# On Training Model Bias of Deep Learning based Super-resolution Frameworks for Magnetic Resonance Imaging

1st Mamata Shrestha
*Department of Computer Science and Computer Engineering*
*University of Arkansas*
Fayetteville, USA
mamatas@uark.edu

2nd Nian Wang
*Department of Radiology and Imaging Sciences*
*Indiana University*
Indiana, USA
nianwang@iu.edu

3rd Ukash Nakarmi
*Department of Computer Science and Computer Engineering*
*University of Arkansas*
Fayetteville, USA
unakarmi@uark.edu

***Abstract*—Super-resolution is an important technique in various fields, particularly in medical imaging, where it plays a crucial role in understanding and analysis of complex, qualitative, and quantitative characteristics of tissues at high resolutions. However, obtaining high-resolution images often faces practical limitations pertaining to acquisition device limitations, patient motion, or longer acquisition times. The remarkable success of deep learning methods has recently opened doors to their application in image super-resolution tasks as well. These deep learning-based methods heavily rely on a substantial amount of data, which is often unavailable, especially in the case of Magnetic Resonance Imaging (MRI) scans. Particularly in magnetic Resonance super-resolution, it is often impossible to have low-resolution and high-resolution training image pairs. To address this, deep learning approaches simulate low-resolution images using many image degradation methods mimicking low-resolution images to create training image pairs from the available few high-resolution images. However, models trained on specific degradation simulations exhibit bias, leading to poor performance in real-world scenarios. In this paper, we hypothesize that such deep learning models trained on specific training image pairs with a specific degradation model are biased, we systematically study such biases with different types of degradation, different deep learning frameworks, and training losses. Finally, we advocate ensuring the diversity of degradation models to generate training image pairs controls such biases resulting in a more robust learning framework for MR image super-resolution.**

***Index Terms*—MRI, Super-resolution, Model Bias, Diverse Degradation model**

## I. Introduction

Super-Resolution (SR) is the process of reconstructing a High-resolution (HR) image from a Low-resolution (LR) input as shown in Figure 1. In MRI applications, high-resolution images are desirable for accurate diagnosis. However, the acquisition of high-resolution images requires sophisticated devices or a longer acquisition time. The resulting images may be sensitive to the patient's motion leading to a low-quality image. Super-resolution plays an important role in MRI [5] [6] [13] by enabling the recovery of high-resolution images without incurring an additional cost in terms of time and facilitating the patient's comfort.

The super-resolution problem can be formulated as:

$$y = \phi(x; \theta_d), \tag{1}$$

where $\phi$ is the degradation model with parameter $\theta_d$ that a desired high-resolution image $x$ undergoes and $y$ is the low-resolution image in Fig. 1. The objective of the super-resolution technique is to find an estimate $\hat{x}$ of the high-resolution image $x$ such that:

$$\hat{x} = \phi^{-1}(y; \theta_r) \tag{2}$$

where $\phi^{-1}(\cdot)$ is the inverse mapping function that maps low-resolution images to high-resolution images.

