

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

# Automatic Windowing for MRI with Convolutional Neural Network

XIAOLE. ZHAO<sup>1</sup>, HANGFEI. LIU<sup>1</sup>, GAIYAN. ZHU<sup>1</sup>, TAO. ZHANG<sup>1,2,3</sup>, XUEMING. ZHOU<sup>1,2,3</sup>

<sup>1</sup>School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

<sup>2</sup>High Field Magnetic Resonance Brain Imaging Laboratory of Sichuan, Chengdu, Sichuan 611731, China

<sup>3</sup>Key Laboratory for NeuroInformation of Ministry of Education, Chengdu, Sichuan 611731, China

Corresponding author: Xiaole. Zhao (e-mail: zxlation@foxmail.com).

This work is supported in part by the National Key Research and Development Program of China under grants 2016YFC0100800 and 2016YFC0100801.

**ABSTRACT** This paper presents a fast, high-precision and fully automatic windowing method based on deep convolutional neural network (CNN) for magnetic resonance imaging (MRI). Displaying a magnetic resonance (MR) image with a data depth of 12/16 bits on regular 8-bit monitors usually needs a windowing process to remap the full range of pixel intensity to a subrange. However, adaptively and automatically adjusting the windowing parameters of MR images under various viewing conditions is a challenging problem in medical image processing due to the low contrast and high grayscale range. We present a novel method based on deep CNNs to estimate the windowing parameters that can match the adjustment of human experts precisely and quickly. The network acts as a typical end-to-end mapping function that takes the raw pixels of the MR images as input, and directly outputs the corresponding estimation of the optimal windowing parameters. To speed up the inference, we utilize a space-to-depth (STD) conversion to reduce the spatial resolution of input images and thus the computing burden of the inference process. Extensive experiments on the dataset annotated by clinicians show that the proposed method can accurately predict the optimal windowing parameters of a MR image with size of  $1024 \times 1024$  in less than 0.01 seconds. Due to the high effectiveness and efficiency of the proposed method, it is highly applicable for various clinical and research purposes.

**INDEX TERMS** Automatic windowing, convolutional neural network, deep learning, window width and window level, magnetic resonance imaging

## I. INTRODUCTION

In the field of magnetic resonance imaging (MRI), a main problem of visualizing magnetic resonance (MR) images is that MR images with a data depth of 12 bits or 16 bits have a much wider full pixel intensity range than the typical 8 bits that the ordinary diagnostic displays can handle [1]. Despite recent studies have demonstrated that a human observer can perceive up to 900 shades of gray, most current viewing applications only support 256 (i.e.,  $2^8$ ) depth levels [2], [3]. To display any 12-bit or 16-bit MR image on commonly used 8-bit computer monitors, the pixel intensity range of this image needs to be remapped in general. A typical remapping process is the so-called *windowing process* that maps the pixel intensity linearly from a subrange  $[y_c - y_w, y_c + y_w]$  to  $[0, 255]$ , where  $y_c$  and  $y_w$  represent window level (WL) and (a half of) window width (WW) respectively. The display quality of MR images greatly depends on these two window-

ing parameters, and improper adjustment of these parameters will seriously affect the visual effect of the images, and even cause the unnecessary loss of valuable diagnostic information in the images [4].

A simple and direct manner is to manually adjust  $y_c$  and  $y_w$  through human operators. But this is time-consuming and energy-intensive, and the adjustment results vary greatly due to the effects of observation conditions and individual differences between human operators. Thus, if these two parameters can be adjusted automatically, it will be beneficial for both research and clinical purposes. However, automatic adjustment of these two parameters is usually a complicated problem for various objective or/and subjective reasons, such as the spatial distribution of pixel intensity, different image types, individual differences of human experts and varying observation conditions, etc [4]. Therefore, it is difficult for automatic adjustment methods to precisely match the results

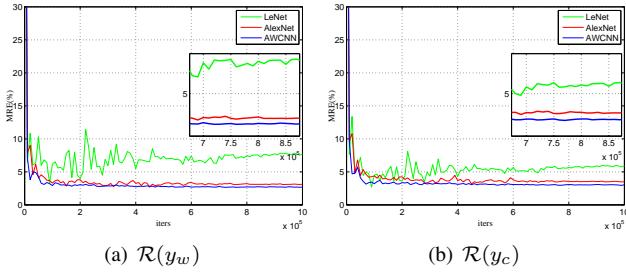


FIGURE 1: Comparison of validation performance between LeNet [5], AlexNet [6] and the proposed AWCNN model. (a) Mean relative error of window width  $\mathcal{R}(y_w)$ . (b) Mean relative error of window level or window center  $\mathcal{R}(y_c)$ .

reported by clinical experts. Nevertheless, if automatic adjustment is almost as good as that set by a human expert and only minor readjustments would be required, then the display in MR systems would be greatly simplified [7].

At present, there have been several techniques developed for automatically adjusting these windowing parameters of medical images, e.g., Gabor-relying adjustment of image levels (GRAIL) [3], the comprehensive hierarchical neural networks (HNN)-based automatic and robust adjustment of the display window [4], the neural network based automatic adjustment of display window [7] and the global contrast stretching (GCS) [9], etc. However, most these traditional methods are based on histogram information of MR images and its derivatives to some extent, with the use of manually extracted features. Consequently, these meaningful and task-related features are specially selected and carefully designed by human experts based on their domain knowledge, making the quality of these features heavily dependent on the domain knowledge and the relevant experience of human experts. This may cause the poor generalization of the method and not suitable for varying MR images. Besides, methods based on specific prior knowledge (e.g., histograms or spatial distributions) or traditional shallow learning (e.g., HNN) have very limited representational ability and cannot well reflect the nonlinear relationship between MR images and their optimal window parameters.

Inspired by the great success of deep learning techniques [10], especially deep convolutional neural networks (CNNs) [5], we explore an end-to-end deep model to predicate the nonlinear relationship between MR images and their optimal windowing parameters, termed as automatic windowing convolutional neural network (AWCNN). Given an image  $\mathbf{x} \in \mathbb{R}^{H \times W}$ , where  $H$  and  $W$  represent the height and width of  $\mathbf{x}$ , automatically windowing the pixel intensity of  $\mathbf{x}$  aims to finding a mapping function  $f(\mathbf{x})$  that describes the relationship between  $\mathbf{x}$  and its windowing parameters  $\mathbf{y} = [y_c, y_w]^T$ :

$$\mathbf{y} = f(\mathbf{x}; \theta), \quad (1)$$

where  $\theta$  is the set of model parameters. Once the parameters of the model are determined, predicting the windowing parameters is easy and forthright. Then, remapping process is

simply implemented through linearly stretching or shrinking the subrange of the full pixel intensity range  $[y_c - y_w, y_c + y_w]$  to the display range  $[0, 255]$ :

$$\hat{x} = \begin{cases} 0, & x < a \\ 255 \times \frac{x-a}{b-a}, & a \leq x \leq b \\ 255, & x > b \end{cases}, \quad (2)$$

where  $a = y_c - y_w$  and  $b = y_c + y_w$  are the lower and upper bounds of the subrange, respectively.  $x$  is a pixel in the MR image  $\mathbf{x}$  and  $\hat{x}$  is the remapping of  $x$  that forms the image  $\hat{\mathbf{x}}$  to be displayed. It is worth noting that  $a$  does not have to be greater than or equal to 0, and  $b$  does not have to be smaller than or equal to the upper bound of the full pixel intensity range  $2^{12} - 1$  or  $2^{16} - 1$ .

A remarkable difference between the proposed method and the previous methods is that it does not require to design and extract representational features manually due to the ability of CNN models to autonomously extract features from the original data. It takes the raw pixels of the MR image as input and generates the desired windowing parameters  $\mathbf{y}$ . Profiting from the powerful representational capacity of deep CNN models, the proposed model is able to accurately match the adjustments of human experts (Fig.1), greatly simplifying the operation of displaying MR images on common monitors. In addition, we also utilize a space-to-depth (STD) conversion (Fig.2(a)) before feeding the image to the model to compress the inference of the model into low resolution (LR) image space, which significantly reduces computation and memory consumption. Interestingly, the STD layer also improves the prediction accuracy to a certain extent although it seems counterintuitive. Consequently, the proposed model is highly efficient and can process a MR image of size  $1024 \times 1024$  in less than 0.01 seconds with the support of a regular NVIDIA GeForce GTX 1080 Ti GPU (Table 1), implying its valuable practicability.

The remaining of this paper is organized as follows. We first present some previous contributions related the present work in section II, and then describe the details of the proposed method in section III. Next, section IV gives the quantitative and qualitative results of experimental analyses. Finally, we discuss some related topics and future work in section V and conclude the whole work in section VI.

## II. RELATED WORK

### A. CONVOLUTIONAL NEURAL NETWORK

One of the most successful models for digital image analysis and processing is deep convolutional neural network (CNN) that can date back to the late 1980s [11]. The first successful real-world application of CNN is introduced by LeCun *et al.* [5] for hand-written digit recognition (LeNet). However, the usage of CNNs did not gather momentum until the prominent contribution of Krizhevsky *et al.* [6] to the ImageNet large scale visual recognition challenge (LSVRC) in 2012 [12]. The network explored by them is the famous AlexNet that won the competition by a large margin.

