Three-dimensional Optical Coherence Tomography Image Denoising via Multi-input Fully-Convolutional Networks

Ashkan Abbasi[a], Amirhassan Monadjemi[a,]*, Leyuan Fang[b,]*, Hossein Rabbani[c], and Yi Zhang[d].

[a] *Artificial Intelligence Department, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran. (E-mail: monadjemi@eng.ui.ac.ir)*

[b] *College of Electrical and Information Engineering, Hunan University, Changsha, China. (E-mail: fangleyuan@gmail.com)*

[c] *Department of Biomedical Engineering, Medical Image and Signal Processing Research Center, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran.*

[d] *College of Computer Science, Sichuan University, Chengdu, China*

Asterisk indicates corresponding author.

***Abstract*— In recent years, there has been a growing interest in applying convolutional neural networks (CNNs) to low-level vision tasks such as denoising and super-resolution. Optical coherence tomography (OCT) images are inevitably affected by noise, due to the coherent nature of the image formation process. In this paper, we take advantage of the progress in deep learning methods and propose a new method termed multi-input fully-convolutional networks (MIFCN) for denoising of OCT images. Despite recently proposed natural image denoising CNNs, our proposed architecture allows exploiting high degrees of correlation and complementary information among neighboring OCT images through pixel by pixel fusion of multiple FCNs. We also show how the parameters of the proposed architecture can be learned by optimizing a loss function that is specifically designed to take into account consistency between the overall output and the contribution of each input image. We compare the proposed MIFCN method quantitatively and qualitatively with the state-of-the-art denoising methods on OCT images of normal and age-related macular degeneration eyes.**

Index Terms—Fully convolutional network (FCN), Multi-input FCN, Image denoising, Optical Coherence Tomography (OCT).

1. INTRODUCTION

Optical coherence tomography (OCT) is a noninvasive imaging modality which is widely used for diagnosis and treatment planning of various ocular diseases [1]. However, due to interferometry nature of the image formation process, noise corruption is inevitable during OCT imaging. The presence of noise heavily degrades image quality and complicates image analysis. Quality of OCT imaging could be improved either with the use of higher incident power or longer exposure time [2]. Unfortunately, both of these options cannot be used because: 1) The incident power is limited by the safety guidelines, and 2) The imaging speed is an important factor to avoid motion artifacts from the fixation eye movements [3] or enable 3-D volumetric imaging. Thus, image denoising is an essential step in many OCT image analysis tasks.

A fair amount of various methods have been proposed in the literature for OCT denoising. For example, the early spatial filtering approaches [4] are based on computing local statistics of the degraded image in the spatial domain. We can transform an image content into another domain such as filtering response domain [5] or multi-resolution domain [6,7], where image statistics can be modeled more efficiently. Although promising results have been obtained with transform domain approaches, they are generally known to produce smoothing or unexpected artifacts due to limited modeling ability [8].

The image modeling ability has been improved by the introduction of patch-based approaches. Since patches have lower dimensions compared to the whole image, they are easier to model. Moreover, patches capture image statistics locally, and thus edges and local structures can be better treated. The most successful representative modeling approaches are Markov random field (MRF) [9], sparse representation [10], and Gaussian mixture models (GMM) [11]. Some recent works that successfully applied patch-based sparse representation to OCT image reconstruction include [12–16]. Recently, a variant of GMM [17] was also applied to OCT image denoising with promising results [18]. Most of the mentioned approaches can be enhanced greatly by the use of nonlocal similarity [19] in natural images. However, although excellent success has been achieved by the mentioned approaches, they mostly rely on computationally expensive optimization algorithms in the reconstruction stage. Also, patch aggregation by averaging negatively affects the effectiveness of the image model [10].

Deep learning approaches have been proven to be highly effective in many high-level vision tasks [20,21]. Recently, these approaches have been also successfully applied in medical image recognition tasks, including classification detection and segmentation. The great success of neural networks and the progress made in their training methods pave the way for using neural networks as a promising alternative approach to tackle image denoising problems. For example, [22] have first shown that denoising using a convolutional neural network (CNN) could outperform several well-known methods. In [23], it has been shown that a multi-layer perceptron (MLP) can achieve comparable performance to the benchmark BM3D [24]. However, MLP has a fully connected architecture, thus making both training and inference computationally intensive. Thus, considerable attention has been recently given to CNNs [25,26].

In this paper, following the works in [12–15] where they learned mappings from noisy images to high signal-to-noise (SNR) images by exploiting sparsity, we propose to use a specifically designed CNN for this task. To the best of our knowledge, the proposed method is the first CNN-based OCT denoising method. Also, with the advances in OCT imaging, the acquisition of 3-D volumetric scans of the retina is now in wide clinical use. Therefore, effective use of information from nearby slices is a promising way to reduce noise [14,16]. Although the neural network-based denoising approaches have achieved great success, most of them have focused on 2-D gray-scale image denoising [27]. Therefore, the question of how to effectively use high correlations among nearby

OCT slices to reduce noise by a CNN is not investigated.

Here, we aim to develop a network architecture for OCT denoising that utilizes the high correlations among nearby OCT images using a multiple branch architecture. Each branch can be considered as a fully-convolutional network (FCN) with the aim to reduce the noise of its input. Hence we name our method multi-input FCN (MIFCN). The results of these FCNs are fused together by an intermediate weighted averaging module which is inspired by the nonlocal mean weighting mechanism [19]. This module produces weight matrices for each branch. Then, Hadamard products of the weight matrices and the outputs of each branch are computed to produce the module's output. The structure is followed by another set of convolution layers which produce the final reconstructed image based on the averaging module output. Since the weighting mechanism suppresses useless contributions among nearby slices, we can ensure that the proposed method can capture their correlations while it is insensitive to small variations in the inputs. We show that the parameters of the proposed MIFCN method can be learned by optimizing a loss function that is specifically designed to enable end-to-end training of the overall architecture.

The rest of the paper is organized as follows. In the following section, we briefly review related works. Next, we describe the proposed MIFCN method in Section 3. In Section 4, we describe the training procedure. Experimental results on clinical OCT data are shown in Section 5. Finally, the conclusion and future works are presented in Section 6.

