A Trusted Medical Image Super-Resolution Method based on Feedback Adaptive Weighted Dense Network

Lihui Chen[a], Xiaomin Yang[a,*], Gwanggil Jeon[b,c], Marco Anisetti[d], Kai Liu[e]

[a]College of Electronics and Information Engineering, Sichuan University, Chengdu, Sichuan, 610065, China
[b]Department of Embedded Systems Engineering, Incheon National University, Incheon, 22012, Korea
[c]School of Electronic Engineering, Xidian University, Xi'an 710071, China
[d]Dipartimento di Informatica (DI), Università degli Studi di Milano, Via Celoria 18, Milano (MI) 20133, Italy
[e]College of Electrical Engineering, Sichuan University, Chengdu, Sichuan, 610065, China

Abstract

High-resolution (HR) medical images are preferred in clinical diagnoses and subsequent analysis. However, the acquisition of HR medical images is easily affected by hardware devices. As an effective and trusted alternative method, the super-resolution (SR) technology is introduced to improve the image resolution. Compared with traditional SR methods, the deep learning-based SR methods can obtain more clear and trusted HR images. In this paper, we propose a trusted deep convolutional neural network-based SR method named feedback adaptively weighted dense network (FAWDN) for HR medical image reconstruction in virtual environment. Specifically, the proposed FAWDN can transmit the information of the output image to the low-level features by a feedback connection. To explore advanced feature representation and reduce the feature redundancy in dense blocks, an adaptive weighted dense block (AWDB) is introduced to adaptively select the informative features. Experimental results demonstrate that our FAWDN outperforms the state-of-the-art image SR methods and can obtain more clear and trusted medical images than comparative methods.

Keywords: Medical image super-resolution, trusted medical image reconstruction, deep convolutional neural network, feedback mechanism, adaptive weighting

1. Introduction

Medical images are widely used for clinical diagnosis, in which high-resolution (HR) images are preferred because they can provide much more significant structure and texture details than low-resolution (LR) images [1]. However, the acquisition of HR medical images is limited by hardware devices.

As an effective and trusted alternative method, single image super-resolution (SR) aims to reconstruct the HR image from its LR counterpart. Image SR is initially proposed for natural image. Lots of medical image SR methods have also emerged in recent years, including interpolation-based [2, 3], reconstruction-based [4, 5], and learning-based methods [6, 7]. Although the interpolation-based SR method is fast, the reconstructed HR image of these methods is blurred. The reconstruction-based methods consider image priors, e.g, non-local [8] and self-similarity [9], but these methods always lead to a lot of performance drop when the prior is inconsistent with test images. Conventional learning-based methods, e.g., sparse representation [6], compressed sensing [10], and random forest [11] have limited representation ability. Overall, the reconstructed HR images obtained by above methods are blurred or unreliable because these methods are susceptible to priors.

Recently, convolutional neural network (CNN) based SR methods have achieved significant improvement in the performance. Dong et. al. [12] propose the pioneering CNN-based method, which uses a full convolution neural network for natural image SR. After that, several SR networks focus on deeper or wider network architecture design such as VDSR [13], DRCN [14], SRDenseNet [15], and DRRN [16]. In addition to the CNN based methods, to obtain appealing visual effect, perceptual loss [17] and generative adversarial network (GAN) [18] are introduced into image SR. However, the textures and details generated by GAN may be different from the ground truth. In medical image SR, numbers of deep learning-based medical image SR methods have also been presented recently. Pham et. al. [19] extend SRCNN to 3D cases for brain MR images SR. As for 2D medical image SR, Wei et al. [20] propose a deep network (DDSR) constructed by dense blocks for MR and CT image SR. However, all of the above methods are feedforward networks, in which the information can only pass from the input to the output. It is known that there always exist difference between the reconstruction result and the ground truth. In the feedforward network, its output cannot be transmitted to the input for improving the performance. As a consequence, these methods have limited ability to reconstruct fine textures, especially for the images with complicated details and textures.

To address the limitation of feedforward networks, we introduce a feedback mechanism into SR networks and propose a network denoted as *Feedback Adaptive Weighted Dense Net-*

*Corresponding author. Email:arielyang@scu.edu.cn

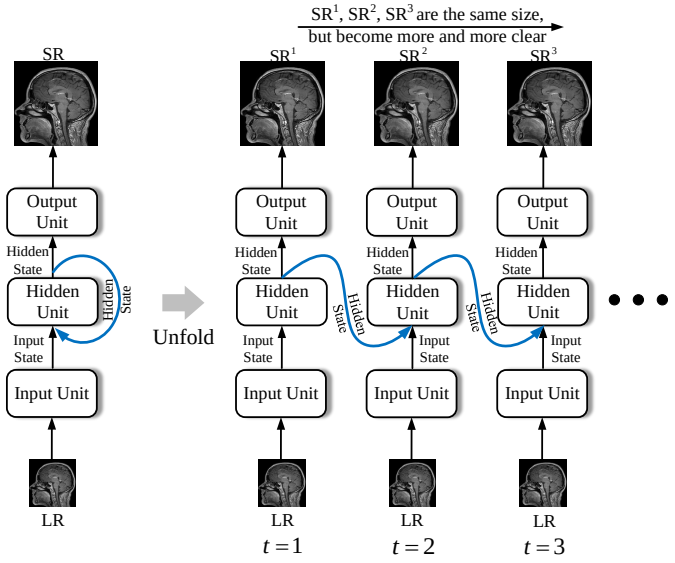


Figure 1: A simple illustration of our FAWDN. The input state is the output of the input unit. The input of the hidden unit is the concatenation of the input state and the hidden state of the previous time step.

work (FAWDN) for trusted medical image SR. The implementation of this feedback mechanism is based on recurrent neural networks (RNNs) since RNN is indispensable for the feedback mechanism [21]. Through the feedback mechanism, our FAWDN can correct the error produced in the preceding time step and obtain a clearer HR image step by step.

As shown in Figure 1, our FAWDN is comprised of three parts: the input unit, the hidden unit, and the output unit, respectively. The input unit is utilized to extract low-level features as the input state. Then, the concatenation of the input state and the hidden state will be fed into hidden unit to obtain a new hidden state, while the initial hidden state is same with the input state. At each time step, the hidden state is passed to output unit to reconstruct HR images. Therefore, a reconstructed HR image is outputted by the FAWDN every time step. These reconstructed images have the same size but will be clearer with the time step. Besides, it will be also transmitted to the hidden unit at the next time step as the feedback signal to achieve the feedback mechanism. To obtain a superior hidden state, an adaptively weighted dense block (AWDB) is introduced into the hidden unit. AWDB is built on dense links [22, 15] because dense links have features reusing and strong representation capability. Furthermore, the dense link can eliminate gradient vanishing. However, in traditional dense links, each convolutional layer equally receives features from all previous convolutional layers, which makes the input features of each convolutional layer extraordinarily redundant. Therefore, we improve the original dense links by adding an adaptive weighting group before every convolutional layer in AWDB. So the convolutional layer can adaptively select informative features and reduce the feature redundancy.

In summary, the contributions of this paper are as follows:

1. To improve medical images resolution, we introduce FAWDN for medical image SR as an alternative method. Experimental results indicate that the introduced FAWDN could obtain superior SR performance over comparative image SR methods.
2. To enhance the quality of reconstructed HR images, a feedback mechanism based on RNN is presented to correct the errors produced by the network at preceding time steps.
3. To reduce the redundancy of input features of convolutional layers in dense link, we introduce adaptive weighting groups into AWDB to adaptively select informative features. The adaptive weighting groups can be trained with the FWADN.

2. Related work

2.1. Feedback mechanism

In computer vision community, feedback mechanisms have been explored for some high-level tasks, such as classification [23], human pose estimation [24], and crowded counting [25]. However, since the targets of high-level and low-level tasks are different, it is inappropriate to directly transfer these methods to low-level vision tasks such as image SR. Furthermore, these networks achieve a feedback mechanism in a top-down manner, in which only the high-level feature is transmitted to the next time step. However, in control theory, the feedback mechanism refers to that a system transmits the output information to its input for correcting the errors in the previous output. Since only high-level feature rather than the output information is transmitted to the low-level convolutional layers in [26, 24, 25], their feedback mechanism is actually not strict. Different from the feedback manner in the above methods, the feedback mechanism in our FAWDN agrees with the strict feedback mechanism, which will be detailed in Section 3.1. Containing the information of the output image, the feedback signal in FAWDN modifies the input of the hidden unit. Accordingly, our FAWDN can obtain a clearer HR image step by step.

The most relevant works to our network are [21] and [27], whereas both of them are introduced to natural image SR. Moreover, [27] transmits multiple high-level features to low-level layers, but only the highest-level features contain information of the output image. In other words, [27] is also incompatible with the strict feedback mechanism. Furthermore, the hidden unit of our FAWDN is completely different from hidden unit of [27] and [21] .

2.2. Deep-learning based image SR

As the first attempt of the DL-based method, SRCNN is limited in the representation ability because of the shallow network. Therefore, subsequent networks, such as VDSR [13], DRCN [14], and DRRN [16], focus on widening or deepening the network to improve the performance. Since there is no learnable upsampling method in early SR networks, all of these networks interpolate the LR input to HR size and extract features in HR space. In other words, these networks super-resolve LR images in the HR space, which results in complicated computations. Not until sub-pixel convolution [28] and

deconvolution [29] were introduced, numbers of networks directly adopted LR images as the input. Although recently proposed SR networks [30, 31, 32, 33, 34] have achieved a significant improvement in performance and computation efficiency, these networks are unable to remove artifacts in complicated images, especially for richly-textured medical images. Therefore, the proposed FAWDN integrates the feedback mechanism, which is capable of correcting the artifacts by the feedback mechanism.

