

Super Resolution of B-Mode Ultrasound Images With Deep Learning

HAKAN TEMİZ¹, (Graduate Student Member, IEEE), AND **HASAN S. BILGE²**, (Member, IEEE)

¹Artvin Vocational High School, Artvin Coruh University, 08000 Artvin, Turkey

²Department of Electrical and Electronics Engineering, Gazi University, 06570 Gazi, Turkey

Corresponding author: Hakan Temiz (htemiz@artvin.edu.tr)

This work was supported by the Scientific and Technical Research Council of Turkey under Grant 119E015.

ABSTRACT Ultrasound offers a safe, non-invasive, and inexpensive way of imaging. However, due to its natural intrinsic imaging characteristics, it produces poor quality images with low resolution (LR) compared to other medical imaging modalities. Various image enhancement techniques have been extensively studied to overcome these shortcomings. Super-resolution (SR) is one of these methods, which endeavor to obtain high resolution (HR) images from LR images while enlarging them. Numerous studies have already utilized different SR techniques in various stages of ultrasound imaging (USI). Unlike other studies, which aimed at obtaining SR in the pre-processing phase or early stages of the post-processing phase of USI, we achieved SR on B-mode ultrasound images, which is the last stage of USI. We constructed a deep convolutional neural network (CNN) and trained it with a very large dataset of B-mode ultrasound images for the scale factors 2, 3, 4, and 8. We evaluated the performance of our proposed model quantitatively with eight image quality measures. The quantitative results revealed that our algorithm is much more successful than other methods at each magnification factor. Furthermore, we also verified that there is a statistically significant difference between our approach and others. Besides, qualitative analysis of the reconstructed images also confirms that it produces much better quality HR images than other methods in terms of the human visual system.

INDEX TERMS Ultrasound, super-resolution, deep learning, convolutional neural network.

I. INTRODUCTION

Ultrasound imaging (USI) has become a standard imaging modality since it enables to execute a safe, non-invasive, and portable way of imaging [1]. Besides, it is an inexpensive modality compared to other imaging modalities, such as computerized tomography (CT), X-ray, and magnetic resonance imaging (MRI). In medical imaging, it is nearly the most preferred method for imaging of the cardiovascular system, abdominals, urology, vascular system, obstetrics, gynecology, and so on. On the other hand, USI suffers from the artifacts emerging during the travel of sound waves through the medium. A typical cause of artifacts is the transmission of ultrasound signals in a short duration in time. Thus, the signals are affected by multiple noise sources [2]. The speckle noise is another fundamental problem destructively affecting ultrasound images. The speckle is the outcome of the coherent interface of constructive and destructive backscattered echoes. Inhomogeneity of tissues also distorts sound waves.

The associate editor coordinating the review of this manuscript and approving it for publication was Gulistan Raja¹.

A variety of techniques have been extensively investigated to overcome the poor image quality in USI. The main goal of these techniques involves the reduction of speckle noise, removal of blurring effect, preserving high-frequency components, and enhancement of the resolution [2]. Improving the resolution is referred to as super-resolution (SR). The SR aims to produce finer details missing in the LR image while enlarging it by a scale factor to obtain a high resolution (HR) image. In this study, we obtained SR on B-mode ultrasound images. Almost 28 thousand ultrasound images were employed for training a deep convolutional neural network (CNN) to produce HR images. We denominate our proposed model as DECUSR¹ (deep convolutional network for ultrasound super-resolution). We compared DECUSR with four well-known interpolation methods (Bicubic, Bilinear, Lanczos, and Nearest interpolation), the method proposed in [3], and two deep learning (DL) models: super-resolution convolutional neural network (SRCNN) [4], and enhanced deep super-resolution network (EDSR) [5]. We used eight

¹Source codes are available at <https://github.com/htemiz/DECUSR>

image quality measures (IQMs) in our quantitative evaluations. We used two separate test sets of images from two different sources. The findings in both test sets reveal that our method surpasses other methods significantly. We confirmed that there is a statistically significant difference between our method and others.

The qualitative comparisons of HR images reconstructed by each algorithm also confirm that our method produces much better quality images than others. In other words, the proposed model greatly surpasses other methods in terms of the human visual system (HVS) as well.

As a summary, the main contributions of this paper are as follows:

- We obtain SR at multiple scales factors (2, 3, 4, and 8).
- We developed a novel CNN model that offers high-quality ultrasound image reconstruction.
- We compared our model with other models by using eight IQMs, and also qualitatively in terms of the human visual system.
- We also checked the statistical significance of our method against other methods.

The rest of the paper is organized as follows. Section 2 gives a summary of related works previously done. In Section 3, the architecture of the proposed model is described in detail. A comprehensive description of the experiments and results is given in Section 4. In the conclusion section, we briefly outline and discuss the contributions of this paper.

II. RELATED WORK

SR is being employed for increasing resolution while preserving or even improving details in images. It can be implemented on still images or multiple images in a sequence. Multi-frame SR algorithms first register the sequences of images to estimate relative motion. The relevant sequences of the same scene are then projected into a coordinate system introducing an interpolation method. As a final step, some image enhancement algorithms are applied to interpolate finer details to obtain better image quality.

On the other hand, a single image super-resolution (SISR) aims at obtaining an HR image from a single LR image. In this study, we adopted a learning-based SISR approach by exploiting a CNN. In the next two sections, we list related works on SR in two different topics: SR in real-world imaging, and USI.

A. SUPER RESOLUTION IN REAL WORLD IMAGING

Plentiful methods exploiting either single or multiple images have been introduced in the last few decades to solve the SR problem. Such as interpolation-based [6], reconstructional [7], frequency domain applications [8], and in recent studies mostly learning-based methods in SISR [9]–[12]. SRCNN network, proposed by Dong *et al.* [4], is the first DL algorithm applied to SR. After then, many different network architectures have been designed for SISR. The former networks were mostly a kind of CNN with some special architecture

adjustments or loss functions. Some of them used residual learning approach; DRCN [13], VDSR [14], DRRN [15], ESPCN [16], and EDSR [5], which has won the NTIRE 2017 challenge [17], have used the sub-pixel convolutional layer structure. The authors of FSRCNN [18] aimed to implement a fast and efficient algorithm. Densely connected layers were used with DenseNet [19] and RDN [20]. MemNet [21] proposed a special architecture consisting of the recursive unit and a gate unit. Laplacian pyramid architecture was used in LapSRN [22]. EnhanceNet [23] has focused on creating realistic textures. Very recent special CNNs are HCNN [24] introducing hierarchical structure, and MGEP-SRCNN [25], which proposes multilabel gene expression programming (MGEP).

Apart from CNN architecture, especially in recent studies, generative adversarial networks (GANs) have exhibited outstanding performance in SISR. The authors of SRGAN [26] intended to recover finer texture details, especially at large upscaling factors, by introducing a perceptual loss function established with an adversarial loss and a content loss. Reference [27] derived ESRGAN from SRGAN. RTSRGAN [28] combined the advantages of ESRGAN and the real-time performance of ESPCN. DGAN [29] used multi generators in the network structure. Cycle-in-cycle GAN is recently used in MCinCGAN [30]. CGAN [31] integrated Laplacian pyramid architecture into GAN. Wasserstein GAN architecture is utilized in WGAN [32]. Feature-guided SR is proposed in FG-SRGAN [33] by pointing out that it is impractical to super resolve LR images due to the absence of ground-truth HR images in the real world. The authors of GCN [34] proposed collaborative network architecture. Gradient Magnitude Similarity Deviation (GMSD) metric is adapted to GMGAN [35] in order to produce HR images in line with the human vision system (HVS). The authors of SRNTT [36] focused on the information loss in LR images, and, proposed a reference-based SR approach for generating better texture details from reference images. A probabilistic generative framework, PGM, which offers the low computational cost and robustness to noise, is proposed in [37]. Reference [38] proposed G-GANISR exploiting the least square loss function instead of cross-entropy. The authors targeted to exhaust all image details without loss of information by gradually increasing the charge of the discriminator.

B. SUPER RESOLUTION IN ULTRASOUND IMAGING

We summarize here the SR studies implemented in the post-processing phase of USI since the pre-processing phase of USI is out of the scope of this paper. A multi-frame SR task was experimented in [39]. The authors mainly focused on the registration step to get a better ultrasound image quality. RF line sequences were used for bilinear deformable block matching (BDBM), which was adapted for motion estimation and small local deformations. Very similar implementation was also performed in [40]. To measure the thickness of intima-media more accurately, it is aimed at [41] to reduce speckle noise while obtaining an HR image by utilizing SR.

For this purpose, anisotropic diffusion regularization was performed by employing a maximum a-posteriori (MAP) approach with transformation information in the frequency domain. In [42], a patch-based Gaussian process regression with multiple annotators (GPRMA) was performed for computing an estimated actual HR image. The authors endeavored to minimize the evidence logs utilizing gradient descend with a radial basis function (RBF) kernel for parameter estimation. Instead of utilizing a single image in breast tumor classification, a multi-image SR approach using complementary information from multiple images before tissue analysis is recommended in [43]. Because ℓ_1 -norm was proven to be more robust in comparison to ℓ_2 -norm, ℓ_1 -norm was chosen as a similarity cost function with total bilateral variation (BTV) regularization. The authors achieved the best results for all texture methods, especially for 5 LR images. Reference [44] proposed a ℓ_p -norm regularizer for ultrasound tissue reflectivity function for motion estimation. The researchers proposed an approach to decimation and blurring operators for solving the related optimization problem.

