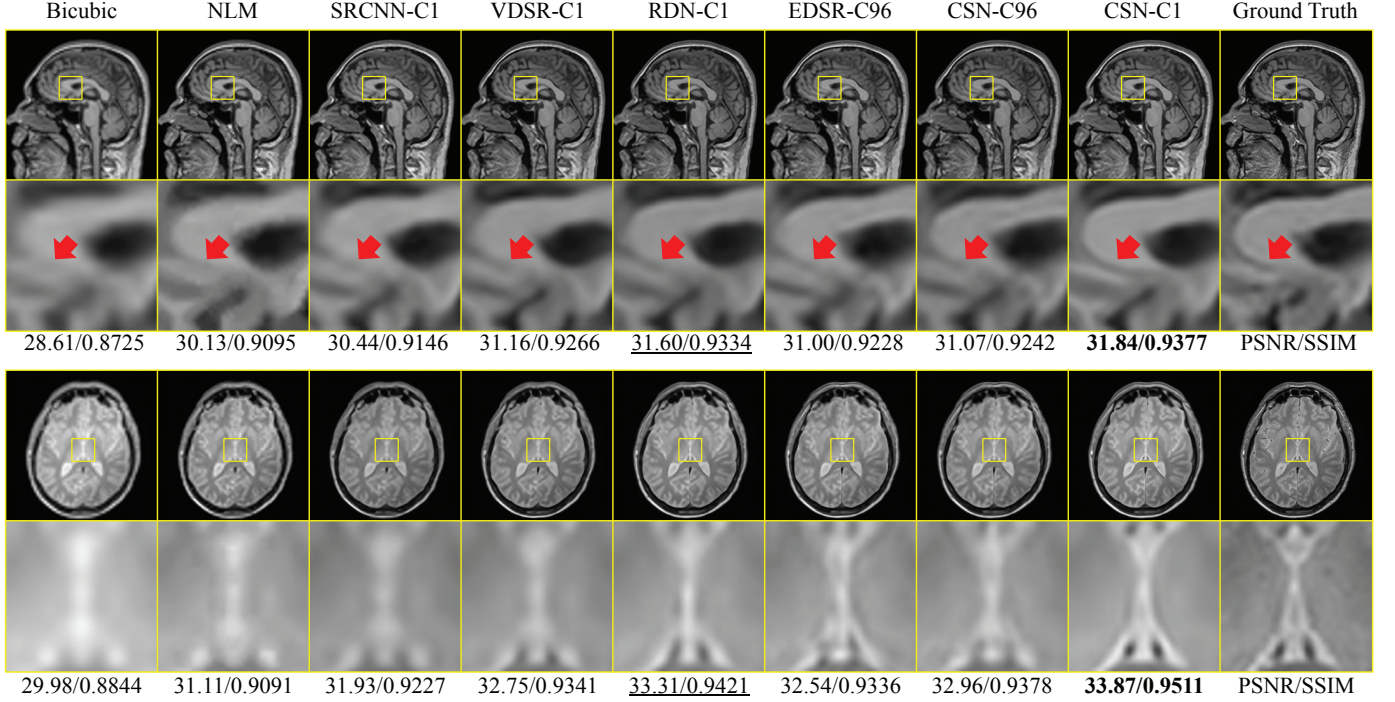
Fig. 9. The visual effect of the compared methods on a T1 image with scaling factor SR×3 (top) and a PD image with scaling factor SR×4 (bottom). The maximal PSNR (dB) and SSIM values for each displayed image are in bold, and the second ones are underlined.

(SRCNN(C1)) or it treats 96 slices as 96 channels (e.g., EDSR (C96)). For fair comparison, we train both CSN (C96) and CSN (C1) to match the cases.

*1) Bicubic Degradation (BD):* Bicubic degradation is a widely used simulation of LR image generation in image SR settings which simply shrinks HR images to a smaller size with the bicubic kernel. We examine this degradation in this section first. Table III shows the quantitative results of the compared methods over the testing datasets of PD, T1 and T2 MR images for SR×2, SR×3 and SR×4. Overall, all the deep learning based methods (SRCNN [17], VDSR [22], RDN [30], EDSR [29] and our CSN) present great advantages over the traditional methods (Bicubic and NLM [55]). However, the proposed CSN model exhibits the best SR performance on all datasets and SR scales in both C1 and C96 scenarios although it has fewer parameters and shallower model structures than RDN [30] and EDSR [29].

Fig.9 presents the visual effects of the compared methods under the bicubic degradation. The top row displays the results

on a T1 image with scaling factor SR×3. It can be observed that the edges of the dark groove (pointed by the red arrows) reconstructed by the proposed CSN model are sharper that of other methods, while the traditional methods (Bicubic and NLM [55]) almost lost this detail information. The bottom row gives the results on another PD image with scaling factor SR×4. We can see that Bicubic, NLM [55], even SRCNN [17] and VDSR [22] cannot reconstruct a clear and reliable result. Despite RDN [30] gives a significantly better result than other methods, the edges in the restored image are still relatively blurry. Instead, our CSN model exhibits a clear and sharp image that successfully recovers the block holes at both top and bottom ends of the image patch, making it more faithful to the ground truth.