Despite our FAWDN is based on RNN, it is different from existed RNN based SR network [14, 16, 35]. The first RNN-based SR networks DRCN [14], achieved by a single recurrent convolution layer, aims to decrease the parameters of the network. To mitigate gradient dilemma in deep RNN, i.e, gradient vanishing and gradient exploding, Tai et. al. [16] adopt a multi-path residual learning and global residual learning in their DRRN. Exploring dual hidden state for SR, Han et. al. [35] present DSRN that transmits two states, i.e., LR state and HR state, over each time step. Our FAWDN is different from above RNN methods in two aspects. First, the above presented RNN-based SR networks have no feedback mechanism. One reason is that information of the output image is scarcely contained in the output of the hidden state. The other reason is that only the hidden state is fed into the hidden unit, which means that no input state can be affected by the feedback signal. Second, our FAWDN is different from these methods in the hidden unit. The hidden unit of our FAWDN consists of two 1×1 convolutional layers and a AWDB, while DRCN and DRRN respectively consist of a single convolutional layer and an enhanced residual block. The proposed AWDB can not only eliminate the gradient dilemma but also adaptively choose informative features for reusing.

2.3. GAN for Image super-resolution

In the image SR community, current deep learning-based image SR methods can be roughly divided into two types, which are the restoration-oriented methods and perceptual-oriented methods, respectively. The restoration-oriented methods aim to reconstruct a HR image that is as close as possible to the ground truth. All methods mentioned in Section 2.2 are restoration-oriented methods. Contrastly, the perceptual-oriented method aims to recover an appealing and photo-realistic HR image, which contains more details or textures.

To obtain a perceptually appealing image, Johnson et al. [17] introduced perceptual loss into image SR. The perceptual loss consists of feature reconstruction loss, which is implemented by minimizing the $L2$ norm of the difference of feature maps outputted by a pretrained network with input of the SR image and the ground truth. To further improve the perceptual quality of SR image, Christian et al. [18] integrated the GAN loss into the perceptual loss, in which the GAN loss is used to force the SR image distribution similar to natural image distribution. Furthermore, texture matching loss was proposed in EnhanceNet [36] to enforce the local similarity between the SR image and the ground truth. To utilize semantic information to super-resolve images, Wang et al. [37] proposed SFTGAN, in which a image semantic segmentation map obtained by a pretrained semantic segmentation network is fed into the generator along with the LR images. Also, Wang et al. [38] improved the generator by residual-in-residual dense block and used the relativistic GAN as the discriminator in their ESRGAN. To further improve the visual quality of reconstruction images, Zhang et al. [39] trained a ranker to optimize generator in the direction of indifferentiable perceptual metrics (i.e., PI [40], NIQE [41], and Ma [42]). Recently, a gradual learning process [43] was introduced into GAN for image SR to accelerate training and improve the stability of GAN. To enhance the robustness of the GAN, Zareapoor et al. [44] proposed a DualGAN for image SR by the game between two generators and two discriminators.

Although GAN models can reconstruct appealing details and textures, these methods always obtain lower PSNR and SSIM than the restoration-oriented methods because the recovered details or texture may be inconsistent with the ground truth [18, 36, 37, 38]. However, for trusted medical image SR, one expects the reconstructed HR image is as close as possible to the ground truth. Therefore, we proposed a restoration-oriented method (i.e., the proposed FAWDN) for medical image SR rather than a perceptual-oriented method.

2.4. Super-resolution for medical image

The early medical image SR methods reconstruct HR images with multiple LR images. For instance, Irani et al. [45] applied iterative back projection (IBP) to several spatially shifted LR images to restore HR images. After that, POCS [46] as well as IBP [45] were utilized to produce 3D MR volume from numbers of 2D slices. However, these methods are complicate attributed to the registration and fusion of multiple LR images. Therefore, Rousseau et al. [47] proposed the first single image based medical image SR method in which some extra information of another HR image was introduced to super-resolve the LR image. However, these early medical image SR methods achieve blurred reconstruction images and even introduce extra artifacts, because these methods lack effective image priors and features.

Recently, various of learning based methods including conventional learning based methods [48, 49, 50] and DL-based methods [19] have been proposed. The conventional learning based medical image SR, such as sparse coding [48, 49] and compressive sensing [51] -based method, have limit ability to recover image with complicated details or textures. Benefiting from the strong feature extracting ability of deep neural networks , some DL-based SR methods have also emerged in the medical image processing community. Pham et al. [19] proposed SRCNN3D for brain MR images SR. Since the shallow layers of SRCNN3D, the representation and mapping ability are insufficient for medical image SR. To obtain better performance, Wei et al.[20] proposed a deep dense network (DDSR) for MR image SR. Although DDSR utilized a deep architecture, its input is interpolated to HR size, which will take more computing resources and time. A common deficiency in these two methods is that they rarely consider the difference between natural images and medical images, which only extend the SR networks for natural images to medical images. Differently, as the self-similarity of the medical image is more obvious than the natural image, the combination of the feedback mechanism

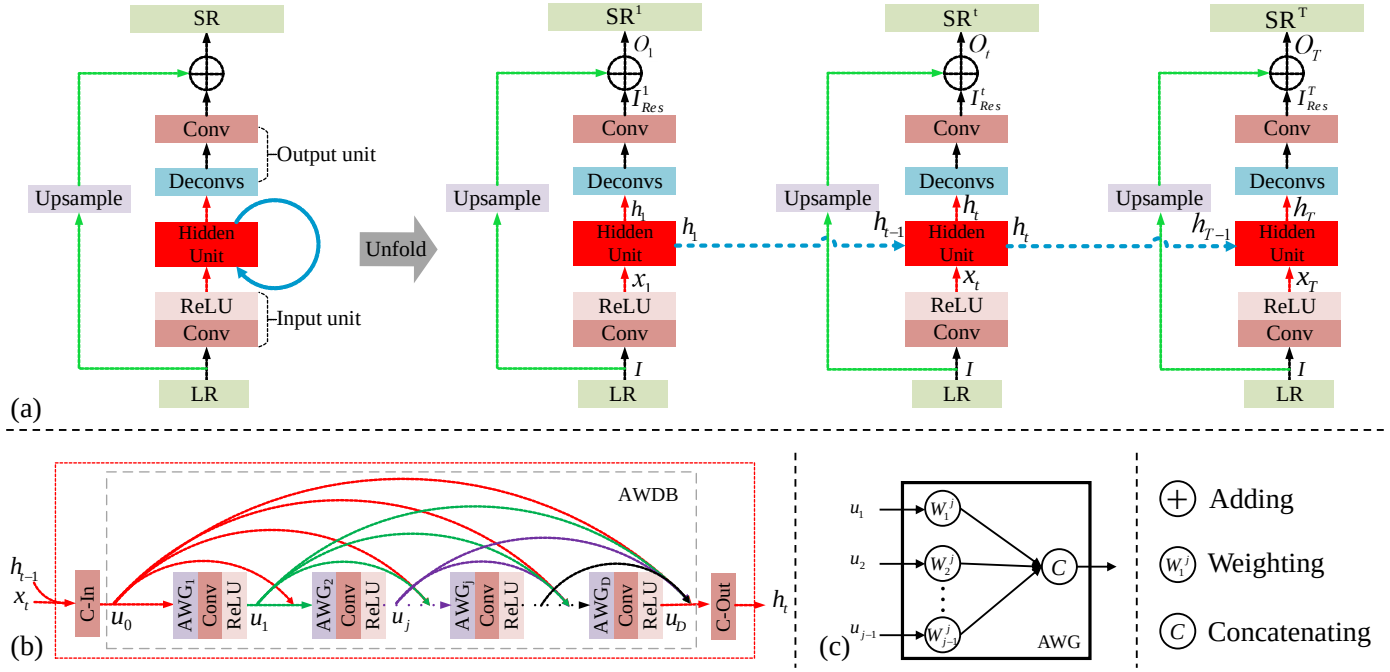


Figure 2: The architecture of our FAWDN.

and AWDB can adaptively use informative features and similar structures to construct HR images.

3. Feedback adaptive weighted dense network

3.1. Network architecture

To achieve a strict feedback mechanism in neural networks, three indispensable conditions are required to be met. First, the network inevitably is recurrent over time so that the feedback signal could be routed to the input at next time step. Second, the feedback signal has to contain the information of the output image, thereby satisfying the condition of output to input. Third, the low-level features are combined with feedback signals as the total input. This is because no input feature can be revised by feedback signal if there is no low-level feature.

As described in above, RNN is a necessary condition of feedback mechanism, so we build our FAWDN by RNN with time steps of T. As shown in Figure 2-(a), the unrolled FAWDN contains T sub-networks. Each sub-network can be regarded as an independent network at a specific time step. Each sub-network includes three parts: the input unit, the hidden unit, and the output unit. Similar to other RNN, the parameters of these three parts in FAWDN are shared across time steps. To achieve the strict feedback mechanism, the output of the network naturally needs to be transmitted to the input of the network. However, since the output size (i.e., the width and height) is inconsistent with the input size, the output is unable to be directly routed to the input of the network as the feedback signal. Having the same size with input image, the hidden state is an alternative as long as it contains the information of the output image. To obtain a superior feedback performance, the hidden state is routed to the input state rather than the input image. In this way, the hidden state can directly refine low-level features. Besides, the input state is a perfect alternative of the input image as it is constant in a feedforward propagation. Since the parameters of the input unit are shared across time step, once the input image is determined, the input state is a constant in a forward propagation. Therefore, the feedback mechanism in our FAWDN is achieved by transmitting the hidden state to the input state.