The authors in [45], implemented a two-dimensional homomorphic deconvolution filter, ensuring real-time processing to the first and second harmonics in the RF image array. As a result, the second-harmonics resulted in better spatial resolution than the first-harmonics.

A sparse coding scheme was implemented in [46] with non-local structural similarity regularizer based on the fact that medical images often have many repetitive image structures. To improve the quality of the constructed HR image, the researchers adapted a non-local similarity constraint into the patch aggregation process after finding similar patches by the Gaussian neighborhood. The proposed algorithm was applied to MRI, CT, and USI. Another research for SR with sparse representation was carried out in [42] for the reconstruction of ultrasound images. Reference [47] employed the idea that RF signal envelopes have to be improved to increase the resolution of images. The authors implemented a multi-dimensional autoregressive (AR) modeling using inverse Fourier transform of non-demodulated In-phase and Quadrature (IQ) signals, instead of directly estimating the envelopes of RF signals. Another experiment in [48] was performed with the Alternating Direction Method of Multipliers (ADMM) technique on IQ signals.

Yoon and Ye [49] utilized a CNN model for achieving SR on randomly subsampled RF data, rather than using every bit of it since there is only a little incremental change in the sequences of RF signals. In this way, they achieved fast run-time reconstruction of ultrasound images. Reference [3] employed a weighted regression for SR for a 2x scale. Another DL architecture akin to the optical flow method was designed in [50] for implementing the multi-frame SR approach with motion estimation. The researchers specifically focused on the motion estimation step of the multi-frame SR task for enhancing the quality of images. Reference [51] employed an SRGAN network with sub-pixel feature channels for generating intravascular ultrasound images.

Reference [52] also used SRGAN with small modifications to achieve SR on the lateral axis of USI. The authors obtained SR on RF data before scan conversion to B-mode for a 4x scale. An unsupervised learning-based method with dilated CNN and residual learning for a single ultrasound image has been proposed by Lu and Liu [53]. The method does not require a training dataset since it learns the LR-HR map from the given image. Reference [54] utilized repetitive data in the non-local neighborhood of samples by first denoising image and then applying a Bayesian approach for SR. Another Bayesian approach with the SR technique is implemented in [55] for ultrasound image restoration.

III. METHOD

In this study, we attained the SR of ultrasound images by using a CNN. We empowered our model to learn the functional relationship between LR and HR images by training it with a large training set of B-mode ultrasound images. We will explain the entire process in detail in the next sections.

A. SUPER RESOLUTION

The super-resolution (SR) is a process of obtaining high resolution (HR) images from a single or multiple low resolution (LR) images generated from either multiple observations of the same scene or produced synthetically from HR images. It strives to produce finer details as well while improving the actual resolution. Thanks to successful results, it has been used for many different applications, such as medical image processing, image conversion from standard-definition television (SDTV) to high definition television (HDTV), image mosaicking, aerial, and satellite imaging, and image enhancement [56]. There are two main attitudes of the SR task: software-based methods and hardware-based methods. One of the hardware-based methods aims to decrease the pixel size beyond a certain threshold. Current technologies have already reached the limit of such threshold sizes. Besides, decreasing the pixel size gives rise to a drop in the amount of light reaching the corresponding cell of the pixel on the sensor. Subsequently, it causes an increase of the shot noise. The other member of the hardware-based method aims to increase the capacitance of the imaging device. That is, it finally slows down the charge transfer rate. In addition to the issues mentioned above, hardware-based approaches are generally expensive, especially in large-scale imaging devices. Therefore, algorithmic-based methods are more preferred than hardware-based approaches.

The task of obtaining LR images from HR images is formulated with the following definition:

$$x(i, j) = D(B(M(y))) + \eta(i, j) \quad (1)$$

where x and y are LR and HR images, respectively. M is a warping function, B is a blurring function, and D is a downsampling function. η stands for additive noise. As seen from (1), the SR task is an inverse problem that HR images can be estimated from LR images. From given HR images,

LR images are simulated with a series of operations defined in Eq. (1). Afterward, the missing information between LR and HR images is being estimated using the relationships between them. Very recent algorithms try to recover these relations by exploiting a variety of learning algorithms. Likewise, the proposed model is also a learning algorithm that aims to comprehend these relationships between LR and HR images.

B. OUR PROPOSED METHOD

We provide theoretical background, implementation, and training details of our proposed algorithm in the next sections.

1) MOTIVATION

A convolutional neural network is comprised of convolutional layers with a particular activation function and parameters $\Theta = \{\mathcal{W}, \mathcal{B}\}$. Where \mathcal{W} is a set of convolution kernels (filters), and \mathcal{B} is a set of biases. At each layer, the input image \mathbf{x} is convolved with a set of K kernels $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_K\}$ and added biases $\mathcal{B} = \{b_1, b_2, \dots, b_K\}$. The convolution operation at each layer generates a new feature map \mathbf{x}_k . Afterward, these features are subjected to an activation function, which results in an element-wise non-linear transform. Such that, $a = \sigma(\mathbf{W} * \mathbf{x} + b)$. Here, $*$ and σ respectively denote the convolution operation and activation function.

The same process is repeated for \mathbf{x} and Θ at every convolutional layer l :

$$\mathbf{x}_k^l = \sigma(\mathbf{W}_k^{l-1} * \mathbf{x}^{l-1} + b_k^{l-1}) \quad (2)$$

Consequently, a convolutional neural network with an L number of layers is given as:

$$\begin{aligned} f(\mathbf{x}; \Theta) \\ = \sigma(\mathbf{W}^L * \sigma(\mathbf{W}^{L-1} \dots \sigma(\mathbf{W}^0 * \mathbf{x} + b^0) + b^{L-1}) + b^L) \end{aligned} \quad (3)$$

where, \mathbf{W}^m represents the K number of kernels in layer n .

Our proposed model, as a CNN, maps a given LR image \mathbf{x} to estimated HR image $\hat{\mathbf{y}} = f(\mathbf{x}; \Theta)$ that is closed to the ground-truth image \mathbf{y} . For the optimization of the network error over given a training dataset $D \{x^{(i)}, y^{(i)}\}_{i=1}^N$, we define a loss function such that:

$$L(\mathbf{y}, \hat{\mathbf{y}}; \Theta) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i - f(\mathbf{x}_i; \Theta)\|^2 \quad (4)$$

where N is the number of training samples. In (4) we introduce the mean-squared-error (MSE) for computing the loss value by quantifying the discrepancy or distance between each predicted image $\hat{\mathbf{y}}$ and its corresponding ground-truth image \mathbf{y} . MSE defined as follows:

$$MSE = \frac{1}{PR} \sum_{i=1}^P \sum_{j=1}^R \|y(i, j) - \hat{y}(i, j)\|^2 \quad (5)$$

where P and R are pixel sizes in both direction, height, and width. So, it is required here to minimize the difference between $\hat{\mathbf{y}}$ and \mathbf{y} . That is, the network must learn to accurately predict the actual output \mathbf{y} from given LR image \mathbf{x} as much as possible.

Including a regularization term, we define an objective function such that;

$$\mathcal{J}(\Theta) = \frac{1}{N} \sum_i^N \mathcal{L}(\mathbf{x}^i) + \lambda r(\Theta) \quad (6)$$

where r is a regularization term with weight $\lambda > 0$, and \mathcal{L} is the loss function defined in (4). Now that our goal is to minimize the objective function \mathcal{J} . In other words, we would like to find our network's parameters $\Theta = \{\mathcal{W}, \mathcal{B}\}$, such that make the objective function be the minimum. This optimization problem could be solved with the gradient descent algorithm.

Since our training set D is too large for computation, we use a mini-batch training method that makes our objective function as follows:

$$\mathcal{J}(\Theta) \approx \frac{1}{|M|} \sum_i^{|M|} \mathcal{L}(\mathbf{x}^i) + \lambda r(\Theta) \quad (7)$$

Each time we draw $M \ll N$ number of samples (128 samples, in our case) from D , and compute \mathcal{L} in the forward pass, and the gradient $\nabla \mathcal{L}$ in the backward pass. The parameter update $\nabla(\mathcal{W}, \mathcal{B})$ is found from the error gradient $\nabla \mathcal{L}$ and regularization gradient ∇r . For updating the parameters $\Theta = \{\mathcal{W}, \mathcal{B}\}$, we used a well-known gradient-based optimization method, ADAM [57]. It is a first-order gradient-based optimization algorithm for stochastic objective functions.

2) IMPLEMENTATION OF THE NETWORK

We dominate our method as DECUSR (Deep Convolutional Network for Ultrasound Super Resolution) for brevity. It essentially consists of three different components: feature extraction block (FEB), repeating blocks (RBs), and upsampling layers. There are three different types of layers in the network: convolution, upsampling, and concatenation. We present the architecture in Fig. 1. Each convolution layer has 16 filters except for the last one, which has only one 1×1 filter. For a convolutional layer in our network, we can write (2) as follows since we use ReLU as the activation function:

$$\mathbf{x}_k^l = \text{ReLU}(\mathbf{W}_k^{l-1} * \mathbf{x}^{l-1} + b_k^{l-1}) \quad (8)$$

where $k = 1$ for the last layer and $k = 16$ for the rest.