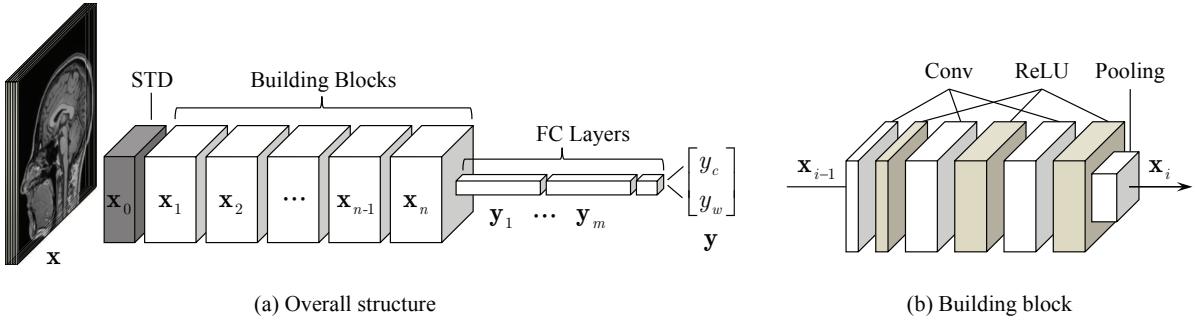


FIGURE 2: The schema of the proposed AWCNN and a building block. (a) The AWCNN consists of  $n$  convolutional building blocks and  $m$  fully connected (FC) layers. A space-to-depth (STD) layer is used to reduce the spatial dimension of the image before it is fed into the model. (b) Each convolutional building blocks consists of the repeated application of three  $3 \times 3$  convolutional layers (zero-padded), each followed by a rectified linear unit (ReLU) [8]. A  $2 \times 2$  max pooling layer with a stride of 2 is used for downsampling. The number of feature maps in each block is doubled by the first convolutional layer.

In the next few years, with the emergence of advanced computing devices and high-quality large-scale datasets (e.g., ImageNet [12], DIV2K [13] etc.), as well as the development of related theories and techniques for efficiently training deep networks, deep CNNs have shown an explosive popularity in many fields and greatly promoted the best state of various computer vision tasks, such as image classification [6], [14]–[17], object detection [18]–[20], face recognition [21]–[23], image generation (image super-resolution [24]–[31], image denoising [32]–[34] etc.). However, this is attributed to not only the outstanding performance of deep CNNs, but also the convenience to extract features from data automatically. The proposed automatic windowing method for MR images also benefits from these advantages to a large extent.

### B. AUTOMATIC WINDOWING FOR MRI

Most of previous methods for auto adjustment of windowing parameters are very restricted to certain type of MR images or operate very differently from human adjustment. They usually perform the calculation based on some prior knowledge about MR images, such as perceptual metric [3], graylevel histogram [7] and MR image type [35] etc. These methods apply finite information to estimate the windowing parameters according to hand-crafted features and specific calculation, resulting in limited or even impractical applications. Some more advanced methods based on multi-layer perceptrons (MLPs) were also developed. An early work is introduced by Ohhashi *et al.* [7] which utilized a three-layer HNN to model the relationship between the input MR image  $\mathbf{x}$  and its windowing parameters  $[y_c, y_w]^T$ . However, they simply extracted a group of six features (combined with sine and cosine transforms) from the graylevel histogram of  $\mathbf{x}$  for model training. Further, Lai and Fang [4] proposed a comprehensive HNNs based algorithm for automatic and robust adjustment of display parameters. They combined wavelet histogram and spatial statistical information of the MR image for feature generation, which overcomes the problem of using pure histogram information only. They also presented an

adaptive display window adjustment system with the online adaptation capability [36].

The features utilized by these methods need to be carefully designed and manually extracted from the input data. But they are usually not robust enough and cannot adequately represent the discriminative information of the raw data [37]. Moreover, the traditional shallow models have very limited representational ability and can hardly match the adjustment by human experts with high accuracy. On the other hand, the proposed method optimizes an end-to-end mapping between the MR image  $\mathbf{x}$  and its optimal windowing parameters  $\mathbf{y} = [y_c, y_w]^T$ , without manual feature extraction. Besides, it is capable of estimating more accurate results of windowing parameters due to the excellent performance of CNNs. Further, the STD conversion makes it not only a quantitatively superior method, but also a faster and practically useful one.

## III. PROPOSED METHOD

### A. OVERALL NETWORK STRUCTURE

The target of automatic windowing for MRI is to estimate the desired windowing parameters  $\mathbf{y} = [y_c, y_w]^T$  of the input MR image  $\mathbf{x}$ , which match the adjustment of human experts as precisely as possible. The complicated nonlinear relationship between  $\mathbf{x}$  and  $\mathbf{y}$  is modeled as a deep CNN architecture that forms an end-to-end mapping from  $\mathbf{x}$  to  $\mathbf{y}$ , i.e.,  $\mathbf{y} = f(\mathbf{x}; \theta)$ . The overall structure of the proposed AWCNN model is simple and outlined in Fig.2. It consists of three parts: (1) a STD layer to reduce the spatial resolution of the original input images; (2) a nonlinear mapping composed of several building blocks to build the hierarchical features; and (3) a linear transform to convert these features into the final estimation, which consists of a series of fully connected (FC) layers.

Given an input image  $\mathbf{x}$ , it is first converted into a LR image  $\mathbf{x}_0$  through the STD layer:

$$\mathbf{x}_0 = \mathcal{F}_{\text{STD}}(\mathbf{x}, r), \quad (3)$$

where  $\mathcal{F}_{\text{STD}}(\cdot)$  denotes the function corresponding to the STD layer, and  $r$  is the rate of the STD conversion. The main

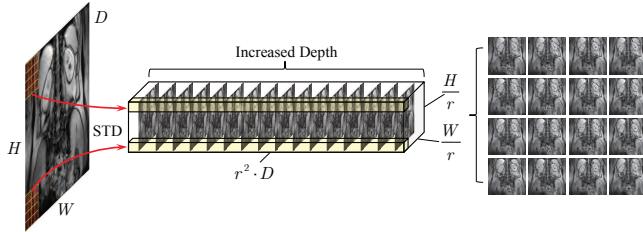


FIGURE 3: The illustration of a space-to-depth (STD) layer. The original image is converted into a small image with  $r^2 \cdot D$  channels ( $D = 1$  for 2D MR images). The pixels in each  $r \times r \times D$  patch of  $\mathbf{x}$  are rearranged into the shape of  $1 \times 1 \times (r^2 \cdot D)$ . The figure shows the case of  $r = 4$ .

purpose of the STD layer is to reduce the spatial resolution of  $\mathbf{x}$  so as to improve the computational efficiency of model inference. Then, several building blocks are used to extract the hierarchical features from  $\mathbf{x}_0$ :

$$\mathbf{x}_i = \mathcal{F}_i(\mathbf{x}_{i-1}), \quad i = 1, 2, \dots, n, \quad (4)$$

where  $n$  is the number of building blocks, and  $\mathcal{F}_i(\cdot)$  is the function of the  $i$ -th building block. Iteratively, the output of the entire nonlinear mapping from the spatially converted image  $\mathbf{x}_0$  can be obtained as follows:

$$\mathbf{x}_n = \mathcal{F}_n(\mathbf{x}_{n-1}) = \mathcal{F}_n(\mathcal{F}_{n-1}(\dots(\mathcal{F}_1(\mathbf{x}_0))\dots)), \quad (5)$$

where  $\mathbf{x}_n = \mathbf{y}_0$  is also the input of the linear transform that consists of  $m$  FC layers. Denote the linear function of the  $j$ -th FC layer as  $\tilde{\mathcal{F}}_j(\cdot)$ , i.e.,  $\mathbf{y}_j = \tilde{\mathcal{F}}_j(\mathbf{y}_{j-1})$ ,  $j = 1, 2, \dots, m$ , we can also get the iterative formula for the linear transform in a similar way:

$$\mathbf{y}_m = \tilde{\mathcal{F}}_m(\mathbf{y}_{m-1}) = \tilde{\mathcal{F}}_m(\tilde{\mathcal{F}}_{m-1}(\dots(\tilde{\mathcal{F}}_1(\mathbf{y}_0))\dots)). \quad (6)$$

Each FC layer in the linear transform phase can be equipped with different numbers of neurons, but there are only 2 neurons in the last FC layer, which are used to map the features to the label space of the MR images, i.e.,  $[y_c, y_w]^T$ .