Given the input image I_t of t-th time step, the output of corresponding time step of the network is denoted as O_t. As shown in Figure 2-(a), the output is obtained by the sum of the residual image I^t_{Res} and the upsampled image. For implementing the feedback mechanism, I_t has to be constant (i.e., $I_1 = I_2 = \cdots = I_T$) in a forward propagation. Benefiting from the feedback mechanism, O_t will be clearer and get closer to the HR image as time step approaches T. Therefore, the proposed FAWDN can be formulated as follows:

$$O_t = N(\theta; I_t) + I_{up} \quad (1 \le t \le T), \tag{1}$$

where $N(\theta; I_t)$ is our FAWDN with parameters of θ and the input image of I_t, and I_{up} is a bilinear upsampled image of the input image. Since T time steps is included, there are T outputs in total. The size of these T outputs is same with that of the ground truth, but the outputs will be closer to the ground truth over time step due to the feedback mechanism. As aforementioned, O_t will become clearer step by step. Consequently, only O_T is chosen as the final reconstructed HR images, and other outputs will be tied to the loss functions to make the hidden state contain the information of the output image.

For each sub-network at a specific time step, the input unit only consists of a single convolutional layer with kernel size of 3×3 and stride of 1 to extract low-level features. Thus, the

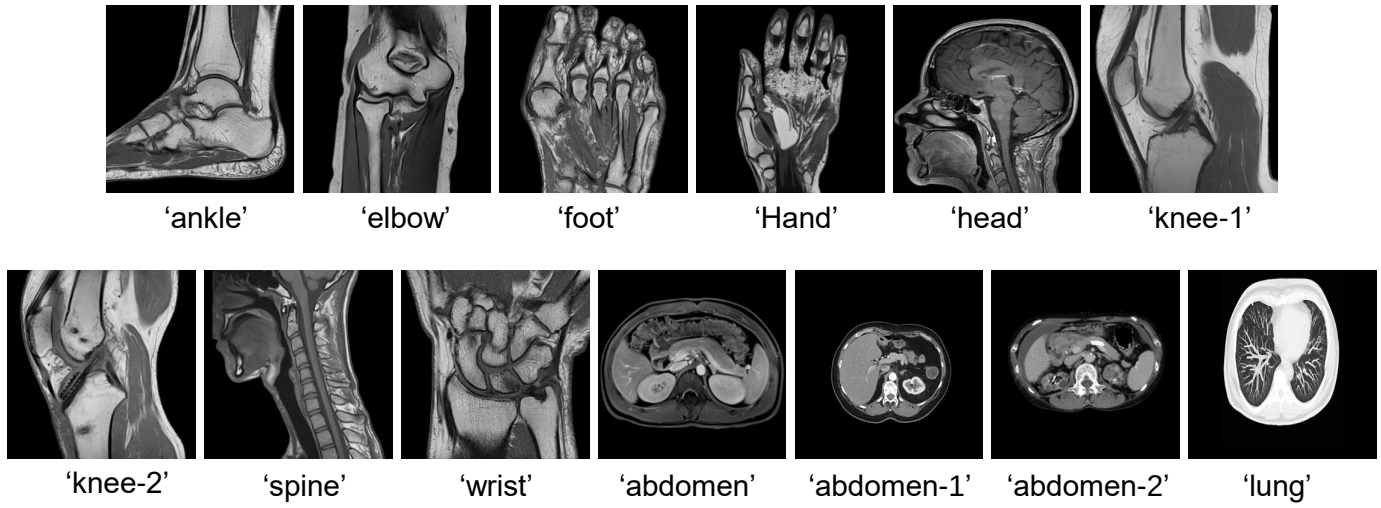


Figure 3: The MR images in the dataset MRI13.

input state can be obtained by

$$x_t = f_{in}(I_t) = max(W * I_t + b, 0), \tag{2}$$

where x_t and $f_{in}(\cdot)$ are the input state of t-th time step and the combined function of convolution and activation function, respectively. The W and b are respectively the weight and bias of the input unit. $max(\cdot, 0)$ and $*$ denote ReLU activation function and convolutional operation, respectively. Since the parameters of input unit is shared across time steps and $I_t = I$, x_t is also constant across time steps. Therefore, we can select the input state x_t as an alternative input for achieving feedback mechanism.

To implement the feedback mechanism, the hidden unit needs to receive both the input state and the feedback signal. In the proposed FAWDN, we use the previous hidden state h_{t-1} as the feedback signal of recurrent time step t. Therefore, the hidden unit can be formulated as follows:

$$h_t = f_h(\theta_h; [x_t, h_{t-1}]), \tag{3}$$

where $f_h(\theta_h; \cdot)$ represents the function of the hidden unit with parameters of θ_h. $[x_t, h_{t-1}]$ denotes concatenation of x_t and h_{t-1}, where $h_0 = x_1$.

In order to make the hidden state contain the information of the output image, it is necessary to feed the hidden state into the output unit so that a output image can be generated by the output unit. Then, the information of the output image can be restricted in hidden state by tying a loss function to the output image. The output unit is comprised of one or two deconvolutional layers followed by a convolutional layer. For x2 and x3 upscale, a single deconvolutional layer is used to upsample the features. For x4 upscale, two deconvolutional layers for x2 upscale are stacked to upsampled the features. Therefore, we can obtain the output of FAWDN at t-th time step by

$$O_t = f_o(\theta_o; h_t), \tag{4}$$

where the $f_o(\theta_o, \cdot)$ refers to the function of the output unit with parameters of θ_o.

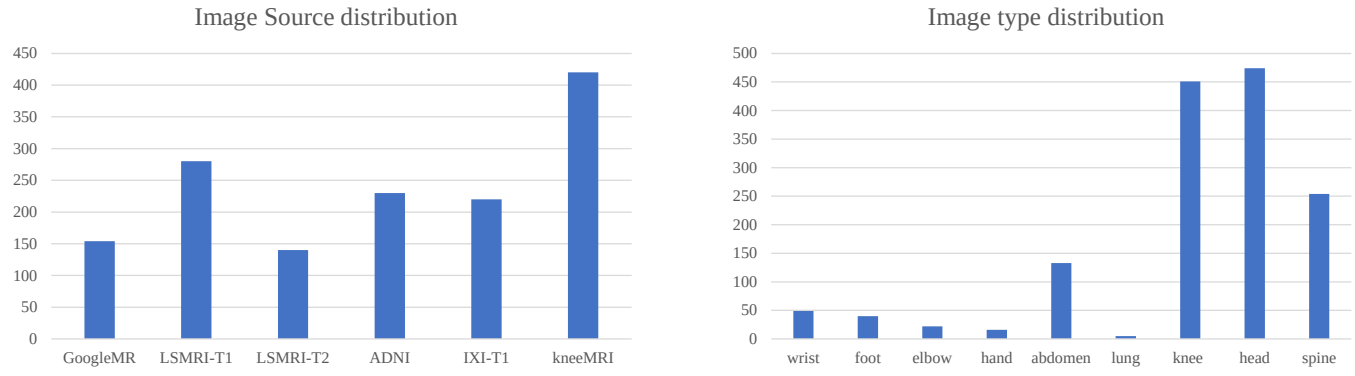
If the loss function only tied to the last time step, the hidden state of previous time steps cannot obtain the information of the output image, thereby failing to achieve the feedback mechanism. When the loss function only tied to the last time step, only the last hidden state h_T can be used to reconstruct HR images, while other hidden states are unable to be used to reconstruct HR images. In that case, these intermediate hidden states have no information of the output image, which is inconsistent with the strict feedback mechanism. As the hidden state is fed into the output unit to generate the HR image every time step, we tie a loss to the HR image every time step to make all hidden states contain the information of the output image. In our FAWDN, the $L1$ loss function is employed. The loss of t-th time step can be obtained by:

$$L_t = \frac{1}{N} \sum_{i=1}^{N} \left\| {O_t}^i - y^i \right\|_1, \tag{5}$$

where N and y_i respectively denote the number of batch size and the i-th HR image in the batch. For FAWDN, the total loss function can be obtained by $L = \frac{1}{T} \sum_{i=1}^{T} (W_t L_t)$, where the W_t is the weight of t-th loss function, and every W_t is set to 1 in our experiments.

3.2. Hidden Unit

As shown in Figure 2-(b), the hidden unit consists of two 1×1 convolutional layers (i.e., C-In and C-Out) and an adaptive weighted dense block (AWDB). To achieve the feedback mechanism, the hidden unit simultaneously receives both the input state and the hidden state. C-In is used to modify the low-level feature in input state with the hidden state. Then, AWDB is utilized to further extract more informative and advanced features. These features are adopted to reconstruct the HR image at current time step as well as modify the low-level feature at the next time step. Since numerous features are produced by AWDB, finally, these features are compressed by C-Out as the hidden state.



(a) (b)

Figure 4: The collected medical image SR dataset. (a) the distribution of images from various data sources; (b) the distribution of image types.

To correct the low-level feature in the input state, the hidden state is routed to the input state as the feedback signal. First, the low-level features is modified with the feedback signal by C-In as follows:

$$u_0 = f_{cin}([x_t, h_{t-1}]), \tag{6}$$

where $f_{cin}(\cdot)$ denotes the function of C-In. $[x_t, h_{t-1}]$ represents the concatenation of the input state and the hidden state of the previous time step. u_0 represents the modified low-level feature.

Then, the modified feature u_0 is fed into AWDB to extract informative high-level features by adaptive weighted dense links. As shown in Figure 2-(b), the dense links in AWDB are weighted by the adaptive weighting group before each convolutional layer. Through the adaptive weighting group, each convolutional layer in AWDB can adaptively select the informative features according to its demands. Due to the dense link, lots of feature maps $\{u_1, u_2, \cdots, u_D\}$ are produced in AWDB as follows:

$$\{u_1, u_2, \cdots, u_D\} = f_{awdb}(u_0) \tag{7}$$

where $f_{awdb}(\cdot)$ denotes the function of AWDB. D and u_j are the number of layers and the output of the j-th layer in AWDB, respectively. As shown in Figure 2-(b), every layer in AWDB is comprised of an adaptive weighting group (see Figure 2-(c)), a convolutional layer, and an activation function ReLU [52]. The output of each layer contains G (also named growth rate in [15, 32]) feature maps and can be obtained by

$$u_j = max(f_j([w_0^j \cdot u_1, w_1^j \cdot u_2, \cdots, w_{j-1}^j \cdot u_{j-1}]), 0) \tag{8}$$

where $f_j(\cdot)$ and $max(\cdot, 0)$ denote the function of the j-th convolutional layer and the ReLU activation function, respectively. $[\cdot]$ represents the operation of concatenation. $w_c^j (0 \le c < j)$ denotes the weight for the output of the c-th features inputted to the j-th adaptive weighting group.

