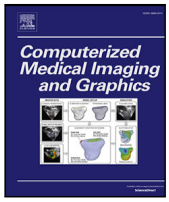
Computerized Medical Imaging and Graphics 108 (2023) 102280

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

journal homepage: www.elsevier.com/locate/compmedimag



Super-resolution of magnetic resonance images using Generative Adversarial Networks

João Guerreiro [a,*], Pedro Tomás [a], Nuno Garcia [b], Helena Aidos [b]

[a] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
[b] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Medical imaging
MRI acceleration
Super-resolution
Generative Adversarial Networks

ABSTRACT

Magnetic Resonance Imaging (MRI) typically comes at the cost of small spatial coverage, high expenses and long scan times. Accelerating MRI acquisition by taking less measurements yields the potential to relax these inherent forfeits. Recent breakthroughs in the field of Machine Learning have shown high-resolution (HR) images could be recovered from low-resolution (LR) signals via super-resolution (SR). In particular, a novel class of neural networks named Generative Adversarial Networks (GAN) has manifested an alternative way of conceiving models capable of generating data. GANs can learn to infer details based on some prior information, subsequently recovering missing data. Accordingly, they manifest huge potential in MRI reconstruction and acceleration tasks. This paper conducts a review on GAN-based SR methods, exhibiting the immersive ability of GANs on upscaling MRIs by a scale factor of ×4 while at the same time maintaining trustworthy and high-frequency details. Despite quantitative results suggesting SRResCycGAN outperforms other popular deep learning methods in recovering ×4 downgraded images, qualitative results show Beby-GAN holds the best perceptual quality and proves GAN-based methods hold the capacity to reduce medical costs, distress patients and even enable new MRI applications where it is currently too slow or expensive.

1. Introduction

Magnetic resonance imaging (MRI) is a state-of-the-art medical imaging technique (Lauterbur, 1973) that is predominantly necessary across patients' diagnoses and medical tracking of ongoing diseases. The detailed information of organs, soft tissues, and bones extracted from an MRI scan allows physicians to effectively evaluate, adjust and control treatments, providing patients a better and more comprehensive care. A relevant problem that arises is the prolonged MRI acquisition time, which subsequently rises costs and leaves many patients on hold. Moreover, patient movement during scans may impact results, potentially requiring retesting. Hence, patients have to lie still in the scanners and even hold their breath for thoracic or abdominal imaging (Funk et al., 2014), since even the slightest movement may compromise the image quality. Therefore, the slow acquisition of MRI scans manifests discomfort among subjects and presents inconvenience in healthcare.

The rationale behind the slow MRI acquisition rate is that it needs to capture detailed information capable of providing proper reasoning for the radiologists. Additionally, the process has to be calibrated to the patient and is based on very strict requirements. During the acquisition, hundreds of slices are recorded from several directions to be pieced together in order to compose a volume. However, the time it takes to acquire these slices can vary greatly depending on the type of imaging being done. For instance, in 3D imaging, if one slice takes around 4 seconds, then to produce one volume of 150 slices the resulting acquisition time is $4 \times 150 = 10$ minutes. On the other hand, in multi-slice 2D acquisitions, the sequence time is determined not so much by the number of slices and desired in-plane spatial resolution, but by the time required for magnetization recovery between consecutive excitations necessary to achieve the desired image contrast. Besides, the duration of the whole process may increase substantially, depending on the pulse sequence type performed (Jackson et al., 1997), the size of the area being scanned and the required number of different weighted scans, which provide different contrasts. Times range from as low as 50 milliseconds to tens of minutes. Consequently, MRI is not used in emergencies as often as would be desired when quick results are needed, such as when there is a serious injury or stroke.

