International Journal of Innovative Computing, Information and Control
Volume **15**, Number **2**, April **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **725–739**

DEEP CONVOLUTIONAL NETWORKS FOR MAGNIFICATION OF DICOM BRAIN IMAGES

Kok Swee Sim and Fawaz Sammani*

Faculty of Engineering and Technology
Multimedia University
Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia
kssim@mmu.edu.my; *Corresponding author: sksbg2018@gmail.com

Received June 2018; revised October 2018

Abstract. *Convolutional neural networks have recently achieved great success in Single Image Super-Resolution (SISR). SISR is the action of reconstructing a high-quality image from a low-resolution one. In this paper, we propose a deep Convolutional Neural Network (CNN) for the enhancement of Digital Imaging and Communications in Medicine (DICOM) brain images. The network learns an end-to-end mapping between the low and high resolution images. We first extract features from the image, where each new layer is connected to all previous layers. We then adopt residual learning and the mixture of convolutions to reconstruct the image. Our network is designed to work with grayscale images, since brain images are originally in grayscale. We further compare our method with previous works, trained on the same brain images, and show that our method outperforms them.*

Keywords: Deep convolutional networks, Single image super-resolution, Magnification, DICOM images

1. **Introduction.** Single Image Super-Resolution (SISR) intends to reconstruct a High-Resolution (HR) image from a Low-Resolution (LR) image. It aims to focus on reconstructing the high-frequency information from the image, which is typically lost when the image is resized to another shape. It is mainly used in applications [1,2] where the high-detail information is greatly desired. Figure 1 illustrates the problem that SISR addresses.

Early methods of SISR include interpolation such as nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation, and other methods that utilize statistical image priors [3,4]. More recent methods include sparse coding [5,6], an external example-based method [7], that uses a learned dictionary based on sparse signal representation. Example-based method includes various phases. For the image pre-processing phase, the first step is to extract overlapping patches from the input image and normalize them. Then, these patches are encoded by a low-resolution dictionary. The sparse coefficients are then passed into a high-resolution dictionary for recovering the high-resolution patches. The overlapping recovered patches are combined to generate the final output image. These stages are shared by most external example-based methods, which focus on optimizing the dictionaries or constructing mapping functions. Moreover, random forests have also been used [8], and showed improvements in accuracy. Recently, due to the excellence of deep learning models, and especially Convolutional Neural Networks (CNN) with their powerful learning ability to images, CNNs have been immensely used to address the problem of SISR, and have shown superiority over the previous methods mentioned above

DOI: 10.24507/ijicic.15.02.725

FIGURE 1. The effect of resizing an image

[9-12]. They can be used to learn an end-to-end mapping from a low-resolution image to a high-resolution image.

DICOM images play an important role in medical images as many medical institutions use them for diagnoses illness of patients. Many researchers have focused on enhancing brain images, including contrast enhancement of brain images [13,14]. Important features included in the image to detect the illness. Therefore, the quality of the image plays a crucial role in detecting these features. If the DICOM image or part of it needs to be magnified for further observation, the quality will then be degraded, and important features of the image may not be detected. Thus, magnification of the DICOM image with quality-preserving is necessary.

The majority of super-resolution studies focus on grayscale or single-channel image. For color images, the image is first transferred to another color space, such as YCbCr, and the super-resolution is performed only on the luminance channel. The other channels are then concatenated along with the reconstructed luminance channel to reconstruct the colored image. However, there are also studies that apply super-resolution to all channels simultaneously. In this study, we will apply our super-resolution network on grayscale images, since brain images are originally in grayscale.

Our network adopts some of the features used in existing studies of super-resolution, as well as features that have significantly improved convolutional networks. Generally, our network comprises the following features.

- **Multi-scale Training:** The network is designed and trained on a single network to tackle the multiple scale super-resolution problem in an efficient manner, rather than designing a separate network to handle a single scale individually. Our network is trained on scales $\times 2$, $\times 3$ and $\times 4$. It also turns out that different scales help each other, and that makes the network more efficient.
- **Global Residual Learning:** In Global Residual Learning, the output image is the result of addition of both the input and the residual image from the final layer. The residual image usually contains the high-frequency components of the image that are lost when the image is resized. Our network learns to estimate the residual image which is composed of the high-frequency components.
- **Local Residual Learning:** In Local Residual Learning, the network learns from its previous layers within the network. In typical convolutional networks, important image details may be lost after so many layers. We solve this problem by imposing identity branches, where the identity branch carries rich image details to late layers, and helps gradient flow.
- **Dense Feature Extraction:** In the first branch of our network, we densely extract the features of the image by applying filter concatenation to every new layer. Each new layer in the feature extraction branch consists of both its output and previous layers' output. The final layer of this branch includes all the layer outputs along with the input image itself. The network densely extracts important features and texture

information from the image that will later act as an input to other convolutional networks.

- **Mixture of Convolutions:** We generate multiple convolutional networks branches with different parameters, with the dense features extracted from the previous layer as their inputs. We then concatenate their outputs together, and perform other convolution operations on the concatenated outputs, before we add the final residual image to the input image.
- **Context:** We utilize contextual information along large image regions. Usually, for images that are largely scaled, information within a small patch is not sufficient for detail recovery. To tackle this problem, our network uses a large receptive field and considers a large image context.

In summary, we train the network on multiple scales of a large receptive field. We first densely extract features of the image, and then feed those features to branches of convolutional networks. We model them to learn the residual image, and finally add the residual image to the original low-resolution image and reconstruct it to a high-resolution one.

This paper is divided into the following sections. In Section 1, we give an introduction of image super-resolution and an introduction of our network. In Section 2, we discuss the traditional methods used for image super-resolution in deep learning. In Section 3, we discuss the proposed network architecture, namely DCNMD. In Section 4, we discuss the training details used in our architecture. In Section 5, we show the results of our method and discuss them, as well as compare them with the traditional methods. Finally, we conclude this paper in Section 6.

2. **Related Work.**

2.1. **Super-Resolution using deep Convolutional Neural Networks (SRCNN).**

SRCNN model [9] was the first proposed deep learning model using convolutional neural networks for image super-resolution. It has achieved better results than the previous state-of-the-art methods, including A+, RFL and SelfEx. The SRCNN model consists of three layers, which are patch extraction or representation, non-linear mapping and reconstruction. The filter sizes are 9×9, 1×1 and 5×5, respectively. SRCNN uses a receptive field of 13×13. Patches are first extracted from both the low-resolution input and its corresponding ground truth output. The network is then trained on these patches, and directly learns the high resolution image. In other words, SRCNN carries the input image to the final layer and reconstructs the high-frequency details at the same time. Therefore, the training time might be spent on learning to reconstruct the same input image, while ignoring to learn the high-frequency components of the image, which acts as the crucial part for super-resolution applications. Moreover, SRCNN is trained on a single scale factor and only copes with that specific scale. Therefore, if we want to perform multiple image scaling, we need to have a trained network for every scale we want to perform. The SRCNN model is shown in Figure 2.