## B. BUILDING BLOCKS

Each building block in the nonlinear mapping contains three convolutional layers, each followed by a nonlinear rectified linear unit (ReLU) [8] layer. Let  $\mathbf{x}_{i,0} = \mathbf{x}_{i-1}$  denote the input feature map of the  $i$ th building block, the basic Conv + ReLU operations in the block can be represented as:

$$\mathbf{x}_{i,j} = \max \left( \mathbf{w}_{i,j-1} * \mathbf{x}_{i,j-1} + \mathbf{b}_{i,j-1}, 0 \right), \quad j = 1, 2, 3 \quad (7)$$

where  $\mathbf{w}_{i,j}$  and  $\mathbf{b}_{i,j}$  are the convolution kernels and biases of the  $j$ th convolution layer in the  $i$ th block, and “ $*$ ” denotes the convolution operation. Besides, the number of feature maps in each building block is doubled by the first convolutional layer. Therefore, the number of the feature maps in  $\mathbf{x}_{i,j}$  is twice that of  $\mathbf{x}_{i,0} = \mathbf{x}_{i-1}$ .

Next, a  $2 \times 2$  max pooling layer is used to reduce the spatial dimension of feature maps at the end of the block.

$$\mathbf{x}_i = \mathcal{P}(\mathbf{x}_{i,3}), \quad (8)$$

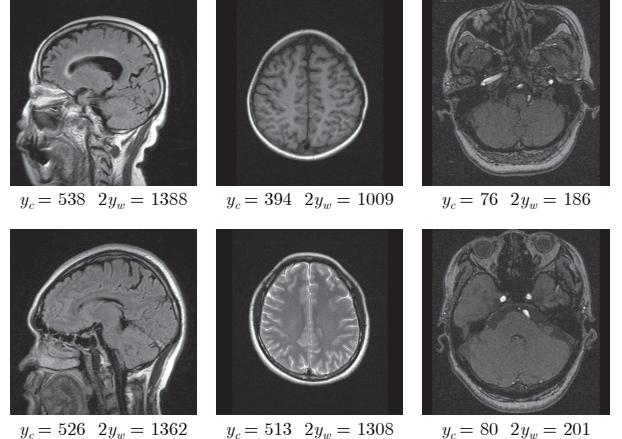


FIGURE 4: Some example images from the dataset. These images are displayed according to the windowing parameters set by clinical experts.

where  $\mathcal{P}(\cdot)$  corresponds to the  $2 \times 2$  max pooling, and  $\mathbf{x}_i$  is the output of the  $i$ th building block. The entire structure of each building block is shown in Fig.2(b). This mapping process of gradually reducing the spatial dimensions and increasing the depth of the hierarchical features is similar to the left part of U-Net [38].

## C. SPACE-TO-DEPTH CONVERSION

To reduce the computational cost and memory occupation of the model inference, and thus improve the practicability of the algorithm, we apply a space-to-depth (STD) conversion to the original image before feeding it into the model, as shown in Fig.3. Given an input MR image  $\mathbf{x} \in \mathbb{R}^{H \times W \times D}$ , where  $D$  indicates the number of channels and usually equals to 1 for 2D MR slices, the STD layer converts  $\mathbf{x}$  into another image with decreased spatial resolution and increased channel number in the following way:

$$[\mathbf{x}]_{H, W, D} \xrightarrow{\mathcal{F}_{\text{STD}}(\cdot)} [\mathbf{x}]_{H/r, W/r, r^2 \cdot D} = \mathbf{x}_0, \quad (9)$$

where  $\mathbf{x}_0 \in \mathbb{R}^{(H/r) \times (W/r) \times (r^2 \cdot D)}$  is the output tensor of the STD layer and  $r$  is the rate of STD conversion. This implies that a small image patch of size  $r \times r \times D$  in  $\mathbf{x}$  is rearranged into a single voxel in  $\mathbf{x}_0$  with the shape of  $1 \times 1 \times (r^2 \cdot D)$ . It is worth noting that this operation seems to prevent the model from dealing with the images of any size because it requires that  $H$  and  $W$  must be divisible by  $r$ . However, we can solve it by simply cropping or zero-padding the image to the expected size that matches the STD conversion.

Although the STD conversion destroys the spatial integrity of the images, the impact on the performance of the model is acceptable in the scenario of automatic windowing parameter predication for MRI. This is because the slight change in the prediction accuracy of the model will not significantly affect the display effect of the images and the readjustment of human experts. Besides, the reduced spatial information is shifted to channel direction rather than totally lost, which

TABLE 1: Quantitative comparison between the models with and without STD layer. The results in columns 2 ~ 5 are evaluated on MR images with size of  $512 \times 512$ . Note that both of them have almost the same model parameters and network depth.

|                     | Test error ( $512 \times 512$ ) |                    |                    |                    | Test efficiency (ms / example) |                  |                  |                    |
|---------------------|---------------------------------|--------------------|--------------------|--------------------|--------------------------------|------------------|------------------|--------------------|
|                     | $\mathcal{R}(y_c)$              | $\mathcal{R}(y_w)$ | $\mathcal{A}(y_c)$ | $\mathcal{A}(y_w)$ | $128 \times 128$               | $256 \times 256$ | $512 \times 512$ | $1024 \times 1024$ |
| wo/ STD ( $r = 0$ ) | 8.93%                           | 8.46%              | 47.58              | 121.24             | 6.57                           | 7.81             | 13.29            | /                  |
| w/ STD ( $r = 4$ )  | 5.77%                           | 5.49%              | 37.89              | 96.79              | 6.06                           | 6.17             | 6.44             | 8.21               |

TABLE 2: Quantitative comparison between the models with different STD conversion rates. The efficiency is measured in milliseconds per example (ms / example).

| STD      | Test Error         |                    |                    |                    | $512 \times 512$ |
|----------|--------------------|--------------------|--------------------|--------------------|------------------|
|          | $\mathcal{R}(y_c)$ | $\mathcal{R}(y_w)$ | $\mathcal{A}(y_c)$ | $\mathcal{A}(y_w)$ |                  |
| $r = 0$  | 8.93%              | 8.46%              | 47.58              | 121.24             | 13.29            |
| $r = 2$  | 10.27%             | 9.65%              | 46.35              | 115.31             | 8.38             |
| $r = 4$  | <b>5.77%</b>       | <b>5.49%</b>       | <b>37.89</b>       | <b>96.79</b>       | <b>6.44</b>      |
| $r = 8$  | 6.76%              | 5.88%              | 45.63              | 114.27             | 6.47             |
| $r = 16$ | 6.93%              | 5.83%              | 42.53              | 107.62             | 6.48             |

TABLE 3: Quantitative comparison between the models with different preprocessing methods. The optimal value of each column is in **bold**, and the second one is underlined.

|         | Preprocessing | $\mathcal{R}(y_c)$ | $\mathcal{R}(y_w)$ | $\mathcal{A}(y_c)$ | $\mathcal{A}(y_w)$ |
|---------|---------------|--------------------|--------------------|--------------------|--------------------|
| $r = 4$ | Bicubic       | 6.12%              | <u>5.74%</u>       | <u>38.75</u>       | 100.22             |
|         | Max Pooling   | <u>5.89%</u>       | 5.89%              | 39.44              | 103.48             |
|         | STD           | <b>5.77%</b>       | <b>5.49%</b>       | <b>37.89</b>       | <b>96.79</b>       |

can make up for the issue to some extent. Most importantly, the STD layer can significantly ease the computational burden and memory consumption, improving the feasibility of deployment in practical applications. We will elaborate on it in section IV.

#### D. TRAINING OBJECTIVE

The target of learning an end-to-end mapping function  $f$  that predicates the optimal windowing parameters  $\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta})$  is to determine the parameters  $\boldsymbol{\theta}$  of the model, where  $\hat{\mathbf{y}}$  is the estimation of the windowing parameters  $\mathbf{y} = [y_c, y_w]^T$  set by human experts. This is usually achieved by minimizing the loss function between  $\hat{\mathbf{y}}$  and  $\mathbf{y}$ . Given a training dataset  $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$ , where  $N$  represents the number of training samples,  $l_1$  loss is applied for training the model:

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \|f(\mathbf{x}^{(i)}; \boldsymbol{\theta}) - \mathbf{y}^{(i)}\|_1. \quad (10)$$

Although the  $l_2$  loss is easier to optimize due to the continuity of the first derivative, the  $l_1$  loss is more robust to outliers and more suitable for the problem of automatically adjusting the display parameters of MR images. This is primarily because the optimal windowing parameters of MR images often vary greatly due to the very wide range of pixel intensity, as well as the quality of annotations of human experts is also uneven due to various subjective or objective factors.

## IV. EXPERIMENTAL RESULTS

### A. DATASET AND ANNOTATION

The dataset contains 15745 real MR images collected by Centauri 1.5T scanner of Alltech Medical Systems Co., LTD.

It is annotated by 17 clinicians from three different hospitals to show the best visualization on a regular monitor according to their domain knowledge and relevant experience. The annotation is executed repeatedly and retrospectively to reduce individual difference. Note that the purpose of data annotation is to make MR images have the best overall visual effect on a regular display, rather than focusing on specific tissue structure or pathology. The MR images have the size of  $512 \times 512$  and are generated from different patients (subjects), body parts, and acquisition sequences, as shown in Fig.4. The entire dataset is randomly divided into 3 subsets for training, validation and testing, which we denote as  $\mathcal{D}$ ,  $\mathcal{V}$  and  $\mathcal{T}$  respectively. The training set includes 12715 images ( $|\mathcal{D}| = 12715$ ) and the testing set has 3000 images ( $|\mathcal{T}| = 3000$ ), and the remaining images are used for validation. The validation set  $\mathcal{V}$  contains only a relatively small number of images ( $|\mathcal{V}| = 30$ ) for quick validation during model training.