2. RELATED WORKS

2.1. Convolutional Neural Network

CNN is a multilayer architecture with an input layer, an output layer, and multiple hidden layers. The hidden layers mostly consist of convolutional and pooling layers. The last hidden layers can be fully connected layers for global decision-making. The convolution and pooling layers enable the whole structure to extract a hierarchical representation of the data, in which shallower layers concentrate on low-level features whereas deeper layers represent higher-level features [28]. Each layer composed of a number of feature maps. Each unit in a feature map (neuron) is computed by a local operation, i.e., convolution or pooling, on the previous layer. By contrast, in a fully connected layer, each neuron is connected to every neuron in the previous layer. The local connectivity greatly reduces the number of parameters to be learned and captures local statistics of natural images. Moreover, these local operations can operate over arbitrary-sized inputs. In this base, Long et al. [29] proposed a variant of CNNs by casting fully connected layers into convolutions and named it FCN. This makes the FCN a natural choice for image transformation tasks ranging from low-level to high-level vision tasks.

2.2 Network Architectures

In general, the main purpose of successive convolution and pooling layers in CNNs is hierarchical feature extraction. However, the pooling or sub-sampling, in any forms, results in loss of spatial information [30,31]. Therefore, we have chosen a network architecture without subsampling layers as a building block for implementing our proposed MIFCN method.

The context aggregation network (CAN) was recently proposed for semantic segmentation [30] and it mainly contains convolution layers with dilated convolutions. Dilated convolution provides a mean to aggregate contextual information without the need for any form of subsampling. More recently, a fast and compact instance of CAN has been also successfully applied for approximating a number of image processing operators [31]. We will describe some of the main components of this architecture in the following paragraphs.

2.3 Convolution Layer and Dilated Convolution

Convolution layer is the main ingredient of a CNN architecture [29]. Each convolution layer composed of several feature maps of the same size and every feature map highlights the regions in the input that are most similar to its corresponding filter. These filters are learned in such a way that they eventually activate features suitable for a given task. During the forward pass, feature maps are computed from the previous layer as follows:

$$F_i^l = b_i^l + \sum_j F_j^{l-1} *_d K_{i,j}^l, \tag{1}$$

where F_i^l is the i-th feature map of layer l, F_j^{l-1} is j-th feature map of the previous layer $(l-1)$, $K_{i,j}^l$is a convolution kernel, the operator $*_d$ represents dilation convolution with dilation d, and b_i^l is a bias. By increasing the dilation *d*, the filter can tap locations separated by the factor *d* without losing resolution [30]. Mathematically, the *d*-dilated convolution at location x between a feature map and a kernel can be written as:

$$(F *_r K)(x) = \sum_{a+rb=x} F(a)K(b), \tag{2}$$

2.4 Activation Layer

To enhance representation ability of neural networks, the results of a convolution layer usually pass through a point-wise nonlinearity. A popular activation function is rectified linear function (ReLU). However, since ReLU outputs zero for all negative values, some neurons can die (dying ReLU problem). The leaky variants of ReLU can be used to avoid this problem. Here, we use the leaky ReLU (LReLU) [32] which is defined as $\sigma(x) = \max(\alpha x, x)$, where the constant parameter α (called the leak parameter) determines the slope for negative values.

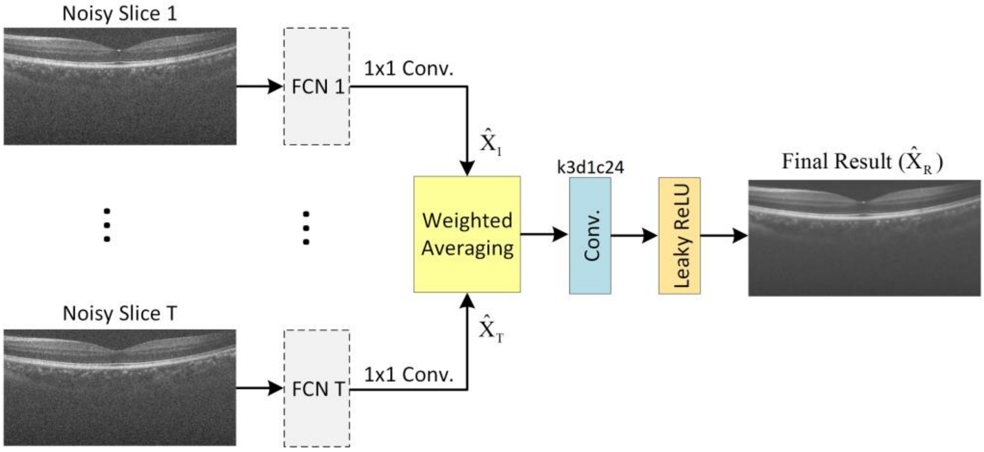

Fig. 1. The overall structure of the proposed MIFCN method for OCT image denoising. The numbers in front of the letters “k”, “d”, and “c” represent kernel size, dilation rate, and number of feature maps, respectively.

3. Proposed MIFCN Method for OCT Image Denoising

Given an OCT image observation Y_1 (main image) with *T-1* number of its nearby OCT images, our goal is to design a network that can effectively utilize the correlations among these T inputs $\{Y_1, \dots, Y_T\}$ and reduce noise of the main image. This problem can be cast as a regression problem [12,13]. In this section, we propose a multi-branch architecture that can be learned in an end-to-end manner. The overall architecture of the proposed MIFCN method is shown in Fig. 1. Each branch mainly consists of convolution layers with the aim to reduce the amount of noise in its input. Thus, each branch is an FCN. Then, a weighted averaging module is used to combine the results based on their similarity to the main image (Y_1). The architecture is followed by another convolution and activation layers to enhance the modeling ability and give more plausible result.

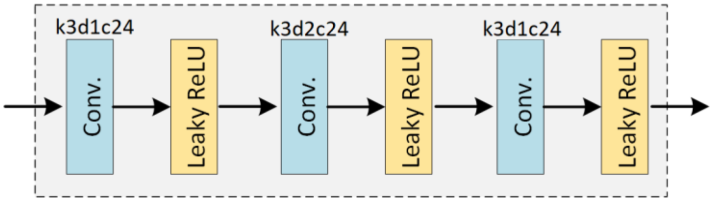

Fig. 2. The structure of each FCN that is used in the overall architecture (Fig. 1) of the proposed MIFCN method. The numbers in front of the letters "k", "d", and "c" represent kernel size, dilation rate, and number of feature maps, respectively.