One main disadvantage of SRCNN is that the network spends most of its time learning to reconstruct the complete image from scratch. Therefore, most of the network parameters focus on reconstructing the general image, and thus fine details of the image such as high-frequency components are lost, and not reconstructed. Another main disadvantage of SRCNN is that a separate network is required for each scale. Therefore, we need three networks if we wish to reconstruct an image of scales 2, 3 and 4.

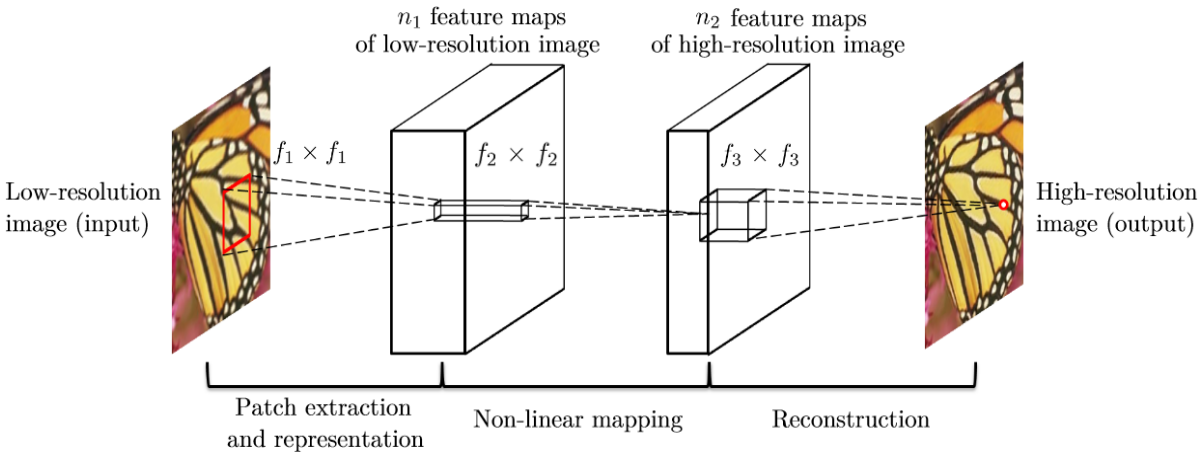

FIGURE 2. Super-Resolution using deep Convolutional Neural Networks model [9]

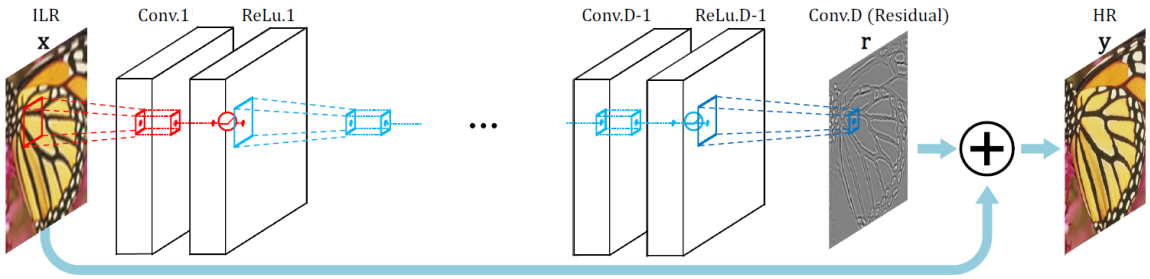

FIGURE 3. Accurate image Super-Resolution using Very Deep convolutional networks model [10]

2.2. **Accurate image Super-Resolution using Very Deep convolutional networks (VDSR).** VDSR model [10] was proposed after SRCNN to tackle the problems of SRCNN, and has achieved better results. VDSR is a deep network, with 20 layers. Except for the first and last layers, each layer consists of 64 filters of size 3×3. The VDSR uses a receptive field of 41×41, in order to cope with image information scaled with large factors. More importantly, the VDSR network learns the residual image only. Therefore, the training time is spent on only learning the residual, or high frequency components of the image, rather than learning the complete image which is composed of the low and high-frequency components, as in the case for SRCNN. After the network learns to predict the residual image, the residual image is added to the interpolated low-resolution image, to give the final high-resolution output. Moreover, VDSR is trained on multiple scale images. Therefore, there is no need to train an individual network for every scale, as in the case for SRCNN. VDSR also uses gradient-clipping to clip the gradients to a pre-defined range, and that greatly speeds up the training. The VDSR model is shown in Figure 3.

However, VDSR does not make use of previous convolutional layers. If the network is deep, then fine details of the image that were presented in previous layers may be lost in further layers. This might create a problem, especially in single image super-resolution networks, where the details and high frequency information of the image are highly desired.

2.3. **Densely connected convolutional networks.** Densely Connected Convolutional Networks (DenseNets) [15] were introduced in 2017. The idea behind DenseNets is that