### B. IMPLEMENTATION DETAILS

Input images are normalized before feeding into the STD layer. Fig.2 shows the detailed configuration of our AWCNN model. We set  $n = 5$  and  $m = 3$  for the model. The number of channels in building blocks is  $\{32, 64, 128, 256, 512\}$  and that of neurons in the first two FC layers is set to 1024, both of which are followed by a dropout operation with a probability of 0.5. We use TensorFlow 1.7.0 to implement and train the proposed AWCNN model and several compared methods. The Adam optimizer [39] is employed to minimize the loss function by setting  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 10^{-8}$ . For all convolutional layers, the kernel size is set to  $3 \times 3$ . We use a mini-batch size of 16 for model training. The deep learning based models are trained and evaluated with a single NVIDIA GTX 1080 Ti GPU on a Omnisys supercomputing workstation equipped with 64GB memory and two Intel Xeon E5-2630 CPUs (2.20 GHz).

In addition, the network parameters are initialized according to Xavier initialization [40]. As an initial attempt, the learning rate is decayed in a piecewise constant way, i.e., it is initialized as  $10^{-4}$  for all convolutional layers and halved at every 200K iterations.

### C. EVALUATION METRICS

We use two typical metrics to evaluate the performance of the model in this paper: mean relative error (MRE) and mean absolute error (MAE) [4], which we denote as  $\mathcal{R}(\cdot)$  and  $\mathcal{A}(\cdot)$  respectively. Given a MR image  $\mathbf{x}^{(i)}$ , the MRE of  $y_c$  and  $y_w$

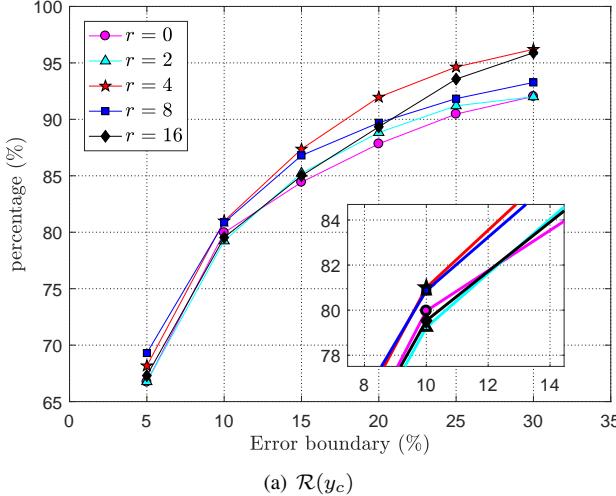
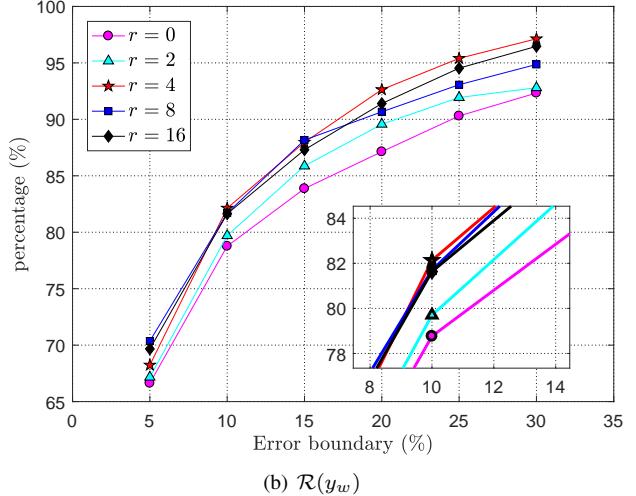
(a)  $\mathcal{R}(y_c)$ (b)  $\mathcal{R}(y_w)$ 

FIGURE 5: Error boundary analysis of the models with different STD conversion rates. The percentages of testing examples in different error ranges are plotted for further illustration on the difference of the model performance.

can be represented as:

$$\mathcal{R}(y) = \frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} \left| \frac{\hat{y}_t - y_t}{y_t} \right| \times 100/\%, \quad (11)$$

where  $y$  represents  $y_c$  or  $y_w$ , and  $\hat{y}_t \in \{\hat{y}_c^{(t)}, \hat{y}_w^{(t)}\}$  denotes an estimate of  $y_t \in \{y_c^{(t)}, y_w^{(t)}\}$ . Correspondingly, the MAE of the model is defined as following:

$$\mathcal{A}(y) = \frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} |\hat{y}_t - y_t|. \quad (12)$$

To further illustrate the superiority of the proposed model over other methods, we also adopt Pearson product-moment correlation coefficient (PPMCC) to evaluate the correlation between model predictions and sample labels.

$$\rho(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\text{Cov}(\hat{\mathbf{y}}, \mathbf{y})}{\sigma_{\hat{\mathbf{y}}} \sigma_{\mathbf{y}}}, \quad (13)$$

where  $\text{Cov}(\hat{\mathbf{y}}, \mathbf{y})$  is the covariance between  $\hat{\mathbf{y}}$  and  $\mathbf{y}$ , and  $\sigma_{\hat{\mathbf{y}}}$  and  $\sigma_{\mathbf{y}}$  are the standard deviation of  $\hat{\mathbf{y}}$  and  $\mathbf{y}$ , respectively.

#### D. MODEL ANALYSIS

In this section, we study several components of the proposed AWCNN model, including the STD conversion and its scale factors, decays of learning rate and loss functions.

##### 1) Space-to-Depth Conversion

To investigate the impact of STD on the performance of the model, we implement a reference model with the same settings as our AWCNN except that it has no STD layer ( $r = 0$ ). Thus, although the reference structure has approximately the same number of parameters and network depth as AWCNN, the feature maps are in higher dimensional space and take more computation and memory occupation. Table 1 collects the related performance indexes of the models, from

which we can observe that AWCNN ( $r = 4$ ) not only has higher efficiency than the reference model ( $r = 0$ ), but also performs better. In fact, the STD layer is also in favor of quick training. It takes about 10 hours to train the AWCNN model ( $r = 4$ ), while training reference model ( $r = 0$ ) takes about 60 hours.

This phenomenon may be due to the fact the small images obtained by the STD conversion layer shares some information with the original input image  $\mathbf{x}$  to a large extent, such as the overall brightness, morphological characteristics etc., which affects the decision-making of human experts on WW and WL adjustment. More importantly, reducing the spatial size of the input image helps the model “see” a larger region of the input image. However, the STD layer is different from the pooling layer in that the pooling operation will lose information of the input, but the STD conversion will not.

##### 2) Patch Size

The STD conversion rate  $r$  is also the size of the image patch where the pixels are converted from space to depth in the original MR image, as shown in Fig.3. We compare the performance of the proposed AWCNN model with different patch sizes in this section. Table 2 shows the performance comparison between these models, including the reference structure described in section IV-D1 ( $r = 0$ ). It can be seen that the model has the lowest testing error and average time consumption when  $r = 4$ . The performance of the model becomes worse when  $r$  is less or greater than 4. The possible reason for this phenomenon is that, on the one hand, the STD layer will destroy the spatial integrity of the input image, thus reducing the prediction accuracy of the model; on the other hand, it will expand the field of vision of the model on the input image and improve the performance of the model. The positive and negative effects of the STD layer reach a good compromise when  $r = 4$ . Another thing worth explaining is that the time consumption seems to be increasing when

TABLE 4: Quantitative comparison between the models trained with different loss functions. The minimum value of each column is in **bold**, and the second one is underlined ( $r = 4$ ).

| losses | $\mathcal{R}(y_c)$ | $\mathcal{R}(y_w)$ | $\mathcal{A}(y_c)$ | $\mathcal{A}(y_w)$ |
|--------|--------------------|--------------------|--------------------|--------------------|
| $l_1$  | <b>5.77%</b>       | <b>5.49%</b>       | <b>37.89</b>       | <b>96.79</b>       |
| $l_2$  | <u>5.87%</u>       | <u>5.53%</u>       | 40.78              | 102.99             |
| Huber  | $\delta = 10^{-1}$ | 6.37%              | 6.18%              | <u>37.94</u>       |
|        | $\delta = 10^{-2}$ | 6.40%              | 6.50%              | <u>97.22</u>       |
|        | $\delta = 10^{-3}$ | 7.18%              | 7.01%              | 38.79              |
|        | $\delta = 10^{-4}$ | 9.77%              | 9.54%              | 38.66              |
|        |                    |                    | 41.70              | 100.98             |
|        |                    |                    |                    | 107.60             |

TABLE 5: Quantitative comparison between the models trained with different learning rate decays. The minimum value of each column is in **bold**, and the second one is underlined.

|     | $\mathcal{R}(y_c)$ | $\mathcal{R}(y_w)$ | $\mathcal{A}(y_c)$ | $\mathcal{A}(y_w)$ |
|-----|--------------------|--------------------|--------------------|--------------------|
| ITD | 6.30%              | 5.95%              | 38.64              | 100.97             |
| CD  | 5.86%              | 6.09%              | 41.46              | 109.98             |
| ED  | <b>5.17%</b>       | <b>4.72%</b>       | <b>37.35</b>       | <u>97.34</u>       |
| PCD | 5.77%              | 5.49%              | <u>37.89</u>       | <b>96.79</b>       |

$r > 4$ . This is mainly because we evaluate the time efficiency of the model on high-performance GPUs, and the efficiency increase brought by reducing spatial dimension is not so obvious when  $r \geq 4$ . Besides, a larger  $r$  will increase the number of kernels in the first convolutional layer, which will also decrease the efficiency of the model to some extent.