As shown in Equation 7, since lots of features are produced by AWDB, it will need a lot of parameters and computational burden to transfer all features to low-level feature. Therefore, we use C-Out (see Figure 2-(b)) to compress the features, and then the compressed features (i.e., the hidden state) are transmitted to the next time step as the feedback signal. Therefore, the hidden unit h_t can be obtained by

$$h_t = f_{cout}([x_0, x_1, \cdots, x_D]) \tag{9}$$

where $f_{cout}(\cdot)$ denotes the function of the second 1×1 convolutional layer.

Table 1: The detail of the proposed FAWDN. $Conv(c_{in}, c_{out}, k, s, p)$ and $Deconv(c_{in}, c_{out}, k, s, p)$ respectively denote the convolutional layer and the deconvolutional (also named transposed convolutional) layer, where c_{in} and c_{out} denote the number of input channel and out channel of the corresponding layer, respectively. k, s, and p respectively represent the kernel size, stride and padding of the convolutional or deconvolutional layer. $W(16d)$ indicates the adaptive weighting group with $16d$ weights. D denotes the total number of convolutional layers in AWDB.

Input Unit	$Conv(3, 16, 3, 1, 1)$	
	$ReLU()$	
Hiden Unit	C-In:	$Conv(16, 282, 1, 1, 0)$
	1st Layer	$W(16 \times 1)$
		$Conv(16\times, 161, 3, 1, 1)$
		$ReLU()$
	2nd Layer	$W(16 \times 2)$
		$Conv(16 \times 2, 16, 3, 1, 1)$
		$ReLU()$
	⋮	⋮
	D-th Layer	$W(16D)$
		$Conv(16 \times D, 16, 3, 1, 1)$
		$ReLU()$
	C-Out:	$Conv(16D, 256, 1, 1, 0)$
Output Unit	×2:	$Deconv(256, 3, 4, 2, 1)$
	or ×3:	$Deconv(256, 3, 7, 3, 2)$
	or ×4:	$Deconv(256, 3, 4, 2, 1)$
		$Deconv(256, 3, 4, 2, 1)$
	$Conv(3, 256, 3, 1, 1)$	

3.3. Architecture and implementation details

Architecture details. The details of the proposed FAWDN is shown in Table 1. Following SRDenseNet [15], we set the number of the output channel of the convolutional layer in both AWDB and the input unit to 16. The number of feature maps of hidden state is set to 256. Therefore, the number of the input channel of C-In (see Figure 2-(b)) is 16+256. We set the kernel size, stride and padding of all convolutional layers in ADWB to 3, 1 and 1, respectively. The kernel size, stride, and padding of C-In and C-Out are set to 1, 1 and 0, respectively. The setting of the deconvolutional layer is shown in Table 1. For the task

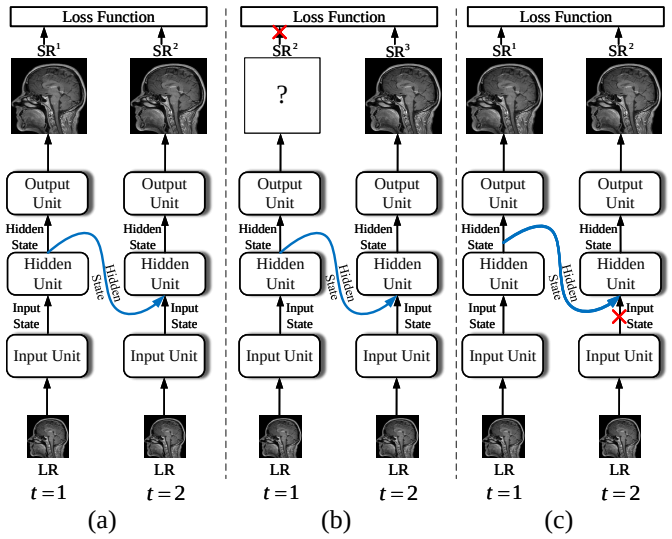


Figure 5: The different version our FAWDN. The red cross indicates that the corresponding connection is broken. (a) The full version of FAWDN; (b) FAWDN only tied loss function to the 2nd time step; (c) FAWDN only with

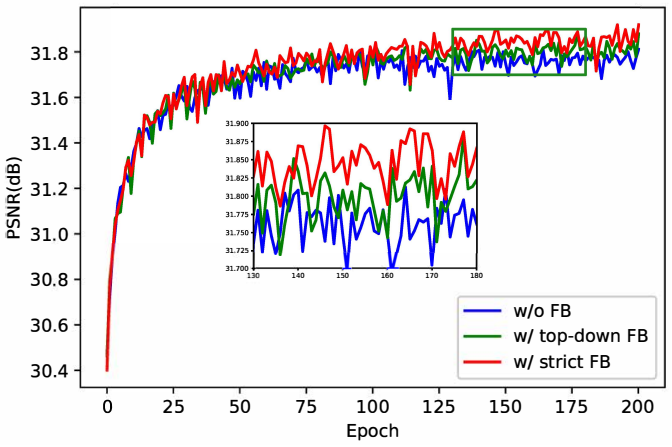


Figure 6: The investigation of the feedback mechanism.

of ×2 and ×4 image SR, the setting is set according to [15]. For the task of ×3 image SR, the parameters of the deconvolution layer is set according to [21]. The number of the input and output channel of FAWDN is set to 3 for both gray and color images. For the gray image, we copy the image three times and then feed the counterpart into FAWDN. To match the number of channel of the ground truth, the ground truth is also copied three-times for training.

Table 2: Study of the feedback mechanism.

Methods	w/o feedback	w/ top-down feedback	w/ strict feedback
PSNR	32.89	32.79	32.80
SSIM	0.9065	0.9063	0.9069

Datasets. To fairly compare with other natural image SR networks [14, 15, 29, 35], we use the DIV2K [53] as the training set and Set5 [54] as the validating set to train our FAWDN. Besides, to investigate the performance of network trained on

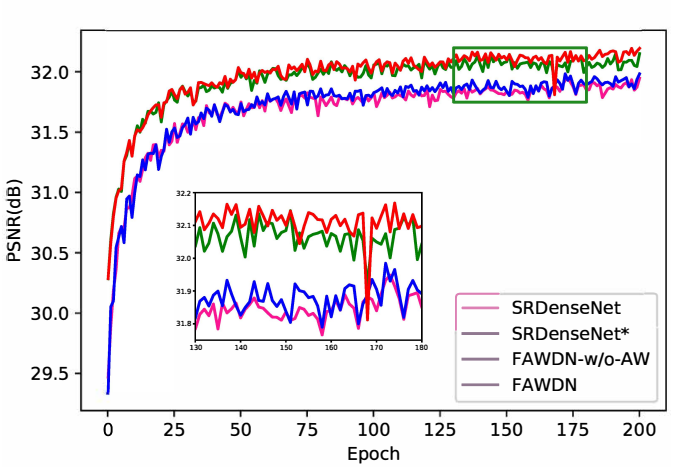


Figure 7: The investigation of adaptive weighting groups.

Table 3: The test results of ×4 upsampling on MRI13. 'Recurrent ' represents if the network is RNN, 'AW' denotes the adaptive weights, and the 'FB' represents the strict feedback mechanism.

Methods	SRDenseNet	SRDenseNet*	FAWDN w/o AW	FAWDN
Recurrent	×	×	✓	✓
AW	×	✓	×	✓
FB	×	×	✓	✓
PSNR	32.92	32.97	33.03	**33.22**
SSIM	0.9075	0.9076	0.9087	**0.9098**

medical images, we collect a medical image SR dataset (called MRIMP) from GoogleMR[1], IXI [55], ADNI [56], KneeMR [57], and LSMRI [58]. This collected medical image dataset includes different parts of human body, and its distribution is shown in Figure 4. However, the network trained only on the medical image dataset will be overfitting in the training dataset (see Section 4.3). Therefore, we train the proposed network by the mixture of the medical image dataset and 200 natural images from DIV2K, and denote this network as FAWDN+. As we focus on medical image SR, we use medical images utilized in DDSR [20] as the test dataset. This medical dataset denoted as MRI13 includes 13 MR images. As shown in Figure 3, MRI13 covers different parts of the human body, which is a great dataset to test network performance for medical image SR. Furthermore, we respectively collect two other test datasets named ADNI100 and OASIS100 from ADNI [56] and OASIS [59]. Each dataset includes 100 images, which are not contained in the training dataset. In this paper, following SRCNN [12], SRDenseNet [15], and LapSRN [60], we only consider the most common degradation model used in image SR, i.e, Bicubic downsampling. To evaluate different methods objectively, two commonly used metrics, i.e., PSNR and SSIM, are adopted to test result images of different methods. A higher value of PSNR or SSIM indicates a better result.

Training details. In the training stage, we employ the Adam [61] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ as the optimizer, and totally optimizes FAWDN 5×10^5 iterations. The initial

[1]GoogleMR: images crawled from the Google image by key words, such as 'MRI foot' and 'MRI lung'.