Traditional networks generally have a single upsampling layer in the case that the network upscales input image by itself. In contrast, others intake an LR image, which is already-upscaled by an interpolation method. The critical question is, at what stage and how to upscale LR image? E.g., directly upscaling after the input layer, or after a kind of feature extraction process. We think that there will be some amount of information loss in either way. Hence, we employed two separate upsampling layers in our network: feature upscaling layer L_{FUP} and direct upscaling layer L_{DUP} . L_{FUP} enlarges the extracted features generated by FEB, whereas L_{DUP} directly enlarges the LR image. We observed in our experiments that both upsampling layers positively contribute to the network.

Similar to the idea of densely connected layers [19], we adopted densely connected RBs to better propagate the

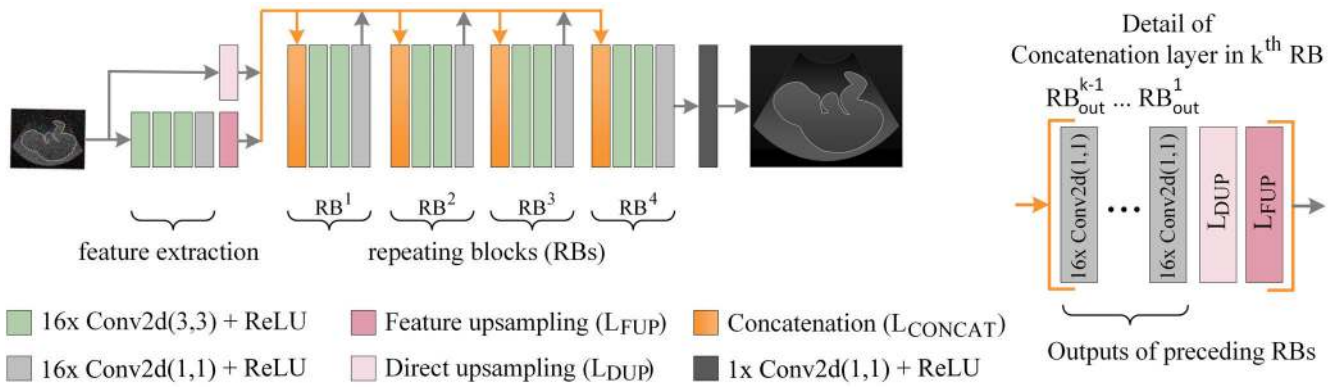


FIGURE 1. The architecture of our proposed model.

information throughout the network. We use concatenation layers in RBs for dense connections. There are also three subsequent convolutional layers in RBs. The first and second convolution layers have $16 \times 3 \times 3$ filters, whereas the last one has $16 \times 1 \times 1$. A concatenation layer (L_{CONCAT}) combines the outputs of both the upsampling layer (L_{DUP} and L_{FUP}) and the outputs of each preceding RB. The accumulated information by the help of the concatenation in the RBs enables the network to learn the non-linear transition from LR to HR images in an efficient and balanced manner. In this way, our model becomes more robust. Besides, it learns much better. In the next section, we explain the mechanism of RB with the concatenation in detail.

a: REPEATING BLOCKS AND CONCATENATION

Let \mathcal{X}_l and $\mathcal{T}_l \in \mathbb{R}^{w \times h \times k}$ respectively be the output and the feature map (a tensor in our case) of ℓ^{th} layer in a network. Here, w , h , and $k \in \mathbb{Z}^+$, and respectively denote the width, height, and depth of the layer. Typically, each layer in ordinary CNN implements a non-linear transformation $\mathcal{F}(\cdot)$. A traditional CNN deems the output of the ℓ^{th} layer as the input of the $(\ell + 1)^{\text{th}}$ layer. Such that, the output of the ℓ^{th} layer can be formulated as $\mathcal{X}_l = \mathcal{F}_l(\mathcal{X}_{l-1})$. Unlike ordinary CNN layers, a concatenation layer stacks its inputs into a single tensor in the same order. For a n number of input feature maps, the output of a concatenation layer can be written as $\mathcal{X}_{concat} = \mathcal{F}_{concat}(\mathcal{T}_1^{w \times h \times k_1}, \mathcal{T}_2^{w \times h \times k_2}, \dots, \mathcal{T}_n^{w \times h \times k_n})$, which produces a new feature map $\mathcal{T}^{w \times h \times (k_1 + k_2 + \dots + k_n)}$.

The first layer in an RB is a concatenation layer, which stacks L_{FUP} , L_{DUP} , and the outputs of each preceding RBs in the given order. Namely, k^{th} repeating block RB^k is connected to L_{FUP} , L_{DUP} , and $RB_{out}^{k-1}, \dots, RB_{out}^1$ via its concatenation layer. We initialed the baseline model with three RBs and endeavored in our experiments to find out the optimum number of them. According to our evaluations, our network achieves the top performance with four RBs. We discuss this later in the next sections.

b: DISSECTING THE COMPONENTS

Before finding out how many RBs should exist, we examined the contributions of the components of the network. For this

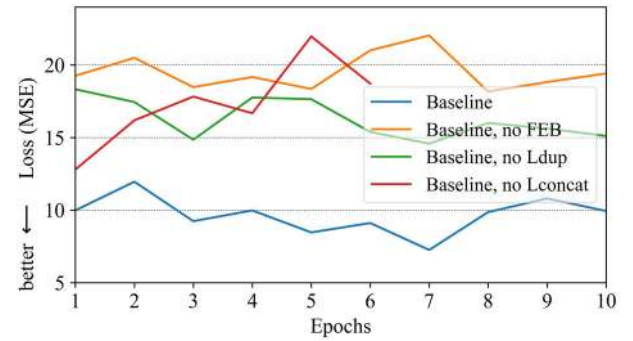


FIGURE 2. Performances of the baseline model (3 RBs) in the presence or absence of its components.

TABLE 1. Components of the baseline model (with 3 RBs) and their contributions to the performance.

Component	✓	✓	✗	✓
Concatenation layer (L_{CONCAT})	✓	✓	✗	✓
Feature extraction block (FEB)	✓	✗	✓	✓
Direct upsampling layer (L_{DUP})	✗	✗	✗	✓
PSNR (dB)	43.8772	43.8147	43.5736	43.9777

purpose, we trained our baseline model (with 3 RBs) in the presence and/or absence of FEB, L_{FUP} , L_{DUP} , L_{CONCAT} for a 2x scale. We give training graphs for each scenario in Fig. 2. The MSE loss of the baseline model in the absence of FEB, L_{DUP} , and L_{CONCAT} is shown with orange, green, and red colors, respectively. The blue line shows the baseline in the presence of all components. The loss is minimum when all components are present. The figure clearly shows that their presence positively contributes to the network.

Table 1 also shows PSNR scores of each scenario obtained in TestSet-I. According to the results, L_{DUP} slightly improves the network's performance (0.1 dB). PSNR score is 43.5736 (dB) in the absence of the layers L_{CONCAT} and L_{DUP} , whereas 43.8772 (dB) only in the absence of L_{DUP} . These results mean that the concatenation layers considerably contribute to the network's performance (by 0.3036 dB PSNR). Likewise, PSNR score is 43.8147 (dB) in the

TABLE 2. Layer details of our network with four RBs.

Layer (type)	Output Shape s(wxh)xFeature #	Parameter #	Connected to
main_input	(wxh)x1	0	
Conv2D_01	(wxh)x16	160	main_input
Conv2D_02	(wxh)x16	2320	Conv2D_01
Conv2D_03	(wxh)x16	2320	Conv2D_02
Conv2D_04	(wxh)x16	272	Conv2D_03
L_FUP	s(wxh)x16	0	Conv2D_04
L_DUP	s(wxh)x1	0	main_input
L_CONCAT_1	s(wxh)x17	0	L_FUP, L_DUP
Conv2D_05	s(wxh)x16	2464	L_CONCAT_1
Conv2D_06	s(wxh)x16	2320	Conv2D_05
Conv2D_07	s(wxh)x16	272	Conv2D_06
L_CONCAT_2	s(wxh)x33	0	L_FUP, L_DUP, Conv2D_07
Conv2D_08	s(wxh)x16	4768	L_CONCAT_2
Conv2D_09	s(wxh)x16	2320	Conv2D_08
Conv2D_10	s(wxh)x16	272	Conv2D_09
L_CONCAT_3	s(wxh)x49	0	L_FUP, L_DUP, Conv2D_07, Conv2D_10
Conv2D_11	s(wxh)x16	7072	L_CONCAT_3
Conv2D_12	s(wxh)x16	2320	Conv2D_11
Conv2D_13	s(wxh)x16	272	Conv2D_12
L_CONCAT_4	s(wxh)x65	0	L_FUP, L_DUP, Conv2D_07, Conv2D_10, Conv2D_13
Conv2D_14	s(wxh)x16	9376	L_CONCAT_4
Conv2D_15	s(wxh)x16	2320	Conv2D_14
Conv2D_16	s(wxh)x16	272	Conv2D_15
Conv2D_17	s(wxh)x1	145	Conv2D_16
Total parameter #		39,265	

Abbreviations: L_FUP, feature upsampling; L_DUP, direct upsampling; L_CONCAT, concatenation layer; s, scale; w, width; h, height.

absence of FEB. Therefore, it contributes to the network by 0.0625 dB PSNR. As a result, we see that each component contributes more or less positively to our network. Based on these observations, we incorporated all components into our network.

c: THE NUMBER OF REPEATING BLOCKS (RBS)

We constructed the baseline network with 3 RBs at the beginning of our experiments. After then, we trained the network by removing existing RBs or plugging new ones. We checked the network's performance after every addition or reduction. We stopped the procedure when the next two additions did not improve performance. In this way, we obtained six different models with a single RB to six RBs, since five and six RBs could not perform better than four RBs. Figure 3 illustrates the training loss in MSE and PSNR scores in TestSet-I by the different number of RBs. Using less than three RBs severely deteriorates the performance.