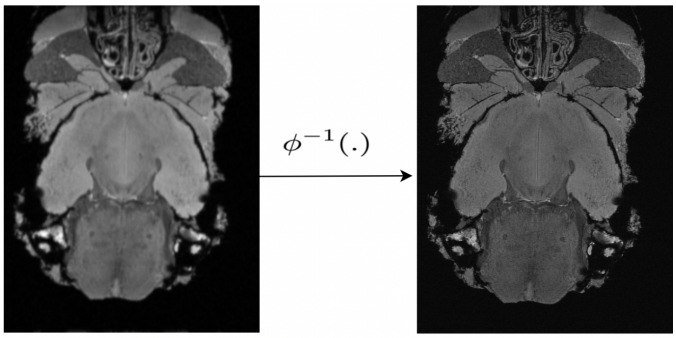


Fig. 1: Single image super-resolution

Recent advancements in Deep Learning (DL) methods have showcased their potential for super-resolution techniques. However, these methods require a large amount of data for effective training and learning purposes. In MRI super-resolution, often, the low-resolution and high-resolution image pair $(y_i, x_i)$ do not exist because i) it is not possible to acquire exact low-resolution and high-resolution image training pairs because of inherent challenges such as the motion of subject between two acquisitions and prohibitively long acquisition times and ii) the degradation model $\phi(\cdot)$ in MRI is non-deterministic. In most state-of-the-art DL frameworks for image super-resolution [10] [12], training examples are generated by simulating a specific degradation method such as Bicubic Downsample, Gaussian Blurring, or Median Filtering. Testing the model on LR images that are mimicked using the same degradation model as in the training set can lead to optimal results. In real-world examples, low-resolution images may have been acquired through a more sophisticated process such as sub-sampling, band-limiting, or accelerated acquisition [1], which are not necessarily accurately represented by one of the downsampling or filtering techniques used for generating training LR image from HR images.

In this work, we argue that it is crucial to have a better representation of various types of degradation in the data to build robust deep learning-based super-resolution frameworks and avoid misleading performance and efficiency metrics.
We hypothesize that:

1) DL-based super-resolution frameworks are biased toward degradation model $\phi(\cdot)$ used to simulate low-resolution images for training DL framework. In other words, $\hat{\phi^{-1}}$ learned by a DL framework is biased towards $\phi(\cdot)$ used to create training image pairs.
2) Enforcing the diversity of the degradation models in the training data set improves generalization and reduces the bias of the DL super-resolution framework.

To examine our first hypothesis, in our first set of experiments, we trained an SRDenseNet architecture using six separate training data sets representing a different type of degradation. Our results demonstrate that the model is only able to produce HR images with improved resolution when the test set images have the same type of degradation as the training set. To the best of our knowledge, this study is the first to investigate model bias in the field of deep learning-based super-resolution approaches.
We also conducted additional experiments with different DL architectures and objective functions to investigate how this bias is affected. To support our second hypothesis, we trained a DL architecture using a combined data set consisting of equal numbers of samples from multiple degradations while maintaining the same numbers of training image pairs (training data size) same as in the first set of experiments.

## II. RELATED WORK

### A. Convolutional Neural Network (CNN) based methods

Deep learning-based CNN architecture was first introduced in [3] for super-resolution tasks, named SRCNN. Since then, various work has tried to improve the result by exploring the network architecture design and learning method [7][2] [9][8]. For instance, VDSR [7] used higher learning rates to learn only the residual between low-resolution and high-resolution images and employed adjustable gradient clipping to mitigate the gradient explosion problem. FSRCNN [2] proposed a compact hourglass-shaped CNN structure to reduce model complexity and improve efficiency for super-resolution. LapSRN [9] utilized a deep Laplacian pyramid structure to progressively reconstruct the sub-band residuals of high-resolution images at multiple pyramid levels to reduce computational cost. DRCN [8] introduced recursive networks with skip connections to mitigate the vanishing gradient problem. In DRCN, the output from each internal layer is skip-connected to the final convolutional layer, and the final high-resolution output is obtained by averaging all outputs from internal layers. Subsequent improvements over DCNN optimize the network by removing unnecessary modules in conventional residual networks such as the Batch Normalization (BN) layer to reduce computation and memory usage while exploiting the inter-relation of learned features for each scale using a new multi-scale model that efficiently reconstructs high-resolution images for various scales [11].
The choice of the objective function in deep learning training significantly affects performance. In super-resolution research, different loss functions have been explored to enhance image quality [10] [17]. The author investigates different loss functions (L1 loss, MSE loss, SSIM, and Multi-scale Structural Similarity Index Measure (MS-SSIM)) for image restoration tasks in [17]. Super-resolution Generative Adversarial Netwotk (SRGAN) [10] introduced an adversarial loss to move the output image toward the natural image manifold. Subsequent work based on GAN [15] made several improvements over SRGAN to achieve perceptually convincing results. They introduced a new architecture, Residual-in-Residual Dense Block Network (RRDBNet) with residual scaling and excluding BN layers.