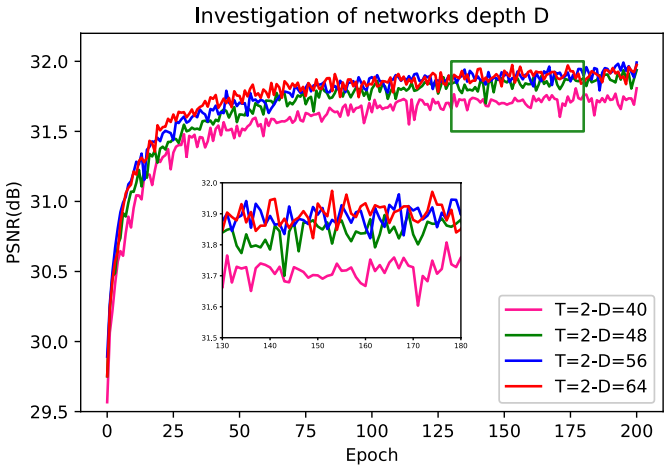


Figure 8: The investigation of the network depth D.

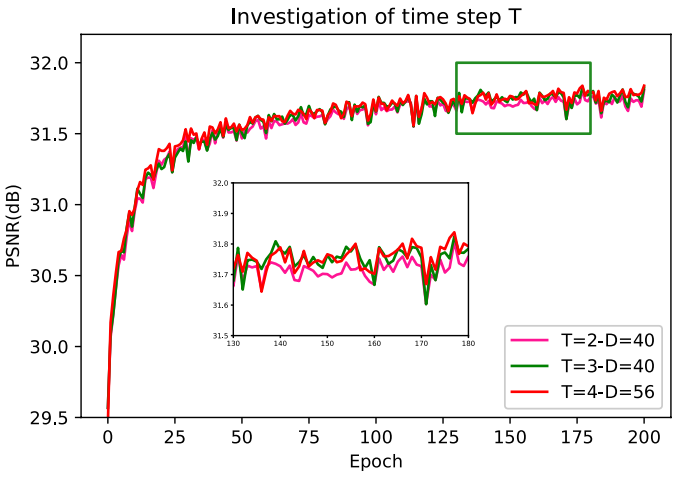


Figure 9: The investigation of the time step T.

Table 4: The average results of different number of convolutional layers on MRI13.

D	40	48	56	64
PSNR	32.82	32.97	32.94	32.96
SSIM	0.9062	0.9079	0.9078	0.9078

learning rate of Adam is set to 0.0002, and it is halved every 2×10^5 minibatch iteration. The minibatch size and LR size are set to 16 and 40 for FAWDN, respectively. For FAWDN+, since the resolution of some medical images is lower than 160×160, the LR size cannot be set to 40×40. Therefore, we set LR size to 30×30 and batch size to 16 for training FAWDN+. Following [62], data augmentation is conducted in our experiments. All experiments of FAWDN are conducted based on Pytorch 1.1.0 in desktops with an Nvidia GTX1080Ti/RTX2080Ti GPU.

4. Experiments and results

4.1. Ablation study

Effectiveness of the feedback mechanism. To verify the effectiveness of the feedback mechanism, we conduct a comparison experiment about three networks, which are illustrated in Figure 5. The first network is our FAWDN with network depth $D = 40$ and growth rate $G = 8$ because small D and G can reduce training time. The second network has the same structure as the first one except for the loss function. We removed the loss function at the 1st time step of the second network so it no longer satisfies the condition of the strict feedback mechanism, but meets the top-down feedback mechanism. In the third network, only the hidden state is fed into the hidden unit at the 2nd time step. Therefore, the third network cannot meet the condition of the strict feedback mechanism. As shown in Figure 6, the curve of PSNR on Set5 reveals the network with the feedback mechanism is superior to the other two networks. The test results on MRI13 are listed in Table 2. It is interesting that the network without feedback mechanism achieves the best results. We guess it is resulted from the difference of natural and medical images, since these networks are trained on natural images but test on medical images. It can be verified by the results of the validation dataset that the network with strict feedback mechanism achieves the best results.

Table 5: The average results of different time step on MRI13 of ×4 upsampling task.

T	2	3	4
PSNR	32.82	32.79	32.83
SSIM	0.9062	0.9060	0.9063

Effects of adaptive weighting groups. To demonstrate the effectiveness of the adaptive weighting groups, we also compare two groups of networks. In the first group, one network is SRDenseNet, and another is the modified SRDenseNet denoted as SRDenseNet* in this paper. SRDenseNet* is obtained by replacing the dense block in SRDenseNet by our AWDB. In the second group, one of the networks is FWADN, while another is a counterpart of FWADN but without the adaptive weighting group before every convolution layer. The curves of PSNR on Set5 are shown in Figure 7. The test results of these two groups are also listed in Table 3. From Figure 7 and Table 3, we can observe that the adaptive weighting groups can improve the performance of the network with the feedback mechanism as well as the network without the feedback mechanism. This is because the adaptive weighting group is capable of selecting the most useful features for various networks. In addition, we can find that the improvement benefiting from adaptive weights to the network with the feedback mechanism is more significant than that to SRDenseNet. It is because the adaptive weights can select more informative features as the feedback signal.

4.2. Investigation of D and T

To explore the trade-off between networks capacity and performance, we study the performance of various network depth D and time step T.

Since a small change of D will result in inconspicuous variation in performance, we set the variation stride of D to 8. The

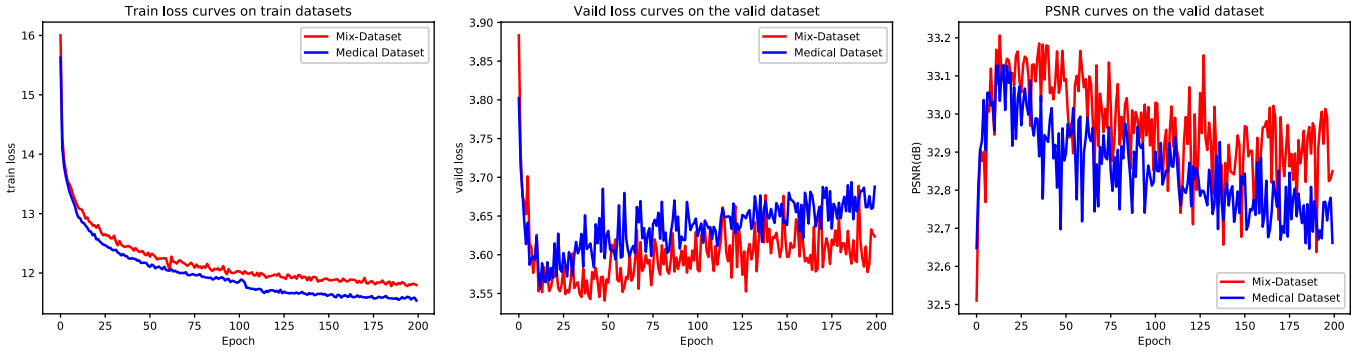


Figure 10: From left to right, curves of train loss, valid loss, and PSNR values are presented.

$I_{hr} - I_{up}$ $I^1_{1,res}$ (w/o FB) $I^2_{1,res}$ (w/o FB) $I^1_{2,res}$ (w/ FB) $I^2_{2,res}$ (w/ FB)

Figure 11: The residual image outputted by the output unit. (a) the outputs at different time step of the network only tied loss function to the last time step; (b) the outputs at different time step of the network tied loss function to all time step. The left column plots the output at the 1st time step and the right column plot the output at the 2nd step time.

Table 6: The performance of the proposed network trained on different training sets. FAWDN, FAWDN+, and FAWDN++ are trained with DIV2K, MRIMP+, and MRIMP, respectively.

	MRI13	ADNI100	OASIS100
FAWDN	32.22/0.9098	28.30/0.8117	30.05/0.8957
FAWDN+	32.21/0.9086	28.81/0.8259	30.38/0.8895
FAWDN++	32.13/0.9079	28.87/0.8228	31.57/0.9075

validation results on Set5 are illustrated in Figure 8. We can observe that deeper networks always achieve better results. However, this phenomenon barely appears in the MRI13 as shown in Table 4. It can be found that the performance of the network is saturated as the depth of AWDB reaches more than 48 layers. Nevertheless, the deeper network achieves better results on validation dataset. To further improve the performance, we set the depth of AWDB to 64.

To investigate the time step T, we train three FAWDNs, of which the time steps are respectively set to 2, 3, and 4. In this experiment, we set D to 8 for fast training. The results of validation dataset and test dataset are shown in Figure 9 and Table 5, respectively. According to the two groups results, we can find the performance of FAWDN is insensitive to the time step. A latent reason is FAWDN without long-term memory so that FAWDN can only correct the output of the nearest time step. Therefore, the time step is set to 2 in FAWDN.

4.3. Investigation of the training dataset

Since most existing medical image datasets only include images of one part of the human body, these datasets are not suitable to train a SR network for the reconstruction of images of multiple parts of the human body. Therefore, we collect a medical image dataset named MRIMP. MRIMP includes images from GoogleMR, IXI [55], ADNI [56], KneeMR [57], and LSMRI [58]. As shown in Figure 4, this medical image dataset includes images of many parts of the human body. We investigate the networks trained on three different datasets: DIV2K, MRIMP, and MRIMP+ (i.e., the mixture of MRIMP and 200 images from DIV2K). As shown in Figure 10, despite images of multiple parts included in this medical image dataset, the training loss decreases with the number of the training iteration while validating loss increasing with the number of the training iteration. It means that the network is overfitting in the training dataset. Besides, it can be observed that the training loss on MRIMP is lower than that on MRIMP+. However, the valid loss presents the opposite results, which means the overfitting is much worse on the medical image dataset than on MRIMP+. In other words, adding extra natural images can mitigate the overfitting when medical images are insufficient.