To further illustrate the performance difference between these models, we also measured the percentage of samples in different error ranges, as shown in Fig.5. Overall, the model still presents the best performance when  $r = 4$ , and the worst performance when  $r = 0$ . This further verifies the superiority of the STD layer in improving the prediction accuracy of the proposed AWCNN model. However, in the low error range ( $\leq 15\%$ ), the model performs best when  $r = 8$ . But in the high error range, the model presents better performance at  $r = 4$ . In summary, the model gives higher efficiency and accuracy when  $r = 4$ . It is therefore the optional setting when performing the subsequent experiments.

### 3) Spatial Dimensionality Reduction

The STD conversion provides a manner to reduce the spatial dimensionality of the original input MR image  $\mathbf{x}$ . In general, there are also two other typical methods to shrink the spatial dimension of feature maps in deep neural networks, i.e., resizing based on interpolation and pooling layer. Compared with the STD layer, both of them only reduce the spatial size of feature maps and do not increase the number of feature channels. So intuitively, they will cause more information loss than the STD conversion.

We also compare these three methods to reduce the spatial dimension of feature maps. Table 3 collects the results of the quantitative comparison when  $r = 4$ . It can be observed that the STD conversion gives the lowest MRE and MAE for both  $y_c$  and  $y_w$ . Moreover, the bicubic resizing exhibits slightly better results than the max-pooling operation, which is consistent with the intuition.

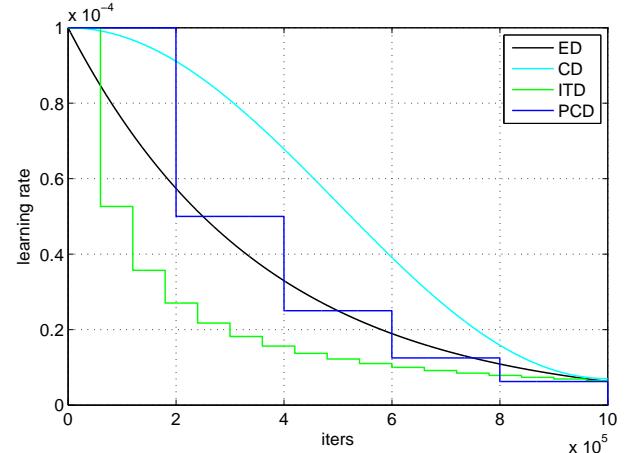


FIGURE 6: Four decays of learning rate. The learning rate is initialized as  $10^{-4}$  and stops at almost the same position. ED: exponential decay, CD: cosine decay, ITD: inverse time decay, and PCD: piecewise constant decay.

### 4) Objective Functions

Learning the end-to-end mapping function for predication the windowing parameters  $\mathbf{y}$  from the original MR image  $\mathbf{x}$  requires the determination of network parameters  $\theta$ , which is usually done by minimizing the loss between the predication of the model and the annotation by human experts. Despite the  $l_1$  loss in (10) is used to train our model, we also check the impact of other loss functions on the performance of the model, including the  $l_2$  loss in (14) and Huber losses with different values of  $\delta$  in (15):

$$l_2(\theta) = \frac{1}{N} \sum_{i=1}^N \|f(\mathbf{x}^{(i)}; \theta) - \mathbf{y}^{(i)}\|_2, \quad (14)$$

$$h(\theta) = \frac{1}{N} \sum_{i=1}^N h_i(\theta; \delta), \quad (15)$$

where

$$h_i(\theta; \delta) = \begin{cases} \frac{1}{2}e_i^2, & |e_i| \leq \delta, \\ \delta|e_i| - \frac{1}{2}\delta^2, & |e_i| > \delta. \end{cases} \quad (16)$$

Here  $e_i = f(\mathbf{x}^{(i)}; \theta) - \mathbf{y}^{(i)}$  and  $\delta$  is the point where the Huber loss function changes from a quadratic to linear.

Table 4 displays the performance comparison between the models trained with different loss functions when  $r = 4$ . As can be seen, the model trained with  $l_1$  loss shows the best predication ability of windowing parameters. This implies that  $l_1$  loss function favors low MRE and MAE and improves the performance of deep models w.r.t. the problem of automatic windowing parameter predication. Besides, the Huber loss is used in robust regression and less sensitive to outliers in data than  $l_2$  loss. From Table 4, we can see that the performance of the model decreases as  $\delta$  decreases. But on the whole, the performance of the model trained with Huber losses is still inferior to that of the model trained with the

TABLE 6: Quantitative evaluation of the compared automatic windowing methods. The optimal values of each column are in **bold**, and the second ones are underlined.

| Methods \ Metrics | Testing Error      |                    |                    |                    | Correlation Index                        |  | Running Time (ms / example) |             |             |
|-------------------|--------------------|--------------------|--------------------|--------------------|--|--|-----------------------------|-------------|-------------|
|                   | $\mathcal{R}(y_c)$ | $\mathcal{R}(y_w)$ | $\mathcal{A}(y_c)$ | $\mathcal{A}(y_w)$ | $\rho(\hat{\mathbf{y}}_c, \mathbf{y}_c)$ | $\rho(\hat{\mathbf{y}}_w, \mathbf{y}_w)$ | 128×128                     | 256×256     | 512×512     |
| GCS [9]           | 45.75%             | 18.32%             | 328.30             | 337.09             | 0.8619                                   | 0.6832                                   | <b>2.23</b>                 | <b>3.46</b> | <b>5.22</b> |
| GRAIL [3]         | 44.52%             | 16.24%             | 202.65             | 202.72             | <u>0.9058</u>                            | 0.9041                                   | $\gg 1000$                  | $\gg 1000$  | $\gg 1000$  |
| ALLTECH           | 11.08%             | 11.49%             | 64.76              | 174.89             | 0.6531                                   | 0.6430                                   | 2.89                        | 4.17        | 6.71        |
| LeNet [11]        | 8.63%              | 8.57%              | 51.89              | 134.34             | 0.8846                                   | 0.8978                                   | 6.23                        | 6.47        | 6.98        |
| AlexNet [6]       | 6.33%              | 6.08%              | 41.84              | 108.10             | 0.8973                                   | 0.9126                                   | 6.30                        | 6.72        | 7.10        |
| AWCNN [Ours]      | <b>5.17%</b>       | <b>4.72%</b>       | <b>37.35</b>       | <b>97.34</b>       | <b>0.9571</b>                            | <b>0.9583</b>                            | 6.06                        | 6.17        | 6.44        |

$l_1$  loss. This is because our evaluation metrics are MRE and MAE, which are formally similar to  $l_1$  loss.

### 5) Learning Rate Decays

The manner of the learning rate decay is another important factor that affects the convergency of model training. In this section, we investigate the impact of several different decays of learning rate on the performance of the model. As shown in Fig.6, it includes inverse time decay (ITD), cosine decay (CD), exponential decay (ED) and piecewise constant decay (PCD). In each case, the learning rate is initialized as  $10^{-4}$  and stops at almost the same position to minimize the impact of different scales of learning rate.

Table 5 displays the result of training the AWCNN model with these methods of learning rate decay when  $r = 4$ . It can be observed that the model performs relatively better when the learning rate decays in the way of ED, which even moves beyond the model trained with PCD adopted by our previous experiments. To further illustrate the performance difference among these learning rate decays, we also conducted the error boundary analysis on the results obtained from different learning decays, as shown in Fig.7. It can be observed that in all error ranges, the model trained with the learning rate that decays in the way of ED presents obviously better performance than the models trained with other learning rate decay methods.

### E. COMPARISON WITH OTHER METHODS

In this section, we compare the proposed AWCNN method with several other windowing methods quantitatively and qualitatively, including global contrast stretch (GCS) [9], Gabor-relying adjustment of image levels (GRAIL) [3] and the method adopted by the Alltech medical systems Co. LTD (ALLTECH). However, all these methods belong to traditional methods. To further illustrate the advantages of the proposed methods, we also include two methods based on deep learning [10], i.e., LeNet [11] and AlexNet [6]. We modify the original LeNet and AlexNet by substituting the last FC layer with a bielemental output layer, similar to Fig.2(a). For the sake of fairness, we always use the same hyperparameter settings, such as batch size, learning rate decay and parameter initialization, optimizer etc. to train these deep models.