The network for each branch is designed as shown in Fig. 2. We have used a similar network structure for all branches. This structure consists of three convolution layers. All of these layers have the same number of feature maps C = 24. The dilation rate (d) for each of hidden layers are set to 1, 2, and 1, respectively. Then, the output of each convolution layer is passed through an activation layer. Let us indicate the input to the t-th branch as $F^0 = Y_t \in R^{M \times N}$, where $t \in \{1,2,\dots,T\}$. Therefore, as explained in the previous sections (2.3 and 2.4), each feature map in the convolution layers are computed as follows:

$$F_i^l = \sigma(b_i^l + \textstyle\sum_{j=1}^{C} F_i^{l-1} *_d K_{i,j}^l), \tag{3}$$

where $l \in \{1,2,3\}$ indicates the layer number, and $K_{i,j}^l$ is a 3×3 kernel. Then, another convolution layer without activation layer can be used to reconstruct an output image at the end of each branch:

$$F^4 = b_i^4 + \textstyle\sum_{j=1}^{C} F_i^3 *_1 K_{i,j}^4 = \hat{X}_t, \tag{4}$$

where $K_{i,j}^4$ is a 1×1 convolution kernel.

During the forward pass, the outputs of branches result in a set of noise reduced images $\{\hat{X}_1, \dots, \hat{X}_T\}$. Since the inputs are nearby images, the outputs of branches have some spatial correlations. However, there might be slight variations between these images. Thus, simple pixel by pixel averaging can result in blurring artifacts [33]. Motion compensation algorithms can be used with the expense of high computational cost and tolerating error (even for noise-free images) [34]. Here, inspired by the nonlocal mean (NLM) weighting mechanism [19], which is almost robust to slight variations in patches [33,35], we propose a module that compares each pixel of the main branch output ($\hat{X}_1$) to the corresponding pixels from the other outputs $\{\hat{X}_2, \dots, \hat{X}_T\}$. The proposed module assigns weights to pixels of denoised nearby images. Each pixel in the output of the proposed module is simply formed by a weighted combinations of a pixel from the main branch output and the corresponding pixels from the other outputs. Concretely, the output of the averaging module is computed as follows:

$$\bar{X} = \sum_{t=1}^{T} \hat{X}_t \, o P_t, \tag{5}$$

where $\bar{X}$ is the output of the weighted averaging module, $\hat{X}_t$ is the output at the end of each branch (Fig. 1), P_t is a weight matrix with the same size as $\hat{X}_t$, and the operator o represents Hadamard product. Each entry in the weight matrix (P_t) reflects the similarity between the corresponding pixels in the main branch output ($\hat{X}_1$) and the *t*-th branch output ($\hat{X}_t$). Therefore, the maximum scores are always assigned to the main branch pixels and the entries of P_1 are bigger than the others. To compute the weight matrix, we need firstly to compute the differences between corresponding pixels from denoised nearby images:

$$D_t = \left(\hat{X}_1 - \hat{X}_t\right)^2, \; t \in \{1,2,\ldots,T\}, \tag{6}$$

where D_t is a matrix of intensity differences whose elements are differences between pixels from the main branch ($\hat{X}_1$) and pixels from the *t*-th branch ($\hat{X}_t$). Then, this matrix can be used for computing exponentially decaying weights for pixels of an image:

$$W_t = \exp(-D_t \backslash h), \tag{7}$$

where W_t is the matrix of weights associated for each pixel in the *t*-th branch prediction, and h is a constant parameter. To ensure that these weights sums to one, we can normalize the weight matrix by:

$$P_t = W_t \backslash (\textstyle\sum_{t=1}^{T} W_t), \tag{8}$$

Next, the result of the weighted averaging module is processed by another set of network layers. As shown in Fig. 1, the output of the averaging module ($\bar{X}$) is fed into a convolution followed by LReLU activation layers to obtain the last hidden feature map (F_i^5). Then, this feature map (F_i^5) is converted to the final image using a 1x1 convolution layer:

$$F^6 = b_i^6 + \sum_{j=1}^{C} F_j^5 *_1 K_{i,j}^6 = \hat{X}_R, \tag{9}$$

where $\hat{X}_R$denotes the final result of our 3-D reconstruction method.

Before concluding this section, it is worth mentioning that unlike the original NLM weighting mechanism [19,33] which is based on comparing small patches around each pixel, here, we compute weights based on comparing pixels between denoised nearby images. We have experimentally found that this pixel by pixel averaging module is strong enough to provide plausible results and avoid extra computations. The reasons can be explained as follows: 1) noise is expected to reduce in each branch, thus pixel by pixel comparison is more robust compared to such a comparison between pure noisy images, and 2) we can compare our module with local filtering approaches [4]. In local filtering approaches, a pixel is denoised based on weighted average of pixels around it. Here, instead of considering a neighborhood around each pixel, we use corresponding pixels from the nearby images that have some correlations. Therefore, it is reasonable to apply the same principles.

4. Learning the Parameters

Since FCN is not sensitive to the input size, we can train an FCN for denoising using only patches. In fact, training an FCN for denoising using patches have empirically shown to be helpful [25,26]. This is because it enables the network to see more local information. As shown in Fig. 1, the proposed MIFCN architecture has *T* branches. To train the parameters of all branches simultaneously, we need *T* patch pairs. However, in contrast with the test dataset, the training dataset does not include nearby images (the training and test datasets will be described in Section 5.1). Since the training procedure is based on using patches, we can gather similar patches for each patch. Therefore, following a simple procedure, we can create a dataset of patches and their similar ones as a training dataset. First, N patches of size $\mathrm{p}_1 \times \mathrm{p}_2$ pixels are extracted from all high SNR images. Then, for each patch in an image, the T most similar patches (including the patch itself) are collected by nonlocal searching [19] in that image. By extracting the corresponding noisy patches for each patch, we have a training set of the following form:

$$D = \left\{\left\{\left(y_1^{(j)}, x_1^{(j)}\right), \dots, \left(y_T^{(j)}, x_T^{(j)}\right)\right\}\right\}_{j=1}^{N}, \tag{10}$$

where $y_t^{(j)}$ is the *t*-th similar patch for the j-th noisy patch and $x_t^{(j)}$indicates its corresponding high SNR patch.