In addition, we test the networks trained on the three datasets using the test datasets. Although a smaller patch size is used in the network trained on MRIMP or MRIMP+, these two network achieves better results than the one trained on DIV2K. The network trained on MRIMP achieves the best results on ADNI100 and OASIS100 while much worse results on MRI13. Nevertheless, both ADNI100 and OASIS only contain MR images of the head, while MRI13 includes MR images of various parts of the human body. This phenomenon also indicates that the generalization of the network trained on MRIMP is worse than the one trained on MRIMP+. Therefore, it is suggested to

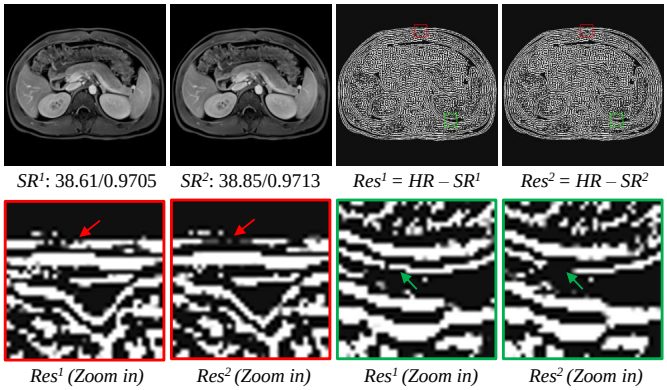


Figure 12: The visual comparison of reconstructed 'abdomen' in MRI13 at different time step. From left to right in the top row, there are the result at the 1st time step SR^1, the result at the 2nd time step SR^2, the contrast enhanced residual image between the HR image and SR^1, and the contrast enhanced residual image between the HR image and SR^2. In the bottom row, we zoom in two areas in corresponding residual images.

train a network with MR images when the medical images are sufficient, but to train a network with a mixture of MR images and natural images when medical images are insufficient.

Table 7: The average PSNR/SSIM of FAWDN's outputs on MRI13 at different time step.

Methods	the 1st step	the 2nd step
PSNR(dB)/SSIM	33.15/0.9090	33.22/0.9098

4.4. Insight into feedback mechanism

To gain insight into the feedback mechanism, we prove that tying the loss function to all time steps is an indispensable condition in our FAWDN for achieving the feedback mechanism. As discussed in Section 3.1, the feedback signal containing the information of the output is a necessary condition to accomplish the feedback mechanism. Therefore, we prove it by two steps as follows:

First, according to the strict definition of the feedback mechanism described in Section 3.1, we can derive that the network will fail the feedback mechanism if the feedback signal routed to the input does not contain the information of the output image.

Second, it cannot be guaranteed that the feedback signal (i.e., the hidden state in our FAWDN) contains the information of the output image if the loss function only tied to the last time step. Suppose the I_{hr} and I_{up} denote the ground truth and the bilinear upsampled image of the input image, respectively. The hidden state h^t would contain the information of the output image only when it can be used to reconstruct the approximate residual image (i.e., $I_{hr} - I_{up}$). In other words, the hidden state has to meet the equation:

$$h^t = f_h([x_t, h_{t-1}]) \quad s.t. \quad f_o(h^t) \approx I_{hr} - I_{up} \quad (1 \le t \le T) \quad (10)$$

where $f_h(\cdot)$ and $f_o(\cdot)$ represent the function of the hidden unit and the output unit, respectively. T is the total number of time step. To make the h_t contain the information of the output image, we tie a loss function to each output (i.e, $f_o(h^t) + I_{up}$) of the network as follows:

$$L(h^t, I_{hr}) = \left\| (f_o(h^t) + I_{up}) - I_{hr} \right\|_1 \quad (11)$$

However, if the loss function only tied to the last time step, it can only ensure $I^T_{sr} \approx I_{hr} - I_{up}$ but cannot ensure $I^t_{sr} \approx I_{hr} - I_{up}$ when $\{1 \le t \le (T-1)\}$. Therefore, it is not guaranteed that all hidden states h^t contain the information of the output image in the network only tied a loss function to the last time step.

As shown in Figure 11, it can be observed that the output residual images $I^1_{1,res}$ of the output unit at the 1st time step is quite different from the residual (i.e., $I_{hr} - I_{up}$) between the ground truth image and the bilinear upsampled image. Besides, checkerboards effects appears in the most area of $I^1_{1,res}$. It looks more like a feature map outputted by convolutional layer rather than a residual image.

In addition to above proof, to verify our FAWDN can obtain a clearer HR image step by step, we list the results of the 1st time step and the 2nd time step. As listed in Table 7, we can find that the result of the 2nd time step are obviously better than that of the 1st time step. Besides, we conduct a visual comparison of outputs of different time steps. Because the difference between the output image of the 1st time step and the 2nd time step is small, we compare the residual images Res^t instead of the original output image. The residual images Res^t is the difference between the ground truth and the output of the t-th time step, which can be obtained by:

$$Res^t = I_{hr} - (f_o(h^t) + I_{up}), \quad (12)$$

where Res^t denotes the residual image. As shown in Figure 12, we can see that Res^1 has more values than Res^2. In other words, compared with the output of the 2nd time step, the output of the 1st time step is further away from the ground truth. Therefore, the output image of the 2nd time step is clearer to the ground truth than that of the 1st time step.

4.5. Comparison with state-of-the-art methods

To validate the effectiveness of our FAWDN, we compare FAWDN with 14 state-of-the-art SR methods. The comparative methods include one traditional medical image super-resolution method LRTV [5], ten distortion-oriented CNN based methods [12, 13, 14, 15, 20, 60, 31, 62], and three GAN-model-based methods [18, 38, 39]. All results of the comparison methods are obtained by public codes of corresponding authors. Since we focus on super-resolving 2D medical images, none of 3D medical image SR methods is selected for comparison.

The quantitative results of different methods on test datasets are listed in Table 8. Our FAWDN+ achieves better results than the comparative methods on all task of ADNI100. Besides, our FAWDN and FAWDN+ outperform other methods except EDSR in other test datasets. Although EDSR achieves better results than our FAWDN+ on some datasets, its parameters and runtime are 613% and 282% (see Section 4.6) of our FAWDN+, respectively. In addition, the LR size used to train

Table 8: The test results of different methods. FAWDN+ means the proposed network trained with MRIMP+. FAWDN++ represents the network trained only with MRIMP. The best result shown in **red**, and the second best result shown in blue. '-/-' in compared methods represents that the corresponding models are not tested because the models are unavailable.

Datasets	MRI13			ADNI100			OASIS100		
Upscale	x2	x3	x4	x2	x3	x4	x2	x3	x4
Bicubic	37.95/0.9677	32.70/0.9126	29.90/0.8591	30.74/0.8943	27.95/0.8063	26.37/0.7298	33.54/0.9585	28.97/0.8919	26.38/0.8159
LRTV	40.64/0.9796	34.39/0.9329	31.19/0.8821	31.75/0.9176	28.69/0.8379	27.06/0.7663	35.34/0.9714	30.35/0.9174	27.53/0.8497
SRCNN	40.76/0.9820	35.03/0.9393	31.75/0.8914	32.49/0.9258	29.29/0.8527	27.49/0.7838	35.52/0.9717	31.27/0.9225	28.47/0.8621
FSRCNN	40.90/0.9809	35.35/0.9406	32.17/0.8934	32.72/0.9276	29.48/0.8562	27.66/0.7876	34.47/0.7778	30.91/0.7568	28.49/0.6900
VDSR	41.78/0.9835	35.43/0.9449	32.44/0.9027	33.09/0.9328	29.83/0.8661	28.00/0.8028	36.70/0.9759	31.60/0.9308	28.96/0.8748
DRCN	42.15/0.9838	35.83/0.9449	32.64/0.9034	33.22/0.9336	29.92/0.8672	28.04/0.8029	37.19/0.9779	32.18/0.9371	29.29/0.8830
LapSRN	41.99/0.9840	-/-	32.68/0.9072	32.96/0.9317	-/-	27.99/0.8027	36.98/0.9772	-/-	29.26/0.8837
SRDenseNet	42.97/0.9846	36.25/0.9469	32.97/0.9075	33.33/0.9348	30.06/0.8704	28.18/0.8079	37.69/0.9788	32.64/0.9477	29.65/0.8900
DDSR	41.92/0.9833	35.49/0.9433	32.45/0.9029	33.00/0.9318	29.68/0.8638	27.91/0.8011	36.97/0.9766	31.72/0.9330	29.15/0.8798
EDSR	**43.61/0.9853**	36.71/**0.9484**	**33.31/0.9107**	33.50/0.9359	30.27/0.8735	28.42/0.8136	**38.18**/0.9797	**33.26**/0.9444	30.48/0.9003
SRMD	42.26/0.9841	35.91/0.9456	32.83/0.9060	33.15/0.9335	29.91/0.8676	28.05/0.8044	37.27/0.9780	32.29/0.9385	29.66/0.8884
SRMDNF	42.76/0.9848	36.09/0.9465	32.69/0.9062	33.30/0.9348	30.02/0.8698	28.13/0.8080	37.69/0.9794	32.52/0.9405	29.74/0.8908
SRGAN	-/-	-/-	29.94/0.8406	-/-	-/-	24.63/0.7127	-/-	-/-	27.42/0.8296
ESRGAN	-/-	-/-	29.10/0.8263	-/-	-/-	24.10/0.7020	-/-	-/-	27.06/0.8336
RankSRGAN-PI	-/-	-/-	29.45/0.8254	-/-	-/-	24.42/0.7059	-/-	-/-	26.92/0.8140
FAWDN	43.35/0.9850	36.60/0.9481	33.22/0.9098	33.41/0.9352	30.16/0.8719	28.30/0.8117	37.91/0.9791	33.00/0.9429	30.05/0.8957
FAWDN+	43.59/0.9851	**36.73**/0.9479	33.21/0.9086	**33.87/0.9400**	**30.75/0.8839**	28.81/**0.8259**	38.10/**0.9798**	33.19/**0.9450**	30.38/0.8895
FAWDN++	-/-	-/-	33.13/0.9079	-/-	-/-	**28.87**/0.8228	-/-	-/-	**31.57/0.9075**

