Deep Wavelet for Medical Image Super-resolution

Sahar Almahfouz Nasser
(194072001)

Department
Electrical Engineering
Indian Institute of Technology Bombay

Contents

List of Figures

Introduction

Any super-resolution algorithm aims to restore the high-resolution image from the low-resolution image or images.

We can establish a taxonomy of the super-resolution algorithms based on the number of the input low-resolution images required to reconstruct the high-resolution image. The first category is known as single-image super-resolution (SISR), while the second one is called multi-image super-resolution (MISR).

The super-resolution problem is an ill-posed one, because of the high-resolution image, associated with the low-resolution image (1), might be one of many possible solutions [28].

$$y = (x \circledast k) \downarrow_s + n \tag{1}$$

Where y is the low-resolution image, x is the corresponding high-resolution image convolved with the blurry kernel k. $\downarrow_s$ represents a downsampling with a factor s. And n is the added noise.

Any SISR algorithm falls into one of the following divisions: interpolation-based methods, reconstruction-based methods, and learning-based methods.

Bicubic Interpolation [11] and Lanczos Resampling [7] are good examples for interpolation-based methods. Although these methods are simple and fast, they are suffering from a deficiency of the accuracies represented by blurry outputs. In contrast, reconstruction based methods [3, 19] generate sharp-details rich outputs by assuming some underlying prior knowledge to confine the solution space. Such algorithms become time-consuming as the scale factor increases.

The last category is the learning-based methods that overpower the previous ones concerning both the computation time and the performance. These methods are machine learning-based methods that try to find a statistical relationship between the low-resolution image and the high-resolution image based on observing the training data. The Markov Random Field algorithm [8] and Neighbor Embedding [2] methods are well-known examples of learning-based algorithms.

1 Deep Architectures for SISR

Many SISR deep learning-based algorithms have been proposed in the last few years, in this chapter we will shed the light on some of the influential algorithms.

1.1 SRCNN

In 2015 Dong et al [5] proposed the first deep learning-based SISR architecture which defeats all the conventional methods.

As Figure (1) depicts, the architecture consists of three layers with filter sizes ($64\times1\times9\times9$), ($32\times64\times5\times5$) and ($1\times32\times5\times5$) respectively from left to right. With mean square error as the loss function. Although this architecture proved the power of deep learning in the field of single image super-resolution, it encountered some problems. Firstly, the bicubic input image is an over smoothed, leading to an erroneous approximation of the high-resolution image. Besides, the interpolation process itself is time-consuming and error-prone, especially if the downsampling kernel is unknown. Secondly, the architecture is very shallow, so the deeper network might give better results.

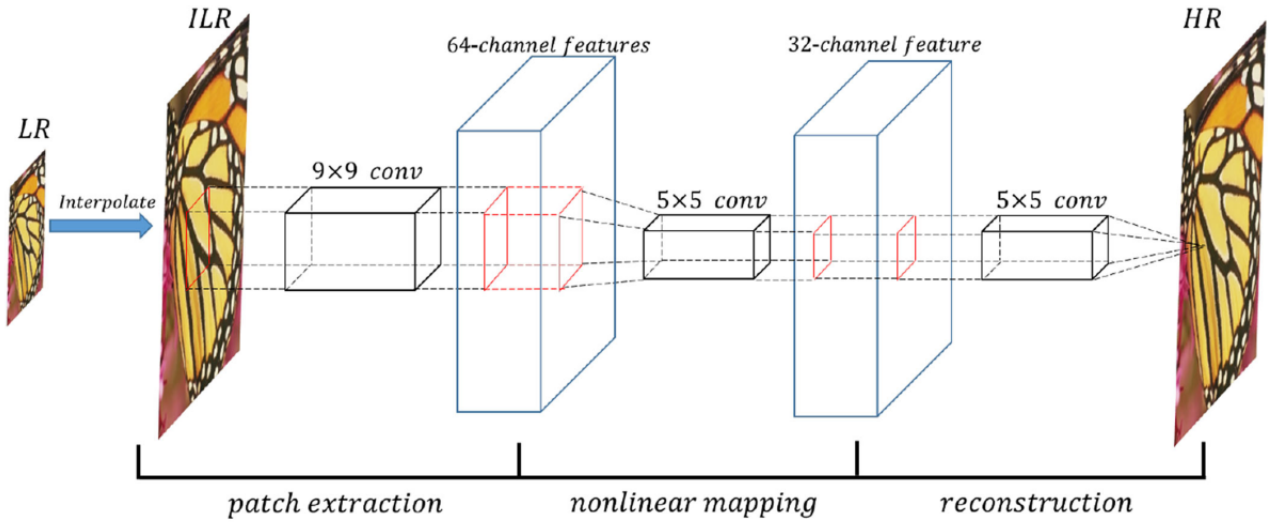


Figure 1: The architecture of the SRCNN.
[28]

1.2 FSRCNN

Fast Super-resolution Convolutional Network (FSRCNN) [4], unlike SRCNN, upsamples the feature maps at the end of the network using a simple deconvolution layer. This deconvolution layer consists of nearest-neighbor interpolation followed by convolution. This upsampling technique reduces the computation cost compared to SRCNN. See the figure(2).

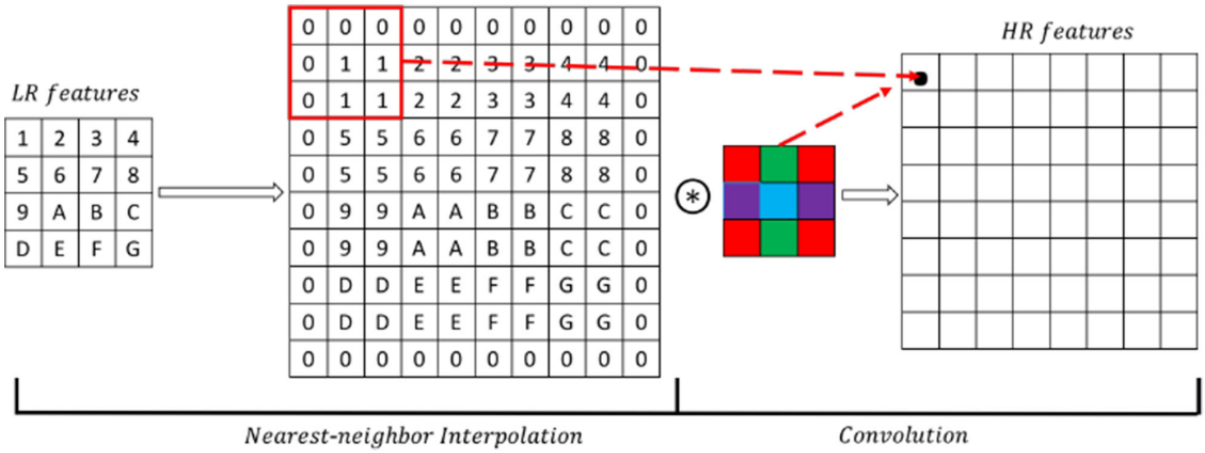


Figure 2: The diagram of the deconvolution layer used in FSRCNN.
[28]

1.3 ESPCNN

However, FSRCNN technique introduces some redundancy. In [21] Shi et al proposed an efficient sub-pixel convolutional neural network (ESPCNN) that uses a subpixel convolution layer given by the equation (2).

$$I^{SR} = f^L(I^{LR}) = PS(W_L * f^{L-1}(I^{LR}) + b_L) \tag{2}$$

where PS is a periodic shuffling operator that reshapes the elements of a $(H \times W \times C \times r^2)$ tensor to a $(rH \times rW \times C)$ tensor,where r is the upsampling factor, see figure(3).

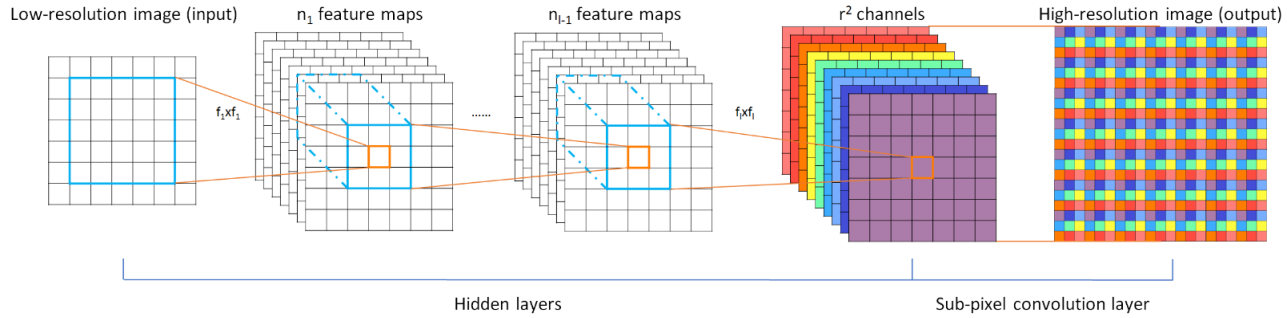


Figure 3: The architecture of ESPCNN.
[21]

1.4 VDSR

In 2016 Kim et al [12] proposed the first very-deep neural network for SISR called VDSR.

VDSR architecture is based on VGG net [22], it consists of twenty layers with a global skip connection between the low-resolution input and the high-resolution output.

Two tricks were used to facilitate the training and boost the convergence, the high learning rate, and the gradient clipping. VDSR uses bicubic interpolation to upsample the input low-resolution image at different scales before feeding it to the network, see figure(4-a).