### B. Transformer based methods

Transformer has achieved state-of-art results in Natural Language Processing. Due to its advantage of global attention and long range dependency inclusion, several works have focused on incorporating transformer architecture for computer vision problems. Transformer was first adopted for the image classification task in [4]. In the work [12], the author proposed a novel network that combines CNN with a transformer layer, achieving a significant result in super-resolution tasks. This model utilizes CNN to extract valuable features and the Transformer layer to capture long-range dependencies.

Prior approaches have investigated network architecture and learning techniques for accomplishing super-resolution tasks. However, the influence of the data set quality on the model's performance has been overlooked in these studies. In this work, we study the influence of quality of data on the performance of deep learning models and propose a method to improve it to generate a robust model.

## III. METHODS

### A. Network Architecture

In our experiment, we trained three distinct deep-learning architectures with varying complexity. The first architecture, SRCNN, was introduced in [3], marking the initial application of deep learning architecture in super-resolution tasks. SRCNN comprises three convolution layers with ReLU activation and kernel sizes of 9, 5, and 5, respectively.

The second architecture, SRDenseNet [14], utilizes dense connections, where the feature maps from each layer are skip-connected to every subsequent layer. This promotes better information flow and mitigates the vanishing gradient problem in deeper architectures. SRDenseNet consists of four types of blocks, consisting of initial convolution blocks, dense blocks, a bottleneck layer, and a reconstruction layer. Each of these components involves a single convolution layer except the dense block. The bottleneck layer aggregates the information along the channel dimension before passing it to the reconstruction layer. In our study, the network utilizes five dense blocks, each containing five convolution layers with a growth rate of 5. The activation function for all layers, except the last one, is Leaky ReLU with a negative slope of 0.2. The last layer employs the Tanh activation function.

Finally, we trained RRDBNet, which shares the same architecture as [15], with the exception that we employ only three residual blocks to reduce the model complexity.

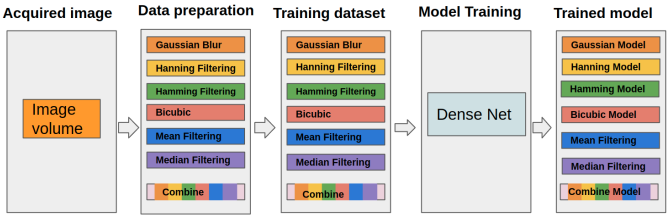


Fig. 2: Training and data preparation pipeline

### B. Loss Function

To further ascertain the biases in DL-based Super-resolution frameworks are from the degradation model not from the network architectures or loss functions, we trained all three deep learning architectures using three different loss functions: Mean Squared Error (MSE) loss, Structural Similarity Index Measure (SSIM) loss, L1 loss. MSE loss is defined as:

$$L_{MSE} = \Sigma_{i=1}^{n}(x_i - \hat{x_i})^2 \tag{3}$$

where $x_i$ is the label image, $\hat{x_i}$ is the output image and n is the number of training examples.

L1 loss is an alternative to MSE loss which measures the sum of absolute differences between pixels in the output and label images. L1 loss is defined as:

$$L_{l1} = \Sigma_{i=1}^{n}|x_i - \hat{x_i}| \tag{4}$$

SSIM [16] is used as a metric to evaluate image quality based on structural information and has been used as a loss function for super-resolution tasks [17]. The SSIM measures the difference between two images in terms of structure, luminance, and contrast. The SSIM metric is calculated as :

$$SSIM(x, \hat{x}) = \frac{2\mu_x\mu_{\hat{x}} + C_1}{\mu_x^2 + \mu_{\hat{x}}^2 + C_1} . \frac{2\sigma_{x\hat{x}} + C_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2} \tag{5}$$

where $\mu_x$ and $\mu_{\hat{x}}$ are pixel sample mean of image $x$ and $\hat{x}$ respectively. $\sigma_x$, $\sigma_{\hat{x}}$ and $\sigma_{x\hat{x}}$ are the variance of $x, \hat{x}$ and covariance of $x$ and $\hat{x}$ respectively. Also, $C_1$ and $C_2$ denote the brightness of two images.

