

Image Super-Resolution for MRI Images using 3D Faster Super-Resolution Convolutional Neural Network architecture

Vanita Mane^{1,*}, Suchit Jadhav^{2,**}, and Praneya Lal^{3,***}

¹Ramrao Adik Institute of Technology

²Department of Computer Engineering

³Mumbai, India

Abstract. Single image super-resolution using deep learning techniques has shown very high reconstruction performance over the last few years. We propose a novel three-dimensional convolutional neural network called 3D FSRCNN based on FSRCNN, which reinstates the high-resolution quality of structural MRI. The 3D neural network generates output brain images of high-resolution (HR) from a low-resolution (LR) input image. A simple design ensures less time complexity and high reconstruction quality. The network is trained using T1-weighted structural MRI images from the human connectome project dataset which is a large publicly available brain MRI database.

Index Terms—MRI, Super Resolution, SRCNN, FSRCNN

1 Introduction

Acquisition of high-resolution MRI images using effective techniques faces many obstacles. MRI or CT imaging becomes expensive as the quality of imaging increases with that the time required for the imaging also increases. There is, therefore, a need to recreate high-resolution image output from the input with low-resolution (LR) to effectively solve these problems. Super-resolution (SR) is the process of high-resolution (HR) image estimation from one or more low-resolution (LR) images. But because of its underdetermined nature SR is a challenging issue. After resolution degradation a countless number of HR images will produce the same SLR data. This drives us to solve this challenge using our thinking. The 3D scans give minute specifics as opposed to 2D models, presenting a memory consumption challenge and massive expense, making them less realistic. This motivated us to offer a solution to the difficulty of super-resolution for MRI images that are created in a 3D structure. One of the conventional methods for SR is random forest [1]. It generates many trees and merges their outputs, unlike a decision tree. To implement it, the algorithm requires the much-needed varied computational power and resources for a positive outcome. Also, compared to other methods, this method does not know the mapping between images of low-resolution and high-resolution. Super Resolution Convolutional Neural Network (SRCNN) [2] shows state-of-the-art performance and reduces many steps required in conventional methods. In [3], CNN layers represent steps in SR and has outstanding performance. In [4] authors have

explained about advanced deep models, high precision has been achieved for SISR, applying these models to real-world scenarios is still challenging, primarily due to massive parameters and computation. In this work, we discuss how CNNs can be used for reconstructing 3D brain MRI images with the perfect trade-off between complexity and performance. We investigate various approaches related to CNNs and finally propose a neural network called 3D Faster Super-Resolution Convolutional Neural Networks based (3D-FSRCNN) on Faster Super-Resolution Convolutional Neural Networks architecture (FSRCNN) [5].

2 Super Resolution Neural Network

2.1 Mathematical background

Super-resolution, specifically focusses at the reconstruction of HR image Y from one LR image X. It can be formulated as:

$$Y = F(x) \quad (1)$$

Where F is a transformation that reconstructs the high resolution image Y from low resolution input image X. Dong et al. [3] shows that this transformation can be achieved using Convolutional Neural Network (CNN) that minimizes difference between reconstructed and ground truth images.

2.2 Previous deep learning techniques

2.2.1 SRCNN AND ITS LIMITATIONS

One simple but very effective Convolutional Neural Network (SRCNN) [3] was proposed for 2D natural images that yield high performance. One approach extends the

*e-mail: vanitamane1@gmail.com

**e-mail: suchit966@gmail.com

***e-mail: praneya0028@gmail.com

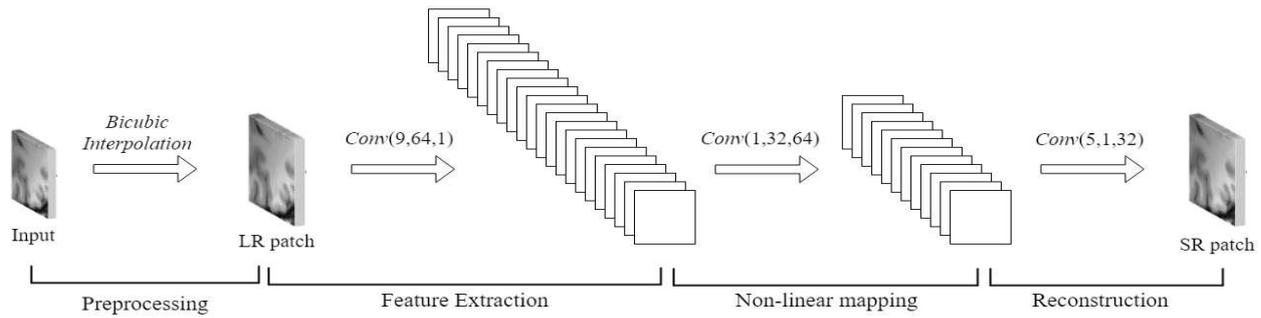


Figure 1. 3D SRCNN Architecture.