Given the training dataset D, we need a loss function to train the parameters θ of the network architecture. The set of parameters θ include kernels and biases of all feature maps. The widely used mean squared error (MSE) is the common choice for image reconstruction purposes. However, in our experiment with the proposed MIFCN architecture, we could not train the proposed MIFCN architecture using pure MSE, particularly because MIFCN has multiple branches. Pure MSE only takes into account the error between predicted outputs and desired outputs. Since our proposed MIFCN architecture has multiple branches, learning its parameters using only MSE might result in a useless branch which outputs zeros. Instead, we have designed a loss function which also takes into account the last feature maps of branches. This loss function enables training the architecture in an end-to-end manner:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(\hat{X}_t^{(i)} - X_t^{(i)}\right)^2 + \frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\hat{X}_1^{(\mathrm{i})} - \hat{X}_R^{(i)}\right)^2, \tag{11}$$

where *N* is the number of training patches, *T* is the number of branches, $\hat{X}_t^{(i)}$ is the output of the last feature map in the *t*-th branch for the i-th noisy training patch, $\hat{X}_R^{(i)}$ is the final output of the architecture for the i-th noisy training patch, and $X_t^{(i)}$ is the corresponding high SNR patch.

In Equation (11), the first term encourages the similarity between the result of the last feature map in each branch for its input and the corresponding high SNR patch. This is because the last feature map should be a noise reduced version of its input. The second term encourages the similarity between the final output and the prediction of the first branch (or main branch). In this way, we can ensure that slight variations in the inputs cannot negatively affect the final output.

For training, the loss function $J(\theta)$ can be minimized using a gradient descent based optimizer [36]. We trained the network with augmented data generated by horizontal and vertical flipping, and +90-degree rotation. All of the training data was presented for 60 epochs. In the first 30 epochs, the learning rate was set to 0.0001. Then, it was set to 0.00001 for the remaining epochs. The total training time for this architecture was less than two hours. More epochs do not seem to improve the performance.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results of the proposed MIFCN method. We compare the proposed MIFCN methods with some of the well-known state-of-the-art denoising methods. The source code of our method will be made publicly available on the website (https://github.com/ashkan-abbasi66/MIFCN). Also, all of visual results of the proposed method and compared methods are now available on the website.

5.1 Datasets

To train and evaluate our proposed MIFCN method, we have used the SDOCT datasets that were made publicly available by [12–14]. All of these images were captured by Bioptigen SDOCT imaging (Durham, NC, USA) from 28 subjects with normal and age-related macular-degeneration (AMD) eyes. In the training part, there are 10 high SNR images with their corresponding noisy images. These high SNR images and their noisy correspondences can be used for training. The high SNR images were acquired by registration of azimuthally repeated OCT images from the fovea [12–14]. The rest of images are used as test dataset. Therefore, there are 18 images in the test part. However, for each image in the test dataset, four noisy nearby OCT images are also provided along with a high SNR image. Thus, an image denoising algorithm can exploit one or more OCT images to reconstruct a high quality OCT image and the final reconstruction results can be compared with the corresponding high SNR images.

5.2 Quantitative Metrics

The performance of the proposed MIFCN method is assessed by various image reconstruction metrics. Specifically, we adopt the peak signal-to-noise-ratio (PSNR), mean-to-standard-deviation ratio (MSR) [37], contrast-to-noise-ration (CNR) [38], and equivalent number of looks (ENL) [6]. Since we have high SNR images, we compute the PSNR as a widely accepted metric in this scenario. This metric is defined based on the intensity differences between the output and a reference image. The other metrics do not need the reference images but they are computed locally. Therefore, we need to select a few regions of interest (ROIs) from the images. The contrast between foreground regions (e.g. red box #2-#6 in Fig. 3) and background noise is measured by the CNR metric. The background noise is computed in the background region (e.g. red box #1 in Fig. 3). The CNR metric is large when ROIs contain prominent features with bigger mean than background’s mean, but with small variance. The MSR is a sign of good feature recovery without taking into account background regions. The ENL evaluates smoothness in background regions. Large ENL indicates a stronger noise smoothing in background areas [6]. In addition, we have used the Wilcoxon signed-rank test to show the statistical differences between the proposed MIFCN method and the compared methods.

5.3 Compared methods

The proposed MIFCN method is compared with some of the well-known state-of-the-art denoising methods from the literature. The comparison methods include: K-SVD denoising algorithm [10], block matching and 3-D filtering (BM3D) [24], spatially adaptive iterative singular-value thresholding (SAIST) [39], patch group based Gaussian mixture model (PG-GMM) [40], block matching and 4-D filtering (BM4D) [41], and segmentation based sparse reconstruction (SSR) [14].

The K-SVD denoising algorithm [10] is a celebrated sparse representation based image denoising method. In this method, the sparse representation over a learned dictionary is used to remove noise. The benchmark BM3D [24] combines the benefits of sparsity based image modeling and nonlocal similarity within each group of similar patches. The BM4D [41] is an extension of BM3D for volumetric data. In BM4D, groups of similar cubes are collaboratively filtered to reduce noise. Therefore, BM4D can naturally capture correlation between multiple images. The SAIST [39] uses a low-rank approach to characterize local and nonlocal variations in a group of similar patches. In contrast to the existing nonlocal image restoration methods which use only nonlocal similarity of corrupted image, the PG-GMM [40] learns a nonlocal similarity prior from an external training dataset. Gaussian mixture models are used to learn the prior based on groups of similar patches. The SSR [14] is a recently proposed OCT image reconstruction algorithm which performs sparse representation over learned dictionaries for each layer. The dictionary for each layer is learned/selected using a segmentation algorithm. Thus, SSR combines a good modeling approach (i.e., sparse representation over learned dictionaries) and a good model selection strategy.

5.4 Algorithm parameters

Most of the parameters of the proposed MIFCN method, including kernels and biases of feature maps, are learned automatically from the training data. The constant parameter α of LReLU function was set to 0.2 and the identity initialization method [30] were used to initialize kernels and biases of feature maps. The number of branches (T) is, in general, dictated by the imaging configuration (more specifically, it is based on the azimuthal resolution of the OCT volume). This is because we want to avoid the contributions of images with large difference in contents. However, there is also a constant parameter h in the exponential weight function (7) that can be used to control the amount of contributions from nearby images. In Section 5.6, we will show the effect of different values of the parameter h on the performance of the proposed MIFCN method. Our test dataset (Section 5.1) has five images per subject. Therefore, we set the number of input branches T to 5. The constant parameter h was experimentally set to 400. Setting a smaller value for h weakens the contributions of nearby images.