In case of SSIM loss, we want to increase SSIM value, therefore, SSIM loss is defined as:

$$L_{SSIM} = \Sigma_{i=1}^{n}(1 - SSIM(x_i, \hat{x}_i)) \tag{6}$$

## IV. EXPERIMENTS

### A. Dataset

All experiments were carried out in compliance with the local Institutional Animal Care and Use Committee. Five Alzheimer's disease (AD) with 5xFAD background mice brains were acquired on a 30-cm bore 9.4 T magnet (Bruker-BioSpec 94/30, Billerica, MA). A 3D gradient echo (GRE) pulse sequence was performed at the spatial resolution of 25× 25 × 25 $mm^3$, FOV = 18.0 mm × 12.8 mm × 7.6 mm, flip angle = 45°, bandwidth (BW) = 125 kHz, and TR = 100ms. We used these acquired volumes of dimension 720*512*360 from 4 different subjects with an isotropic resolution of 25-micron as high-resolution samples. We took the image slices along the last dimension to obtain high-resolution images of dimension 720*512.

We simulate low-resolution degradation using six different degradation methods: Bicubic Downsample, Gaussian Blurring, Mean, Median, Hanning, and Hamming Filtering to obtain the corresponding low-resolution image from its high-resolution counterparts. Each of these data sets contains 720 training, 167 validation, and 73 test examples. Furthermore, we create a combined data set of size 720 consisting of an equal number of random images from each of those six data sets to create a training data set with diverse degradation. To ensure fairness in our experiments and comparisons, we maintained an equal number of train, test, and validation samples in all degradation data sets. The complete data preparation and training pipeline is shown in Figure 2.

We evaluate the performance of the model using Peak Signal To Noise Ratio (PSNR), SSIM, Normalized Root Mean Squared Error (NRMSE), and High-Frequency Error Norm (HFEN) values. A Higher value for PSNR and SSIM, and a lower value for NRMSE and HFEN indicate a better result. PSNR is widely used metric to determine the quality of image. PSNR is calculated as:

$$PSNR = 10\log_{10}\left(\frac{L}{\frac{1}{M}\Sigma_{i=1}^{M}(x_i - \hat{x_i})^2}\right), \tag{7}$$

where $L$ is maximum pixel value and $M$ is the number of pixel.

### B. Training Details

We trained the SRDenseNet model for 500 epochs, employing MSE loss as the objective function. The learning rate is initially set to 0.0001 and reduced by a factor of 0.05 at every 30 and 50 epochs. The input low-resolution images and label high-resolution images were normalized to the range of [0, 1]. The training is performed with a batch size of 4, utilizing the Adam optimizer. We repeat the experiment for each of the six data sets and a combined data set as outlined in the dataset section.

In subsequent experiments, we maintain the same configuration regarding to settings such as learning rate, number of epochs, and batch size. However, we use different loss functions. Specifically, we used L1 loss and SSIM loss. Furthermore, we separately train the SRCNN and RRDBNet models on the six data sets and the combined data set, using MSE loss as the objective function.

### C. Results

Table I lists the results of SRDenseNet trained on six data sets and the Combine data set as described in the data set section. In all these experiments, all other training settings remained the same, except the degradation model. When the test set images have the same degradation model, SRDenseNet trained on a single degradation achieves the best results. However, for all other test sets with a different degradation model than that of the training, the results were substantially poor. For instance, a Bicubic model trained on a Bicubic Downsampled training set achieves an average PSNR of 33.82 dB for test images that had undergone Bicubic degradation but its performance drops to 28.53dB on the same test set images but had gone through a Gaussian blur degradation to mimic LR images as depicted in the fourth row of the table I. Similar observations could be made for all other rows as well. This support our first hypothesis that the model is biased on the training data set degradation method. In contrast, the combined model trained on the Combine data set achieves comparable PSNR between 30.82 dB to 34.46 dB for all six types of degradation test sets. This supports our second hypothesis and ascertains that incorporating diversity into the training data set helps in reducing model bias and enhancing generalization capability. The representative visual example to depict the model bias on training model degradation is shown in Figure 4. We can see that only the Bicubic model and the combined model have better PSNR and SSIM compared to other models because, for the Bicubic model, the training and test degradation model is the same, suggesting that DL SR framework are biased on the degradation model and hence supporting our first hypothesis. The robust performance of the combined model supports our second hypothesis.