On the other hand, the network with four RBs converges rapidly in the second epoch and stabilizes the convergence throughout the training. Five RBs also exhibit a very similar learning curve as four RBs, but with slightly lower in the performance. The performance deteriorates noticeably with six RBs. Based on these results, we decided to employ four RBs in our network.

TABLE 3. Training parameters.

Scale	Stride	patch size	Learning rate	Min. L. rate	Batch size	Epoch #	Stopping patience
2	6	16	1.0E-03	1.0E-07	128	10	5 epochs
3	4	11					
4	3	8					
8	1	4					

IV. EXPERIMENTS AND RESULTS

In this section, we discuss the details of the network's architecture, datasets, training, and evaluation.

A. DATASETS

We benefited from two different web sites to create training and validation sets and two separate test sets. In total, we downloaded 36149 B-mod ultrasound images from the former web site, <http://www.ultrasoundcases.info>. We excluded images that are not in good condition. From the remaining 29696 images, we composed the validation and test set 'TestSet-I' by randomly selecting a thousand images for each set, without replacement. Finally, the training set consisted of the remaining 27696 images.

The second test set, 'TestSet-II,' consisted of the images only from the following web site <http://www.ultrasound-images.com>. We cropped randomly selected 500 images so that their size is 600×450 pixels. In this way, we eliminated and/or minimized unnecessary dark regions at sides, or marks printed by imaging devices. The training and validation sets consisted of the images collected only from the former website.

B. TRAINING

We constructed our model with KERAS.² It consists of a total of 39265 parameters. In III-B.2.c, we delineate the entire network architecture in detail. All convolution layers are initialized with Glorot uniform initializer. We employed a rectified linear unit (ReLU) as the activation function in the convolutional layers. Padding is set to 'same' so that the size of the output image does not change.

We obtained LR images dividing each training image by 255.0 after downscaling by bicubic interpolation. The images were cropped to make their size is multiple of the scale. We trained our model on NVIDIA GeForce RTX 2080Ti GPU at most ten epochs with batch size 128, learning rate 1×10^{-3} , minimum learning rate 1×10^{-7} , early stopping patience 5, and weight decay 1×10^{-6} . The learning rate was halved after the training is not improved for three epochs. All training parameters are summarized in Table 3.

The size of input image patches is 16, 11, 8, and 4 for the scales 2, 3, 4, and 8, respectively. A similar approach has been applied in determining the stride values. We used stride values 6, 4, 3, and 1 for scales 2, 3, 4, and 8, respectively. In this way, we obtained approximately the same number

² <https://keras.io>

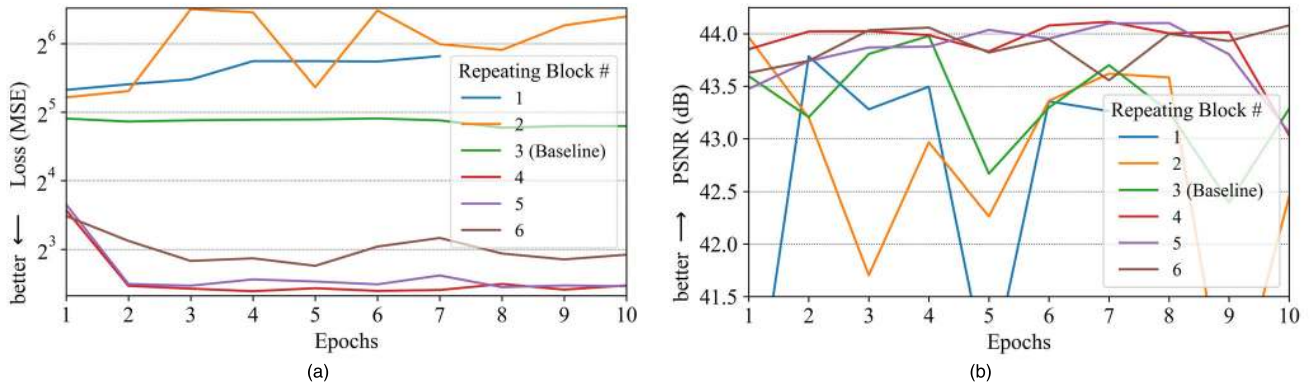


FIGURE 3. Performance change of Decusr by the number of repeating blocks (RBs). (a) training performance; (b) PSNR scores in TestSet-I. When the number of RBs is four (red line), Decusr converges very rapidly and preserves its success throughout the training. Furthermore, it scores the highest.

of training image patches for each scale. The images in the training set were randomly reordered at the beginning of each epoch. Similarly, the image patches were also extracted in random order from each input image.

C. EVALUATION

The quality of HR images obtained from each method was quantitatively measured by the following image quality metrics: peak signal to noise ratio (PSNR), structural similarity index (SSIM) [58], multi-scale structural similarity index (MS-SSIM) [59], feature similarity index (FSIM) [60], mean absolute deviation (MAD), Haar wavelet-based perceptual similarity index (HaarPSi) [61], gradient magnitude similarity deviation (GMSD) [62], and visual fidelity information (VIF) [63]. Since PSNR is one of the most widely used measures for image quality assessment, we present the definition of PSNR as follows:

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right) \quad (9)$$

where L is the maximum intensity value in images (255 for this case), and MSE is defined in (5).

The performance of the proposed method is compared with Bicubic, Bilinear, Lanczos and Nearest interpolation methods, SRCNN, EDSR (baseline model with approximately 1.5M parameters), and the method proposed in [3]. We shortened the name of this method as RWRUSR to easy reference. The benchmarks were performed for all scales.

We did our best to obtain the best results from RWRUSR. Since the authors did not mention how they upsampled the LR image before constructing the design matrix in the regression procedure, we tried three different ways of upscaling the LR image in each direction (horizontal or vertical) for each pass of regression. In the first approach, we just repeated existing rows and columns in the new rows and columns, respectively. In the second approach, instead of repeating the same values, we fill in the average values of adjacent rows or columns. In the last approach, we used the bicubic interpolation method for upscaling LR images. Since it offered much better results than others, the first technique was employed for the scales

2, 4, and 8. The authors state in their work that magnification by multiples of two can be yielded by repeatedly implementing the algorithm. Namely, the authors used the algorithm only for the magnification factors that are multiples of two. Therefore, for scale 3, we introduced the bicubic interpolation method for upscaling LR images since it yielded better results than the first two approaches.

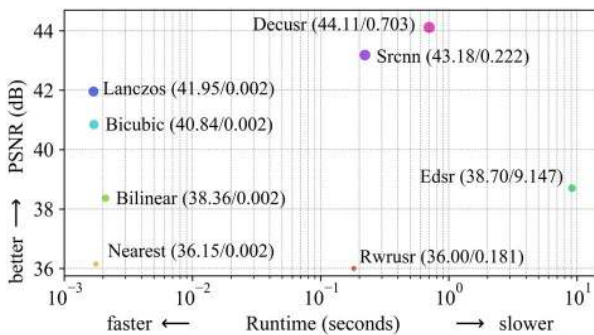
D. RESULTS

We evaluated the performance of each method in quantitative and qualitative terms. We present in Table 4 the quantitative results of the methods in both test sets. The highest score is marked with red color, whereas the second one with blue. According to most IQMs, our method outperforms other algorithms by a noticeable margin at all scale factors. The second-best performance is mostly achieved by EDSR in TestSet-I, and by SRCNN in TestSet-II. Except for FSIM measure at scales 3, 4, and 8, our method achieves the highest scores in TestSet-II as well. SRCNN surpassed EDSR in this test by mostly achieving the second-highest scores. In both tests, no method other than DL methods ranked the top two. Experiments show that Lanczos interpolation outperforms other interpolation methods. We observed in our experiments that EDSR yields a very low PSNR score at each scale. To us, this may be due to the use of ℓ_1 loss. The authors trained EDSR with ℓ_1 loss, instead of ℓ_2 loss, which maximizes PSNR, for the reason that it offers fast convergence. Therefore, it yields too low PSNR score compared to others. However, this does not apply to other metrics. It mostly offers the second-best performance in other IQMs in TestSet-I. It even achieves the highest FSIM score in TestSet-II for scales 3, 4, and 8, as we mentioned above.