For training, we had 10 pairs of noisy and high SNR images (Section 5.1). We extracted patches of size 15×15 pixels. The bigger patch sizes may cause blurring artifacts because of losing small details and smaller patch sizes are more likely to capture noise [26]. Since there are a large background portion in OCT images, we manually cropped a portion containing the retina from each training image. Then, patches were extracted with as less overlap as possible to increase the variety of training samples. Specifically, we extracted 400 patch pairs from each training image pair. Flipping and rotation were also used to augment the training data by a factor of 3 (for further details, see Section 4). Therefore, in our experiments, the number of training samples (N) in Equation (10) was 400*10*3. After training the model, all of the mentioned parameters kept unchanged during experiments. The parameters of the compared methods were optimally assigned or set according to their original papers.

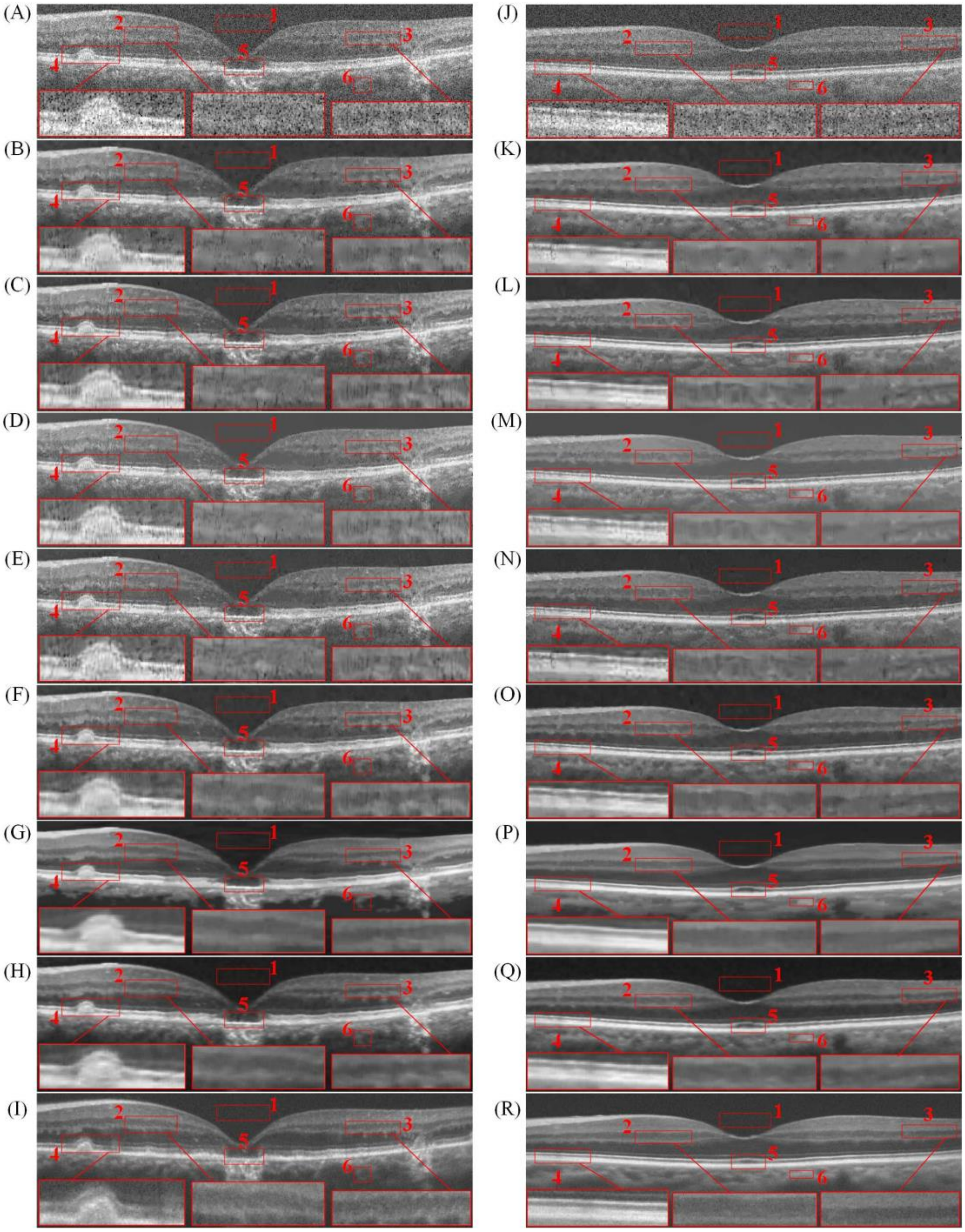

Fig. 3. Visual comparison of two denoised images by the compared methods. First column: (A) Original Noisy Image; (B) KSVD Denoising (PSNR = 26.05); (C) BM3D (PSNR = 26.25); (D) SAIST (PSNR = 26.01); (E) PG-GMM (PSNR = 26.1); (F) BM4D (PSNR = 26.48); (G) SSR (PSNR = 26.89); (H) The proposed MIFCN method (PSNR = 27.49); (I) The registered and averaged images. Second column: (J) Original Noisy Image; (K) KSVD Denoising (PSNR = 26.13); (L) BM3D (PSNR = 26.02); (M) SAIST (PSNR = 26.16); (N) PG-GMM (PSNR = 25.89); (O) BM4D (PSNR = 26.54); (P) SSR (PSNR = 27.06); (Q) The proposed MIFCN method (PSNR = 27.56); (R) The registered and averaged images.