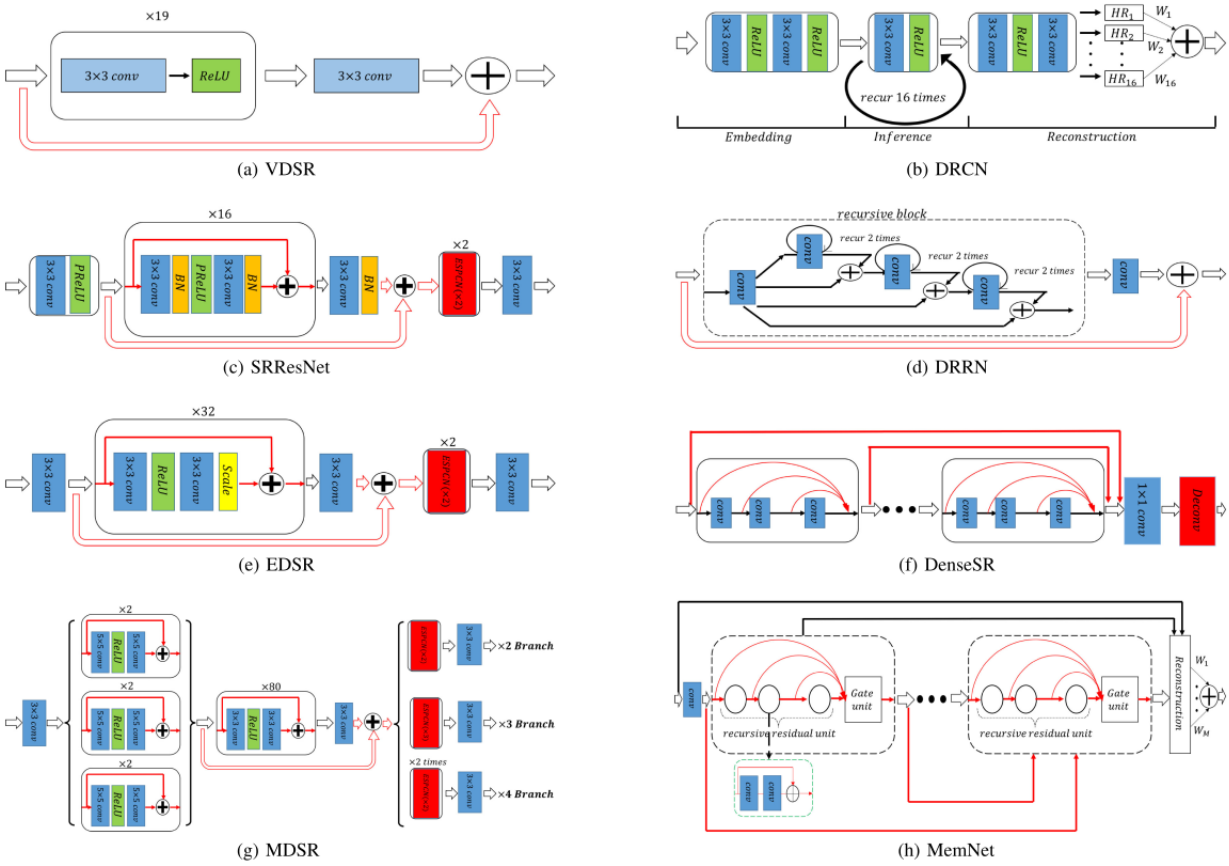


Figure 4: Very deep architectures for single image super-resolution.
[28]

1.5 DRCN

The kernels used in VDSR were very similar, thus to reduce the number of weights, the same authors proposed a Deeply-recursive Convolutional Network (DRCN)[13].

The suggested architecture comprises three parts. The first part is the embedding network f_1 which is two convolutional layers with ReLU activation functions(3).

$$\begin{aligned} \boldsymbol{H}_{-1} &= max(0, \boldsymbol{W}_{-1} \times \boldsymbol{x} + \boldsymbol{b}_{-1}) \\ \boldsymbol{H}_0 &= max(0, \boldsymbol{W} \times \boldsymbol{H}_{-1} + \boldsymbol{b}_{-1}) \\ f_1(\boldsymbol{x}) &= \boldsymbol{H}_0 \end{aligned} \tag{3}$$

The second part is the inference network f_2 which passes f_1 through a recursive layer to get the output f_2. The recursive layer is a convolutional layer with ReLU activation function. Due to sharing the parameters between the recursive layers, adding more recursive layers will not change the number of parameters though it will widen the receptive field only(4).

$$\begin{aligned} \boldsymbol{H}_d &= g(\boldsymbol{H}_{d-1}) = max(0, \boldsymbol{W} \times \boldsymbol{H}_{d-1} + \boldsymbol{b}) \\ f_2(\boldsymbol{H}) &= (g \circ g \circ ....\circ) g(\boldsymbol{H}) = g^D(\boldsymbol{H}) \end{aligned} \tag{4}$$

The final part is the reconstruction network which outputs the high-resolution image (5).

$$\begin{aligned} \boldsymbol{H}_{D+1} &= max(0, \boldsymbol{W}_{D+1} \times \boldsymbol{H}_D + \boldsymbol{b}_{D+1}) \\ \hat{\boldsymbol{y}} &= max(0, \boldsymbol{W}_{D+2} \times \boldsymbol{H}_{D+1} + \boldsymbol{b}_{D+2}) \\ f_3(\boldsymbol{H}) &= \hat{\boldsymbol{y}} \end{aligned} \tag{5}$$

As figure(4-b)depicts, the final result is a linear combination of sixteen intermediate results with learnable coefficients sum up to one. The flaws of this algorithm are summarized in the difficulty of the training due to the vanishing gradient, and the invariability of the learnable scalars with different input images.

1.6 SRResNet

To overcome the problem of vanishing gradient, researchers came up with skip connections-based architectures such as SRResNet [15]. This architecture is composed of sixteen residual blocks [10] with a batch normalization layer in each block to stabilize the training. See figure(4-c).

1.7 DRRN

A deep recursive residual network [23]is a stack of residual blocks. The convolutions of each residual block are recursive, thus increasing the number of residual blocks will not increase the number of parameters. Figure (4-d).

1.8 EDSR

In the Enhanced Deep Residual Network [16] the authors removed the batch normalization layers from the residual blocks. As ResNet proposed to solve classification problems, the inner feature maps are abstract, so the shift caused by the batch normalization will not affect the convergence. In contrast, the SISR is a regression problem where the input and the output images are very related, so such a shift will affect the performance. Figure(4-e)

1.9 SRDenseNet

SRDenseNet [25] exploited the fact that the residual blocks allow the re-usage of the features while the dense blocks investigate new features by concatenating the features from different blocks before the deconvolution layer. Figure(4-f)

1.10 MemNet

The novelty of this paper [24] revealed in the idea of creating a sort of short term memory by adding dense connections within the block. Similarly, the dense connections from previous blocks represent a long term memory.Figure(4-h).

1.11 LapSRN

The performances of the previous approaches deteriorate when the upsampling factor increases. For big upsampling factors such as eight, priors are used to restrict the solution space.

Lai et al proposed Deep Laplacian Pyramid Networks for Super-Resolution [14] which comprises two parts the extractor and the reconstructor. For instance, at a level S the extractor contains d convolutional layers and one deconvolutional layer to upsample the extracted features by a factor of two. At the reconstructor, these features will be added pixel-wise to an upsampled version of the input image. See Figure(5).

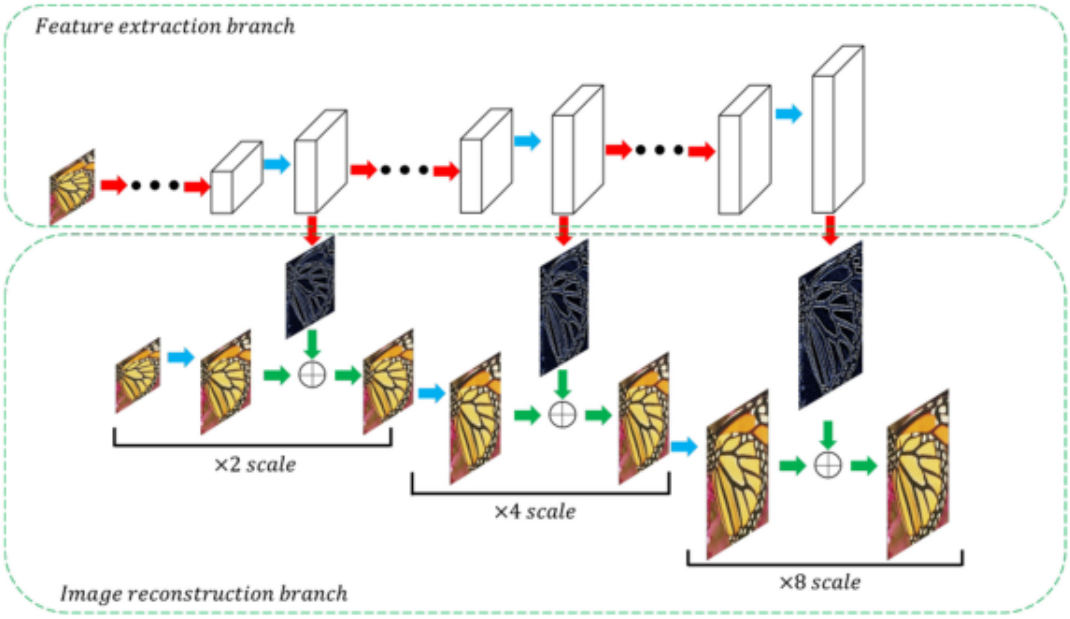
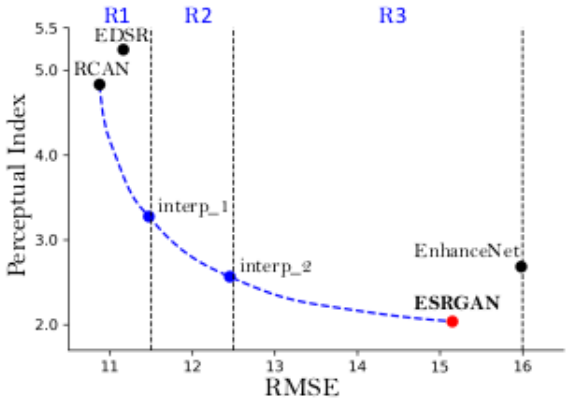


Figure 5: The architecture of Laplacian Pyramid Networks for Sigle Image Super-Resolution. [28]

1.12 ESRGAN

Enhanced Super-Resolution Generative Adversarial Network is a perceptual-driven approach proposed by Wang et al. [27] won the PIRM2018-SR Challenge, see Figure(6).

This method focuses on increasing the high-resolution details in the super-resolved image, unlike the PSNR-oriented methods that output over-smoothed results, see Figure (7).