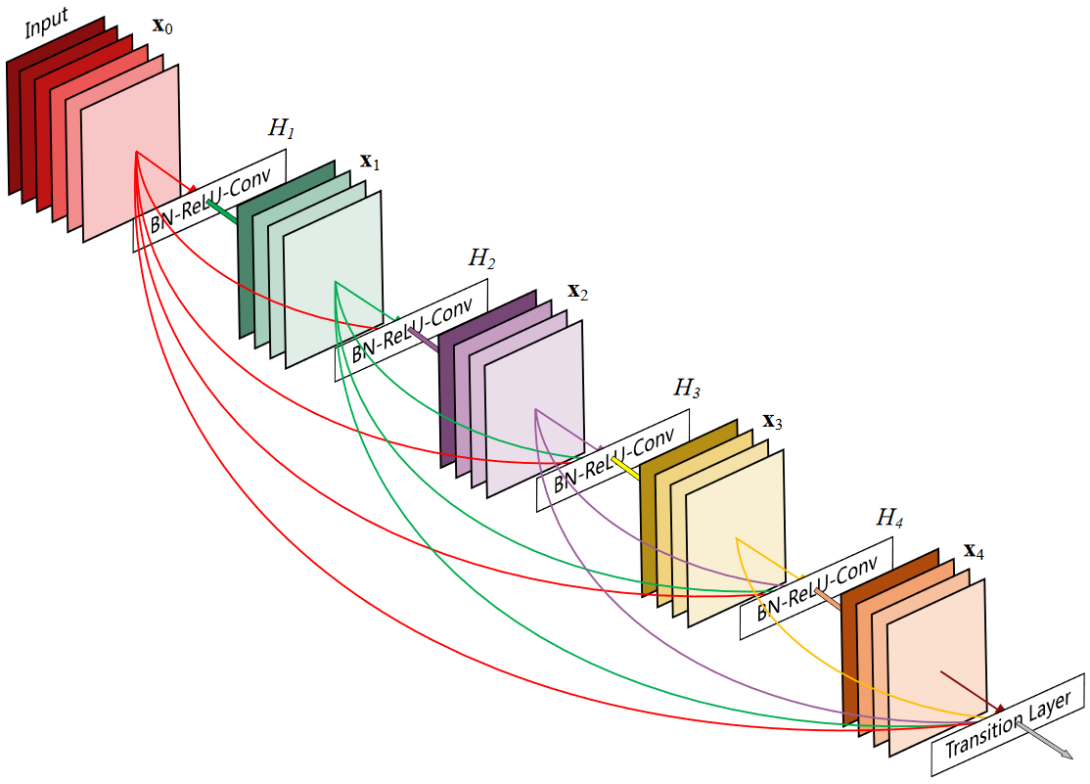

FIGURE 4. DenseNets [15]

each layer is connected to every other layer in a feed-forward fashion. In other words, each layer contains all previous feature-maps as input. DenseNets can tackle the vanishing-gradient problem, enhance feature propagation, re-utilize features, and significantly decrease the number of parameters. In this study, we utilize DenseNets to perform dense feature extraction at the first stage of our network. DenseNets are shown in Figure 4.

3. **Deep Convolutional Networks for Magnification of DICOM Brain Images.** We propose a deep convolutional network for the purpose of single image super-resolution. Our network learns an end-to-end mapping. It takes an interpolated low-resolution image, and outputs a high-resolution image. We utilize the concept of residual learning [16] and model the network to only learn the residual, since brain images are characterized with high image details and texture, and the big challenge is to fully recover them. We first densely extract features from the input image, and then perform a mixture of convolutional networks and concatenate their outputs together. Finally, we perform other convolution operations on the concatenated outputs, before we add the final residual image to the input image to generate the high-resolution output image. Moreover, we train the network on multiple scales to avoid using multiple networks each trained separately for a specific scale. We also use a receptive field of 41×41 to preserve image information when the image is largely scaled. Our images are all in grayscale, since that is the originality of brain images. Therefore, there is no need to extract specific channels (or all) and run them through the network to get the output. The network is then trained to achieve and end-to-end mapping between the low-resolution and ground-truth high-resolution patch. In the image post-processing stage, a slight thresholding is performed to the output image for evaluation purposes. Modelling our architecture with the above techniques described makes the network more efficient and robust, achieving better results than existing architectures.

3.1. **Image pre-processing.** In our image pre-processing stage, we perform the following.

- **Image conversion**
  Usually for colour images (RGB), they are converted to a different colour space (ex. YCbCr), and then the luminance channel (Y) is extracted and fed into the network. The output of the network is then concatenated with the other two channels to construct the complete colour high-resolution image. For the case of brain images, they are originally in grayscale. Thus, there is no need to extract the luminance channel and perform the operation as mentioned above. Reducing the number of channels of the original DICOM brain image from 3 to 1 is sufficient. The grayscale image is then directly fed into the network.
- **Extracting overlapping patches**
  We choose to extract overlapping patches of size 41×41, with a stride of 41, from both the low-resolution image, and their corresponding ground-truth high-resolution image. This is usually the case when working with super-resolution. For each low-resolution training patch x of size 41×41, there is a ground-truth high-resolution patch y of the same size (i.e., 41×41). The network is then trained on these patches.
- **Multi-scale generation**
  When generating the image patches as described in the previous step, each patch is scaled with 3 different scales: $\times 2$, $\times 3$ and $\times 4$. This is to enable our single network to work with multi-scale images, rather than training a separate network for each scale. Moreover, different scales help each other while training, making the network more efficient.
- **Image augmentation**
  Image augmentation is the act of generating different versions of an image from itself, by using various image transformations such as rotation, translation and flipping. This makes the network much more efficient, since it will be trained to respond on all possible appearances of the image. Image augmentation includes image rotation, image shifting, image flipping and other types of image transformations. In this study, we have chosen to perform rotation by 90° for each of the training images. Therefore, if it is necessary to rotate the brain image by 90°, the network can still perform well.

3.2. **Network architecture.** Figure 5 shows the proposed network architecture. As mentioned, dense features of the image are first extracted, and then passed to 4 branches of convolutional networks. A series of concatenation and convolutional operations is then carried on, and the residual image is predicted. Finally, the input image is added to the residual image to reconstruct the high-resolution image.

3.2.1. *Network parameters.*

- **Number of Filters:** We have used 64 filters for all the layers in the dense feature extraction stage of the network, denoted as C1 in Figure 5. This is to obtain as much necessary features from the image before passing those features to other networks. We have also used 64 filters for the first branch, denoted as C2 in Figure 5. For the rest of the branches, denoted as C3, C4 and C5, 32 filters are used. Finally, one filter for all the 1×1 convolutions is used.
- **Padding:** To prevent the output of each layer from shrinking after every convolution operation, we have padded each layer's output with zeros in order for the output size to be equal to the input size. That is mainly referred to as the 'same' padding. We have performed the 'same' padding for all the layers in our network.

- **"HE Normal" Weights Initialization:** To avoid the vanishing/exploding gradient problem, we choose to initialize our weights using "HE normal" initialization [17]. This technique initializes random weights drawn from a normal (Gaussian) distribution with mean zero and a standard deviation of $\sqrt{2/n_{in}}$, where n_{in} represents the number of inputs in the previous layer. This makes the network much more efficient, and reduces the training time.