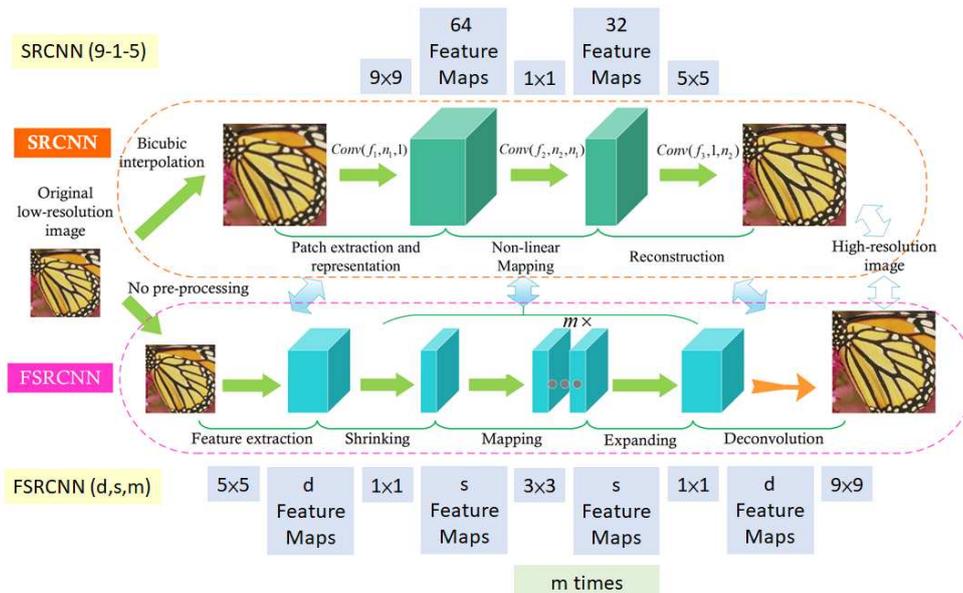


Figure 2. Comparison between SRCNN and FSRCNN [3]

idea of SRCNN and applies it to 3D brain MRI images [2] and another approach extends SRCNN to CT imaging [7], it demonstrates that 3D SRCNN successfully keeps the structure and contours in 3D MRI and CT scans. The limitation of the 3D SRCNN is that it requires more time to predict the SR scan. After analyzing the architecture of the SRCNN, it is found that there are some steps in the model which are costly. The preprocessing step while training delays the training process as it allows the input to be interpolated to the desired output size using methods such as bicubic interpolation, which is costly when we consider three-dimensional images and when testing the interpolated LR images adds further to the complexity. Also, as studied by [5] the non-linear mapping step in SRCNN is the most expensive and causes an increase in time complexity. Complexity of 3D SRCNN:

$$O(f_1^3 n_1 + n_1 f_2^3 n_2 + n_2 f_3^3) S_{HR} \quad (2)$$

Where f_i and n_i are size of the filter and filter number of the three layers in SRCNN respectively. S_{HR} is the size of the

output/HR image. In [2] $f_1 = 9, f_2 = 1, f_3 = 5$ and $n_1 = 64, n_2 = 32$. As seen in figure 1, the complexity is dependent on the size of the HR image, and the non-linear mapping layer adds the most to the constraints of the network [5].

2.2.2 FSRCNN FOR 3D SUPER RESOLUTION

To solve these complex issues with SRCNN, [5] proposed a method to accelerate the SRCNN model called FSRCNN. In this, the proposed 2D FSRCNN model showed a major upgrade from the previous SRCNN model. The FSRCNN model does not have the costly interpolation step which makes it faster while training as well as while testing. One of the features of this 3D FSRCNN model is that we can have the desired output size without interpolating the input image. Also, the non-linear mapping method in SRCNN which was found as most expensive was replaced with three steps - shrinking, mapping and expanding, reducing complexity.