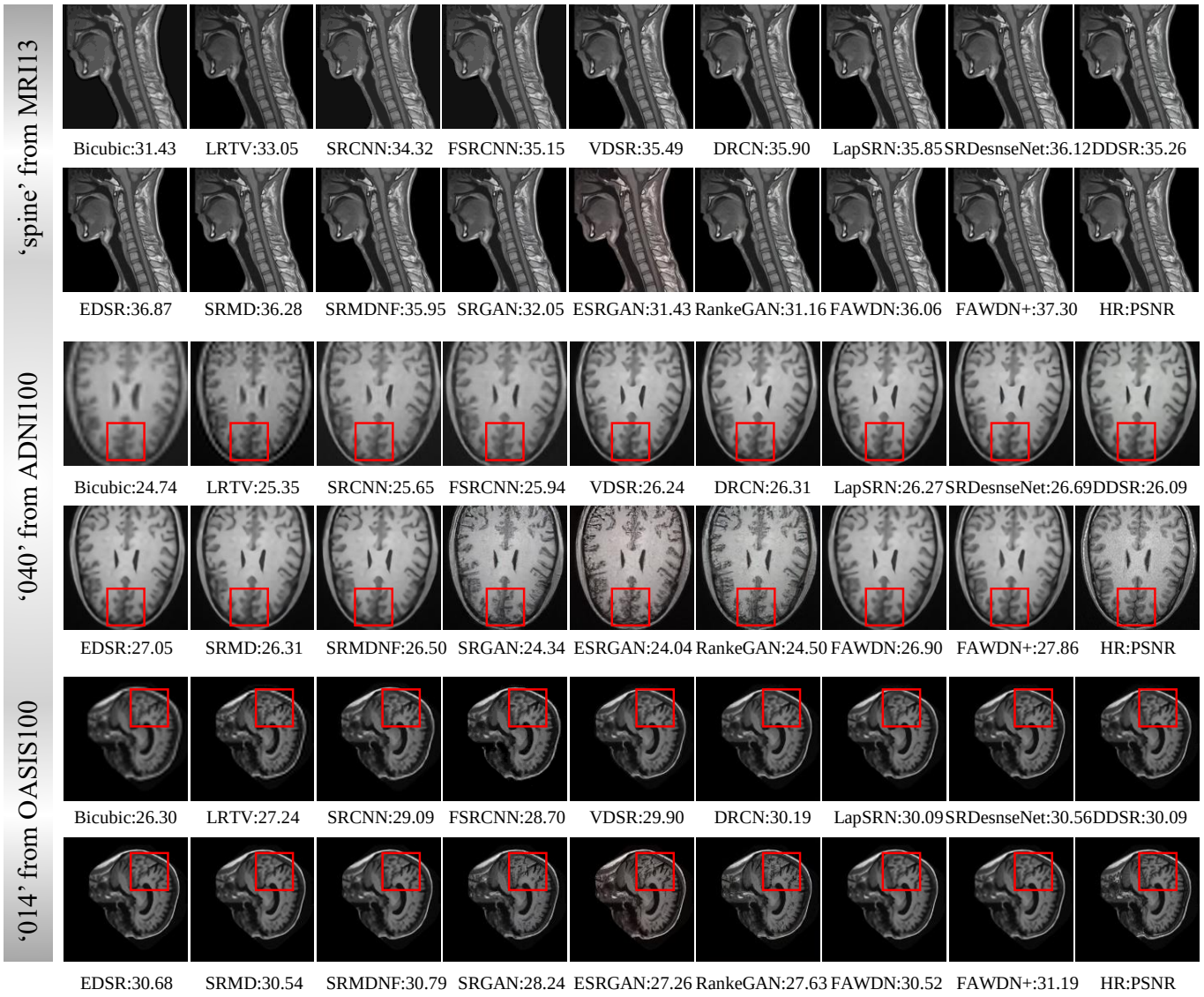


Figure 13: The visual results of different methods on test datasets of x4 upscale task.

Table 9: Average running time of different methods on MRI13 on three test datasets for all upsampling task. The method with * means that it is implemented with Matlab. The one without * means that it is implemented with Python. The best result shown in **red**, and the second best result shown in blue. '-/-' in compared methods represents that the corresponding models are not tested because the models are unavailable.

Methods	Para.	Time(×4)	Time(×3)	Time(×2)
LRTV*	-	25.4162	21.58	13.97
SRCNN*	7K	6.8366	6.2943	6.3318
FSRCNN*	**2K**	2.2574	3.3475	6.8299
VDSR*	0.66M	0.1139	0.1022	0.1489
DRCN*	1.77M	1.5899	1.5436	1.5466
LapSRN*	0.81M	0.1206	-	0.1343
SRDenseNet	7.16M	0.0169	0.0556	0.0551
DDSR	2.02M	0.0125	0.0483	0.0495
EDSR	43.7M	0.2035	0.2457	0.3723
SRMD*	1.50M	0.0165	**0.0186**	**0.0383**
SRMDNF*	1.50M	**0.0119**	0.0187	0.0384
SRGAN	1.52M	0.0156	-	-
ESRGAN	16.7M	0.0431	-	-
RankSRGAN-PI	1.55M	0.0123	-	-
FAWDN+	7.17M	0.0722	0.1497	0.2819

Table 10: The weighted average performance on the three test sets. The best result shown in **red**, and the second best result shown in blue. '-/-' in compared methods represents that the corresponding models are not tested because the models are unavailable.

Method	×2	×3	×4
Bicubic	32.49/0.9589	28.72/0.8964	26.59/0.8309
LRTV	33.98/0.9720	29.82/0.9198	27.53/0.8598
SRCNN	34.42/0.9737	30.57/0.9261	28.21/0.8711
FSRCNN	34.04/0.8823	30.51/0.8492	28.32/0.7914
VDSR	35.32/0.9768	31.00/0.9335	28.72/0.8835
DRCN	35.63/0.9780	31.34/0.9365	28.91/0.8877
LapSRN	35.40/0.9776	-/-	28.87/0.8898
SRDenseNet	35.97/0.9788	31.65/0.9426	29.16/0.8932
DDSR	35.41/0.9770	30.99/0.9336	28.77/0.8858
EDSR	36.31/0.9797	32.07/0.9420	29.69/0.8999
SRMD	35.64/0.9781	31.39/0.9375	29.10/0.8915
SRMDNF	35.94/0.9792	31.56/0.9390	29.16/0.8930
SRGAN	-/-	-/-	26.26/0.8276
ESRGAN	-/-	-/-	25.79/0.8221
RankSRGAN-PI	-/-	-/-	25.90/0.8128
FAWDN	36.13/0.9792	31.89/0.9410	29.42/0.8972
FAWDN+	**36.45/0.9799**	**32.26/0.9426**	29.82/0.8946
FAWDN++	-/-	-/-	**30.40/0.9025**

our FAWDN+ is 30×30, which is much smaller than its LR size of 48 × 48. Compared with EDSR, the proposed network can obtain competitive performance in reconstructed image quality, while having superiority over its computational complexity and parameters. As for GAN model based methods, it can be observed that all the GAN model-based achieve low PSNR and SSIM. Overall, benefiting from the feedback mechanism and the adaptively weighted dense link (see Section 4.1), our FAWDN and FAWDN+ are capable of obtaining a more trusted HR medical image with fewer parameters compared with other methods.

Moreover, we conduct a visual comparison as shown in Figure 13. In the red box, we can discover the textures and details in the result images of GAN model based methods look unnatural and are inconsistent with that in the ground truth. Although DDSR, SRDenseNet, EDSR and SRMDNF get natural results, their edges are not clear as ours, as shown in the red box of the '040' from ADNI100 in Figure 13 . Contrarily, benefiting from the feedback mechanism, our FAWDN can recover the textures and details by correcting the error produced in the previous time step (see Section 4.4). Furthermore, the recovered details and textures are close to those of the ground truth. In other words, our methods can obtain a more trusted HR medical image than the comparison methods.

4.6. Analysis of model efficiency and performance

To investigate efficiency of these methods, we analyzed the model size and inference time of different methods for image SR. Since the number of parameters and the computing complexity of FAWDN, FAWDN+ and FAWDN++ are the same, we only test the running time of FAWDN+. As shown in Table 9, although the parameters of SRCNN, FSRCNN, VDSR, and LapSRN are small, their average performance (see Table 10) on the three datasets is much worse than networks with much more parameters. Compared with SRDenseNet, our FAWDN and FAWDN+ obtain much better results while keeping similar parameters with SRDenseNet. Although EDSR achieves better results than our FAWDN+ on MRI13 and OASIS100. However, its parameters and running time for ×4 task are 613% and 282% of our network. Besides, as shown in Table 8, the results obtained by EDSR on MRI13 and OASIS are only a little higher than FAWDN+, while the results obtained by FAWDN+ are much higher than EDSR. The number of images in MRI13 is also much fewer than the other test sets. To compare EDSR and FAWDN in a global manner, we calculate the weighted average result of all test sets on all upsampling tasks. The weighted average results are obtained by

$$\begin{aligned} PSNR_w &= \frac{N_m * PSNR_m + N_a * PSNR_a + N_o * PSNR_o}{N_m + N_a + N_s}, \\ SSIM_w &= \frac{N_m * SSIM_m + N_a * SSIM_a + N_o * SSIM_o}{N_m + N_a + N_s}, \end{aligned} \tag{13}$$

where N_m, N_a, and N_o are the numbers of images in MRI13, ADNI100, and OASIS100, respectively. $PSNR_m/SSIM_m$, $PSNR_a/SSIM_a$, and $PSNR_o/SSIM_o$ are the values of PSNR/SSIM on MRI13, ADNI100, and OASIS100, respectively. The weighted average results of all methods are listed in Table 10. From Table 10, we can observe that the proposed method outperforms other methods on all tests. Therefore, overall, the proposed method obtains better result than the compared methods.

5. Conclusion

In this paper, we proposed a trusted medical image SR method denoted as FAWDN. To achieve the strict feedback mechanism, the hidden state is transmitted to the input state.