Results on PIRM self val dataset

Method	PI	RMSE
ESRGAN	**2.040**	15.15
interp_2	2.567	12.45
EnhanceNet	2.688	15.99
interp_1	3.279	11.47
RCAN	4.831	10.87
EDSR	5.243	11.16

Figure 6: The results of PIRM2018-SR Challenge. [27]

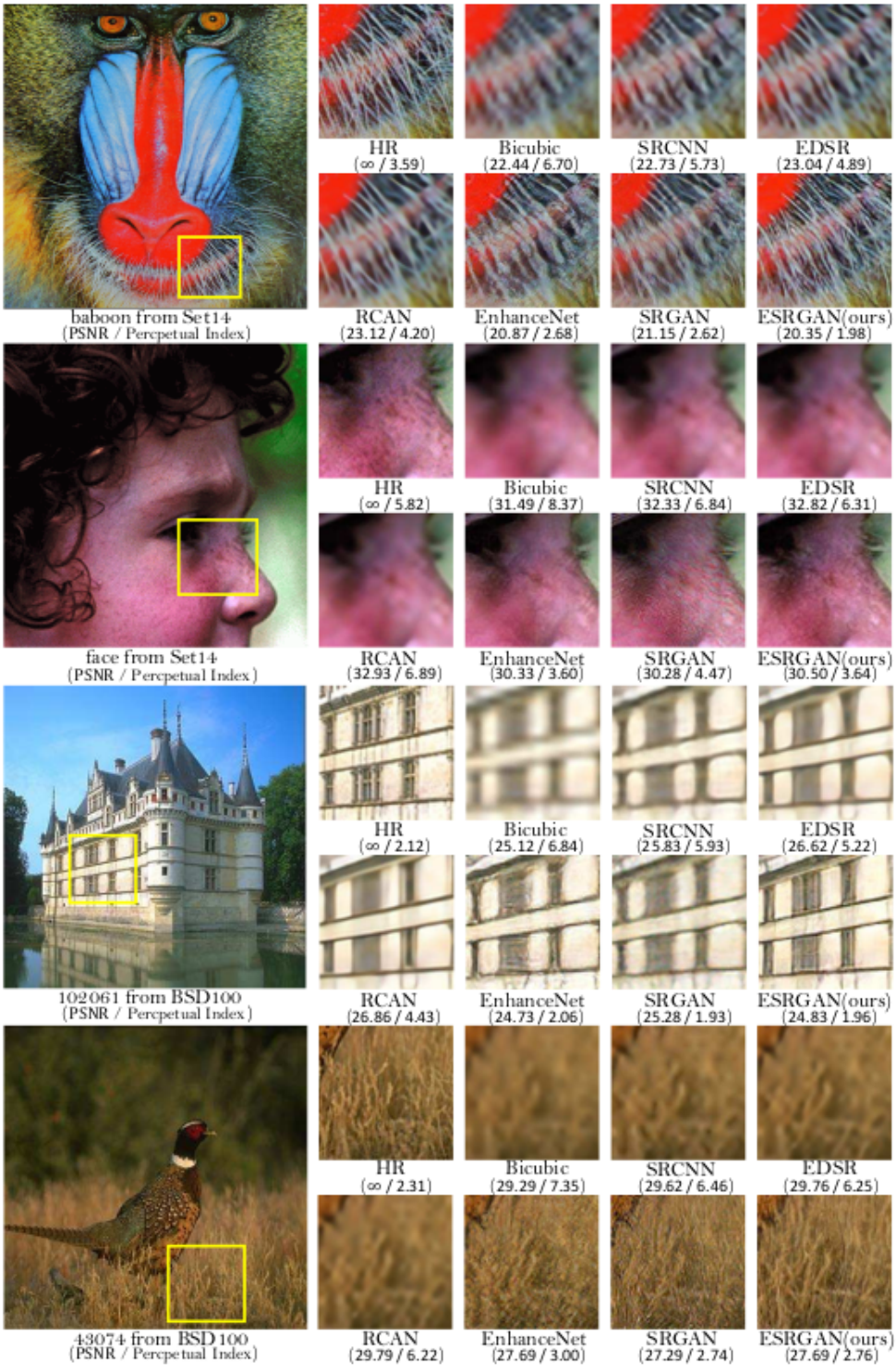


Figure 7: A comparison between the ESRGAN and other methods for SISR.
[27]

This paper presented many new ideas such as using residual-in-residual dense blocks, taking out the

batch normalization layers, using relativistic average cross-entropy loss, and finally using the feature maps before activation in the VGG network to compute the perceptual loss.

Figure (8) shows the architecture of the generator, which is a sequence of twenty-three residual-in-residual dense blocks followed by upsampling and convolution layers.

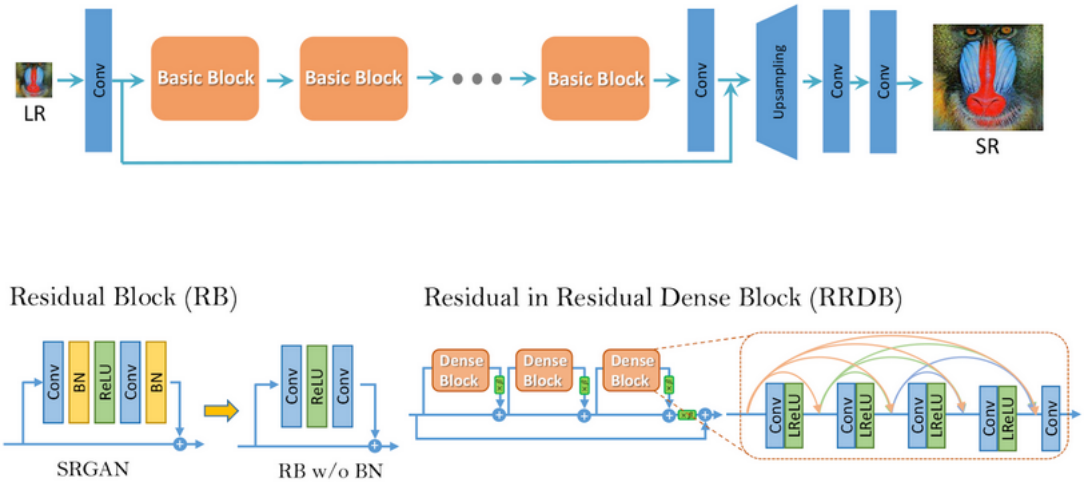


Figure 8: The architecture of the generator of the ESRGAN . [27]

The final loss function of the generator is a weighted combination of three losses. The first loss is the perceptual loss, which is the mean square error between the feature maps of the pre-trained VGG generated when the input is the super-resolution image and the ones generated when the input is the generated(fake) image. Figure(9) shows the results of using the feature maps before and after the activation. The second loss is the relativistic average cross-entropy loss. And the final loss is the content loss, which is the L_1 norm of the difference between the generated image and the super-resolution image, as stated in (6).

$$\begin{aligned} L_D^{Ra} &= -\mathbb{E}_{xr}[log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{xf}[log(1 - D_{Ra}(x_f, x_r))] \\ L_G^{Ra} &= -\mathbb{E}_{xr}[log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{xf}[log(D_{Ra}(x_f, x_r))] \\ L_G &= L_{percep} + \lambda L_G^{Ra} + \eta L_1 \\ L_1 &= \mathbb{E}_{xi}||G(x_i) - y||_1 \end{aligned} \tag{6}$$

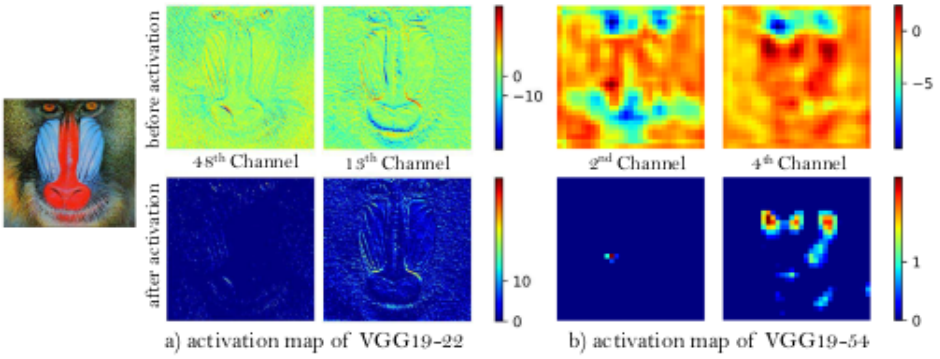


Figure 9: Using the feature maps before activation to compute the perceptual loss preserves more details than using the maps after the activation .

[27]

2 Deep Wavelet Super Resolution

The wavelet-based deep neural networks proved their efficiency in the domain of single image super-resolution and image denoising. The core idea of these architectures is about replacing the pooling layers with wavelet transform (WT). On the one hand, pooling layers are used in deep neural networks to avoid overfitting and reduce the computation burden. On the other hand, these layers cause loss of information. In 2017, Bae et al[1] showed that replacing the pooling layers with (DWT) improves the reconstruction results, see figure 10. In the same year Guo et al[9] proposed deep wavelet super-resolution network. This network predicts the missing details of the wavelet coefficients of the low-resolution images to obtain the coefficients of the SR image which serve as input to IDWT to generate the SR image. Figure 11 depicts the procedure of 2dDWT, where they treat the 2D signal $\boldsymbol{x}[n, m]$ (pixel at n column, n row) as 1D signals among the rows $\boldsymbol{x}[n, :]$ and 1D signals among the columns $\boldsymbol{x}[:, m]$. The output of this procedure consists of four sub-bands: average (LL), vertical (HL), horizontal (LH), and diagonal (HH).

$2dDWT \longrightarrow LA, LV, LH, LD := 2dDWTLR$

$\triangle SB = HRSB - LRSB = \{HA - LA, HV - LV, HH - LH, HD - LD\} = \{\triangle A, \triangle V, \triangle H, \triangle D\}$

And the cost is given by the following equation.

$$cost = \frac{1}{2}||\triangle SB - f(LRSB)||_2^2 \tag{7}$$

Essentially, the network learns the differences (residuals) between wavelet sub-bands of LR and HR images, see figure12.