#### 1) Quantitative Evaluation

The quantitative results on the testing dataset are shown in Table 6. It can be seen that the deep learning based methods surpass traditional methods by a large margin in predication accuracy, which illustrate the superiority of deep models over traditional techniques. This is principally because deep CNNs have the ability to adequately extract features from training data and the powerful representational ability to map the complex nonlinear relationship between the original MR images and their optimal windowing parameters. Besides, we can also see that the proposed model outperforms the classic LeNet [11] and AlexNet [6], in terms of accuracy, correlation to sample labels (i.e., PPMCC) and execution efficiency. Our AWCNN model is more expressive than LeNet and AlexNet in that it has more model parameters and a deeper network structure. On the other hand, it still has higher execution efficiency since the STD layer enables the inference process executed in LR image space, thus reducing computational consumption. From this point of view, the effect of the STD layer in automatic windowing of MR images is similar to that of ESPCNN [26] in image super-resolution. For execution efficiency, all deep learning based methods can achieve the similar level near to that of GCS [9], supported by a single NVIDIA GeForce GTX 1080 Ti GPU. It is noteworthy that GRAIL [3] is an iterative optimization-based method, which results in extremely slow execution.

To further illustrate the performance difference between the compared methods, we also conduct error bound analysis on the results of these methods, as shown in Fig.8. The proposed AWCNN model provides the best predication for window center  $y_c$  (Fig.8(a)), followed by LeNet [11] and AlexNet [6]. GCS has the worst performance since it simply map the entire intensity range to  $[0, 255]$  in a linear way. For window width  $y_w$ , our AWCNN model also shows the best performance. In the low error range ( $\leq 15\%$ ), the AWCNN exceeds LeNet [11] and AlexNet [6] significantly, while in the high error range ( $\geq 15\%$ ), the situation is reversed. The GRAIL method [3] performs poorly in predicting window width  $y_w$  (Fig.8(b)), which may be due to its initial design for mammographic images. However, it presents good PPMCC values, indicating that its predication results are correlated with sample labels well. ALLTECH is a traditional method with relatively good performance, but it is still far worse than our AWCNN method, especially in the low error range.

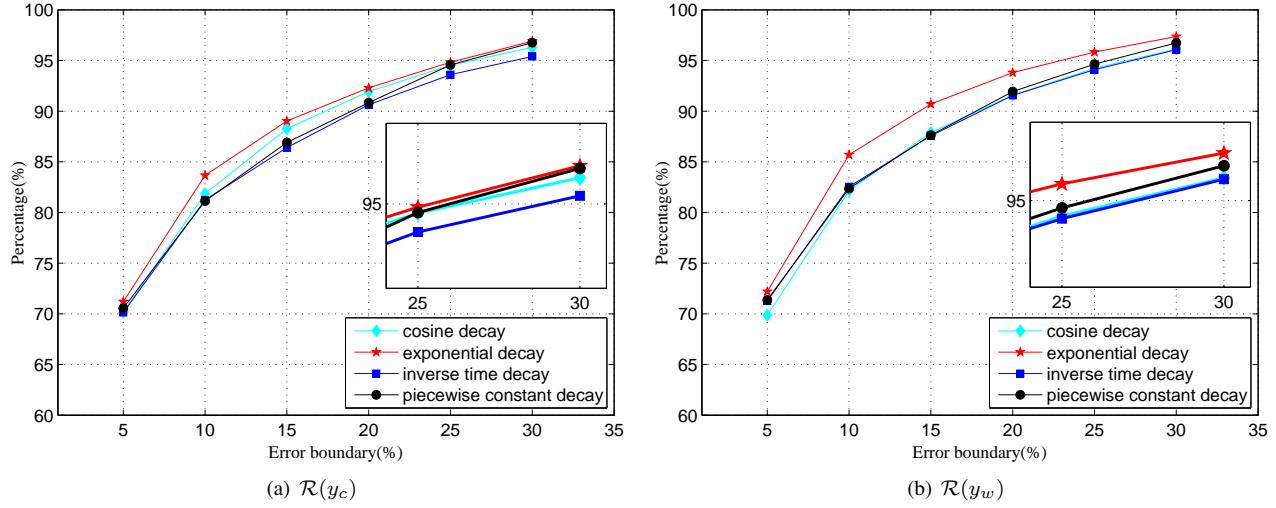


FIGURE 7: Error boundary analysis on the models trained with different learning rate decays. The ratios of testing examples in different error ranges are plotted for further illustration on the difference of model performance.

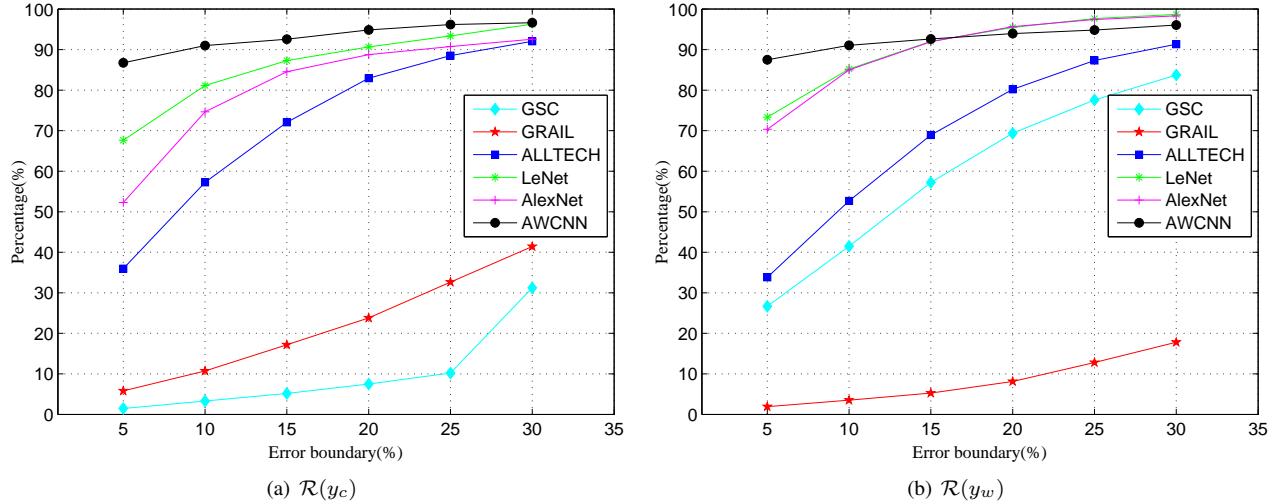


FIGURE 8: Error boundary analysis on the compared models. The ratios of testing examples in different error ranges are plotted for further illustration on the performance difference of the models.

## 2) Qualitative Evaluation

Different windowing parameters will give observers distinct visual experience, especially when these parameters are quite different. Fig.9 displays the visual comparison between the methods mentioned above. The first two rows are two different brain MR images of different types and the last two rows are two spinal MR images of the same type. The last column (ground truth) corresponds to the results adjusted by human clinical experts. As can be observed, the results of GCS [9] tend to be darker than normal brightness range, while those of GRAIL [3] tend to be brighter (similar to overexposure in photography). The results of ALLTECH are relatively better than GCS and GRAIL, but still a little dark on the whole. Since the GCS does not take any consideration of the image content and it simply compresses the intensity of all the pixels linearly to [0,255], there will be an overall darkening

effect. These traditional methods do not take full consideration of image content information, resulting in relatively inferior results. For instance, the GCS method [9] simply stretch the intensity range of all pixels linearly to [0, 255], equivalently for each pixel. The GRAIL [3] method aims at enhancing image contrast, causing many details of the image lost. Besides, it can be seen that all methods based on deep CNNs show obviously better visual effect than the traditional methods. However, our AWCNN model gives more accurate quantitative results than LeNet [11] and AlexNet [6]. This implies that human experts only need to make more minor readjustments according to the results of our AWCNN model to achieve a good display effect.

Further, we show the visual comparison of more body parts in Fig.10, from which we can observe similar results. These MR images have optimal windowing parameters with