Different from the feedback mechanism in a top-down manner, we force the hidden state to contain the information of the output image by a loss function and fed low-level features into the hidden unit of FAWDN every time step. So, the proposed FAWDN met the condition of the strict feedback mechanism. In addition, we presented AWDB to adaptively select informative features outputted from previous convolutional layers. The study of the feedback mechanism demonstrated that the network with the strict feedback mechanism outperformed the one without the strict feedback mechanism in both performance and convergence speed. Experimental results on adaptive weighting groups also indicated that our AWDB could obtain better performance than original dense block. Overall, proved by comparative experiments, the proposed FAWDN could obtain more clear and trusted HR medical images than the state-of-the-art methods. In further research, we will extend the proposed method to 3 dimension cases. Besides, SR networks for various degradation models will be also studied, such as K-space truncation, additive noise, and blur.

Acknowledgments

This work is sponsored by Key Research and Development Project of Science and Technology Commission Foundation of Sichuan Province (grant no.2018FZ0036) and the National Natural Science Foundation of China (grant no. 61711540303 and 61701327). We appreciate the comments from the reviewers and editors, which help us improve the paper a lot. We also appreciate the help from our colleague Wenchi Zhang who help us in the experiments and the illustrations.

References

[1] E. Carmi, S. Liu, N. Alon, A. Fiat, D. Fiat, Resolution ehancement in mri, Magn. Reson. Imaging 24 (2) (2006) 133–154.
[2] D. Gottlieb, C.-W. Shu, On the gibbs phenomenon and its resolution, SIAM review 39 (4) (1997) 644–668.
[3] T. M. Lehmann, C. Gonner, K. Spitzer, Addendum: B-spline interpolation in medical image processing, IEEE T. Med. Imaging 20 (7) (2001) 660–665.
[4] J. V. Manjón, P. Coupé, A. Buades, V. Fonov, D. L. Collins, M. Robles, Non-local mri upsampling, Med. Image Anal. 14 (6) (2010) 784–792.
[5] F. Shi, J. Cheng, L. Wang, P.-T. Yap, D. Shen, Lrtv: Mr image super-resolution with low-rank and total variation regularizations, IEEE T. Med. Imaging 34 (12) (2015) 2459–2466.
[6] J. Yang, J. Wright, T. S. Huang, Y. Ma, Image super-resolution via sparse representation, TIP (2010).
[7] X. Lu, Z. Huang, Y. Yuan, Mr image super-resolution via manifold regularized sparse learning, Neurocomputing 162 (2015) 96–104.
[8] F. Rousseau, A. D. N. Initiative, et al., A non-local approach for image super-resolution using intermodality priors, Med. Image Anal. 14 (4) (2010) 594–605.
[9] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: CVPR, 2015.
[10] N. Kulkarni, P. Nagesh, R. Gowda, B. Li, Understanding compressive sensing and sparse representation-based super-resolution, IEEE T. Circ. Syst. Vid. 22 (5) (2011) 778–789.
[11] S. Schulter, C. Leistner, H. Bischof, Fast and accurate image upscaling with super-resolution forests, in: CVPR, 2015, pp. 3791–3799.
[12] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE TPAMI 38 (2) (2016) 295–307.
[13] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: CVPR, 2016.
[14] J. Kim, J. K. Lee, K. M. Lee, Deeply-recursive convolutional network for image super-resolution, in: CVPR, 2016.
[15] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: ICCV, 2017.
[16] Y. S. Tai, J. X. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: CVPR, 2017, pp. 2790–2798.
[17] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: ECCV, Springer, 2016, pp. 694–711.
[18] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: CVPR, 2017, pp. 4681–4690.
[19] C. Pham, A. Ducournau, R. Fablet, F. Rousseau, Brain mri super-resolution using deep 3d convolutional networks, in: Proc. IEEE Int. Symp. Biomed. Imag. (ISBI 2017), 2017, pp. 197–200. doi:10.1109/ISBI.2017.7950500.
[20] S. Wei, W. Wu, G. Jeon, A. Ahmad, X. Yang, Improving resolution of medical images with deep dense convolutional neural network, Comp. Pra. Exp. e5084.
[21] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, W. Wu, Feedback network for image super-resolution, in: CVPR, 2019.
[22] G. Huang, Z. Liu, V. D. M. Laurens, K. Q. Weinberger, Densely connected convolutional networks, in: CVPR, 2017.
[23] A. R. Zamir, T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, S. Savarese, Feedback networks, in: CVPR, 2017.
[24] J. Carreira, P. Agrawal, K. Fragkiadaki, J. Malik, Human pose estimation with iterative error feedback, in: CVPR, 2016, pp. 4733–4742.
[25] D. B. Sam, R. V. Babu, Top-down feedback for crowd counting convolutional neural network, in: AAAI, 2018.
[26] X. Jin, Y. Chen, Z. Jie, J. Feng, S. Yan, Multi-path feedback recurrent neural networks for scene parsing, in: AAAI, 2017.
[27] Q. Li, Z. Li, L. Lu, G. Jeon, K. Liu, X. Yang, Gated multiple feedback network for image super-resolution, in: BMVC, 2019.
[28] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: CVPR, 2016, pp. 1874–1883.
[29] C. Dong, C. C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: ECCV, 2016, pp. 391–407.
[30] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: ICCV, 2017.
[31] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: CVPR, 2018.
[32] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: CVPR, 2018.
[33] M. Haris, G. Shakhnarovich, N. Ukita, Deep back-projection networks for super-resolution, in: CVPR, 2018.
[34] Z. Hui, X. Wang, X. Gao, Fast and accurate single image super-resolution via information distillation network, in: CVPR, 2018.
[35] W. Han, S. Chang, D. Liu, M. Yu, M. J. Witbrock, T. S. Huang, Image super-resolution via dual-state recurrent networks, in: CVPR, 2018, pp. 1654–1663.
[36] M. S. Sajjadi, B. Scholkopf, M. Hirsch, Enhancenet: Single image super-resolution through automated texture synthesis, in: CVPR, 2017, pp. 4491–4500.
[37] X. Wang, K. Yu, C. Dong, C. Change Loy, Recovering realistic texture in image super-resolution by deep spatial feature transform, in: CVPR, 2018, pp. 606–615.
[38] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C. C. Loy, Esrgan: Enhanced super-resolution generative adversarial networks, in: CVPRW, 2018, pp. 63–79.
[39] W. Zhang, Y. Liu, C. Dong, Y. Qiao, Ranksrgan: Generative adversarial networks with ranker for image super-resolution, in: ICCV, 2019.
[40] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, L. Zelnik-Manor, The 2018 pirm challenge on perceptual image super-resolution, in: ECCV, 2018.
[41] A. Mittal, R. Soundararajan, A. C. Bovik, Making a "completely blind" image quality analyzer, IEEE Signal Process. Lett. 20 (3) (2012) 209–212.
[42] C. Ma, C.-Y. Yang, X. Yang, M.-H. Yang, Learning a no-reference quality metric for single-image super-resolution, Comput. Vis. Image Und. 158 (2017) 1–16.

[43] P. Shamsolmoali, M. Zareapoor, R. Wang, D. K. Jain, J. Yang, G-ganisr: gradual generative adversarial network for image super resolution, Neurocomputing 366 (2019) 140–153.
[44] M. Zareapoor, H. Zhou, J. Yang, Perceptual image quality using dual generative adversarial network, Neural Comput. Appl. (2019) 1–11.
[45] M. Irani, S. Peleg, Improving resolution by image registration, CVGIP: Graphi. Model. Image Process. 53 (3) (1991) 231–239.
[46] H. Stark, P. Oskoui, High-resolution image recovery from image-plane arrays, using convex projections, J. Opt. Soc. Am. 6 (11) (1989) 1715–1726.
[47] F. Rousseau, Brain hallucination, in: ECCV, 2008, pp. 497–508.
[48] Y. Wang, J. Qiao, J. Li, P. Fu, S. Chu, J. F. Roddick, Sparse representation-based mri super-resolution reconstruction, Measurement 47 (2014) 946–953.
[49] O. Oktay, W. Bai, M. J. Lee, R. Guerrero, K. Kamnitsas, J. Caballero, A. De Marvao, S. A. Cook, D. P. O. Regan, D. Rueckert, Multi-input cardiac image super-resolution using convolutional neural networks, Med. Image Anal. (2016) 246–254.
[50] A. Rueda, N. Malpica, E. Romero, Single-image super-resolution of brain mr images using overcomplete dictionaries, Med. Image Anal. 17 (1) (2013) 113–132.
[51] S. Roohi, J. Zamani, M. Noorhosseini, M. Rahmati, Super-resolution mri images using compressive sensing, in: ICEE, 2012, pp. 1618–1622.
[52] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: ICML, 2010, pp. 807–814.
[53] E. Agustsson, R. Timofte, Ntire 2017 challenge on single image super-resolution: Dataset and study, in: CVPRW, 2017, pp. 126–135.
[54] M. Bevilacqua, A. Roumy, C. Guillemot, M. L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: BMVC, 2012.
[55] Ixi-dataset, http://brain-development.org/ixi-dataset/.
[56] Adni, http://adni.loni.usc.edu/.
[57] N. Bien, P. Rajpurkar, R. L. Ball, J. Irvin, A. Park, E. Jones, M. Bereket, B. N. Patel, K. W. Yeom, K. Shpanskaya, et al., Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mrnet, PLoS Med. 15 (11) (2018) e1002699.
[58] S. Sud, K. Ala, Al, N. Friska, M. Hira, A. Nunik, A.-R. Wasfi, B. Mohammad, A.-J. Mohammed, Lumbar spine mri dataset, DOI:http://dx.doi.org/10.17632/k57fr854j2.2#file-1bc6b195-27fc-4ac4-ae43-aaa6bf386912.
[59] Oasis, https://www.oasis-brains.org/.
[60] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep laplacian pyramid networks for fast and accurate super-resolution, in: CVPR, 2017.
[61] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2014.
[62] B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, Enhanced deep residual networks for single image super-resolution, in: CVPRW, 2017.