$SRSB = \{SA, SV, SH, SD\} = LRSB + \triangle SB$

$SR = 2dIDWT\{SRSB\}$

Two years later, Pengju Liu et al[17] developed multi-level wavelet CNN (MWCNN), where they replaced the pooling layers of the U-net encoder with DWT and the upsampling layers at the U-net decoder with inverse discrete wavelet transform (IDWT),see figure 13. The proposed method beat the state-of-the-art architectures in the domains of single image super-resolution, artifacts removal from JPEG images, and image denoising.

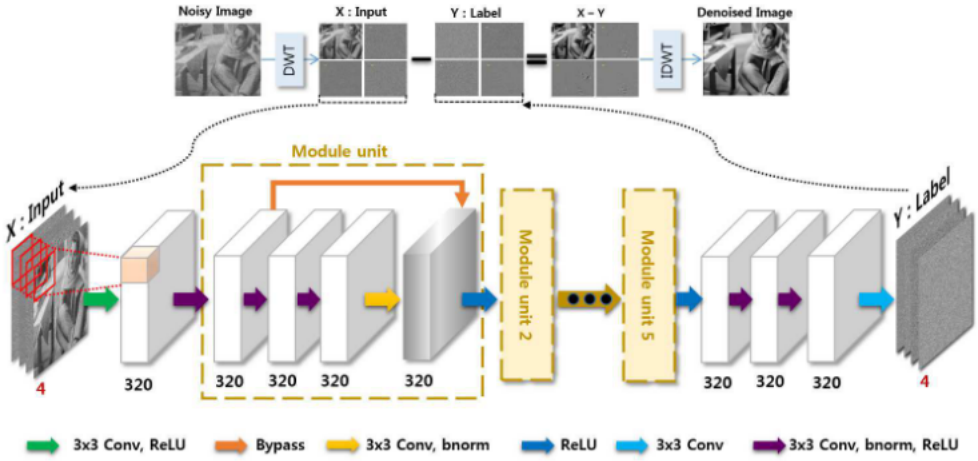


Figure 10: The architecture of wavelet domain deep residual learning network.
[1]

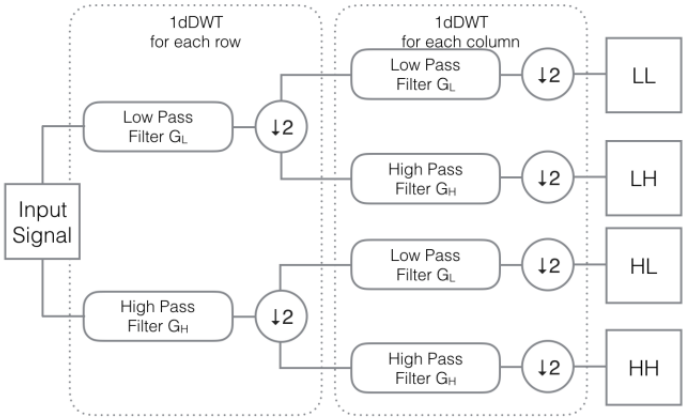


Figure 11: 2dDWT of a 2D image.
[9]

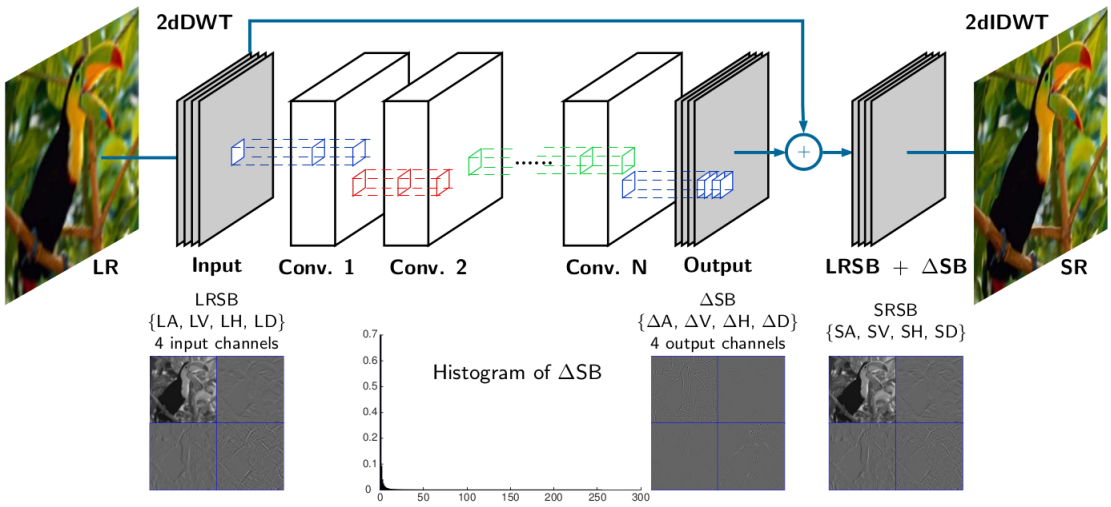


Figure 12: The architecture of deep wavelet super-resolution network. [9]

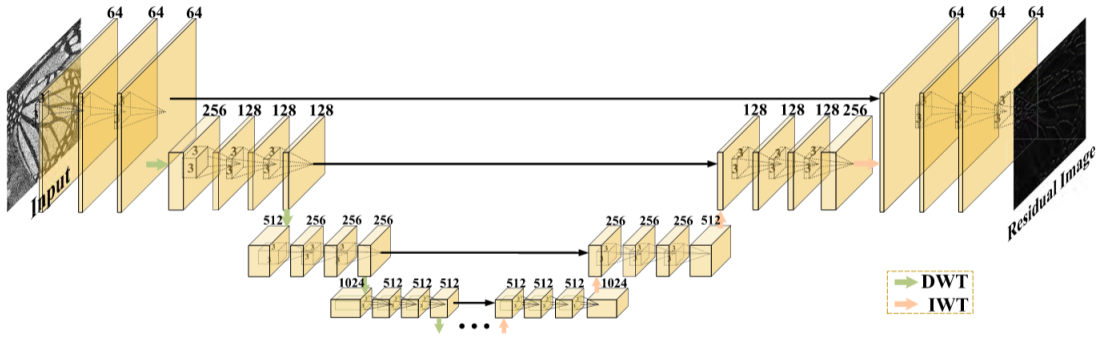


Figure 13: The architecture of multi-level wavelet network. [17]

We can decompose any 2D image $\boldsymbol{x}$ into four subband images $\boldsymbol{x}_{LL}$, $\boldsymbol{x}_{LH}$, $\boldsymbol{x}_{HL}$, and $\boldsymbol{x}_{HH}$ using four orthogonal filters $\boldsymbol{f}_{LL} = [1, 1; 1, 1]$, $\boldsymbol{f}_{LH} = [-1, -1; 1, 1]$, $\boldsymbol{f}_{HL} = [-1, 1; -1, 1]$, and $\boldsymbol{f}_{HH} = [1, -1; -1, 1]$.

The subbands are given by the following equations:

$$\boldsymbol{x}_{LL} = (\boldsymbol{f}_{LL} \circledast \boldsymbol{x}) \downarrow_2 \tag{8}$$

$$\boldsymbol{x}_{LH} = (\boldsymbol{f}_{LH} \circledast \boldsymbol{x}) \downarrow_2 \tag{9}$$

$$\boldsymbol{x}_{HL} = (\boldsymbol{f}_{HL} \circledast \boldsymbol{x}) \downarrow_2 \tag{10}$$

$$\boldsymbol{x}_{HH} = (\boldsymbol{f}_{HH} \circledast \boldsymbol{x}) \downarrow_2 \tag{11}$$

Here $\circledast$ is the convolution operator, and $\downarrow_2$ is a down-sampling with a factor of two.

Thus, the downsampling using the DWT requires four fixed convolution filters with a stride of 2. The (i,j)-th value of the subbands given by the following:

$$\boldsymbol{x}_{LL}(i, j) = \boldsymbol{x}(2i - 1, 2j - 1) + \boldsymbol{x}(2i - 1, 2j) + \boldsymbol{x}(2i, 2j - 1) + \boldsymbol{x}(2i, 2j) \tag{12}$$

$$\boldsymbol{x}_{LH}(i,j) = -\boldsymbol{x}(2i-1,2j-1) - \boldsymbol{x}(2i-1,2j) + \boldsymbol{x}(2i,2j-1 + \boldsymbol{x}(2i,2j) \tag{13}$$

$$\boldsymbol{x}_{HL}(i,j) = -\boldsymbol{x}(2i-1,2j-1) + \boldsymbol{x}(2i-1,2j) - \boldsymbol{x}(2i,2j-1) + \boldsymbol{x}(2i,2j) \tag{14}$$

$$\boldsymbol{x}_{HH}(i,j) = \boldsymbol{x}(2i-1,2j-1) - \boldsymbol{x}(2i-1,2j) - \boldsymbol{x}(2i,2j-1 + \boldsymbol{x}(2i,2j) \tag{15}$$

It is worth mentioning that the average pooling with a stride of 2 is given by:

$$\boldsymbol{x}_l(i,j) = (\boldsymbol{x}_{l-1}(2i-1,2j-1) + \boldsymbol{x}_{l-1}(2i-1,2j) + \boldsymbol{x}_{l-1}(2i,2j-1) + \boldsymbol{x}_{l-1}(2i,2j))/4 \tag{16}$$

And we can reconstruct the image $\boldsymbol{x}$ without lossing of information using IDWT as shown below.

$$\boldsymbol{x}(2i-1,2j-1) = (\boldsymbol{x}_{LL}(i,j) - \boldsymbol{x}_{LH}(i,j) - \boldsymbol{x}_{HL}(i,j) + \boldsymbol{x}_{HH}(i,j))/4 \tag{17}$$

$$\boldsymbol{x}(2i-1,2j) = (\boldsymbol{x}_{LL}(i,j) - \boldsymbol{x}_{LH}(i,j) + \boldsymbol{x}_{HL}(i,j) - \boldsymbol{x}_{HH}(i,j))/4 \tag{18}$$

$$\boldsymbol{x}(2i,2j-1) = (\boldsymbol{x}_{LL}(i,j) + \boldsymbol{x}_{LH}(i,j) - \boldsymbol{x}_{HL}(i,j) - \boldsymbol{x}_{HH}(i,j))/4 \tag{19}$$

$$\boldsymbol{x}(2i,2j) = (\boldsymbol{x}_{LL}(i,j) + \boldsymbol{x}_{LH}(i,j) + \boldsymbol{x}_{HL}(i,j) + \boldsymbol{x}_{HH}(i,j))/4 \tag{20}$$

Wang et al[26] proposed a multi-wavelet residual dense convolutional neural network (MWRDCNN),see figure 14. In the proposed method, they introduced residual dense blocks (RDBs) to the U-net architecture. Their results revealed that introducing RDBs to the (MWCNN) increased the learning efficiency and improved the results especially in removing the Gaussian noise, see figure 15.