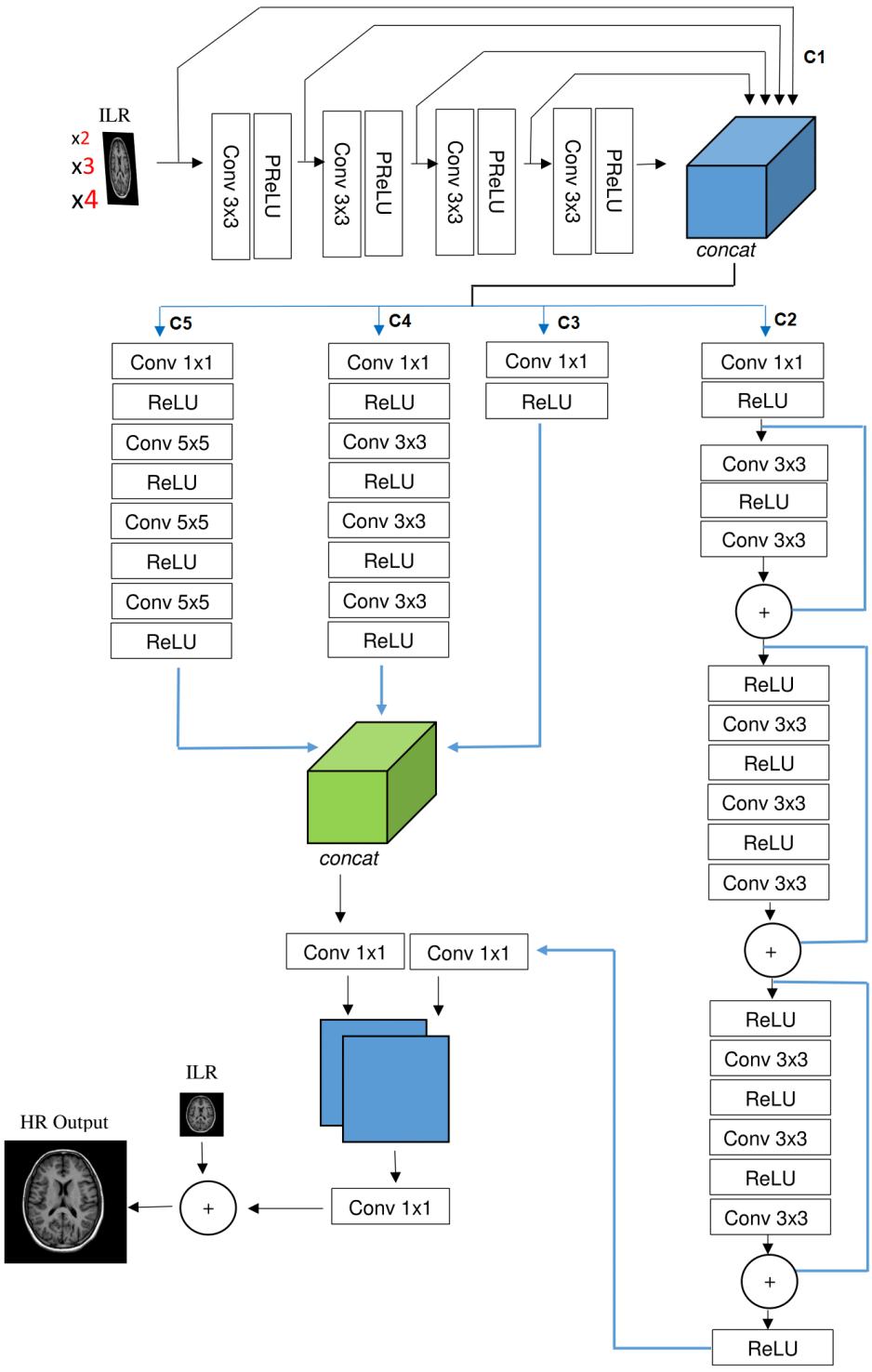

FIGURE 5. Deep Convolutional Networks for Magnification of DICOM (DCNMD) network architecture

3.2.2. *Network activations.*

- **Parametric Rectified Linear Unit (PReLU)**
  PReLU activation [17] is used in the dense feature extraction stage of the network. In PReLU, when the input is positive, the activated output remains the same. However, when the input is negative, the activated output is a learned parameter (a), named the leakage coefficient, multiplied with the input, instead of an output of zero as in the case for ReLU activation. PReLU is claimed by its authors that it enhances model fitting with nearly zero extra computational cost.
- **Rectified Linear Unit (ReLU)**
  ReLU activation function is used in the other stages of the network. The main advantage of ReLU is that it introduces non-linearity to the image. It tackles the vanishing gradient problem. The difference between ReLU from PReLU is that the learned leakage coefficient is zero. Thus, when the input is negative, the activated output is zero. However, it remains the same as in the case of PReLU when the input is positive (i.e., the activated output remains the same). Figure 6 shows the difference between ReLU (left) and PReLU (right).

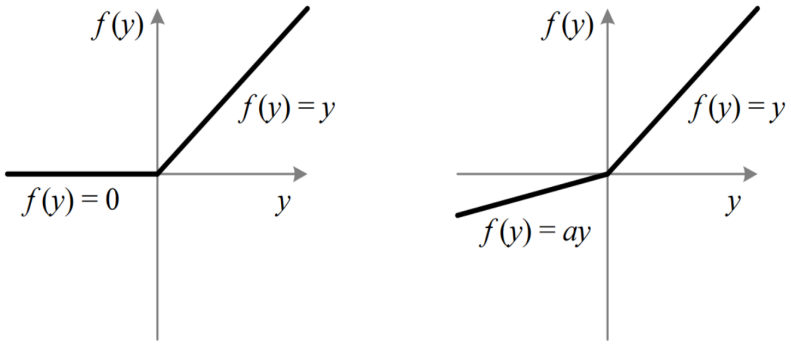

FIGURE 6. ReLU (left) and PReLU (right) [17]

4. **Training.** Our network has been trained on 785 brain images, and tested on 20, with a batch size of 64. The following has been considered for training.

- **Loss Function**
  A loss function in a neural network, and more specifically in a convolutional network, is the difference between the expected output and the actual output. Therefore, the network would be fully trained when the loss function is minimal, implying that the network has learned the expectations of its inputs. For our network, given a set of high-resolution images denoted as $\{Y_i\}$, in order to minimize the loss between the network's prediction $\{h_\theta(x_i)\}$ and the ground-truth high-resolution image $\{Y_i\}$, we use the mean squared error as our loss function, shown in Equation (1):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - h_\theta(x_i))^2 \tag{1}$$

  where n = number of training samples in each batch (i.e., $n = 64$).
- **Optimization**
  In deep learning, optimization is finding convergence, or finding the optimal or minimum of the error function that generalizes well. Normally, normal gradient descent can guarantee convergence to global minimum in a convex error surface, but it is very slow. Stochastic gradient descent is faster, but includes high variance

updates. Mini-batch gradient descent balances between the two. One type of mini-batch gradient descent is Adaptive Moment Estimation (Adam) [18]. Adam is used for optimization of our network. Adam computes the adaptive learning rates for each parameter, and stores an exponentially decaying average of the estimates of the first moment (the mean) and the second moment (the un-centered variance) of the gradients, respectively. The Adam update rule is expressed as in Equation (2):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{2}$$

where $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ and $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$.