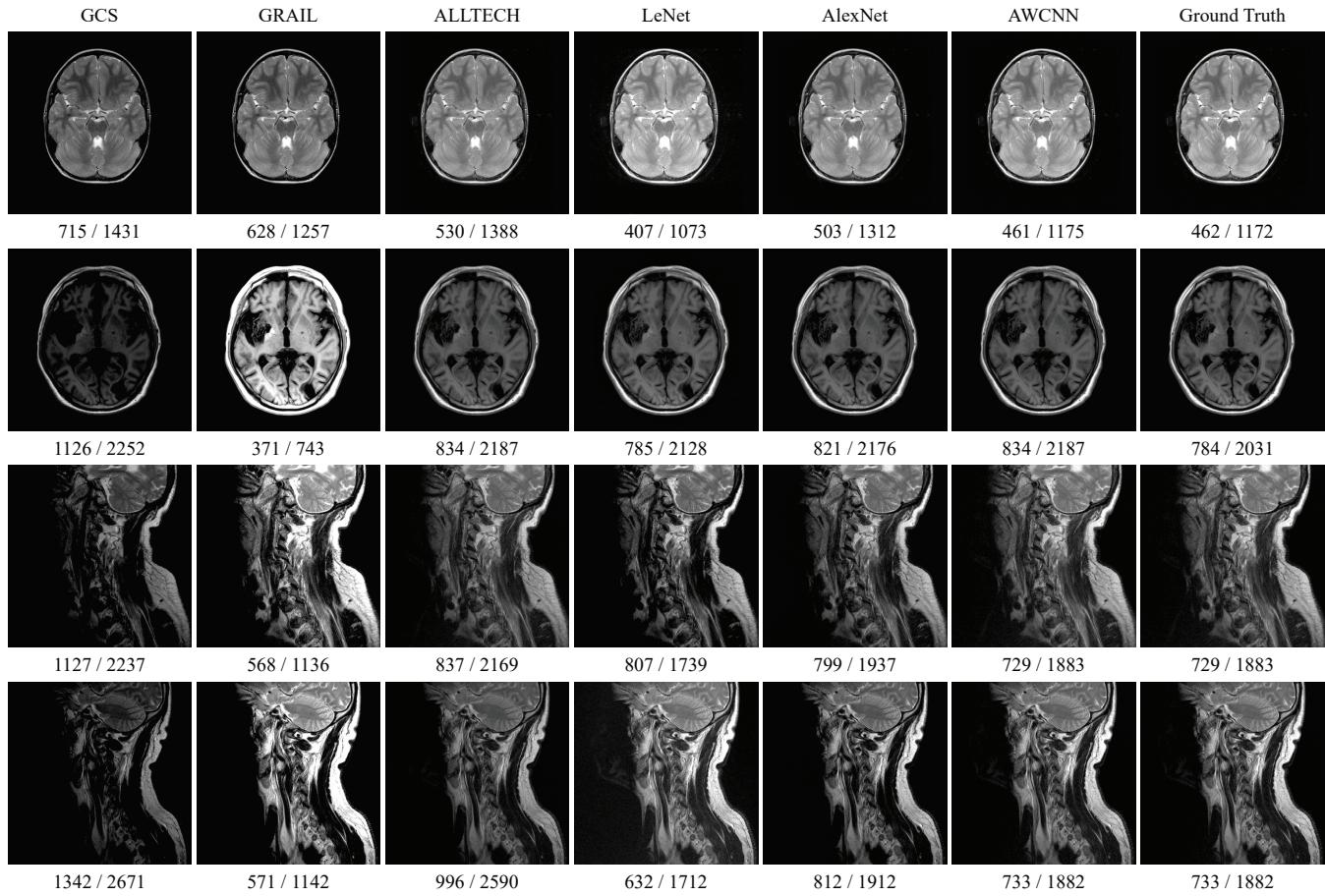


FIGURE 9: The visual comparison between different automatic windowing methods. The first two rows are two different types of brain MR images, the last two rows are two spinal MR images with the same type ( $y_c$  /  $2y_w$ ).

larger floating range (e.g., the first two rows). However, the proposed AWCNN model can still predict these parameters very well. This shows that our AWCNN model can provide a good prediction for different body parts, whose optimal windowing parameters possibly locate in very different data ranges. In addition, since the training samples are collected from different hardware equipment (MRI scanners) and MRI acquisition sequences, indicating that it is also applicable in these different cases.

## V. DISCUSSION AND FUTURE WORK

### A. TRAINING DATA

Typically, the methods based on machine learning (including deep learning) techniques utilize extra information extracted from training data to perform prediction tasks. Therefore, the quality of the training data is critical to the performance of learning models. To improve the quality of training data, the optimal windowing parameters of our MR training images are annotated by professional human experts. The parameters are cross-verified and corrected several times for minimizing annotation error. Even so, the training data still can not match the quality of data for conventional natural image processing

tasks (e.g., image classification and recognition) because the high-quality annotation of medical data often needs specific domain knowledge, but this is usually unnecessary for natural images. In addition, even when training data is annotated by domain experts, the label noise can be a significant limitation in developing deep learning algorithms, whereas in computer vision community, the annotation noise of natural images is typically relatively low [41]. The proposed AWCNN model is also affected by the annotation noise. Therefore, decreasing labeling noise and improving the quality of training data will be part of our future work.

### B. STD CONVERSION

The STD layer before feeding MR images into the model is the key factor to reduce computing consumption and speed up model execution. It also enables the extension of model parameters and network depth, giving the proposed AWCNN model more expressive capacity. Unlike bicubic resizing and pooling operation (max or average pooling), the STD layer shrinks the spatial size of inputs and increases the number of feature channels. Therefore, the spatial information of the input image is transferred to the channel direction instead of

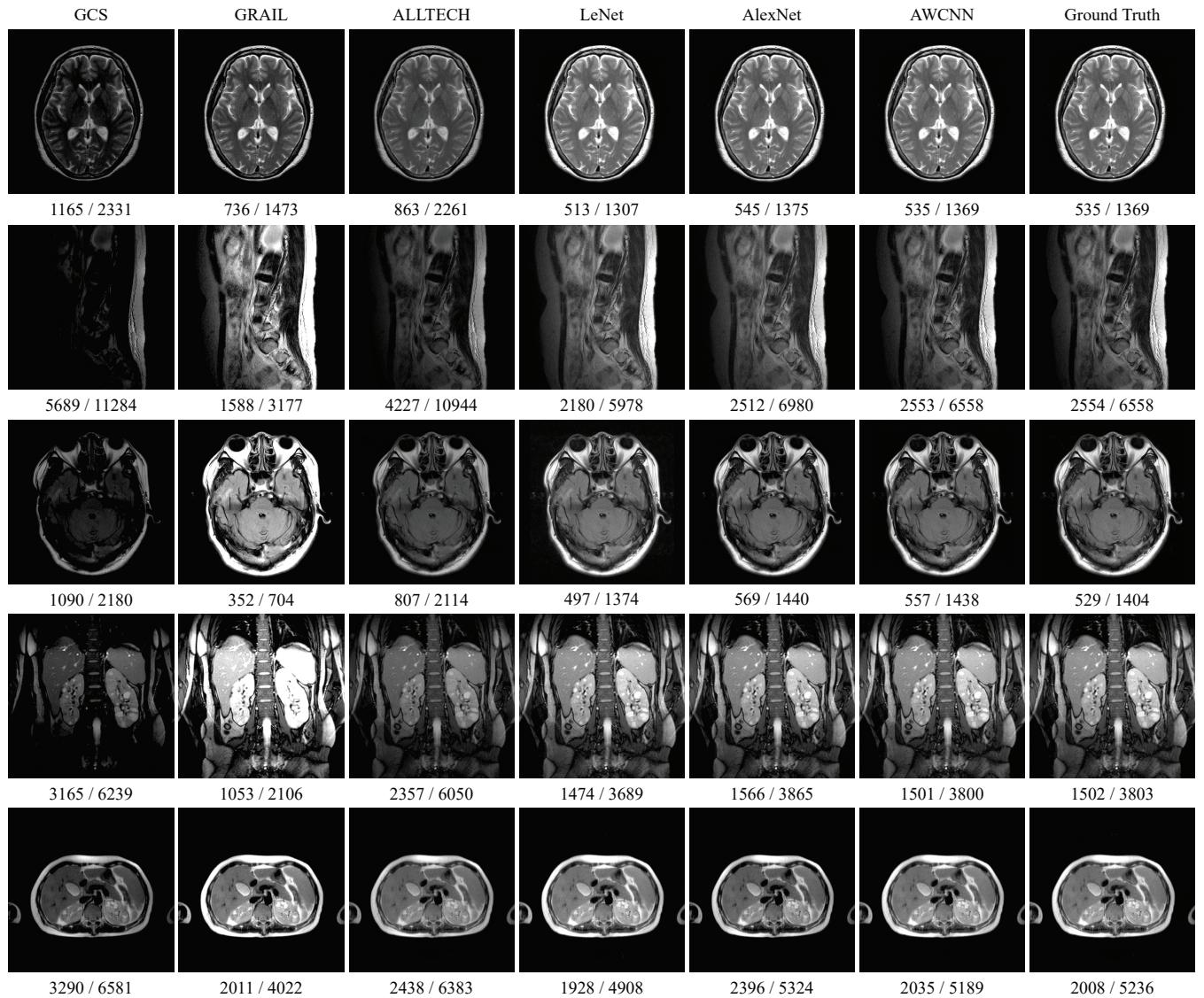


FIGURE 10: Visualization effect according to the windowing parameters calculated by different methods. We covered several typical body parts to illustrate the superiority of the proposed AWCNN method to other methods. Ground truth indicates the parameters set by human clinical experts ( $y_c / 2y_w$ ).

being completely lost, as in the cases of bicubic resizing and pooling operation. The quantitative comparison displayed in Table 3 verifies this conclusion. Thus, an intuitive manner to improve the model performance may be replacing all pooling layers in the model with STD layers. This will increase the number of parameters in the entire model to some extent and improve the representational capacity of the model, which is also part of our future work.

### C. APPLICATION AND DEPLOYMENT

In addition to prediction precision, another important factor to be considered for algorithm application and deployment is the execution efficiency. Table 1 and Table 6 exhibit that the proposed method provides a comparable execution effi-

ciency to ALLTECH, which is a mature and broadly applied method for MRI automatic windowing. Although the results presented are GPU-enabled, we speculate that utilizing CPUs to conduct the computation will not significantly decrease the execution efficiency because once the model is trained and the model parameters are fixed, it only takes one feedforward propagation to test an input. Combined with its high prediction accuracy, it can be seen that the proposed method has valuable potential practical application.