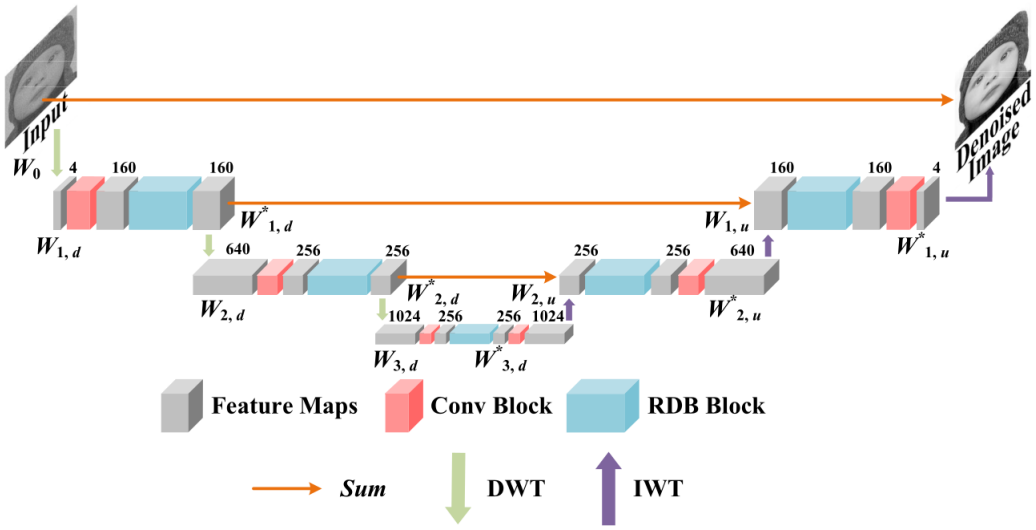


Figure 14: The architecture of MWRDCNN. [26]

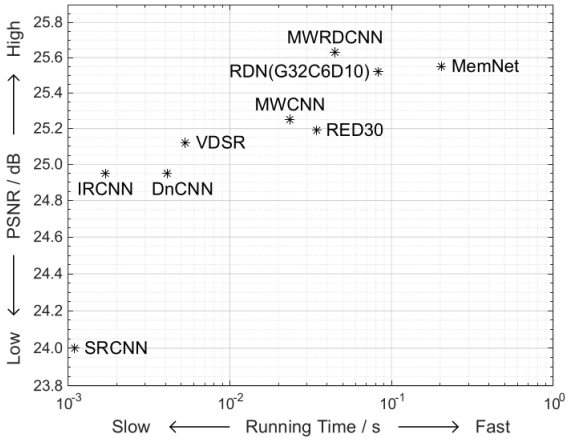


Figure 15: The denoising performance of the proposed MWRDCNN and other 8 state-of-the-art convolutional neural networks.

[26]

Liu et al [18] presented a densely self-guided wavelet network (DSWN) which integrates multi-scale information to extract good local features. Their proposed architecture consists of two parts. The first part is a U-net-like architecture with dense residual blocks. While the second part consists of two branches, one branch focuses on the dark areas of the image, they called it an end-to-end branch, and the other branch focuses on the bright parts of the image, they called it a residual branch, see figure 16. This paper emphasizes the harmful effect of using the batch norm layers on denoising.

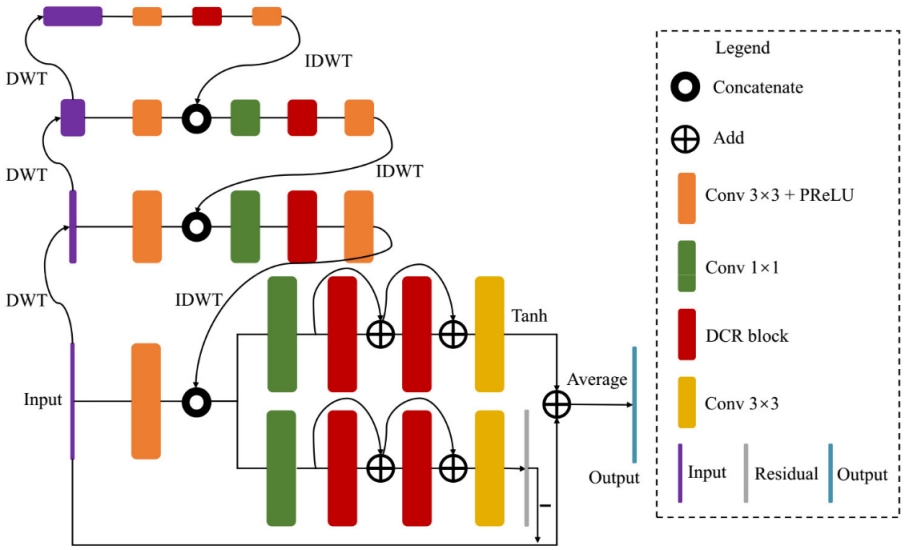


Figure 16: The architecture of the densely self-guided wavelet network(DSWN).
[18]

3 Qualitative criteria for super resolution

Many measures are available to quantify the quality of the super-resolution technique. In this section, we shed the light on some of the most prominent measures.

1. Mean Square Error (MSE): Given two images I and $\hat{I}$, the MSE is given by the following equation 21:

$$MSE = \frac{1}{N}||I - \hat{I}||^2 \tag{21}$$

   Where N represents the number of pixels in I and $\hat{I}$.

2. Peak Signal to Noise Ratio (PSNR): By considering $L = 255$ the PSNR is given by (22).

$$PSNR = 10log_{10}(\frac{L^2}{MSE}) \tag{22}$$

3. Structural Similarity Index Measure (SSIM): It measures the perceptual differences between two images I and $\hat{I}$. It is given by (23):

$$SSIM(I, \hat{I}) = \frac{2\mu_I\mu_{\hat{I}} + k_1}{\mu_I^2 + \mu_{\hat{I}}^2 + k_1} \times \frac{\sigma_{I\hat{I}} + k_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + k_2} \tag{23}$$

   Where μ_I and σ_I are the mean and the standard deviation of the image I. $\mu_{\hat{I}}$ and $\sigma_{\hat{I}}$ are the mean and the standard deviation of the image $\hat{I}$.

4 Data

The Super-resolution of Multi-Dimensional Diffusion MRI (Super MUDI) dataset contains the data of four healthy human subjects with ages range between 19 and 46 years [20]. For each subject 1,344 MRI volumes are provided. The imaging device was clinical 3T Philips Achieva Scanner (Best, Netherlands) with a 32-channel adult head coil. The Super MUDI Challenge comprises two tasks: isotropic, and anisotropic super-resolution. The names of these tasks were derived from the acquisition strategies of the low-resolution MRI data. The objective of using two downsampling strategies is to compare the combinations of the down-sampling methods and the super-resolution approaches that can be the best to be used in a clinical scheme to obtain simulated high-quality and high-fidelity MRI images while reducing the acquisition time. In the anisotropic subsampling the volume has high in-plane resolution(2.5mm $\times$ 2.5mm) but thick axial slice (5mm), while in the isotropic subsampling the volume has low resolution (5mm) in all the directions. For each subject the sizes of the high resolution, isotropic, and anisotropic volumes are (76, 92, 56), (38, 46, 28), and (76, 92, 28) respectively, see figure 17.

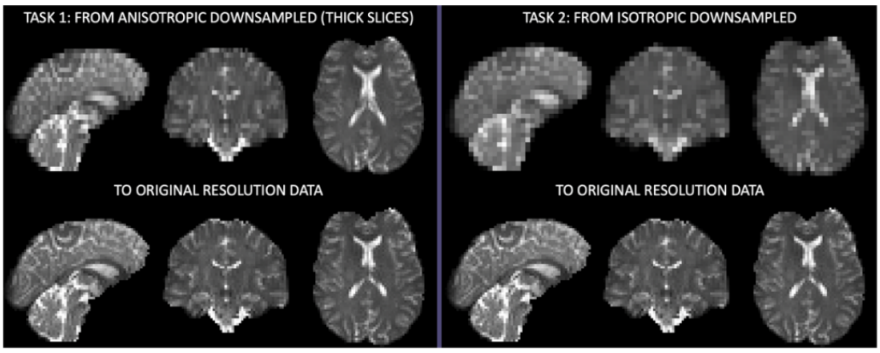


Figure 17: A sample of the data from Super-MUDI challenge.
[20]

5 Proposed Architecture for image super-resolution

My proposed architecture is inspired from [6]. Dilated Convolutional Encoder-Decoder Network (DCED) takes a low-resolution image (LR) as an input, interpolates it using a bicubic interpolation operator. The interpolated image gets passed into two successive convolutional blocks, each consisting of a 2 D dilated convolution layer followed by a ReLU activation function. The output then gets passed into three successive encoder-decoder blocks (EDB). The output of the EDB3 gets concatenated with the input of the EDB1. After upsampling the output of EDB3, it gets added to the bicubic interpolated image. As if DCED network recovers finer details by predicting the residual image which gets added to the bicubic interpolated image to produce the high-resolution (HR) image as the final output of the network.

To exploit the potential of the wavelet transform, I added a discrete wavelet transform block just after the bicubic-interpolated image then I used the subbands as input of the proposed method, see figure 18.