- **Evaluation**

  Usually for evaluating image restoration quality, the Peak Signal-to-Noise Ratio (PSNR) metric is used. We therefore use PSNR as our evaluation metric. The PSNR is defined as in Equation (3):

$$PSNR = 20 \times \log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{3}$$

  Since $MAX = 1$, the PSNR equation simplifies to Equation (4):

$$PSNR = 20 \times \log_{10}\left(\frac{1}{\sqrt{MSE}}\right) \tag{4}$$

- **Learning Rate Decay**

  We have chosen to implement learning rate decay. Through the training, the learning rate decreases from its initial value. Our initial learning rate is set to be 0.001, with a decrease per 20 epochs.

- **Epochs and Early Stopping**

  Normally, it is a good practice to monitor the error on the validation set during training and stop (with some patience) if the validation error does not improve enough. We have used an early stopping with patience of 15. If the validation error does not improve much for 15 times, the network stops training. We choose to use 100 epochs; however, the network stops at 30. Figure 7 illustrates the training and testing validation loss plot.

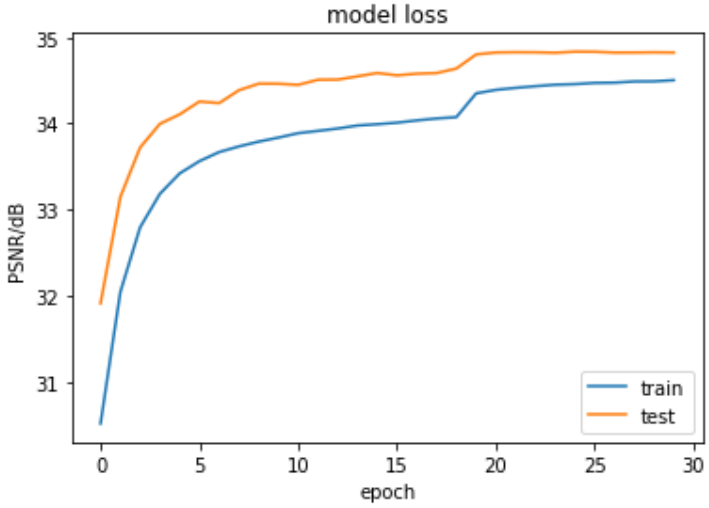

FIGURE 7. The training and testing validation loss plot

5. **Results and Discussion.** VDSR has achieved better results than SRCNN. Therefore, we choose to directly compare our results with VDSR.

5.1. **Comparison with scale 4.** The results of **Structural Similarity (SSIM)** [19] for scale 4 are shown for 5 full testing images in Table 1. Following Table 1, the results of SSIM for patches taken from the image, shown in white, are demonstrated in Figures 8-11.

TABLE 1. SSIM comparison with scale 4 for 5 full testing images

Test Image	Image 1	Image 2	Image 3	Image 5	Image 7
Bicubic	0.828	0.818	0.802	0.818	0.827
VDSR	0.849	0.841	0.825	0.837	0.849
Ours	**0.920**	**0.920**	**0.912**	**0.910**	**0.919**