5.5 Results for OCT Image Denoising

Figure 3 presents two denoising results of our proposed MIFCN method. The corresponding high SNR images are also shown at the bottom of each column. As can be seen, K-SVD, BM3D, SAIST, and PG-GMM attenuate the noise while result in apparent visual artifacts. Exploiting self-similarity in BM3D, SAIST, and PG-GMM allows better reconstruction of retinal layers. The SAIST method utilizes a low-rank approach which is a powerful technique especially for background regions. However, this method results in more artifacts especially in retinal layers compared to other methods (e.g., compare red box#4 in Fig. 3, (D) and (M) with their corresponding boxes in other images). The visual results confirms that BM4D, SSR, and our proposed method can better preserve layer structure due to exploiting correlation among nearby OCT images. Similar to the results of BM3D (e.g. Fig. 3, (C) and (L)), using fixed bases in BM4D limits modeling ability and results in visible artifacts. However, there are less artifacts in the OCT reconstruction results of BM4D compared to the BM3D's results. This might be attributed to finding better matches by grouping 3-D patches in BM4D instead of grouping 2-D patches in BM3D. Although the SSR uses learned dictionaries for reconstruction of each layer, its reconstruction results are too smooth. The layer boundaries are reconstructed very well, but the smoothness can be easily seen from the background regions (vitreous and sclera) and cloudy appearance choroidal region below retinal layers.

The visual results can be validated by the average quantitative results which are reported in Table I and II. In Table I three quantitative metrics (i.e., MSR, CNR, and ENL) which are widely used in the evaluation of OCT reconstruction algorithms [6,14] are reported. The average PSNR results are reported in Table II. These quantitative results reveal that the proposed MIFCN method performs reasonably well in terms of all metrics, except that for the mean of the ENL. This is because the ENL evaluates smoothness in background regions. Therefore, this high ENL value for SAIST can be attributed to the strong noise suppression in background areas due to exploiting a low-rank strategy. For SSR method, smoothing textures is the main cause of achieving high ENL value. However, because SSR can well preserve layers structures, other metrics have relatively high values. All of these qualitative and quantitative results suggest that the proposed MIFCN method outperforms other methods for OCT denoising.

Table I

Mean and standard deviation (SD) of the MSR, CNR, and ENL results for denoising 18 foveal images by the compared methods. Where $p<0.05$, the metrics for each test method are considered statistically significant and were marked by "*". Best results in the mean values are shown in bold.

Method	MSR			CNR			ENL		
	Mean	SD	*p* value	Mean	SD	*p* value	Mean	SD	*p* value
K-SVD	7.53	1.26	8.91E-04*	3.39	0.52	4.80E-07*	776.33	150.54	5.61E-14*
BM3D	6.86	0.96	1.34E-08*	3.21	0.45	2.33E-10*	1228.17	511.56	3.60E-10*
SAIST	7.52	1.45	1.72E-03*	3.19	0.47	4.84E-10*	**5752.29**	1142.22	4.48E-10*
PG-GMM	7.17	1.20	8.64E-06*	3.22	0.48	1.09E-09*	995.42	314.31	5.21E-13*
BM4D	7.07	0.81	1.67E-09*	3.31	0.44	4.73E-10*	1037.98	262.66	4.02E-14*
SSR	8.04	0.92	1.40E-03*	3.57	0.49	5.74E-06*	5225.34	3236.76	5.57E-03*
MIFCN	**8.38**	0.94		**3.75**	0.52		2750.75	400.98	

Table II

Mean and standard deviation (SD) of the PSNR (dB) results for denoising 18 foveal images by the compared methods. Where p<0.05, the metrics for each test method are considered statistically significant and were marked by "*". Best results in the mean values are shown in bold.

Method	PSNR		
	Mean	SD	*p* value
K-SVD	26.21	2.68	1.54E-06*
BM3D	26.18	2.68	1.60E-07*
SAIST	26.15	2.73	1.30E-06*
PG-GMM	26.08	2.67	6.96E-07*
BM4D	26.66	2.74	2.41E-07*
SSR	27.23	2.86	9.48E-01
MIFCN	**27.37**	2.73	

We reported the average run-time (in seconds) for denoising by the compared methods in Table III. All of the reported experiments were conducted on a desktop PC with an Intel® i7-7700K CPU at 4.2 GHz, 16 GB of RAM, and a GPU of NVIDIA GeForce GTX 1080 Ti. According to this table, the proposed MIFCN method has the least run-time both on CPU and GPU.

Table III

Average Runtime (seconds) for denoising 18 foveal images by the compared methods. Best result is shown in bold.

Method	Runtime (seconds)	
K-SVD	28.61	
BM3D	6.89	
SAIST	69.30	
PG-GMM	52.72	
BM4D	46.88	
SSR	16.31	
MIFCN	CPU:	1.008
	GPU:	**0.064**

5.6 Effects of different values of the parameter h

The aim of our proposed MIFCN method is to reduce the noise of an OCT image by exploiting the correlation among some nearby OCT images. To this end, the constant h in Equation (7) controls the amount of contributions from other branches (or nearby images). A few visual results are reported in Fig. 4. If a very small value is selected for h (Fig. 4 (C)), the final result ($\hat{X}_R$) is almost identical to the prediction of the main branch ($\hat{X}_1$). The reason is that very small weights are assigned to each pixel of other branches. As we increase this constant, we can continuously increase the amount of contributions from other denoised nearby images. Comparing Fig. 4 (D) and (F) clearly shows that in Fig. 4 (D) small features become blurry and the reconstruction result suffers from artifacts.

Since the constant value of h has a significant effect on the final result, we experimentally study effects of different values for this parameter.

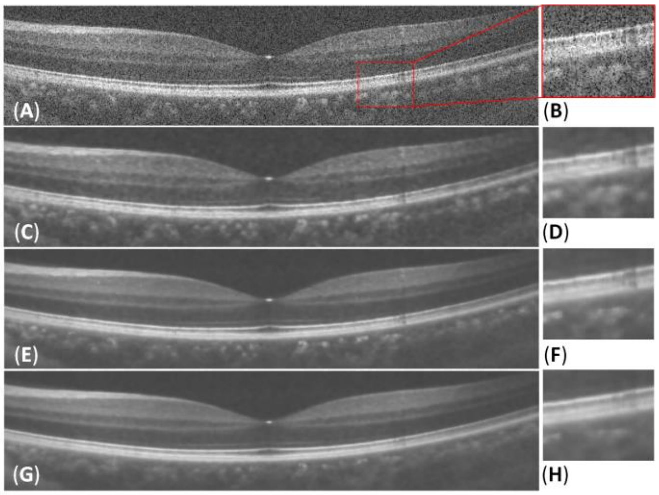

Fig. 4. The effect of different values of the parameter h for denoising a retinal OCT image. The second column shows a magnified region. (A), (B) Original Noisy Image; (C), (D) MIFCN's output using h = 1; (E), (F) MIFCN's output using h = 400; and (G), (H) MIFCN's output using h = 2000. The figure is better seen by zooming on a computer screen.