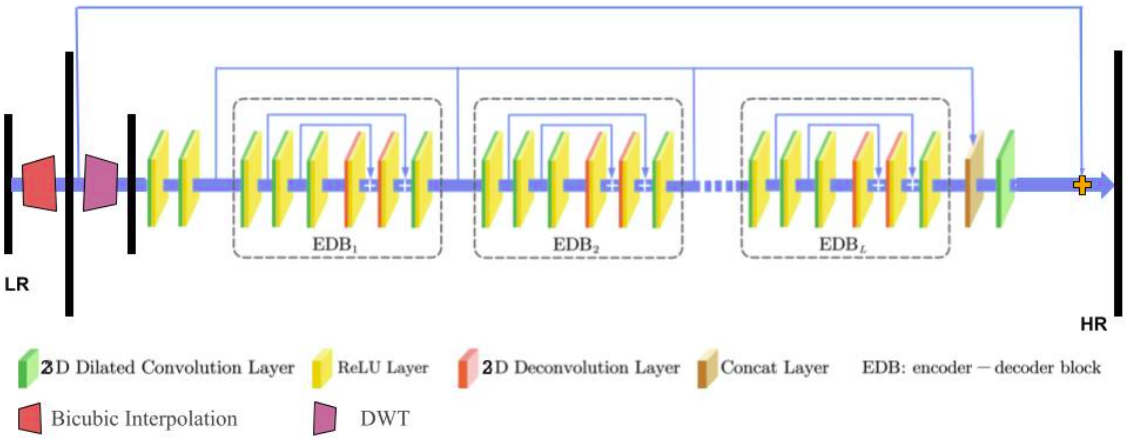


Figure 18: The block diagram of the proposed method.

5.1 Training and Testing

My proposed network consists of 53188 parameters, I trained it on 1 GPU until convergence. The network trained on 1344 slices belong to subject 1, validated on 1344 slices belong to subject 3, and tested on 1344 slices of subject2. For a fair judgment of the proposed method, I compared it with the baseline of [6].

5.1.1 Baseline

In this experiment, I trained the baseline proposed in [6].

Figure 19 shows the loss function which is the mean square error between the reconstructed image and the high-resolution image (GT), the structural similarity (SSIM), and peak signal to noise ratio (PSNR) on the training and validation datasets.

Figure 20 shows a comparison between the baseline and the bicubic interpolation on one slice of the testing dataset.

Table 1 shows the values of the evaluation metrics of the baseline network on the testing dataset.

MSE	SSIM	PSNR
0.0014	0.8612	28.9904

Table 1: The evaluation metrics of the baseline network on the testing dataset.

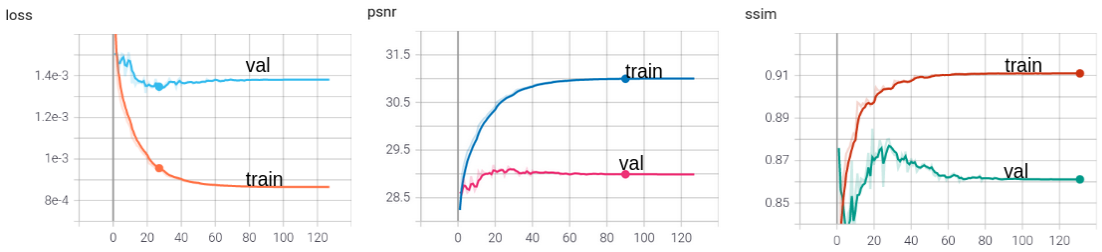


Figure 19: The training/validation graphs of the baseline network. The sub-figures from left to right are MSE, SSIM, and PSNR.

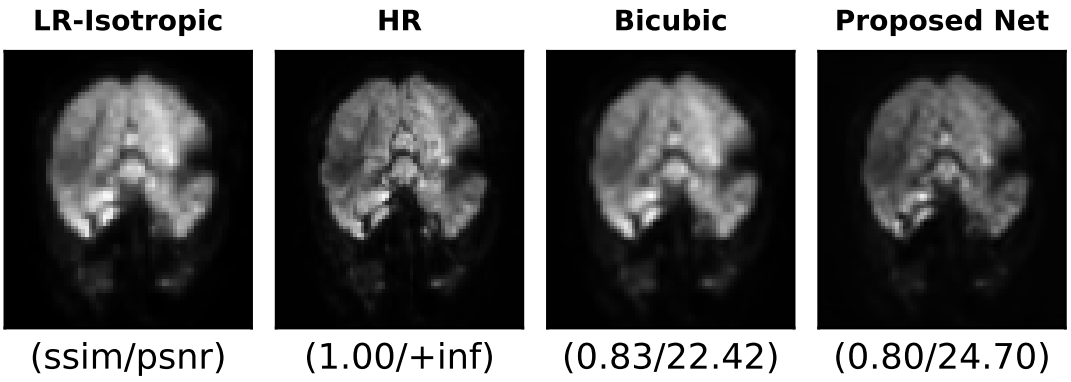


Figure 20: Visualization of the results of the baseline network method on a slice from the testing dataset.

5.1.2 Experiment: 1

In this experiment I trained the proposed method showed in figure 18. The values of the evaluation metrics of my proposed method on the testing dataset are as shown in Table 2.

MSE	SSIM	PSNR
0.0017	0.7907	28.327

Table 2: The evaluation metrics on the testing dataset.

Figure 21 shows the training and validation graphs of the proposed method.

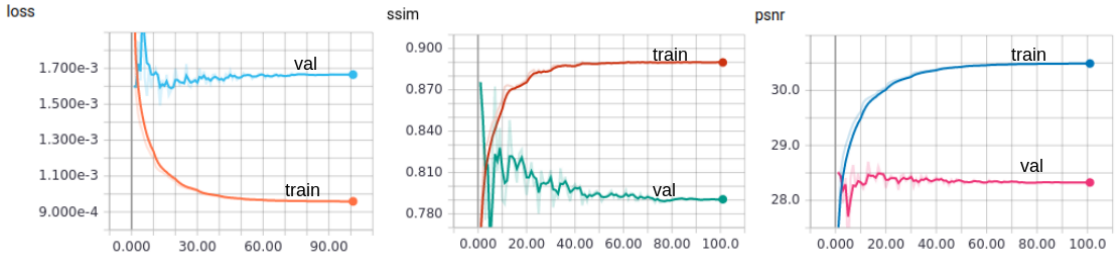


Figure 21: The sub-figures from left to right are MSE, SSIM, and PSNR.

Finally, figure 22 shows a visual comparison between the proposed method and the bicubic interpolation.

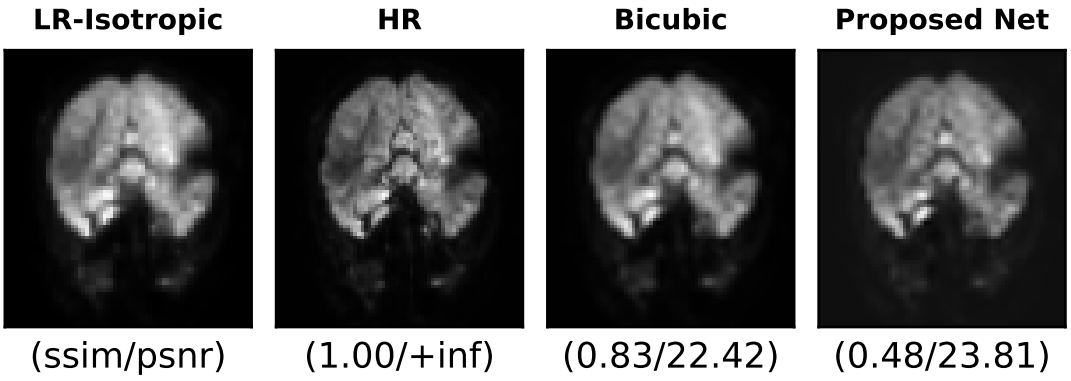


Figure 22: Visualization of the results of the proposed method on a slice from the testing dataset.

As figure 22 depicts my proposed method outperforms the bicubic interpolation in terms of PSNR but it scored a much lower value for SSIM.

5.1.3 Experiment: 2

In this experiment, the first convolution block after the wavelet transform block, see figure 18, was removed.

Figure 23 shows the training/validation graphs of this experiment.

Table 3 shows the values of the evaluation metrics on the testing dataset.

As figure 24 depicts the values of the PSNR and SSIM are better than the ones of the experiment 2.

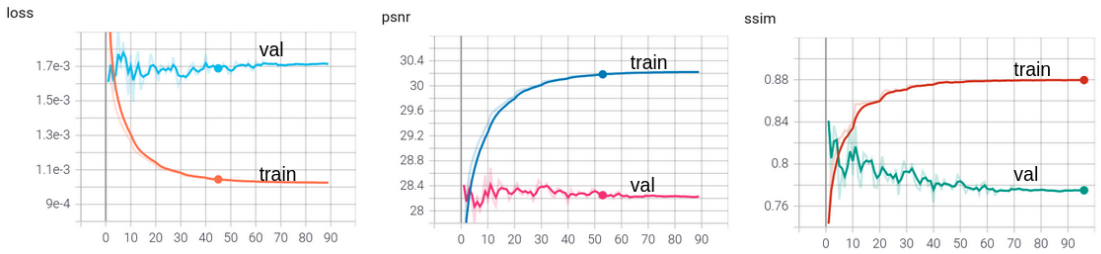


Figure 23: The training/validation graphs.The sub-figures from left to right are MSE, SSIM, and PSNR.