In the next section, we discuss how the FSRCNN can be extended in a three-dimensional setting to produce HR

brain MRI images, its architecture, and parameters related to its architecture.

2.3 Network Architecture

The 3D FSRCNN model proposed in [5] can be broken down into five parts- feature extraction, shrinking, mapping, expanding, and deconvolution as shown in Figure 2. The first four are layers of convolution while the fifth is a layer of deconvolution. We refer to a convolution layer as Conv(f_i, n_i, c_i), and a deconvolution layer as DeConv(f_i, n_i, c_i), where the f_i, n_i, c_i variables represent the filter size, the number of filters and the number of channels, respectively. Since the entire network contains tens of variables, we can not analyze each one of them. Therefore, the insensitive variables are given a reasonable value in advance, leaving the sensitive variables to be unset. When a small change in the variable can have a significant effect on performance[4], we consider a variable sensitive. Such sensitive variables also reflect some important influencing factors in SR, as explained below.

Feature Extraction: This step is similar to feature extraction in SRCNN, the difference being, in FSRCNN it is done on the LR image directly rather than on interpolated image. In SRCNN, filter size (f_1) was 9. As these filters are executed in SRCNN on an upscaled image, using a filter size of 5 will recover similar information as a 9 9 9 filter would in SRCNN. Thus, at this stage $f_1 = 5, c_1 = 1$ and now we need to figure out no. of filters n_i . Let the n_i at this layer be denoted by d as in [z]. The filter number is a sensitive variable, finally, we can represent the first layer as Conv(5, $d, 1$).

Shrinking: Nonlinear mapping in SRCNN follows the extraction of the feature but FSRCNN has a shrinking layer to reduce dimensionality of d . The LR features d is very large and for 3D MRI images it will be larger. Authors in [4] have used 1 1 1 filter to reduce computational cost. Thus, we will follow it and use 1 1 1 filter size i.e., $f_2 = 1$ to reduce the dimension of d . The filter number $n_2 = s$ should be less than d to reduce the LR feature dimension. Thus, the second layer can be represented as Conv(1, s, d).

Non-linear mapping: This step is the most important as it maps the LR image features to the HR images. It has the most share in the super-resolution performance. An comprehensive analysis of deep networks was conducted in [5] authors and found that retaining filter size as 3 is a great trade-off between good performance and network scale. We obey their analysis, and we maintain $f_3 = 3$ and $n_3 = s$. Another sensible variable is the number of layers of mapping denoted by m . So, layers of mapping can be stated as m Conv (3, s, s).

Expanding: This layer undoes the shrinking layer which reduces the LR feature dimension for computational efficiency. Generating images without expanding would generate images with poor quality. To maintain

consistency [4] have used filter size $f_4 = 1$, same as in the shrinking layer. The shrinking layer was Conv(1, d, s) so Conv(1, s, d) would be the expanding layer.

Deconvolution: This layer is responsible for upscaling and aggregating the previous features using deconvolution filters. Deconvolution may be seen as an inverse convolutionary process. As in a convolution operation stride, k outputs the $1/k$ number of times of the input and as the deconvolution is the inverse operation, stride k upscales the input by a factor of k . We can take advantage of this and set the phase $k = n$ as the scaling factor, resulting in a reconstructed, upscaled HR image. By studying [2], authors of [5] have assumed that the filter size f_5 should be 9 which is backed by experiments. So, the last layer can be represented as DeConv(9, 1, d) and stride of n which is set to the scaling required.

Activation: We've used the Rectified Linear Unit (ReLU) for the activation task after every convolution layer. Mathematically, It is set to $R(z) = \max(0, z)$. ReLU is the activation function most normally used in neural networks, especially in CNNs. The calculation is simple, as there are no complex calculations. Hence, the 3D FSRCNN model can take less time to train or test. This converges more rapidly. Linearity means that when x gets big, the slope doesn't "saturate," or plateau. It does not have the problem of the vanishing gradient than other activation functions such as sigmoid or tanh suffer. Figure 3 visualizes the ReLU activation function

Loss and Optimizer: We are following SRCNN[1] and using loss of MSE (mean square error) as cost function. The goal is to minimize the MSE between the reconstructed image \hat{Y}^i and the ground truth image Y_i .