For visual evaluation, we represent these results graphically in Figs. 1 and 2 in Appendix A. The subfigures (a), (b), (c), (d), (e), (f), (g), and (h) respectively show the average score curves of the following IQMs: FSIM, GMSD, HAARPSI, MAD, MS-SSIM, PSNR, SSIM, and VIF. Lower value in GMSD and MAD metrics means better performance, whereas the opposite is true for others. We show our method with a green line in both figures. The stability in the perfor-

TABLE 4. Quantified performances of methods in TestSet-I and TestSet-II. Red shows the highest while blue shows the second highest. Our method exhibits superb performance almost in all IQMs. EDSR and SRCNN perform the second-best in TestSet-I and TestSet-II, respectively.

	TestSet-I									TestSet-II							
	METHOD	FSIM	GMSD	HAARPSI	MAD	MSSSIM	PSNR	SSIM	VIF	FSIM	GMSD	HAARPSI	MAD	MSSSIM	PSNR	SSIM	VIF
2x	Bicubic	0.9535	0.0185	0.8706	1.4998	0.9824	40.8414	0.9719	0.7745	0.9441	0.0557	0.6163	3.6253	0.9391	28.4820	0.9143	0.5735
	Bilinear	0.9501	0.0295	0.8329	1.9883	0.9719	38.3645	0.9574	0.6982	0.9411	0.0632	0.5826	4.1297	0.9216	27.6981	0.8938	0.5138
	Lanczos	0.9552	0.0123	0.8881	1.3238	0.9853	41.9543	0.9767	0.8076	0.9450	0.0509	0.6292	3.4559	0.9447	28.8120	0.9206	0.6009
	Nearest	0.9474	0.0297	0.6048	2.4628	0.9646	36.1492	0.9482	0.6547	0.9420	0.0632	0.4533	4.1949	0.9248	27.6200	0.8971	0.5127
	Rwrrsr	0.9474	0.0298	0.6184	2.5915	0.9632	36.0017	0.9405	0.6524	0.9418	0.0632	0.4583	4.2956	0.9244	27.6036	0.8947	0.5118
	Srrnn	0.9588	0.0071	0.9183	1.2097	0.9885	43.1829	0.9810	0.8414	0.9481	0.0401	0.6729	3.2171	0.9547	29.2911	0.9306	0.6389
	Edsr	0.9607	0.0245	0.9114	1.0206	0.9958	38.7039	0.9843	0.8506	0.9488	0.0922	0.5770	3.6799	0.9703	24.3891	0.9278	0.5905
	Decusr	0.9602	0.0055	0.9304	1.1016	0.9978	44.1133	0.9837	0.8467	0.9495	0.0322	0.7080	2.8020	0.9915	30.5537	0.9404	0.6391
3x	Bicubic	0.9421	0.0492	0.7226	2.7021	0.9475	35.6253	0.9201	0.5786	0.9340	0.0837	0.4824	5.1305	0.8815	26.2538	0.8420	0.4074
	Bilinear	0.9397	0.0596	0.6777	3.0930	0.9320	34.3312	0.9029	0.5229	0.9317	0.0903	0.4618	5.4147	0.8664	25.9196	0.8273	0.3746
	Lanczos	0.9432	0.0435	0.7463	2.5088	0.9540	36.3586	0.9282	0.6133	0.9350	0.0798	0.4916	5.0235	0.8880	26.3999	0.8481	0.4259
	Nearest	0.9095	0.0550	0.4103	3.7437	0.9128	32.3509	0.8780	0.4605	0.9105	0.0841	0.3078	5.7498	0.8592	25.5089	0.8174	0.3513
	Rwrrsr	0.9415	0.0534	0.7122	9.8225	0.9447	27.3613	0.8985	0.6099	0.9343	0.0844	0.4847	6.7489	0.8827	25.5703	0.8284	0.4134
	Srrnn	0.9480	0.0291	0.7910	2.5121	0.9627	37.1659	0.9304	0.6659	0.9401	0.0589	0.5456	4.9067	0.9049	27.1015	0.8459	0.4687
	Edsr	0.9487	0.0344	0.7963	2.0260	0.9902	36.3102	0.9485	0.6858	0.9410	0.0942	0.4952	4.9766	0.9625	24.4698	0.8710	0.4633
	Decusr	0.9486	0.0259	0.8044	2.0758	0.9914	38.5420	0.9470	0.6839	0.9403	0.0564	0.5566	4.4042	0.9793	27.4008	0.8740	0.4821
4x	Bicubic	0.9329	0.0919	0.5852	3.8359	0.8992	32.5970	0.8580	0.4276	0.9261	0.1217	0.4017	6.1552	0.8304	25.1760	0.7837	0.3042
	Bilinear	0.9302	0.1066	0.5441	4.2945	0.8757	31.6114	0.8377	0.3817	0.9242	0.1321	0.3792	6.4900	0.8099	24.8524	0.7676	0.2759
	Lanczos	0.9336	0.0856	0.6103	3.6775	0.9077	33.0334	0.8648	0.4519	0.9270	0.1181	0.4147	6.0722	0.8374	25.2908	0.7880	0.3179
	Nearest	0.8815	0.1008	0.2618	4.7625	0.8606	30.3313	0.8132	0.3338	0.8821	0.1261	0.1985	6.8089	0.8038	24.4282	0.7574	0.2535
	Rwrrsr	0.8829	0.1007	0.2732	4.9647	0.8587	30.2282	0.8007	0.3326	0.8839	0.1260	0.2094	6.9668	0.8029	24.4066	0.7526	0.2523
	Srrnn	0.9394	0.0710	0.6513	3.4470	0.9238	34.0981	0.8738	0.5036	0.9346	0.0943	0.4613	5.8631	0.8593	25.8619	0.7897	0.3560
	Edsr	0.9406	0.0722	0.6630	3.2774	0.9747	32.3757	0.8919	0.5392	0.9355	0.1112	0.4473	5.9889	0.9489	24.2466	0.8110	0.3768
	Decusr	0.9411	0.0664	0.6760	3.2150	0.9764	34.7470	0.8806	0.5397	0.9346	0.0917	0.4746	5.5224	0.9605	26.0336	0.8137	0.3851
8x	Bicubic	0.9078	0.1847	0.3337	6.8447	0.7203	27.7036	0.7029	0.1883	0.9094	0.1925	0.2644	8.4830	0.6951	23.2878	0.6704	0.1476
	Bilinear	0.9047	0.1939	0.3135	7.2258	0.7009	27.2690	0.6925	0.1728	0.9076	0.2014	0.2483	8.8399	0.6783	23.0319	0.6592	0.1366
	Lanczos	0.9095	0.1795	0.3491	6.7492	0.7269	27.8645	0.7042	0.1956	0.9103	0.1876	0.2760	8.3989	0.7007	23.3907	0.6717	0.1527
	Nearest	0.8310	0.2238	0.1504	7.6639	0.6729	26.4055	0.6576	0.1268	0.8348	0.2268	0.1208	9.2193	0.6584	22.6021	0.6375	0.1031
	Rwrrsr	0.8368	0.2190	0.1495	7.9387	0.6732	26.0784	0.6453	0.1253	0.8454	0.2236	0.1225	9.5784	0.6547	22.3502	0.6316	0.1023
	Srrnn	0.9173	0.1615	0.3866	6.2971	0.7573	28.6281	0.7165	0.2178	0.9167	0.1738	0.2973	8.3248	0.7189	23.6237	0.6413	0.1673
	Edsr	0.9212	0.1677	0.3822	6.4011	0.8904	25.4221	0.7387	0.2528	0.9205	0.1918	0.2930	8.7150	0.8579	21.3377	0.6943	0.1936
	Decusr	0.9205	0.1536	0.4118	6.0668	0.8942	28.9230	0.7252	0.2484	0.9200	0.1641	0.3161	7.7954	0.8864	23.7852	0.6963	0.1947

**FIGURE 4.** Run-time performances of methods for 2x scale factor. Values in parenthesis indicate PSNR score and average run-time. Our method surpasses other algorithms with moderate run-time overhead.

mance ranking of the algorithms in both test sets can easily be seen in both figures.

To make a qualitative comparison between the methods, we present in Figs. 5 and 6 the same region of the HR images reconstructed by each method from a given LR image in each test set. The full-size original images with their ground-truth patches are also given on the left. We represent the PSNR and SSIM scores of each method below the images. It is very

clear from the figures that the best quality images, in terms of HVS, are the ones reconstructed by the proposed method. Our method very successfully preserved high-frequency details. The lowest quality images were yielded by Nearest interpolation, and then RWRUSR. The aliasing artifacts yielded by both algorithms may easily be seen in all figures. Particularly at large scales. The images produced by EDSR have some pixel corruptions, especially in dark areas. In contrast, our method yields very good resolution and sharpness. The larger the scale, the higher the difference in image quality between our method and others.