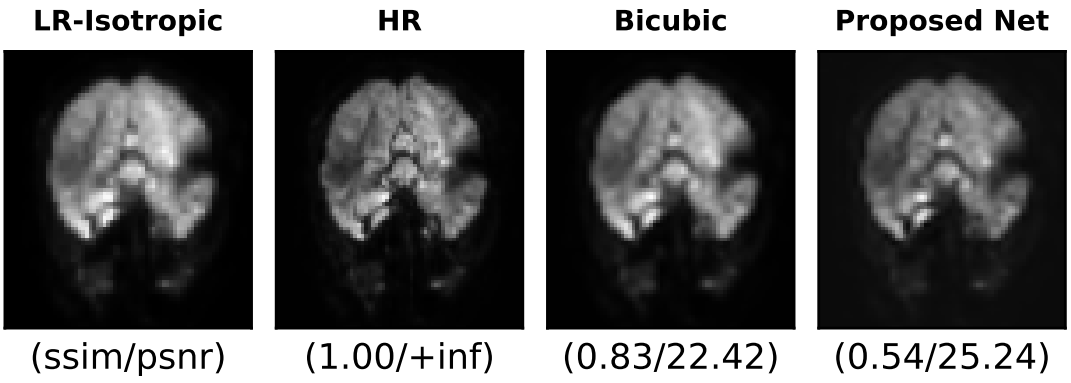


Figure 24: Visualization of the results of experiment 3 on a slice from the testing dataset.

MSE	SSIM	PSNR
0.0017	0.7751	28.2299

Table 3: The evaluation metrics on the testing dataset.

5.2 Discussion

From the tables (1, 2, and 3) it is clear that the baseline performes better on the testing dataset than our proposed method.

However, figure25 shows that replacing one of the convolutional blocks with DWT (Experiment 2) improves the PSNR compared to the baseline, see figure 25.

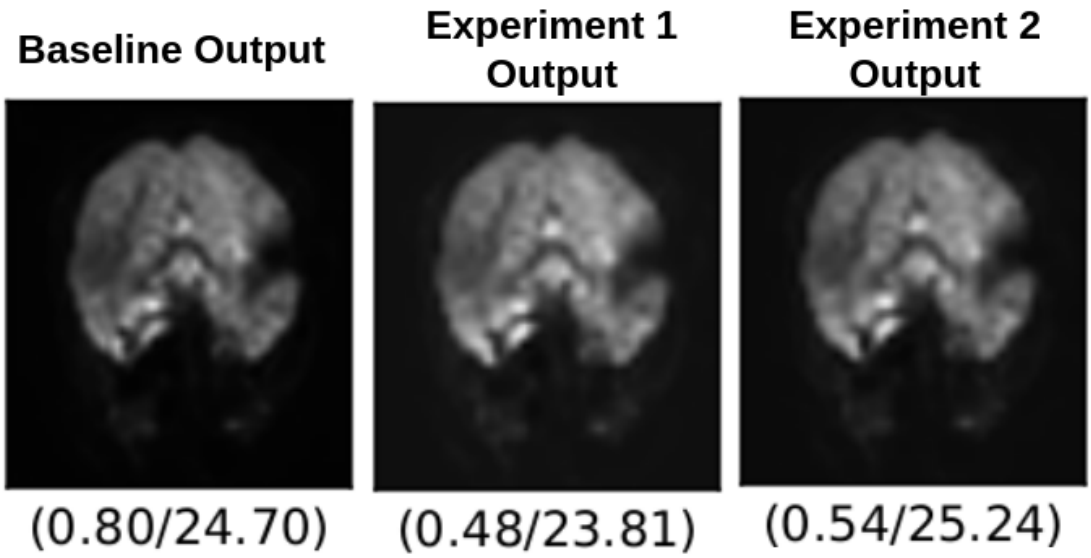


Figure 25: A visual comparison between the outputs of baseline, the experiment1 and the experiment 2 on a slice from the testing dataset.

6 Deep Wavelet Super-resolution Network

The wavelet-based deep neural networks proved their efficiency in the domain of single image super-resolution and image denoising.

Guo et al [9] proposed deep wavelet super-resolution network. This network predicts the missing details of the wavelet coefficients of the low-resolution image to obtain the coefficients of the SR image which serve as input to IDWT to generate the SR image.

The output of the DWT procedure consists of four sub-bands: average (LL), vertical (HL), horizontal (LH), and diagonal (HH).

$2dDWT \longrightarrow LA, LV, LH, LD := 2dDWTLR$

$\triangle SB = HRSB - LRSB = \{HA - LA, HV - LV, HH - LH, HD - LD\} = \{\triangle A, \triangle V, \triangle H, \triangle D\}$

And the cost is given by the following equation.

$$cost = \frac{1}{2}||\triangle SB - f(LRSB)||_2^2 \tag{24}$$

Essentially, the network learns the differences (residuals) between wavelet sub-bands of LR and HR images, see figure26.

$SRSB = \{SA, SV, SH, SD\} = LRSB + \triangle SB$

$SR = 2dIDWT\{SRSB\}$

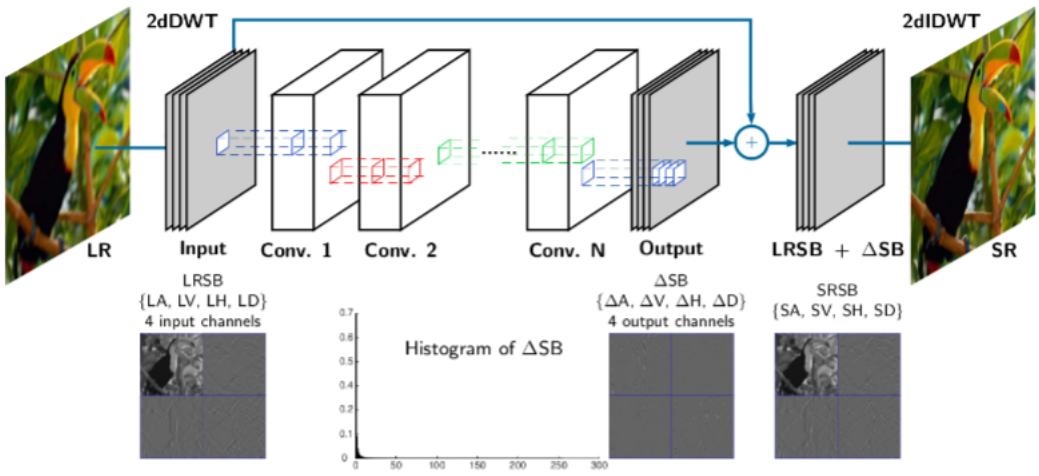


Figure 26: The architecture of deep wavelet super-resolution network [9]

6.1 Training and Testing

This network consists of 4785036 parameters, I trained it on 1 GPU until for 100 epochs. The network trained on 1344 slices belong to subject 1, validated on 1344 slices belong to subject 3, and tested on 1344 slices of subject2.

It is clear from 27 that the network is not able to learn anything as the training loss is constant throughout the training. See figure 28

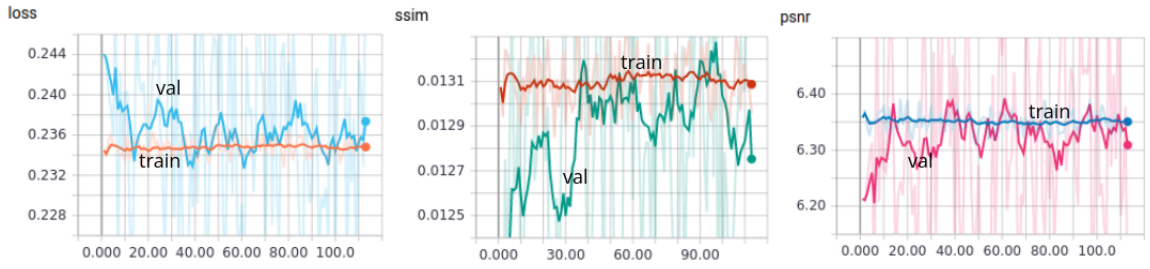


Figure 27: The training/validation graphs of the baseline network. The sub-figures from left to right are MSE, SSIM, and PSNR.[9]

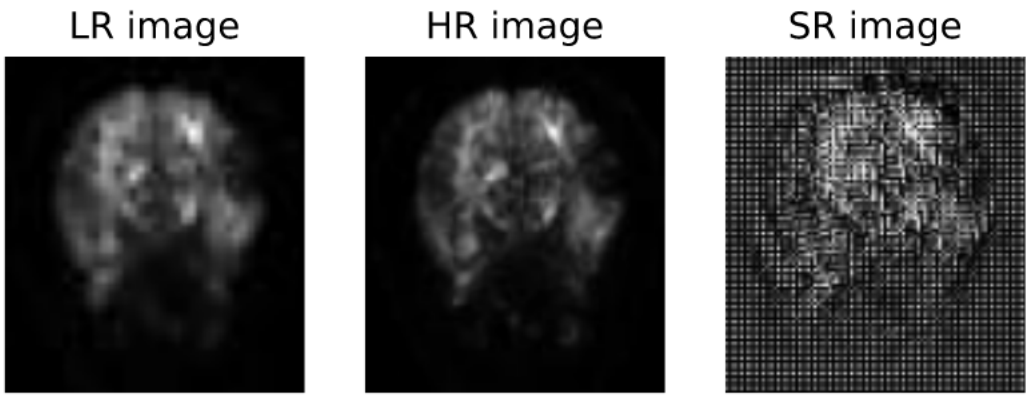


Figure 28: Visualization of the results of the deep wavelet super-resolution network on a slice from the testing dataset.[9]

7 Multi-wavelet Residual Dense Convolutional Neural Network

Wang et al [26] proposed a multi-wavelet residual dense convolutional neural network (MWRDCNN) as in figure 29. In the proposed method, they introduced residual dense blocks (RDBs) to the U-net architecture. Their results revealed that introducing RDBs to the (MWCNN) increased the learning efficiency and improved the results especially in removing the Gaussian noise.