Quantitative metrics may also help in selecting a suitable value for *h*. In Table IV, we reported the mean values of some quantitative metrics which were obtained using different values of the parameter h. However, each metric has its own merits and drawbacks. Therefore, visual inspection is still the most prominent tool.

Table IV

Mean of the MSR, CNR, ENL, and PSNR results for denoising 18 foveal images using different values of the parameter h for the proposed MIFCN method.

h value	MSR	CNR	ENL	PSNR
1	7.40	3.53	735.46	26.77
100	7.87	3.63	2557.01	27.14
200	8.12	3.69	2712.91	27.26
300	8.27	3.72	2744.02	27.33
400	8.38	3.75	2750.75	27.37
500	8.45	3.77	**2753.79**	27.39
600	8.51	3.79	2751.27	27.41
700	8.56	3.80	2748.64	27.42
800	8.59	3.82	2749.03	27.42
900	8.63	3.83	2748.38	27.43
1000	8.65	3.84	2747.28	27.43

For an ideal metric, it is natural to expect that by increasing the value of *h*, that metric's value increases until a certain point, then it starts to fall down. This is because increasing the value of *h* results in more contributions from other nearby images and this might reconstruct an image which is not faithful to the main input image. From Table IV, one can see that the MSR and CNR metrics does not help in our experiment since they are continuously increasing. However, the ENL and PSNR give more relevant results which are more consistent with the visual results in Fig. 4. For example, when the parameter *h* is greater than 500 in Table IV, the ENL value starts to decrease and the PSNR value remains almost unchanged. Figure 4 (F) and (H) also demonstrate that when *h* is much larger than 500, the small features become less visible, and the result becomes blurry. Therefore, for a given dataset of

OCT images, both the quantitative metrics and the visual quality of the reconstruction results must be considered to find a suitable value for h. Overall, we have used h = 400 for all of the experiments that are reported in Section 5.5.

5.7 Effects of the number of layers

In this section, we experimentally show how the number of convolution layers could affect the performance of the proposed MIFCN method. In the proposed MIFCN method's architecture (Figs. 1 and 2), the main ingredients are 3×3 convolution followed by LReLU activation layers, 1×1 convolution layers, and a pixel by pixel averaging module. Here, we are interested in varying the number of 3×3 convolution layers while keeping all other things unchanged.

In Table V, the mean squared error (MSE) of five different configurations are reported. These configurations are indicated by MIFCN-A-B, where A shows the number of 3×3 convolution layers for each branch (Fig. 2) and B shows the number of 3×3 convolution layers right after the pixel by pixel averaging module (Fig. 1). We have used a similar training set (Section 5.1) and training procedure (Section 4) for learning the parameters of each configuration. However, in this section, a different evaluation strategy is used. For evaluating each trained model, we used the learned model for reconstructing both the test and training sets. This is because changing the number of layers might easily lead to over fitting or under fitting. Therefore, evaluating errors for the training and test sets can give us more insights into the performance of a model.

Table V

MSE for the training and test sets. The best test set MSE is shown in bold.

Configuration	Training set	Test set
MIFCN-3-0	84.64	120.55
MIFCN-3-1	84.83	**119.23**
MIFCN-3-2	85.64	120.28
MIFCN-4-1	79.7	120.12
MIFCN-2-1	94.18	123.13

In Table V, the configuration indicated by MIFCN-3-1 shows our main model. This configuration is used in the previous sections. Removing the convolution layer after the pixel by pixel averaging module leads to a model (MIFCN-3-0) with slightly inferior test performance. The configurations MIFCN-3-2 and MIFCN-4-1 show that adding more convolution layers cannot improve the performance of the main model. This is due to lack of training data (as it is explained in Section 5.1, there are only 10 training image pairs). The MIFCN-4-1 configuration clearly shows that the model needs more data since the MSE for the training set decreases significantly but the test error does not improve. The last configuration in Table V shows that removing one convolution layer from each branch leads to a model (MIFCN-2-1) with limited modeling capacity. For this model, the MSEs for the training and test sets are significantly higher than the other configurations. These quantitative comparisons show that the main model (MIFCN-3-1) offer a good trade-off between performance and complexity.

6. CONCLUSION

In this paper, we propose a new deep learning method (named MIFCN) for denoising SDOCT images. The proposed MIFCN method exploits a weighted averaging module which is inspired by the nonlocal mean method [19] to effectively capture useful information from nearby OCT images. We show how the parameters of the proposed MIFCN architecture can be learned in an end-to-end manner. We have also conducted extensive experiments to compare the proposed MIFCN method with some of the well-known state-of-the-art methods. The experimental results show the effectiveness of the proposed MIFCN method over the compared methods. These results also show that the proposed MIFCN method can effectively reduce noise while preserving textures and layer structures. Interestingly, our proposed MIFCN produces less artifacts compared to other competing methods. Therefore, the proposed MIFCN method is not only useful for OCT image quality improvement but also it might be a good preprocessing step for retinal layers segmentation methods. In the future, we would like to incorporate segmentation information into our proposed MIFCN method [14]. Also, we would like to extend the proposed MIFCN method for OCT image interpolation [14,15]. In addition, although we only considered the task of retinal OCT image denoising, the proposed MIFCN method might also be applied to denoising of other medical images.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grants 61501180 and 61771192.

REFERENCES

[1] H. Rabbani, R. Kafieh, Z. Amini, Optical Coherence Tomography Image Analysis, in: Wiley Encycl. Electr. Electron. Eng., John Wiley & Sons, Inc., Hoboken, NJ, USA, 2016: pp. 1–16.

[2] M. Szkulmowski, M. Wojtkowski, Averaging techniques for OCT imaging, Opt. Express. 21 (2013) 9757–9773.

[3] A. Baghaie, Z. Yu, R.M. D’souza, Involuntary eye motion correction in retinal optical coherence tomography: Hardware or software solution?, Med. Image Anal. 37 (2017) 129–145.

[4] A. Ozcan, A. Bilenca, A.E. Desjardins, B.E. Bouma, G.J. Tearney, Speckle reduction in optical coherence tomography images using digital filtering, J. Opt. Soc. Am. A. 24 (2007) 1901–1910.