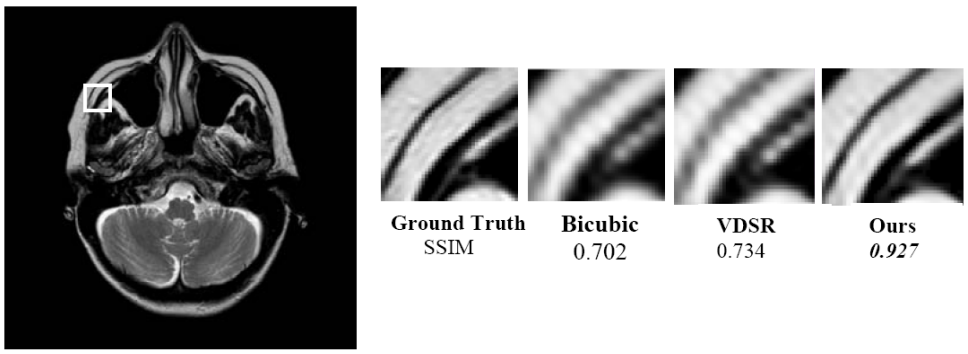

FIGURE 8. Test image 7 patch comparison results for scale 4

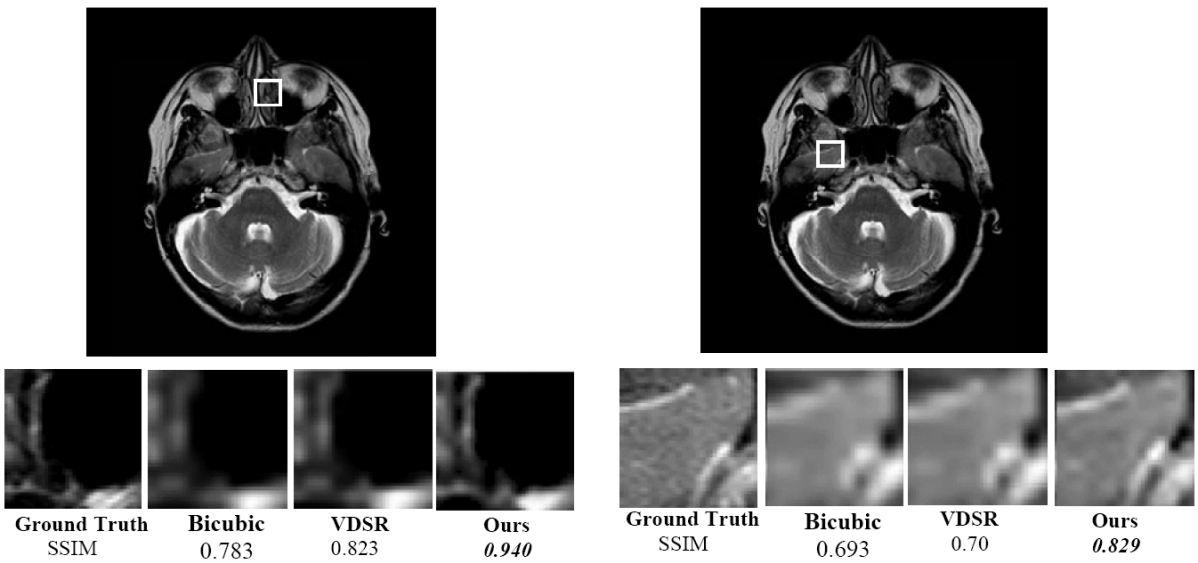

FIGURE 9. Test image 3 patch comparison results for scale 4

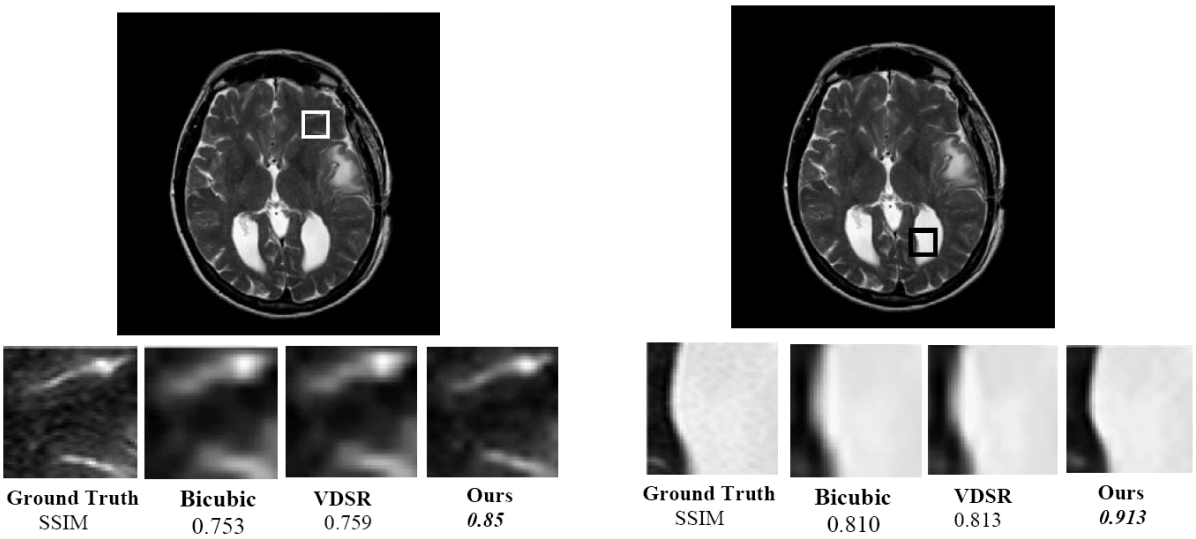

FIGURE 10. Test image 1 patch comparison results for scale 4

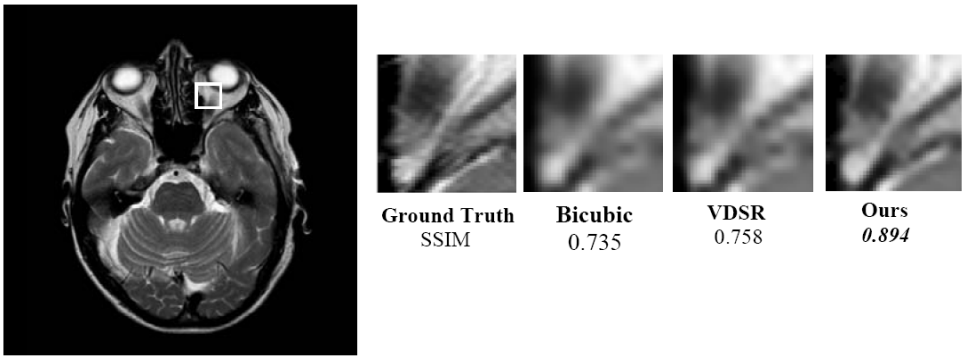

FIGURE 11. Test image 5 patch comparison results for scale 4

5.2. **Comparison with scale 2.** The results of SSIM for patches taken from the image, shown in white, are demonstrated in Figures 12-14 for the image with scale 2.

5.3. **Comparison with scale 3.** The results of SSIM for patches taken from the image, shown in red, are demonstrated in Figures 15 and 16 for the image with scale 3.

In Subsection 5.1, comparison with images of scale 4 is carried on. Firstly, Table 1 is shown to demonstrate the results of the SSIM between the reconstructed image and the original image. It can be seen from Table 1 that the proposed method outperforms the existing methods. For example, test image 2 achieves an SSIM of 0.920 using the proposed method, while VDSR achieves 0.841. Moreover, to further illustrate the comparison and the effectiveness of the proposed method, patches of size 41×41 are chosen from the original image, bicubic interpolated image, reconstructed image using VDSR and the proposed method. The patches are shown and the SSIM calculation for each patch is calculated. It can be seen that the SSIM comparison has outperformed the existing methods. A similar procedure is carried on for images with scale 2 in Subsection 5.2 and for scale 3 in Subsection 5.3.

5.4. **Comparison of the full image with PSNR.** The results of the **Peak Signal-to-Noise Ratio (PSNR)** expressed in dB for 5 of the full test images and for 2 different scales are shown in Table 2.