To deploy our method in conventional MRI scanners, there are several ways to use the trained model. Since we employ TensorFlow to implement the model, one can directly install TensorFlow ( $\geq 1.7.0$ ) on the terminal to deploy it. We can also save the model to the disk (such as \*.pb files) and

use third-party tools to process it. A more radical approach is to take out model parameters and treat them as simple numerical matrices, and then manipulate these parameters independently as required.

## VI. CONCLUSION

In this paper, we present a novel method based on deep CNNs to estimate the optimal windowing parameters of MR images that usually have a much wider full pixel intensity range than the typical 8-bit displays. During the first moments of MR image screening, clinical experts spend a non-negligible amount of time to perform ordinary windowing parameters adjustments, expecting to maximize the presence of all sorts of elements. The proposed AWCNN model is composed of a series of stacked convolutional building blocks, followed by several FC layers. The main novelty of this work is the introduction of the STD layer to compress model inference into LR image space, which reduces computational overhead greatly without significant performance loss. Profiting from the ability of CNNs to automatically extract features from raw data, our AWCNN model takes the original MR images as input and does not require any manual effort for feature engineering. Moreover, the STD layer makes it comparable to traditional methods in term of execution efficiency. This improves the practicability of the proposed method to a large extent. Therefore, it has the potential to be integrated to the diagnosis workflow, allowing radiologists to start screening sessions with better initial windowing parameters and keep on modifying them as needed.

## REFERENCES

- [1] A. D. A. Maidment, R. Fahrig, and M. J. Yaffe, "Dynamic range requirements in digital mammography," *Medical Physics*, vol. 20, no. 6, pp. 1621–1633, 1993.
- [2] T. Kimpe and T. Tuytschaever, "Increasing the number of gray shades in medical display systems—how much is enough?" *Journal of Digital Imaging*, vol. 20, no. 4, pp. 422–432, 2006.
- [3] A. Albiol, A. Corbi, and F. Albiol, "Automatic intensity windowing of mammographic images based on a perceptual metric," *Medical Physics*, vol. 44, no. 4, pp. 1369–1378, 2017.
- [4] S. Lai and M. Fang, "A hierarchical neural network algorithm for robust and automatic windowing of mr images," *Artificial Intelligence in Medicine*, vol. 19, no. 2, pp. 97–119, 2000.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [7] A. Ohhashi, S. Yamada, K. Haruki, H. Hatano, Y. Fujii, K. Yamaguchi, and H. Ogata, "Automatic adjustment of display window (gray level) for MR images using a neural network," *Proceedings of SPIE-International Society for Optical Engineering*, vol. 30, no. 1, pp. 111–120, 1992.
- [8] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.
- [9] S. I. Sahidan, M. Y. Mashor, A. S. W. Wahab, and Z. Salleh, *Local and global contrast stretching for color contrast enhancement on Ziehl-Neelsen tissue section slide images*. Springer Berlin Heidelberg, 2008.
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [11] Y. LeCun, J. S. D. B. Boser, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [13] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1122–1131.
- [14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *13th European Conference on Computer Vision*, vol. 8689, 2013, pp. 818–833.
- [15] C. Szegedy, Y. J. Wei Liu, P. Sermanet, S. Reed, and et al., "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1–9.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [17] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," *Eprint Arxiv*, no. arXiv:1904.09925 [cs.CV], 2019.
- [18] W. Ouyang, P. Luo, X. Zeng, and et al., "Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection," *arXiv preprint*, sep 2014.
- [19] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," *Eprint Arxiv*, no. arXiv:1412.1441 [cs.CV], 2014.
- [20] X. Du, X. Shi, and R. Huang, "RepGN: object detection with relational proposal graph network," *Eprint Arxiv*, no. arXiv:1904.08959, 2019.
- [21] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.
- [22] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: face recognition with very deep neural networks," *Eprint Arxiv*, no. arXiv:1502.00873, 2015.
- [23] Y. Zhu, H. Fan, and K. Yuan, "Facial expression recognition research based on deep learning," *Eprint Arxiv*, no. arXiv:1904.09737 [cs.CV], 2019.
- [24] C. Dong, C. C. Loy, K. He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 38, no. 2, pp. 295–307, 2016.
- [25] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.
- [26] W. Shi, J. Caballero, F. Huszar, A. P. A. Johannes Totz, and et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.
- [27] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1132–1140.
- [28] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," *Eprint Arxiv*, no. arXiv:1802.08797, 2018.
- [29] X. Zhao, Y. Zhang, T. Zhang, and X. Zou, "Channel splitting network for single MR image super-resolution," *Eprint Arxiv*, no. arXiv:1810.06453 [cs.CV], 2018.
- [30] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," *Eprint Arxiv*, no. arXiv:1807.02758 [cs.CV], 2018.
- [31] T. Guo, H. S. Mousavi, and V. Monga, "Adaptive transform domain image super-resolution via orthogonally regularized deep networks," *Eprint Arxiv*, no. arXiv:1904.10082v1 [cs.CV], 2019.
- [32] V. Jain and H. S. Seung, "Natural image denoising with convolutional networks," in *International Conference on Neural Information Processing Systems*, 2008, pp. 769–776.
- [33] T. Wang, M. Sun, and K. Hu, "Dilated deep residual network for image denoising," *Eprint Arxiv*, no. arXiv:1708.05473 [cs.CV], 2017.
- [34] J. Carroll, N. Carlson, and G. T. Kenyon, "Phase transitions in image denoising via sparsely coding convolutional neural networks," *Eprint Arxiv*, no. arXiv:1710.09875 [cs.NE], 2017.
- [35] R. E. Wendt, "Automatic adjustment of contrast and brightness of magnetic resonance images," *Journal of Digital Imaging*, vol. 7, no. 2, pp. 95–97, 1994.
- [36] S. H. Lai and M. Fang, "An adaptive window width/center adjustment system with online training capabilities for mr images," *Artificial Intelligence in Medicine*, vol. 33, no. 1, pp. 89–101, 2005.

- [37] B. Yoshua, C. Aaron, and V. Pascal, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.
- [39] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Eprint Arxiv*, no. arXiv:1412.6980 [cs.LG], 2014.
- [40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.
- [41] G. Litjens, T. Kooi, B. E. Bejnordi, and et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, no. 2017, pp. 60–88, 2017.



**TAO ZHANG** received the BS degree in 1996 and MS degree in 1999 from Shanghai Jiao Tong University, Shanghai, China, and the Ph. D in Mechanical Engineering from Florida State University in 2004. He is currently a professor in the School of Life Science and Technology of the University of Electronic Science and Technology of China. From 2005 to 2012, he worked as a senior scientist at the General Electric (GE) Global Research Center. From 2012, he joined and worked as the Chief Technology Officer and Vice President of Alltech Medical System in China. His research interests include the magnetic resonance imaging (MRI) system technologies and its clinical applications, MRI system hardware development, and the applications of MRI technique in neuroscience and brain research. He has published over 30 technical papers and has been awarded over 20 patents.



**XIAOLE ZHAO** received the B.S. and the M.S. degrees from the School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China, in 2013 and 2016 respectively. He is currently working toward the Ph.D. degree in the High Field Magnetic Resonance Brain Imaging Laboratory of Sichuan and the School of Life Science and Technology, the University of Electronic Science and Technology of China, Chengdu, China. His research interests include computer vision and image processing, machine learning, medical image analysis and deep learning techniques.



**HANGFEI LIU** received the B.S. degree from the School of Biomedical Engineering, Xin Xiang Medical University, Henan, China, in 2015. He is a M.S. candidate with the Neural Information Group of the School of Life Science and Technology, the University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests mainly include magnetic resonance imaging, MR image reconstruction, multi-modal synthesis of medical images by machine learning and

deep learning.



**XUEMING ZOU** received the MS degree in Physics from the Oregon State University (OSU), Corvallis, Oregon, USA in 1986 and the Ph.D. degree in Bio-Medical Physics from the Massachusetts Institute of Technology, Cambridge, in 1990. He is a professor in the School of Life Science and Technology of the University of Electronic Science and Technology of China (UESTC), and the group leader in the High-field Magnetic Resonance Brain Imaging Key Laboratory of Sichuan Province. He worked as the Founder, President and CEO of USA Instruments which scratch to number one in global market share in MRI Coils, from 1993 to 2002. He also served as the Global Vice President and General Manager for Greater China Region of MR Business Division of GE. In 2005, Dr. Zou founded the AllTech Medical Systems Co. LTD, a China Corporation operating in Chengdu and worked as the president. His research interests include MRI system physics, advanced medical diagnostic imaging applications, and big data in medical imaging.



**GAIYAN ZHU** received her B.S. degrees from the School of Information and Communication Engineering, the North University of China (NUC), Taiyuan, China, in 2017. She is currently working to the M.S. candidate with the Neural Information Group of the School of Life Science and Technology, the University of Electronic Science and Technology of China (UESTC), Chengdu, China. Her research interests mainly include magnetic resonance imaging processing.