We also checked the statistical significance of our method's success against other methods by one-way analysis of variance (ANOVA) and Welch's t-test at 0.05 level of significance. We performed 224 measurements (4 scales x 8 IQMs x 7 methods) in total for each test set. We present the full list of p values of the measurements in Table 1 Appendix B. The p values greater than 0.05, which indicate no significant difference between our method and others, are marked in red. Respectively, 211 and 208 measurements performed in TestSet-I and TestSet-II, confirmed the significance of our method. Most of the remaining 29 analyses, which are showing no significance, are the ones performed against EDSR, and then SRCNN. Almost half of these analyses were

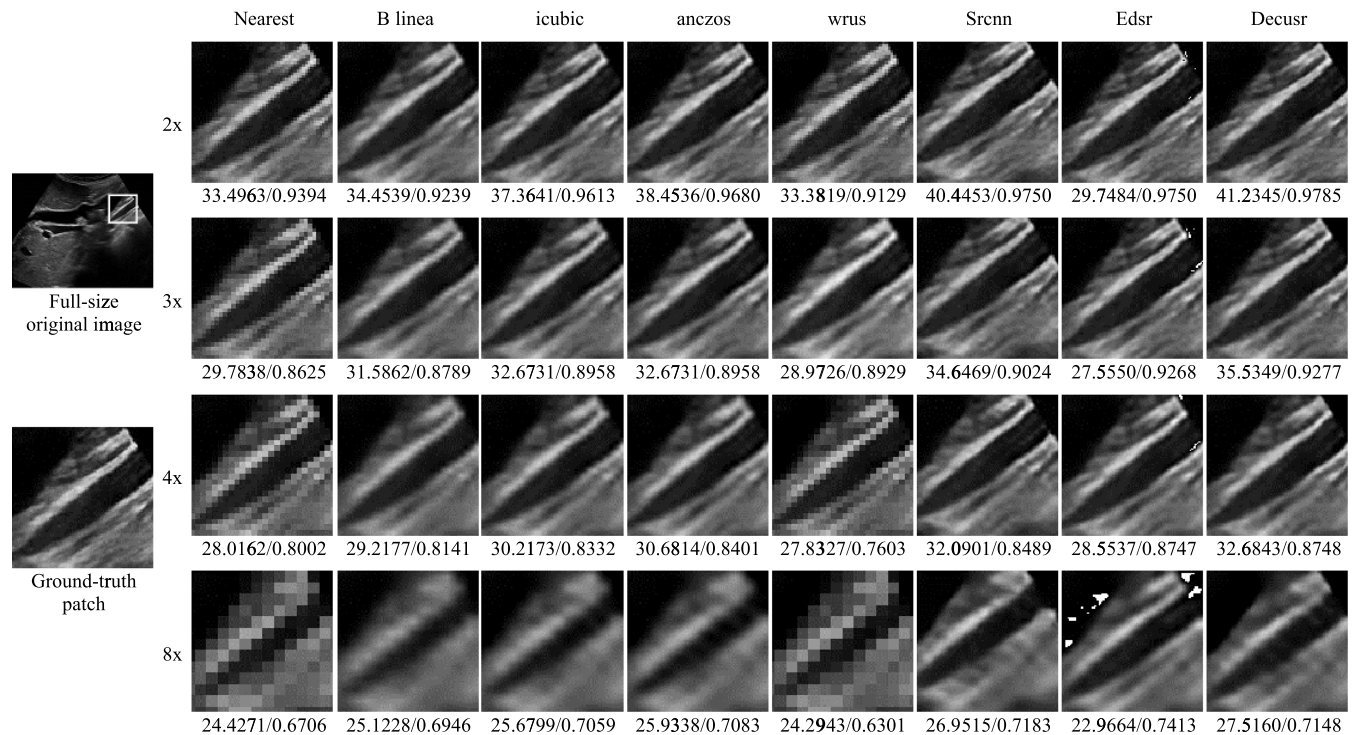


FIGURE 5. Visual comparison of an HR image reconstructed by each method. Values below the images, respectively, designate PSNR and SSIM scores. Our method produces much better images in terms of the human visual system.

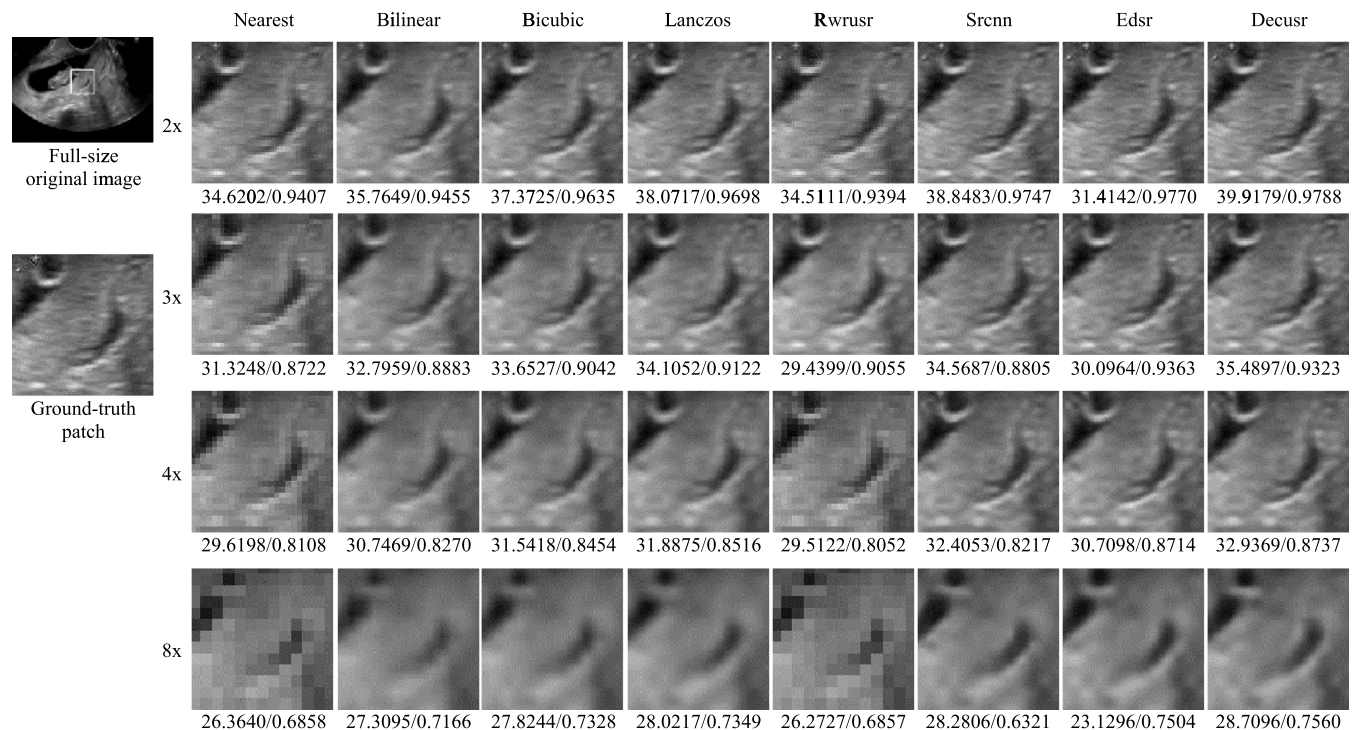


FIGURE 6. Visual comparison of an HR image reconstructed by each method. Values below the images, respectively, designate PSNR and SSIM scores. Our method produces much better images in terms of the human visual system.

performed in FSIM, and the rest are mostly in VIF and SSIM measures. All measurements in HAARPSI and MS-SSIM confirm our method's significance. Apart from only one, two, and three measurements, respectively, all measurements in

GMSD, MAD, and PSNR, also confirm the significance of our method.

In addition to statistical analyses, we present the run-time performances of all methods in Fig. 4. The figure reveals that

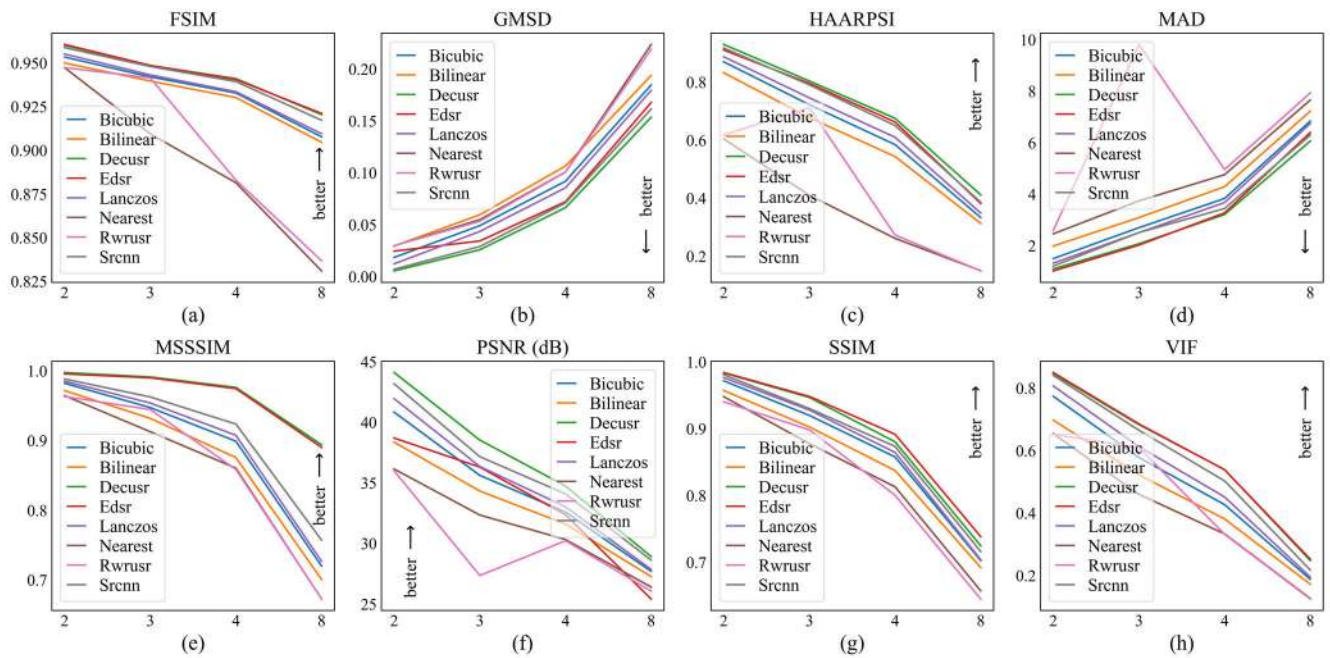


FIGURE 7. Average performances of methods in TestSet-I in terms of (a) FSIM, (b) GMSD, (c) HAARPSI, (d) MAD, (e) MS-SSIM, (f) PSNR, (g) SSIM, and (h) VIF. Lower value in GMSD and MAD measures means better performance, whereas the opposite is true for others. DECUSR (green line) exhibits superb performance almost in all measures compared to other methods.