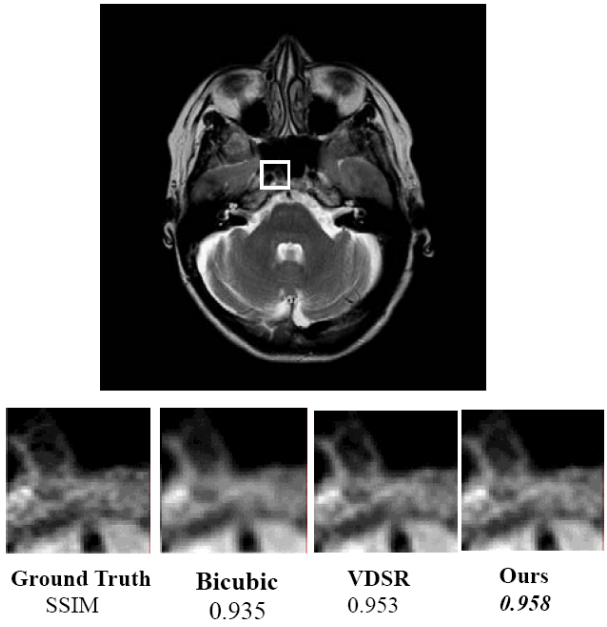

FIGURE 12. Test image 3 patch comparison results for scale 2

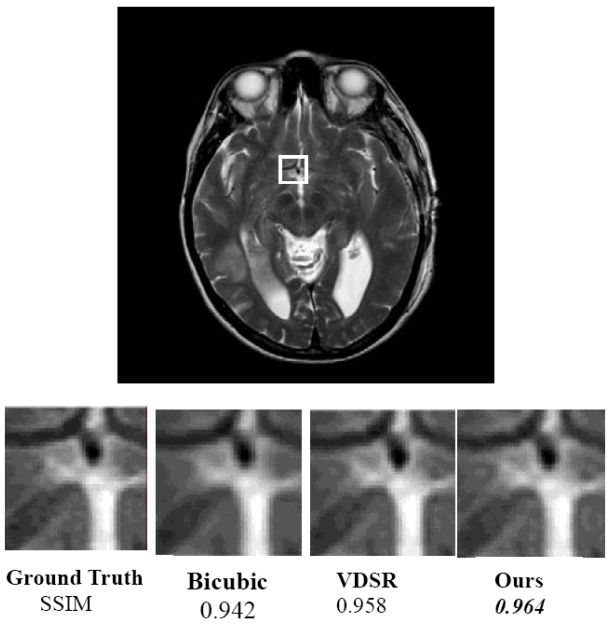

FIGURE 13. Test image 2 patch comparison results for scale 2

TABLE 2. PSNR comparison with scale 2 and 4 for 5 full testing images

Scale	Type	Image 1	Image 2	Image 3	Image 5	Image 7
×4	Bicubic	26.377	26.005	25.389	26.527	26.454
	VDSR	27.287	26.913	26.156	27.150	27.318
	Ours	**31.623**	**31.153**	**30.307**	**30.403**	**31.219**
×2	Bicubic	33.805	33.236	32.438	33.477	33.505
	VDSR	35.479	35.289	34.046	35.138	35.107
	Ours	**36.860**	**36.913**	**35.641**	**35.557**	**35.366**

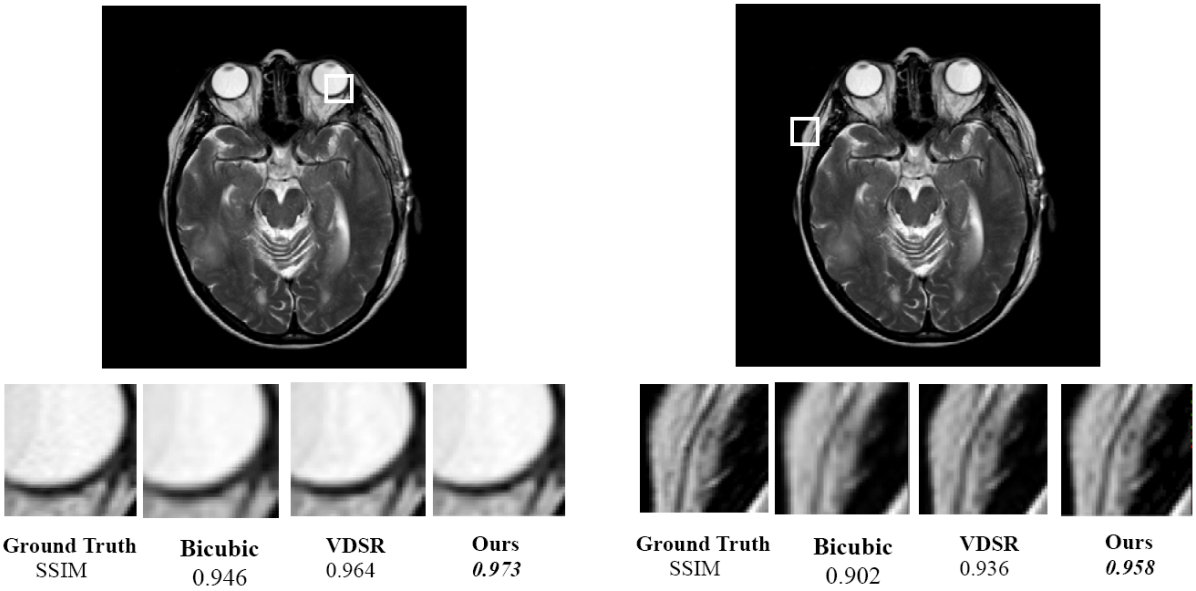

FIGURE 14. Test image 4 patch comparison results for scale 2

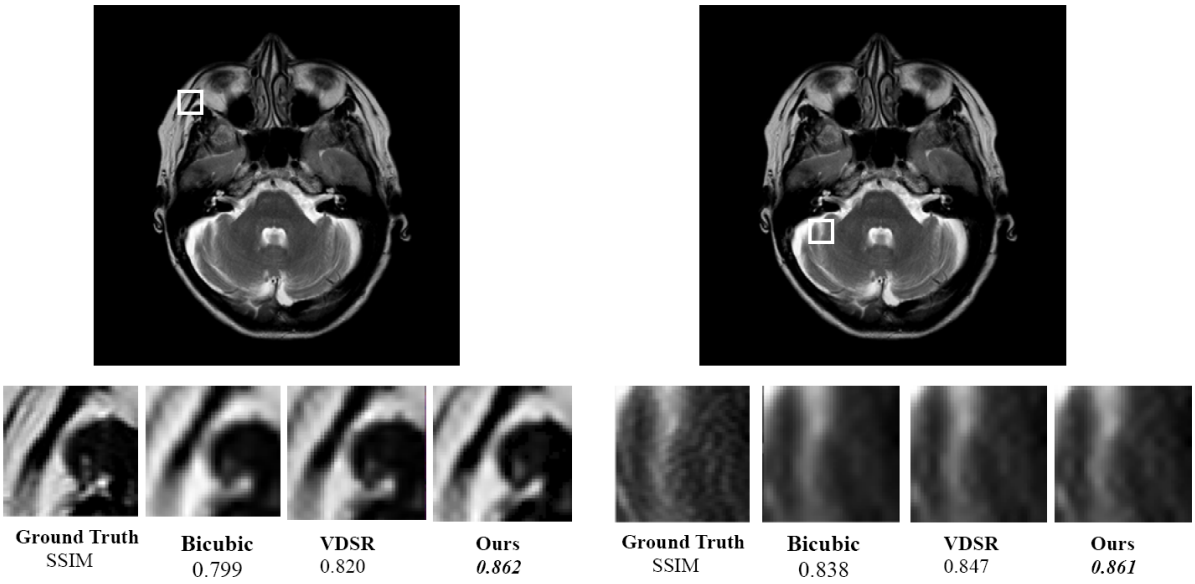

FIGURE 15. Test image 3 patch comparison results for scale 3

It can be clearly observed from the results of the testing images that the proposed method outperforms VDSR in both the structural similarity and PSNR evaluation methods.