Table II summarizes the results of SRDenseNet trained using L1 and SSIM loss. We can observe similar biases in the case of both losses. This demonstrates that model bias exists regardless of the loss function used for training. Moreover, the combined model consistently performs well across all test sets in both cases, highlighting the effectiveness of the diverse data set which again supports our second hypothesis.

Table III displays the results of the bias experiment with different architectures. The RRDBNet and SRCNN models trained on a combined data set achieve robust performance across all test sets. Conversely, models trained on a single degradation show a larger variance in PSNR values when tested across multiple types of degradation test sets. For instance, RRDBNet trained on the Gaussian Blur dataset achieves an average PSNR of 33.65dB on the Gaussian test set, but its performance drops to 30.09 dB, 30.78dB, 27.07 dB, 23.96 dB, and 20.31 dB when tested on the Hanning, Hamming, Bicubic, Mean, and Median datasets, respectively. All of the above experiments, tables, and results we also computed for all three model architectures and all three losses described in the Methods section. Only a few results are presented here because of the space limit. However, our observations on all experiments consistently support both hypotheses.

TABLE I: Average PSNR Value For SRDenseNet Architecture

| Training Set | Test Set | | | | | |
|---|---|---|---|---|---|---|
| | Gaussian | Hanning | Hamming | Bicubic | Mean | Median |
| Gaussian | **34.33** | 31.07 | 31.56 | 26.68 | 25.81 | 23.17 |
| Hanning | 27.34 | **31.92** | 30.08 | 25.53 | 27.03 | 23.86 |
| Hamming | 32.29 | 31.39 | **32.23** | 28.33 | 32.15 | 28.16 |
| Bicubic | 28.53 | 27.71 | 28.23 | **33.82** | 29.43 | 28.68 |
| Mean | 30.61 | 29.16 | 29.06 | 28.99 | **36.44** | 23.47 |
| Median | 30.22 | 28.18 | 29.13 | 31.48 | 32.69 | **33.54** |
| Combine | 32.47 | 30.82 | 31.42 | 32.82 | 34.46 | 32.88 |

TABLE II: Average PSNR Value For SRDenseNet trained using different loss function

| Loss | Training Set | Test Set | | | | | |
|---|---|---|---|---|---|---|---|
| | | Gaussian | Hanning | Hamming | Bicubic | Mean | Median |
| SSIM loss | Gaussian | **34.52** | 31.11 | 31.59 | 25.76 | 22.22 | 19.88 |
| | Hanning | 25.88 | **31.48** | 28.69 | 24.57 | 25.42 | 22.63 |
| | Hamming | 31.72 | 31.42 | **31.92** | 28.3 | 31.49 | 28.04 |
| | Bicubic | 28.42 | 27.66 | 28.15 | **33.41** | 29.23 | 29.39 |
| | Mean | 30.74 | 30.77 | 31.42 | 28.05 | **36.51** | 24.33 |
| | Median | 31.32 | 30.27 | 30.82 | 30.42 | 32.96 | **33.41** |
| | Combine | 31.79 | 30.64 | 31.25 | 31.38 | 33.68 | 32.24 |
| L1 loss | Gaussian | **34.3** | 30.98 | 31.55 | 26.3 | 24.12 | 20.55 |
| | Hanning | 28.17 | **31.83** | 30.86 | 26.26 | 28.16 | 24.45 |
| | Hamming | 29.17 | 31.79 | 30.99 | 26.77 | 28.75 | 25.60 |
| | Bicubic | 28.65 | 27.76 | 28.29 | **33.73** | 29.5 | 18.75 |
| | Mean | 31.01 | 30.62 | **31.26** | 28.69 | **36.03** | 26.04 |
| | Median | 31.18 | 30.14 | 30.69 | 30.53 | 32.71 | **33.21** |
| | Combine | 31.52 | 30.14 | 30.73 | 32.72 | 34.17 | 32.58 |