Adam optimizer can be utilized to minimize the MSE loss.

$$MSE = \frac{1}{n} \sum_0^n (\hat{Y}_i - Y_i)^2 \quad (3)$$

Overall Architecture of the 3D FSRCNN: All the above parts are connected to form the FSRCNN as Conv(5, $d, 1$)-ReLU-Conv(1, s, d)-ReLU- m Conv(3, s, s)- ReLU-Conv(1, d, s)-ReLU-DeConv(9, 1, d). Authors [z] represented the FSRCNN model(d, s, m).

Complexity of 3D FSRCNN: The complexity of 3D FSRCNN model can be calculated as:-

$$O(125d^1 + sd + 27ms^2 + ds729d)S_{LR} = O(27ms^2 + 2sd + 854d)S_{LR} \quad (4)$$

If we compare the complexities of 3D SRCNN and our 3D FSRCNN, the complexity of 3D SRCNN is approximately n times the complexity of 3D FSRCNN, where n is the scaling factor in 3D FSRCNN.

Flow of the algorithm: We input the LR image into the network in the training phase, which is $1/n$ the size of the ground-truth image, where n is the scaling-factor. We opt for $n = 2$ for our experiments. All layers process the image as explained in Figure 4, and generate an SR image as the output. The MSE loss is calculated for every SR image generated and after a batch size, backpropagation is carried out to update the weights of the network. During

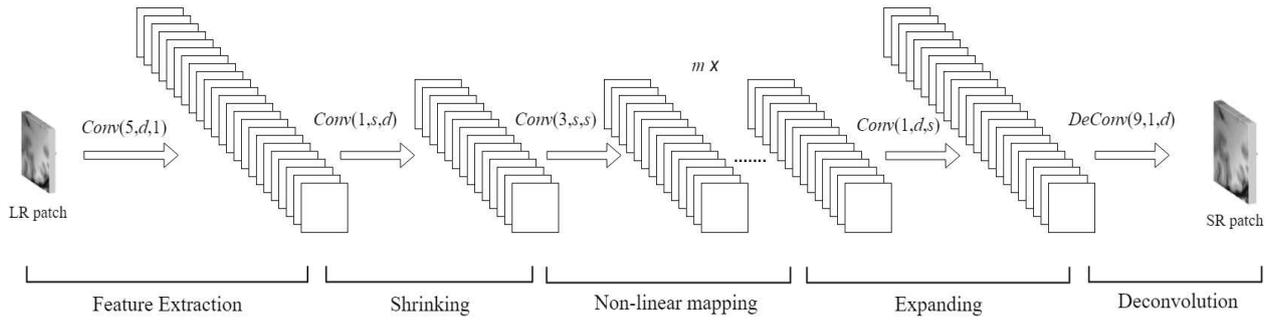


Figure 3. Proposed Architecture of 3D FSRCNN

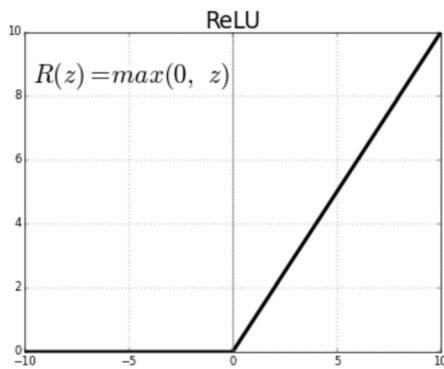


Figure 4. Visualization of ReLU Activation

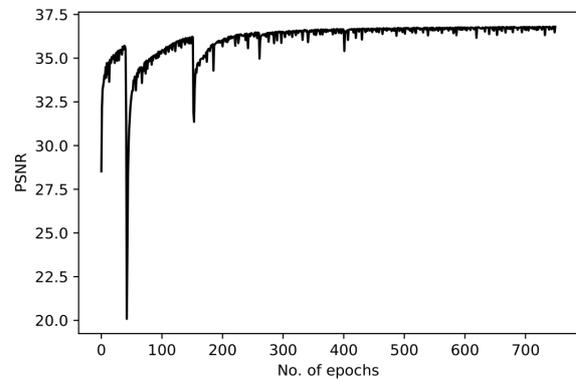


Figure 5. PSNR vs No. of epoch

the test process the test image is divided into 32 cubes and given to the 3D FSRCNN model as input. The SR images produced as output are then averaged and merged in a sliding window manner with the window shifted by half of the cube size. This dividing and merging of patches reduce the prediction time immensely. This was tested on LR images of size 128 X 160 X 160, the difference can be observed in table 1.