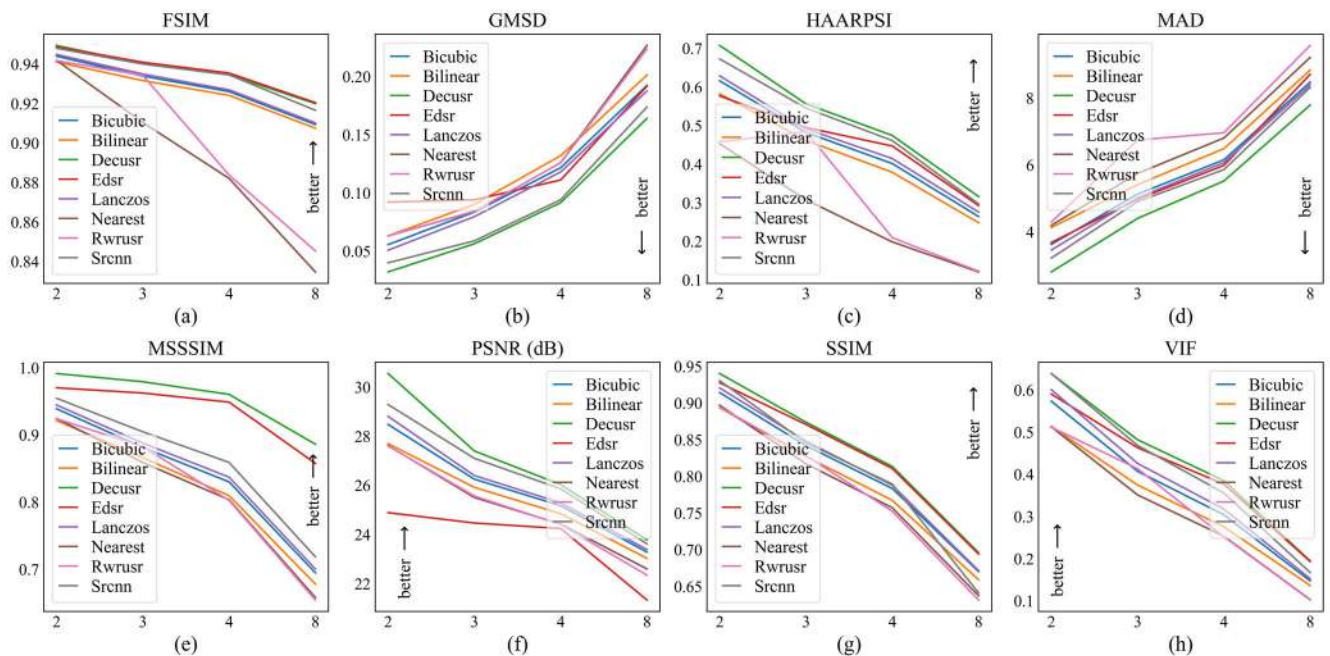


FIGURE 8. Average performances of methods in TestSet-II in terms of (a) FSIM, (b) GMSD, (c) HAARPSI, (d) MAD, (e) MS-SSIM, (f) PSNR, (g) SSIM, and (h) VIF. Lower value in GMSD and MAD measures means better performance, whereas the opposite is true for others. DECUSR (green line) exhibits superb performance almost in all measures compared to other methods.

our method surpasses other algorithms with moderate run-time overhead.

V. CONCLUSION

We implemented in this study a novel deep learning approach to improve the low resolution and quality of ultrasound images. We achieved SR on the B-mode ultrasound images,

which is the last stage of USI, rather than obtaining SR in the pre-processing phase or early stages of the post-processing phase of USI.

The proposed approach utilizes a deep learning algorithm that learns from samples the non-linear relationships between LR and HR ultrasound images. We trained our network with a huge training set of final B-mode ultrasound images to

TABLE 5. Statistical significance analysis of our method against others in terms of ANOVA and WELCH'S t-test at 0.05 level of significance. We round of the p-values up to three digits since it is sufficient. p-values indicating no significant difference are marked in red. Respectively, 211 and 208 out of the 224 measurements in TestSet-I and TestSet-II confirmed the significance of our method.

		FSIM		GMSD		HaarPsi		MAD		MS-SSIM		PSNR (dB)		SSIM		VIF	
		Anova	t-test	Anova	t-test	Anova	t-test	Anova	t-test	Anova	t-test	Anova	t-test	Anova	t-test	Anova	t-test
TestSet-I	2x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.338	0.338	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.179	0.179	0.228	0.228
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.009	0.009	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.078	0.078
	3x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.943	0.943	0.000	0.000	0.029	0.029	0.235	0.235	0.000	0.000	0.000	0.000	0.278	0.278	0.655	0.655
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.304	0.304	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.382	0.382	0.000	0.000	0.000	0.000	0.224	0.224	0.001	0.001	0.000	0.000	0.000	0.000	0.882	0.882
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.004	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	8x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.326	0.326	0.000	0.000	0.000	0.000	0.000	0.000	0.012	0.012	0.000	0.000	0.000	0.000	0.043	0.043
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.006	0.000	0.000	0.007	0.007	0.011	0.011	0.000	0.000
TestSet-II	2x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.282	0.282	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.027	0.027	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.978	0.978
	3x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.302	0.302	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.326	0.326	0.000	0.000
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.761	0.761	0.026	0.026	0.042	0.042	0.000	0.000	0.000	0.000	0.096	0.096	0.000	0.000	0.009	0.009
	4x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.126	0.126	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.465	0.465	0.075	0.075
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.962	0.962	0.054	0.054	0.004	0.004	0.001	0.001	0.000	0.000	0.288	0.288	0.000	0.000	0.000	0.000
	8x Bicubic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Bilinear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Edsr	0.493	0.493	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.684	0.684	0.678	0.678
	Lanczos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.002	0.000	0.000	0.000	0.000
	Nearest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Rwrrsr	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Srnn	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.201	0.201	0.000	0.000	0.000	0.000

obtain SR for the scale factors 2, 3, 4, and 8. We benchmarked our method with four well-known interpolation methods and two state-of-the-art DL networks. We used eight IQMs for quantitative evaluations. We also evaluated the quality of the reconstructed images by each method in terms of HVS.

We created two separate large test sets composed of images from two separate sources. The experiments showed that the proposed model surpasses other methods by a significant margin. Our algorithm achieved much better scores than others almost in all IQMs at each magnification factor. We also confirmed the statistical significance of our method by two statistical analyses. Besides, we demonstrated the success of our proposed method in producing high-quality images in terms of HVS. It reconstructs very good images in many aspects, such as resolution, edge-preserving, and sharpness.

As a conclusion, we showed with this study that our proposed approach exhibits excellent performance in qualitative and quantitative terms. Our method maintains high-quality ultrasound image reconstruction. Since it is software-based, it can be easily adapted to other image processing algorithms or integrated into imaging devices without the need for any hardware modifications or expansions.

APPENDIXES

APPENDIX-A

Visual representations of method's scores in TestSet-I and TestSet-II.

APPENDIX-B

Statistical significance analysis of the proposed method with others.

VI. ACKNOWLEDGMENT

The authors would like to thank Dr. Taco Geertsma, who is the founder of the website www.ultrasoundcases.info, for granting permission to use the images from the website. They would like to thank the owner of the website www.ultrasound-images.com for sharing sample ultrasound images.