## V. CONCLUSIONS

In this paper, we examine the model bias on training data and propose a method to overcome it. Our results on the experiment with multiple types of degradation demonstrate that the model performance is optimistic while tested on the same degradation as the training set. On the other hand, the model trained on data set with diverse samples demonstrates enhanced robustness and achieves comparable performance across various types of degradation. In conclusion, our study highlights the significance of training data quality and suggests that it can be improved by incorporating diverse samples.

TABLE III: Average PSNR Value For different Model Architecture trained using MSE loss

| Model | Training Set | Test Set | | | | | |
|---|---|---|---|---|---|---|---|
| | | Gauss | Hann | Hamm | Bicubic | Mean | Median |
| SRCNN | Gaussian | **34.3** | 31.08 | 31.6 | 27.31 | 27.86 | 25.14 |
| | Hanning | 31.57 | **31.64** | 31.82 | 27.96 | 31.47 | 28.59 |
| | Hamming | 31.55 | 31.63 | **31.84** | 27.89 | 31.46 | 28.37 |
| | Bicubic | 28.711 | 27.86 | 28.37 | **33.64** | 29.54 | 28.73 |
| | Mean | 31.74 | 30.82 | 31.49 | 29.36 | **35.48** | 28.65 |
| | Median | 31.64 | 30.55 | 31.12 | 29.88 | 32.26 | **32.92** |
| | Combine | 32.18 | 31.06 | 31.61 | 29.59 | 33.44 | 32.27 |
| RRDBNet | Gaussian | **33.65** | 30.09 | 30.78 | 27.07 | 23.96 | 20.31 |
| | Hanning | 25.11 | **31.58** | 28.93 | 24.77 | 24.76 | 20.99 |
| | Hamming | 31.72 | 31.1 | **31.84** | 30.19 | 31.92 | 28.26 |
| | Bicubic | 28.58 | 27.76 | 28.28 | **33.8** | 29.44 | 28.83 |
| | Mean | 30.23 | 29.98 | 30.62 | 29.21 | **35.84** | 23.73 |
| | Median | 30.6 | 29.79 | 30.31 | 31.09 | 32.47 | **33.54** |
| | Combine | 32.55 | 30.26 | 31.12 | 33 | 34.92 | 33.09 |

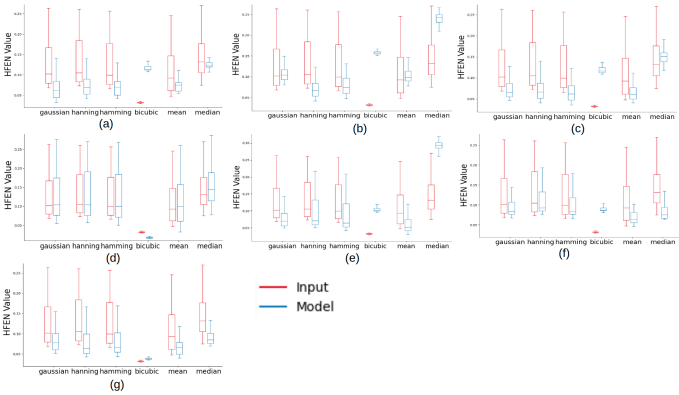


Fig. 3: Box plot for Average HFEN value for scale factor 2 trained on a) Gaussian, b) Hanning, c) Hamming, d) Bicubic, e) Mean, f) Median, and g) Combined datasets. Input label represents the plot for input and label images. Blue box plot lower than red represent better results. From the figure, we can see that model trained on the (g) Combined dataset has robust performance across all six test sets (Gaussian, Hanning, Hamming, Bicubic, Mean, Median and Combine Set) whereas other models performance fluctuates depending on the nature of testing set.