Table 1. Difference between prediction time

	Without Patching	With Patching
Maximum Prediction Time	229.46 s	6.86 s
Minimum Prediction Time	228.86 s	6.45 s
Average Prediction Time	228.9 s	6.73 s

2.4 Investigation of parameters

The 3D FSRCNN model, as explained above have different parameters – sensitive variables and hyperparameters. Both parameters needed experimentation and literature knowledge to decide their values. For the sensitive variable m , [6] concludes that $m = 4$ is the optimal value

where the performance of the non-linear mapping layer is the best. Also, they have provided values $d = 48, 56$ and $s = 12, 16$ leaving us with $2 \times 2 = 4$ experiments with different combination. FSRCNN(48, 12, 4) is the smallest network but has the least performance out of all but still better than SRCNN. Table 2 gives the average loss, average PSNR for all the four combinations. We choose $d = 56$ and $s = 12$, even though the network becomes large with these values and affects the training and prediction time, the output image quality is high. Now, for hyperparameters – learning rate, momentum, epochs, batch size, we had to experiment and produce values for the same. For the learning rate, we found that $lr = 0.001$ was optimal in minimizing the loss at a constant rate. In many CNN implementations, momentum is set between 0.5 and 0.9, following it we set the momentum to 0.9. As seen in figure 5, after training for 750 epochs, the PSNR values seemed to become stable from the 500th epoch. Batch size had less effect on the performance, so considering the limitation of RAM, we trained our 3D FSRCNN model with batch size 32.

Table 2. Difference between prediction time

	FSRCNN(48,12,4)	FSRCNN(48,16,4)	FSRCNN(56,12,4)	FSRCNN(56,16,4)
Average Loss	23.39	19.68	19.53	17.38
Average PSPNR	34.93	35.61	35.71	36.09

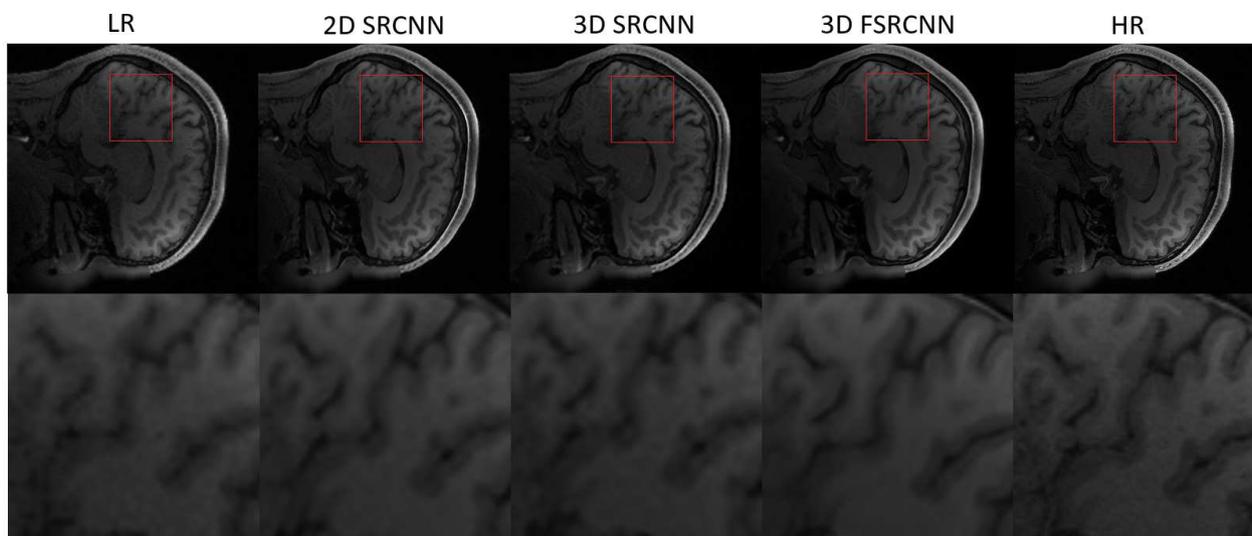


Figure 6. Illustration of 2D SRCNN, 3D SRCNN, 3D FSRCNN reconstructed results and corresponding LR and HR images.