*2) Truncation Degradation (TD):* The $k$-space truncation of the HR image is a process that simulates the real image acquisition process where a LR image is scanned by reducing acquisition lines in phase and slice encoding directions. The missing data, therefore, is in $k$-space and the degradation

TABLE IV
QUANTITATIVE COMPARISON BETWEEN DIFFERENT ALGORITHMS ON $\mathcal{T}$ (:, TD). THE MAXIMAL PSNR (dB) AND SSIM VALUES OF EACH ROW ARE IN BOLD, AND THE SECOND ONES ARE UNDERLINED.

| | scale | Bicubic (2D) | NLM [55] (2D) | SRCNN [17] (C1) | VDSR [22] (C1) | RDN [30] (C1) | CSN (C1) | EDSR [29] (C96) | CSN (C96) |
|---|---|---|---|---|---|---|---|---|---|
| PD | SR×2 | 34.65/0.9625 | 36.18/0.9707 | 38.23/0.9802 | 39.89/0.9850 | 40.39/0.9862 | **41.77/0.9897** | 39.47/0.9837 | 39.50/0.9839 |
| | SR×3 | 30.88/0.9167 | 32.02/0.9324 | 32.90/0.9432 | 34.27/0.9555 | 35.00/0.9609 | **36.09/0.9697** | 33.97/0.9531 | 34.12/0.9540 |
| | SR×4 | 28.82/0.8713 | 29.27/0.8906 | 30.52/0.9078 | 31.69/0.9244 | 32.64/0.9362 | **33.51/0.9489** | 31.44/0.9219 | 31.72/0.9246 |
| T1 | SR×2 | 33.38/0.9460 | 34.71/0.9581 | 36.52/0.9705 | 37.58/0.9760 | 38.08/0.9784 | **38.62/0.9813** | 37.09/0.9741 | 36.99/0.9737 |
| | SR×3 | 29.79/0.8793 | 30.83/0.9027 | 31.72/0.9187 | 32.57/0.9304 | 33.33/0.9416 | **33.68/0.9464** | 32.27/0.9274 | 32.25/0.9266 |
| | SR×4 | 27.96/0.8182 | 28.68/0.8439 | 29.31/0.8616 | 30.14/0.8818 | 31.00/0.9018 | **31.27/0.9092** | 30.04/0.8803 | 30.07/0.8794 |
| T2 | SR×2 | 33.06/0.9541 | 34.56/0.9641 | 37.04/0.9773 | 38.74/0.9823 | 40.02/0.9826 | **40.47/0.9868** | 38.11/0.9803 | 38.20/0.9807 |
| | SR×3 | 29.50/0.9016 | 30.57/0.9197 | 31.80/0.9381 | 33.23/0.9515 | 33.99/0.9576 | **34.95/0.9653** | 32.89/0.9482 | 33.00/0.9490 |
| | SR×4 | 27.60/0.8511 | 28.37/0.8718 | 29.32/0.8960 | 30.51/0.9162 | 31.49/0.9301 | **32.28/0.9421** | 30.31/0.9137 | 30.54/0.9163 |

pattern of the LR image is different from simply shrinking the image size in the image domain by, e.g., bicubic interpolation [25]. Table IV shows the quantitative comparison between different methods under truncation degradation. Again, our CSN model presents the best SR performance in the case of C1, followed by RDN [30]. It is noteworthy that the SR performance of bicubic, NLM [55], SRCNN [17] and VDSR [22] is slightly worse than that of these methods on $\mathcal{T}(:, BD)$ in the same setting, e.g., SR×2 on T2 images. On the contrary, the results of RDN [30] and our CSN on $\mathcal{T}(:, TD)$ are better than that on $\mathcal{T}(:, BD)$, which may imply that they are more suitable for the task of MR image super resolution because the image degradation used to generate the LR images are more compatible with the real MR imaging process.

Fig.10 presents the visual effects of the compared methods on a PD image (top, SR×2), a T1 image (middle, SR×4) and a T2 image (bottom, SR×4). The red arrows (both thick and thin ones) mark the most distinct positions in the small image patches. For the T1 image, other methods fail to reconstruct the white ridge pointed by the thick arrows and the dark line pointed by the thin arrows. However, The proposed CSN model can clearly recover these details. Similar comparison can also be observed from the T2 image. There are two black holes near to the top end of the small image patch, while only our CSN model can give a clear indication of the potential structure. We can also observe other obvious differences, e.g., the black hole at the bottom end of the image patch.

## V. DISCUSSION AND FUTURE WORK

### A. Multiple Branches

Like the original stage mapping in [34], the stage mapping in our CSB can also be easily extended to more branches (⩾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 map makes the model wider, like [34], [35]. This significantly increases model parameters when the network depth is the same. Since EDSR [29] is a typical network with very wide structure and causes training failure, while RDN [30] 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 [32], [31] and DenseNet [33] (or RDN [30]) 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], [29]. The medical images SR framework combined multi-type and multi-scale information is also expected to further improve the SR performance of deep models.

Fig. 10. The visual effect of the compared methods on a PD image (top), a T1 image (middle) and a T2 image (bottom) with scaling factor SR×2, SR×4 and SR×4, respectively. The maximal PSNR (dB) and SSIM values for each displayed image are in bold, and the second ones are underlined.

## 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 block (CSB). 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 improve integrate information, the MAR [34] mapping is also applied to merge the feature maps of 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] 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.

[29] 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.

[30] 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

[31] 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.

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

[33] 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

[34] 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

[35] 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

[36] 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

[37] 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.

[38] 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

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.

[44] 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.

[45] 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.

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

[47] 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.

[48] 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

[49] 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.

[50] 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

[51] 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

[52] 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.

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

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

[55] 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.