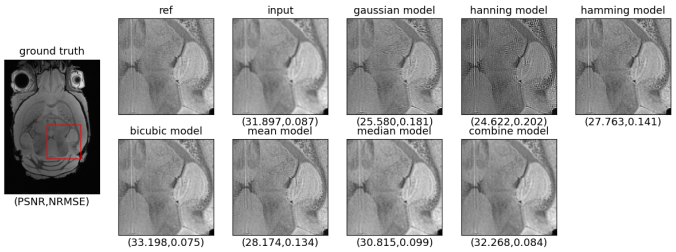


Fig. 4: Example Visual results from bicubic test set for depicting model bias in DL. Input is the test sample image that has gone through Bicubic degradation. Each other image except the ground truth/reference is output from each of the models trained on the training dataset with individual degradation.

## REFERENCES


[1] Kai Tobias Block, Martin Uecker, and Jens Frahm. "Undersampled radial MRI with multiple coils. Iterative image reconstruction using a total variation constraint". In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 57.6 (2007), pp. 1086–1098.

[2] Chao Dong, Chen Change Loy, and Xiaoou Tang. "Accelerating the Super-Resolution Convolutional Neural Network". In: *CoRR* abs/1608.00367 (2016). arXiv: 1608.00367. URL: http://arxiv.org/abs/1608.00367.

[3] Chao Dong et al. "Image Super-Resolution Using Deep Convolutional Networks". In: *CoRR* abs/1501.00092 (2015). arXiv: 1501.00092. URL: http://arxiv.org/abs/1501.00092.

[4] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[5] Hayit Greenspan. "Super-Resolution in Medical Imaging". In: *The Computer Journal* 52.1 (2009), pp. 43–63. DOI: 10.1093/comjnl/bxm075.

[6] Jithin Saji Isaac and Ramesh Kulkarni. "Super resolution techniques for medical image processing". In: *2015 International Conference on Technologies for Sustainable Development (ICTSD)*. 2015, pp. 1–6. DOI: 10.1109/ICTSD.2015.7095900.

[7] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate Image Super-Resolution Using Very Deep Convolutional Networks". In: *CoRR* abs/1511.04587 (2015). arXiv: 1511.04587. URL: http://arxiv.org/abs/1511.04587.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Deeply-Recursive Convolutional Network for Image Super-Resolution". In: *CoRR* abs/1511.04491 (2015). arXiv: 1511.04491. URL: http://arxiv.org/abs/1511.04491.

[9] Wei-Sheng Lai et al. "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution". In: *CoRR* abs/1704.03915 (2017). arXiv: 1704.03915. URL: http://arxiv.org/abs/1704.03915.

[10] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *CoRR* abs/1609.04802 (2016). arXiv: 1609.04802. URL: http://arxiv.org/abs/1609.04802.

[11] Bee Lim et al. "Enhanced Deep Residual Networks for Single Image Super-Resolution". In: *CoRR* abs/1707.02921 (2017). arXiv: 1707.02921. URL: http://arxiv.org/abs/1707.02921.

[12] Zhisheng Lu et al. *Transformer for Single Image Super-Resolution*. 2022. arXiv: 2108.11084 [cs.CV].

[13] Yoshiki Sano et al. "Super-resolution method and its application to medical image processing". In: *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. 2017, pp. 1–2. DOI: 10.1109/GCCE.2017.8229301.

[14] Tong Tong et al. "Image Super-Resolution Using Dense Skip Connections". In: Oct. 2017, pp. 4809–4817. DOI: 10.1109/ICCV.2017.514.

[15] Xintao Wang et al. "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks". In: *CoRR* abs/1809.00219 (2018). arXiv: 1809.00219. URL: http://arxiv.org/abs/1809.00219.

[16] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[17] Hang Zhao et al. "Loss Functions for Image Restoration With Neural Networks". In: *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 47–57. DOI: 10.1109/TCI.2016.2644865.