6. **Conclusion.** We propose a deep convolutional network that is capable of reconstructing high-resolution images of DICOM brain images, from the interpolated low-resolution images. The network learns an end-to-end mapping between the low and high-resolution images. In the first stage of our network, we densely extract features from the input image. We then adopt residual learning and the mixture of convolutions to generate the high-resolution output image. We have shown that our network outperformed both bicubic interpolation, and the previously proposed VDSR model, in which we train on the same brain images. The proposed method has successfully managed to enhance the

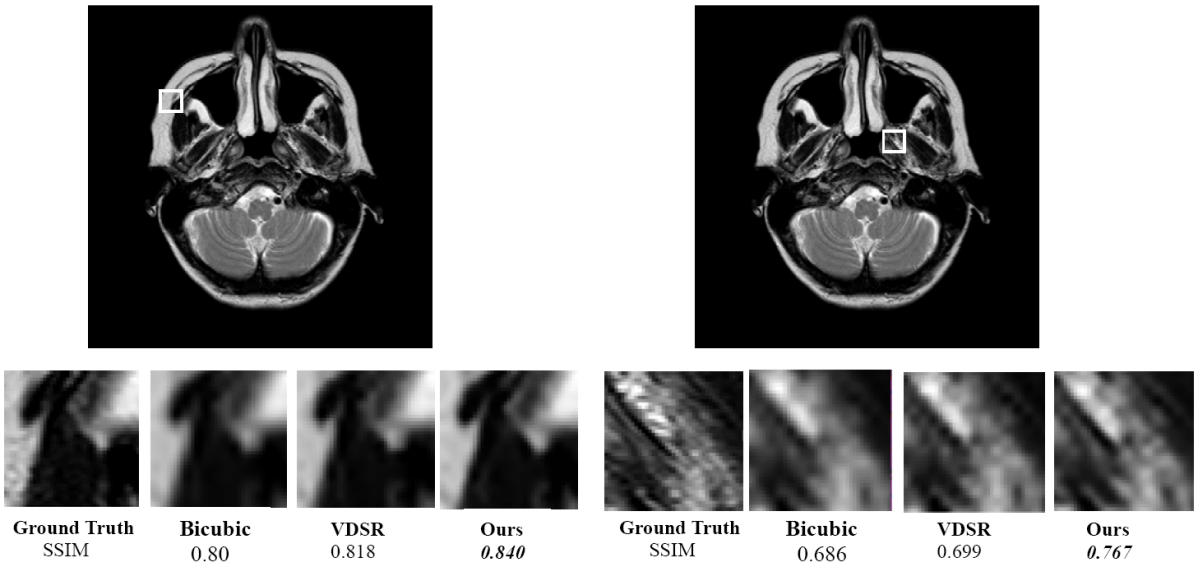

FIGURE 16. Test image 8 patch comparison results for scale 3

quality of the brain image, and outperforms existing methods. Further research may be carried on to focus on recovering all the details of the image such that the comparison with the reference image is nearly equal.

REFERENCES

[1] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. K. Bhatia, A. M. Marvao, T. Dawes, D. P. O'Regan and D. Rueckert, Cardiac image super-resolution with global correspondence using multi-atlas patchmatch, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol.16, pp.9-16, 2013.

[2] W. W. Zou and P. C. Yuen, Very low resolution face recognition problem, *The 4th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp.1-6, 2010.

[3] J. Sun, Z. Xu and H.-Y. Shum, Image super-resolution using gradient profile prior, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[4] K. I. Kim and Y. Kwon, Single-image super-resolution using sparse regression and natural image prior, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.32, pp.1127-1133, 2010.

[5] J. Yang, J. Wright, T. S. Huang and Y. Ma, Image super-resolution as sparse representation of raw image patches, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[6] J. Yang, J. Wright, T. S. Huang and Y. Ma, Image super-resolution via sparse representation, *IEEE Trans. Image Processing*, vol.19, no.11, pp.2861-2873, 2010.

[7] C. Yang, C. Ma and M. Yang, Single-image super-resolution: A benchmark, *European Conference on Computer Vision*, 2014.

[8] S. Schulter, C. Leistner and H. Bischof, Fast and accurate image upscaling with super-resolution forests, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.3791-3799, 2015.

[9] C. Dong, C. C. Loy, K. He and X. Tang, Image super-resolution using deep convolutional networks, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.38, pp.295-307, 2016.

[10] J. Kim, J. K. Lee and K. M. Lee, Accurate image super-resolution using very deep convolutional networks, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1646-1654, 2016.

[11] C. Dong, C. C. Loy and X. Tang, Accelerating the super-resolution convolutional neural network, *European Conference on Computer Vision*, 2016.

[12] W. Lai, J. Huang, N. Ahuja and M. Yang, Deep Laplacian pyramid networks for fast and accurate super-resolution supplementary material, *arXiv:1704.03915*, 2017.

[13] K. S. Sim, C. S. Ee and Z. Y. Lim, Contrast enhancement brain infarction images using sigmoidal eliminating extreme level weight distributed histogram equalization, *International Journal of Innovative Computing, Information and Control*, vol.14, no.3, pp.1043-1056, 2018.

[14] V. Teh, K. S. Sim and E. K. Wong, Contrast enhancement of CT brain images using gamma correction adaptive extreme-level eliminating with weighting distribution, *International Journal of Innovative Computing, Information and Control*, vol.14, no.3, pp.1029-1041, 2018.

[15] G. Huang, Z. Liu, L. V. Maaten and K. Q. Weinberger, Densely connected convolutional networks, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.2261-2269, 2017.

[16] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778, 2016.

[17] K. He, X. Zhang, S. Ren and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, *IEEE International Conference on Computer Vision (ICCV)*, pp.1026-1034, 2015.

[18] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980*, 2014.

[19] W. Zhou, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Processing*, vol.13, no.4, pp.600-612, 2004.