Table 3. Comparison of SR Result

	2D SRCNN			3D SRCNN			3D FSRCNN		
	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
Mean	33.41	0.922	3.84 s	35.75	0.942	9.82 s	37.53	0.967	6.73 s
Min	32.96	0.894	3.63 s	32.87	0.901	7.03 s	33.05	0.756	6.45 s
Max	36.76	0.939	4.06 s	39.88	0.967	12.83 s	48.40	0.975	6.86 s

3 Experiments

3.1 Dataset

To show a better generalization of the 3D FSRCNN model, a publicly available brain MRI dataset, the human connectome project[5], with 1, 113 subjects' T1 weighted anatomical images.. Those 3D images were generated using 32 channel head coil from Siemens 3 Tesla systems. The dimensions of the image are 256x320x320 with a spatial resolution of 0.7 mm isotropic. The images are in high resolution, thus, they were used as ground truth for our training.

3.2 Data Augmentation

The generation of LR MRI images is done by discarding every nth pixel from the image, where n is the scaling factor. For our purpose, the n = 2, so the LR image is downscaled by the factor of 2. HR image of dimension 256x320x320 is used to generate an LR image of dimension 128x160x160. The MRI images in the database are of size 256x320x320, processing an image with such a large size is computationally very expensive. So, we divided the MRI images into small cubes of 64x64x64. Similarly, LR images are divided into patches of 32x32x32.

These patches are used for training the model instead of the whole image.

3.3 Training

We have implemented our 3D FSRCNN models using Keras with TensorFlow as backend and used some other libraries for image processing. All 3D FSRCNN models are trained on Google Colab.

4 Results and Analysis

We tested 2D SRCNN, 3D SRCNN, and 3D FSRCNN on the same data for analysis. The output of the analysis (shown in Figure 6) represent the impact on images applied by these different methods. To better understand the results, we defined it with three metrics – Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM), and time required for reconstructing the output, two in the image domain and one in the time domain (shown in Table 3).The comparison of SR Result found that the output on 3D FSRCNN has the overall best performance impact and it outperforms the 2D SRCNN and 3D SRCNN methods.

5 Conclusion

In this research study, we represented a new deep neural network specifically designed for medical brain MRI images. We studied various super-resolution techniques, such as a 3D mDCSRN-GAN study for MRI, which claims to allow a 4-fold reduction in scanning time while maintaining the same image resolution and quality[8]. Another study re-designed the SRCNN model to explore a more effective network structure so as to increase running speed without impacting the quality[9]. Based on the FSRCNN architecture, we created a 3D model which successfully reconstructs the low-resolution images with very high PSNR value. We believe that more efficiency can be gained by maintaining a proper balance between the deep network and high performance. One of the problems we came across is the increase in time complexity as the network becomes deep. A study on this topic is a potential future work. Still the results showed that our model outperforms many interpolations and deep leaning methods.

References

- [1] S. Schuler, C. Leistner and H. Bischof, “Fast and accurate image upscaling with superresolution forests”, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3791-3799. doi: 10.1109/CVPR.2015.7299003.
- [2] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution”, in Computer Vision–ECCV 2014, pp. 184–199. Springer, 2014.
- [3] C. Dong, C. C. Loy, K. He, and X. Tang, “Image superresolution using deep convolutional networks”, IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 2, pp. 295–307, 2016.
- [4] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue and Q. Liao, “Deep Learning for Single Image Super-Resolution: A Brief Review”, in IEEE Transactions on Multimedia. doi:10.1109/TMM.2019.2919431.
- [5] Dong, Chao, C.C. Loy, and Xiaou Tang. “Accelerating the super-resolution convolutional neural network”. European Conference on Computer Vision. 2016.
- [6] David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E.J. Behrens, Essa Yacoub, Kamil Ugurbil, for the WUMinn HCP Consortium. (2013). The WU-Minn Human Connectome Project: An overview. *NeuroImage* 80(2013):62-79.
- [7] CT-image Super Resolution Using 3D Convolutional Neural Network by Yukai Wang, Qizhi Teng, Xiaohai He, Junxi Feng, Tingrong Zhang (Jun 2018).
- [8] Yuhua Chen, Feng Shi, Anthony G. Christodoulou, Zhengwei Zhou, Yibin Xie, Debiao Li, “Efficient and Accurate MRI Super-Resolution using a Generative Adversarial Network and 3D Multi-Level Densely Connected Network”, (Jun 2018).
- [9] Chao Dong, Chen Change Loy, Xiaou Tang, “Accelerating the Super-Resolution Convolutional Neural Network”, (Aug 2016).