[5] A. Paul, D.P. Mukherjee, S.T. Acton, Speckle Removal Using Diffusion Potential for Optical Coherence Tomography Images, IEEE J. Biomed. Heal. Informatics. (2018), Early Access.

[6] A. Pizurica, L. Jovanov, B. Huysmans, V. Zlokolica, P. De Keyser, F. Dhaenens, W. Philips, Multiresolution Denoising for Optical Coherence Tomography: A Review and Evaluation, Curr. Med. Imaging Rev. 4 (2008) 270–284.

[7] H. Rabbani, R. Nezafat, S. Gazor, Wavelet-Domain Medical Image Denoising Using Bivariate Laplacian Mixture Model, IEEE Trans. Biomed. Eng. 56 (2009) 2826–2837.

[8] Z. Amini, H. Rabbani, Classification of medical image modeling methods: a review, Curr. Med. Imaging Rev. 12 (2016) 130–148.

[9] S. Roth, M.J. Black, Fields of Experts, Int. J. Comput. Vis. 82 (2009) 205–229.

[10] M. Elad, M. Aharon, Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries, IEEE Trans. Image Process. 15 (2006) 3736–3745.

[11] G. Yu, G. Sapiro, S. Mallat, Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity, IEEE Trans. Image Process. 21 (2010) 2481–2499.

[12] L. Fang, S. Li, Q. Nie, J.A. Izatt, C.A. Toth, S. Farsiu, Sparsity based denoising of spectral domain optical coherence tomography images, Biomed. Opt. Express. 3 (2012) 927–942.

[13] L. Fang, S. Li, R.P. McNabb, Q. Nie, A.N. Kuo, C.A. Toth, J.A. Izatt, S. Farsiu, Fast Acquisition and Reconstruction of Optical Coherence Tomography Images via Sparse Representation, IEEE Trans. Med. Imaging. 32 (2013) 2034–2049.

[14] L. Fang, S. Li, D. Cunefare, S. Farsiu, Segmentation Based Sparse Reconstruction of Optical Coherence Tomography Images, IEEE Trans. Med. Imaging. 36 (2017) 407–421.

[15] A. Abbasi, A. Monadjemi, Optical coherence tomography retinal image reconstruction via nonlocal weighted sparse representation, J. Biomed. Opt. 23 (2018) 036011-1–11.

[16] R. Kafieh, H. Rabbani, I. Selesnick, Three Dimensional Data-Driven Multi Scale Atomic Representation of Optical Coherence Tomography, IEEE Trans. Med. Imaging. 34 (2015) 1042–1062.

[17] M. Niknejad, H. Rabbani, M. Babaie-Zadeh, C. Jutten, Image Restoration Using Gaussian Mixture Models With Spatially Constrained Patch Clustering, in: IEEE Trans. Image Process., IEEE, 2015: pp. 3624–3636.

[18] Z. Amini, H. Rabbani, Statistical Modeling of Retinal Optical Coherence Tomography, IEEE Trans. Med. Imaging. 35 (2016) 1544–1554.

[19] A. Buades, B. Coll, J.-M. Morel, A Non-Local Algorithm for Image Denoising, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2005: pp. 60–65.

[20] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Proc. Adv. Neural Inf. Process. Syst., 2012: pp. 1097–1105.

[21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016: pp. 770–778.

[22] V. Jain, S. Seung, Natural Image Denoising with Convolutional Networks, in: Proc. Adv. Neural Inf. Process. Syst., 2009: pp. 769–776.

[23] H.C. Burger, C.J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with BM3D?, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012: pp. 2392–2399.

[24] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering, IEEE Trans. Image Process. 16 (2007) 2080–2095.

[25] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, IEEE Trans. Image Process. 26 (2017) 3142–3155.

[26] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning Deep CNN Denoiser Prior for Image Restoration, ArXiv Prepr. (2017).

[27] J. Wu, R. Timofte, Z. Huang, L. Van Gool, On the Relation between Color Image Denoising and

Classification, ArXiv Prepr. (2017).

[28] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature. 521 (2015) 436–444.

[29] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2015: pp. 3431–3440.

[30] F. Yu, V. Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, ArXiv Prepr. (2015).

[31] Q. Chen, J. Xu, V. Koltun, Fast Image Processing with Fully-Convolutional Networks, in: Proc. IEEE Int. Conf. Comput. Vis., IEEE, 2017: pp. 2516–2525.

[32] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier Nonlinearities Improve Neural Network Acoustic Models, in: Proc. Int. Conf. Mach. Learn., 2013.

[33] A. Buades, B. Coll, J.M. Morel, Denoising image sequences does not require motion estimation, in: Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill., 2005: pp. 70–74.

[34] M.D. Robinson, S.J. Chiu, C. a Toth, J. a Izatt, J.Y. Lo, Novel Applications of Super-resolution in Medical Imaging, in: P. Milanfar (Ed.), Super-Resolution Imaging, CRC Press, 2010: pp. 384–412.

[35] M. Protter, M. Elad, H. Takeda, P. Milanfar, Generalizing the nonlocal-means to super-resolution reconstruction, IEEE Trans. Image Process. 18 (2009) 36–51.

[36] D. Kingma, J. Ba, Adam: A method for stochastic optimization, ArXiv Prepr. (2014).

[37] G. Cincotti, G. Loi, M. Pappalardo, Ultrasound Medical Images With Wavelet Packets, IEEE Trans. Med. Imaging. 20 (2001) 764–771.

[38] P. Bao, Lei Zhang, Noise reduction for magnetic resonance images via adaptive multiscale products thresholding, IEEE Trans. Med. Imaging. 22 (2003) 1089–1099.

[39] W. Dong, G. Shi, X. Li, Nonlocal image restoration with bilateral variance estimation: A low-rank approach, IEEE Trans. Image Process. 22 (2013) 700–711.

[40] J. Xu, L. Zhang, W. Zuo, D. Zhang, X. Feng, Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising, in: Proc. IEEE Int. Conf. Comput. Vis., IEEE, 2015: pp. 244–252.

[41] M. Maggioni, V. Katkovnik, K. Egiazarian, A. Foi, Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction, IEEE Trans. Image Process. 22 (2013) 119–133.