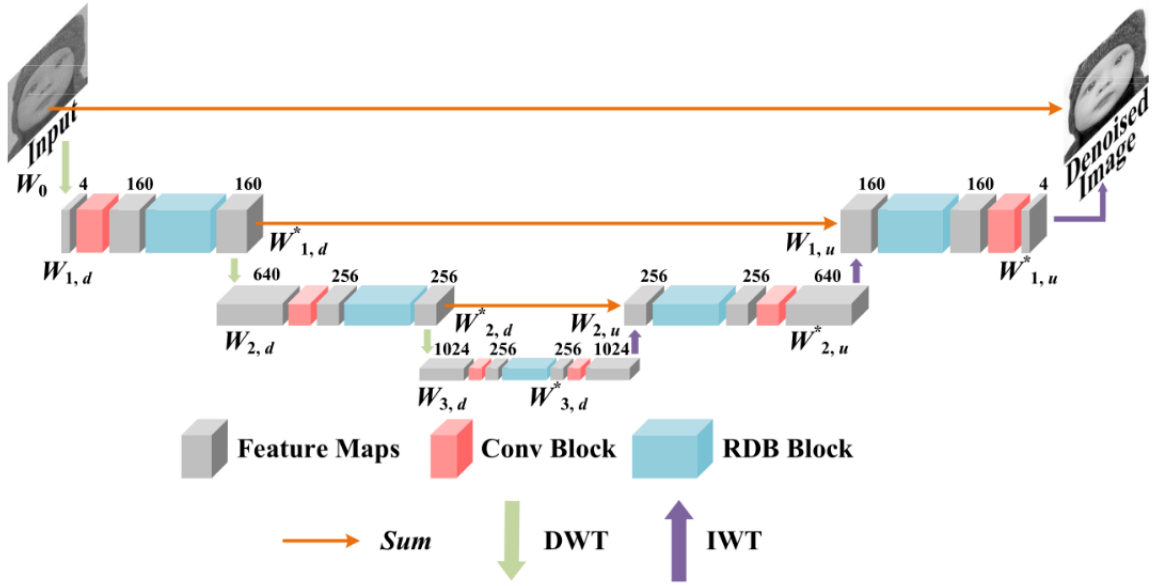


Figure 29: The architecture of multi-wavelet residual dense convolutional neural network (MWRDCNN).[26]

I could not implement this architecture as it has too many parameters and it requires many GPUs.

My network is a standard U-Net with a dilated convolution layers instead of max-pooling layers. The input of the network is the output of the DWT of the LR image, and the output is the super-resolved image. See figure 30.

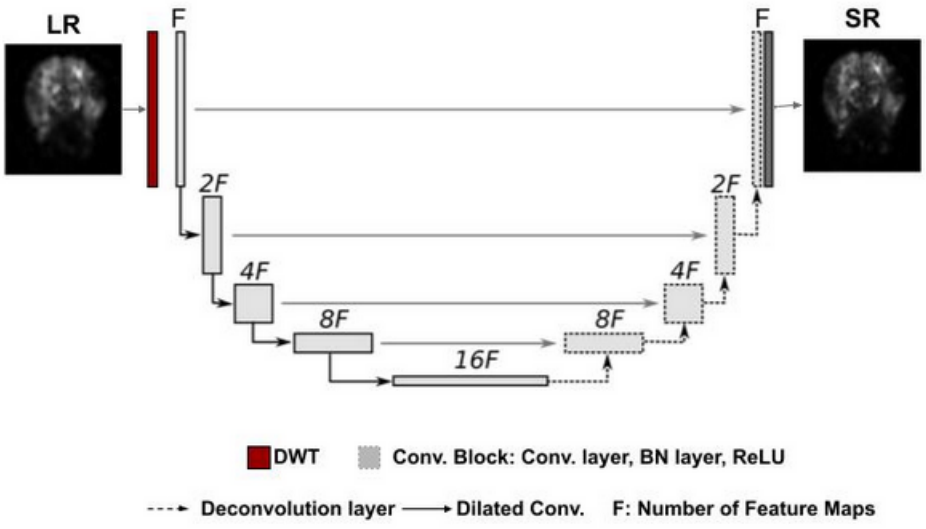


Figure 30: The proposed U-Net architecture for image super-resolution.

7.1 Training and Testing

This network consists of 19158234 parameters, I trained it on 1 GPU until for 300 epochs. The network trained on 1344 slices belong to subject 1, validated on 1344 slices belong to subject 3, and tested on 1344 slices of subject2.

Figure 31 shows the training and validation graphs.

Table 4 shows the values of the evaluation metrics on the testing dataset.

Figure 32 shows a comparison between the results of the proposed method and the bicubic interpolation. The PSNR of the proposed method is higher than the PSNR of bicubic interpolation.

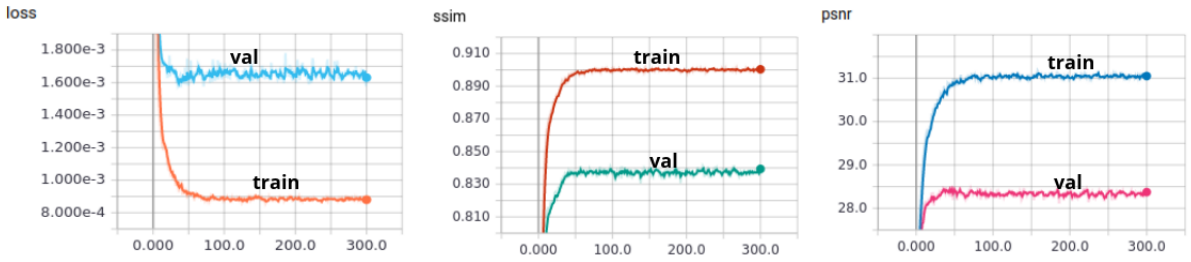


Figure 31: The training/validation graphs.

MSE	SSIM	PSNR
0.0016	0.8385	28.39

Table 4: The evaluation metrics on the testing dataset.

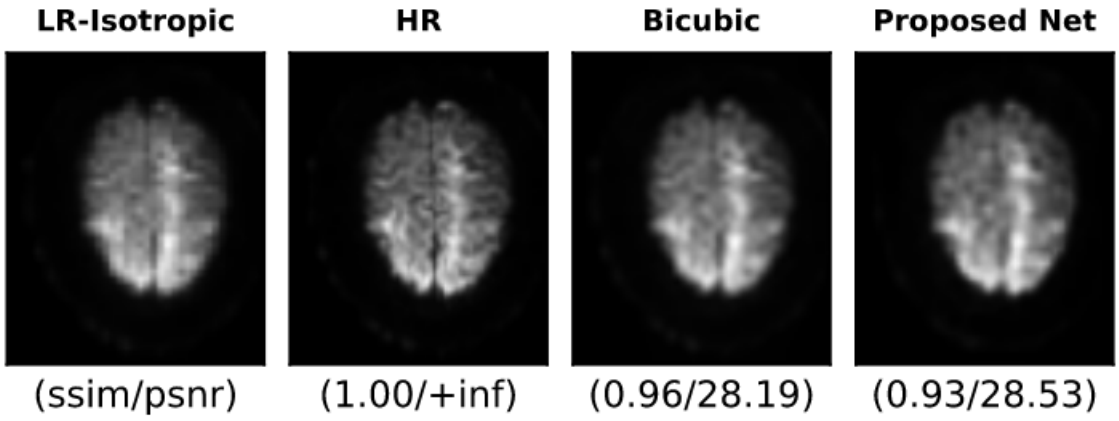


Figure 32: A comparison between the performance of the proposed method and bicubic interpolation.

References

[1] Woong Bae, Jaejun Yoo, and Jong Chul Ye. "Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 145–153.

[2] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. "Super-resolution through neighbor embedding". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. IEEE. 2004, pp. I–I.

[3] Shengyang Dai et al. "Softcuts: a soft edge smoothness prior for color image super-resolution". In: *IEEE transactions on image processing* 18.5 (2009), pp. 969–981.

[4] Chao Dong, Chen Change Loy, and Xiaoou Tang. "Accelerating the super-resolution convolutional neural network". In: *European conference on computer vision*. Springer. 2016, pp. 391–407.

[5] Chao Dong et al. "Image super-resolution using deep convolutional networks". In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[6] Jinglong Du et al. "Brain MRI super-resolution using 3D dilated convolutional encoder–decoder network". In: *IEEE Access* 8 (2020), pp. 18938–18950.

[7] Claude E Duchon. "Lanczos filtering in one and two dimensions". In: *Journal of applied meteorology* 18.8 (1979), pp. 1016–1022.

[8] William T Freeman, Thouis R Jones, and Egon C Pasztor. "Example-based super-resolution". In: *IEEE Computer graphics and Applications* 22.2 (2002), pp. 56–65.

[9] Tiantong Guo et al. "Deep wavelet prediction for image super-resolution". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 104–113.

[10] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[11] Robert Keys. "Cubic convolution interpolation for digital image processing". In: *IEEE transactions on acoustics, speech, and signal processing* 29.6 (1981), pp. 1153–1160.

[12] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1646–1654.

[13] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Deeply-recursive convolutional network for image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1637–1645.

[14] Wei-Sheng Lai et al. "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[15] Christian Ledig et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[16] Bee Lim et al. "Enhanced deep residual networks for single image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[17] Pengju Liu et al. "Multi-level wavelet convolutional neural networks". In: *IEEE Access* 7 (2019), pp. 74973–74985.

[18] Wei Liu, Qiong Yan, and Yuzhi Zhao. "Densely self-guided wavelet network for image denoising". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 432–433.

[19] Antonio Marquina and Stanley J Osher. "Image super-resolution by TV-regularization and Bregman iteration". In: *Journal of Scientific Computing* 37.3 (2008), pp. 367–382.

[20] Marco Pizzolato et al. *Super-resolution of Multi Dimensional Diffusion MRI data*. Mar. 2020. DOI: 10.5281/zenodo.3718990. URL: https://doi.org/10.5281/zenodo.3718990.

[21] Wenzhe Shi et al. "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[22] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[23] Ying Tai, Jian Yang, and Xiaoming Liu. "Image super-resolution via deep recursive residual network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3147–3155.

[24] Ying Tai et al. "Memnet: A persistent memory network for image restoration". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4539–4547.

[25] Tong Tong et al. "Image super-resolution using dense skip connections". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4799–4807.

[26] Shuo-Fei Wang, Wen-Kai Yu, and Ya-Xin Li. "Multi-wavelet residual dense convolutional neural network for image denoising". In: *IEEE Access* 8 (2020), pp. 214413–214424.

[27] Xintao Wang et al. "Esrgan: Enhanced super-resolution generative adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 0–0.

[28] Wenming Yang et al. "Deep learning for single image super-resolution: A brief review". In: *IEEE Transactions on Multimedia* 21.12 (2019), pp. 3106–3121.