REFERENCES

- [1] R. Prevost, M. Salehi, S. Jagoda, N. Kumar, J. Sprung, A. Ladikos, R. Bauer, O. Zettinig, and W. Wein, "3D freehand ultrasound without external tracking using deep learning," *Med. Image Anal.*, vol. 48, pp. 187–202, Aug. 2018.
- [2] S. H. Contreras Ortiz, T. Chiu, and M. D. Fox, "Ultrasound image enhancement: A review," *Biomed. Signal Process. Control*, vol. 7, no. 5, pp. 419–428, Sep. 2012.
- [3] W. Sharabati and X. Bowie, "Robust weighted regression for ultrasound image super-resolution," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2018, pp. 1–6.
- [4] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 184–199.
- [5] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.
- [6] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
- [7] H. A. Aly and E. Dubois, "Image up-sampling using total-variation regularization with a new observation model," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1647–1659, Oct. 2005.
- [8] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," *Adv. Comput. Vis. Image Process.*, vol. 1, no. 2, pp. 317–339, 1984.
- [9] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [10] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2004, p. 1.
- [11] Y. Romano, J. Isidoro, and P. Milanfar, "RAISR: Rapid and accurate image super resolution," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 110–125, Mar. 2017.
- [12] Y. Tang, P. Yan, Y. Yuan, and X. Li, "Single-image super-resolution via local learning," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 1, pp. 15–23, Mar. 2011.
- [13] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1637–1645.
- [14] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.
- [15] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3147–3155.
- [16] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.
- [17] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1110–1121.
- [18] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9906, 2016, pp. 391–407.
- [19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
- [20] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
- [21] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4539–4547.
- [22] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Dec. 2017, p. 5.
- [23] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch, "EnhanceNet: Single image super-resolution through automated texture synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4501–4510.
- [24] B. Liu and D. Ait-Boudaoud, "Effective image super resolution via hierarchical convolutional neural network," *Neurocomputing*, vol. 374, pp. 109–116, Jan. 2020.
- [25] J. Tang, C. Huang, J. Liu, and H. Zhu, "Image super-resolution based on CNN using multilabel gene expression programming," *Appl. Sci.*, vol. 10, no. 3, p. 854, 2020.
- [26] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
- [27] X. Wang, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018.
- [28] X. Hu, X. Liu, Z. Wang, X. Li, W. Peng, and G. Cheng, "RTSRGAN: Real-time super-resolution generative adversarial networks," in *Proc. 7th Int. Conf. Adv. Cloud Big Data (CBD)*, Sep. 2019, pp. 321–326.
- [29] M. Zareapoor, M. E. Celebi, and J. Yang, "Diverse adversarial network for image super-resolution," *Signal Process., Image Commun.*, vol. 74, pp. 191–200, May 2019.
- [30] Y. Zhang, S. Liu, C. Dong, X. Zhang, and Y. Yuan, "Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution," *IEEE Trans. Image Process.*, vol. 29, pp. 1101–1112, 2020.
- [31] X. Zhang, H. Song, K. Zhang, J. Qiao, and Q. Liu, "Single image super-resolution with enhanced Laplacian pyramid network via conditional generative adversarial learning," *Neurocomputing*, Jul. 2019.

- [32] F. Wu, B. Wang, D. Cui, and L. Li, "Single image super-resolution based on wasserstein GANs," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 9649–9653.
- [33] S. Lian, H. Zhou, and Y. Sun, "FG-SRGAN: A feature-guided super-resolution generative adversarial network for unpaired image super-resolution," in *Proc. Int. Symp. Neural Netw.*, 2019, pp. 151–161.
- [34] M. E. A. Seddik, M. Tamaazousti, and J. Lin, "Generative collaborative networks for single image super-resolution," *Neurocomputing*, Oct. 2019.
- [35] X. Zhu, L. Zhang, L. Zhang, X. Liu, Y. Shen, and S. Zhao, "Generative adversarial network-based image super-resolution with a novel quality loss," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Dec. 2019, pp. 1–2.
- [36] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7982–7991.
- [37] Z. Wang, B. Chen, H. Zhang, and H. Liu, "Variational probabilistic generative framework for single image super-resolution," *Signal Process.*, vol. 156, pp. 92–105, Mar. 2019.
- [38] P. Shamsolmoali, M. Zareapoor, R. Wang, D. K. Jain, and J. Yang, "G-GANISR: Gradual generative adversarial network for image super resolution," *Neurocomputing*, vol. 366, pp. 140–153, Nov. 2019.
- [39] R. Morin, A. Basarab, M. Ploquin, and D. Kouamé, "Post-processing multiple-frame super-resolution in ultrasound imaging," *Med. Imag. Ultrason. Imag., Tomogr. Imaging*, vol. 8320, Feb. 2012, Art. no. 83201G.
- [40] R. Morin, A. Basarab, S. Bidon, and D. Kouamé, "Motion estimation-based image enhancement in ultrasound imaging," *Ultrasonics*, vol. 60, pp. 19–26, Jul. 2015.
- [41] Y. Dai, B. Wang, and D. Liu, "A fast and robust super resolution method for intima reconstruction in medical ultrasound," in *Proc. 3rd Int. Conf. Bioinf. Biomed. Eng.*, Jun. 2009, pp. 1–4.
- [42] Q. Zhang, B. Li, M. Shen, and J. Yang, "A novel ultrasonic image zooming algorithm based on sparse representation," in *Proc. 3rd Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2016, pp. 861–865.
- [43] M. Abdel-Nasser, J. Melendez, A. Moreno, O. A. Omer, and D. Puig, "Breast tumor classification in ultrasound images using texture analysis and super-resolution methods," *Eng. Appl. Artif. Intell.*, vol. 59, pp. 84–92, Mar. 2017.
- [44] N. Zhao, Q. Wei, A. Basarab, D. Kouame, and J.-Y. Tourneret, "Single image super-resolution of medical ultrasound images using a fast algorithm," in *Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2016, pp. 473–476.
- [45] R. Jirik and T. Taxt, "High-resolution ultrasonic imaging using two-dimensional homomorphic filtering," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 53, no. 8, pp. 1440–1448, Aug. 2006.
- [46] S. Yang, Y. Sun, Y. Chen, and L. Jiao, "Structural similarity regularized and sparse coding based super-resolution for medical images," *Biomed. Signal Process. Control*, vol. 7, no. 6, pp. 579–590, Nov. 2012.
- [47] M. Ploquin, A. Basarab, and D. Kouame, "Resolution enhancement in medical ultrasound imaging," *J. Med. Imag.*, vol. 2, no. 1, 2015, Art. no. 17001.
- [48] R. Morin, A. Basarab, and D. Kouame, "Alternating direction method of multipliers framework for super-resolution in ultrasound imaging," in *Proc. 9th IEEE Int. Symp. Biomed. Imag. (ISBI)*, May 2012, pp. 1595–1598.
- [49] Y. H. Yoon and J. C. Ye, "Deep learning for accelerated ultrasound imaging," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6673–6676.
- [50] M. Abdel-Nasser and O. A. Omer, "Ultrasound image enhancement using a deep learning architecture," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.*, 2017, pp. 639–649.
- [51] Y. Wu, F. Yang, J. Huang, and Y. Liu, "Super-resolution construction of intravascular ultrasound images using generative adversarial networks," *J. Southern Med. Univ.*, vol. 39, no. 1, pp. 82–87, 2019.
- [52] W. Choi, M. Kim, J. HakLee, J. Kim, and J. BeomRa, "Deep CNN-based ultrasound super-resolution for high-speed high-resolution B-Mode imaging," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2018, pp. 1–4.
- [53] J. Lu and W. Liu, "Unsupervised super-resolution framework for medical ultrasound images using dilated convolutional neural networks," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2018, pp. 739–744.
- [54] P. Khavari, A. Asif, and H. Rivaz, "Non-local super resolution in ultrasound imaging," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–6.
- [55] M. C. Özkul, Ü. E. Mumcuoğlu, and I. T. Sancak, "Single-image Bayesian restoration and multi-image super-resolution restoration for B-mode ultrasound using an accurate system model involving correlated nature of the speckle noise," *Ultrason. Imag.*, vol. 41, no. 6, pp. 368–386, Nov. 2019.
- [56] K. Nasrollahi and T. B. Moeslund, "Super-resolution: A comprehensive survey," *Mach. Vis. Appl.*, vol. 25, no. 6, pp. 1423–1468, Aug. 2014.
- [57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [59] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, 1999, pp. 1398–1402.
- [60] L. Zhang, L. Zhang, X. Mou, D. Zhang, and others, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [61] R. Reisenhofer, S. Bosse, G. Kutyniok, and T. Wiegand, "A Haar wavelet-based perceptual similarity index for image quality assessment," *Signal Process., Image Commun.*, vol. 61, pp. 33–43, Feb. 2018.
- [62] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 684–695, Feb. 2014.
- [63] H. R. Sheikh and A. C. Bovik, "A visual information fidelity approach to video quality assessment," in *Proc. 1st Int. Workshop Video Process. Qual. Metrics Consum. Electron.*, 2005, pp. 23–25.



and data mining.

HAKAN TEMİZ (Graduate Student Member, IEEE) received the B.Sc. degree in machine education from Marmara University, the bachelor's degree in business administration from Anadolu University, and the B.Sc. and M.Sc. degrees in computer engineering from Ahmet Yesevi University. He is currently pursuing the Ph.D degree with Gazi University. He is a Lecturer with Artvin Coruh University. His research interests include deep learning, machine learning, computer vision,



HASAN S. BILGE (Member, IEEE) received the B.Sc. degree in computer engineering from Yildiz Technical University, the M.Sc. degree in electrical and electronics engineering from Kirikkale University, and the Ph.D. degree in electrical and electronics engineering from Baskent University. He is an Associate Professor with the Department of Electrical and Electronics Engineering, Gazi University. His research interests include image processing, pattern recognition, face recognition, FPGA, ultrasound imaging, and deep learning.

• • •