The desired image quality also impacts the acquisition time. The decrease in acquisition time is proportional to the spatial resolution reduction. If an MRI is acquired with half the resolution, then the acquisition time is practically halved (Chen et al., 2020) (excluding scanning preparation and/or pre-scanning time). Furthermore, spatial resolution highly impacts the achieved image signal-to-noise ratio

* Corresponding author.
E-mail address: joao.l.carrilho.guerreiro@tecnico.ulisboa.pt (J. Guerreiro).

https://doi.org/10.1016/j.compmedimag.2023.102280
Received 15 March 2023; Received in revised form 30 June 2023; Accepted 26 July 2023
Available online 31 July 2023

(SNR). In 3D imaging, increasing the resolution by a factor of two (e.g., from 2 mm isotropic to 1 mm isotropic) results in an eight-fold decrease in the SNR. To maintain the same SNR level at the higher spatial resolution, acquiring 64 repetitions would be necessary, leading to a dramatic increase in the acquisition time. Therefore, the ability to infer a high-resolution (HR) image from a low-resolution (LR) image yields a massive impact on the performance of image analysis and MRI acceleration.

Additionally, MRI scans are heavily expensive for medical clinics as a result of equipment, installation, and maintenance costs. Consequently, such costs raise patient expenses. An alternative is low-field MRI scanners (Marques et al., 2019), which are significantly less expensive than their high-field counterparts, thus making MRI technology more accessible to everyone. However, images acquired using low-field MRI scanners tend to be of relatively low resolution, as signal-to-noise ratios are lower. Once again, the ability to improve the spatial resolution of MRIs manifests substantial value.

A convenient concept in Machine Learning was introduced, called Image Super-Resolution (SR), referred to as the task responsible for the reconstruction of an image from low to high resolution. MRI assisted by Artificial Intelligence (AI) has the potential to attain faster results detained with proper quality conditions for medical use. Therefore, after running the MRI scan faster and gathering less raw data, an SR method can be exploited to reconstruct the MRI. Since collecting that data is what makes MRI so slow, this concept can speed up the scanning process significantly.

In general, SR methods are currently based on Generative Adversarial Networks (GANs), which were introduced by Goodfellow et al. (2014) and have recently gained a lot of attention. GANs introduce an alternative way of conceiving models capable of generating data, entitled generative models, and recently they have been used for several image-based applications.

Motivated by the convenience of recovering high-resolution images from low-resolution ones, this work conducts a comparative and benchmark study that focuses on investigating different GAN architectures that can achieve superior performance in MRI spatial resolution enhancement.

Several surveys (Mahapatra et al., 2019; Gupta et al., 2020; Li et al., 2020; Hüsem and Orman, 2020; Anwar et al., 2020; Tian et al., 2022) have addressed the subject of SR. However, they usually lack experiments in the context of MRI, fail to mention relevant state-of-the-art models, or do not mention GAN framework problems and strategies. Distinctively, this work employs rigorous experiments over an MRI dataset using state-of-the-art models and contributes with solutions for GAN problems. Additionally, this work intends to validate the proficiency of GANs in providing extremely detailed anatomical information appropriate to accommodate reliable diagnoses.

Succeeding a rigorous analysis of the state-of-the-art, several GAN-based models were selected based on a comprehensive selection criteria that took into consideration key aspects, such as the performance under multiple applications and the publication date. Subsequently, the most recent models that manifest state-of-the-art performance were selected. The performance of these models is evaluated on the FastMRI dataset (Zbontar et al., 2018). Meanwhile, SRGAN is not considered in the experiments due to the lack of performance, as subsequent models have already surpassed it by some margin.

Regarding the outline of this paper, Section 3 reviews state-of-the-art GANs for SR. Subsequently, it is carried out a discussion about GAN-based MRI reconstruction problems and optimization strategies intended to minimize error when fitting SR algorithms. Afterwards, Section 6 describes experiments performed over FastMRI to exhibit the effectiveness of GANs among medical image reconstruction and processing. Ultimately, Section 6 holds an extensive discussion with quantitative and qualitative results.

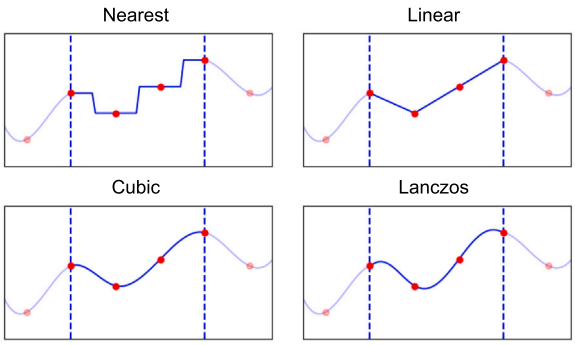


Fig. 1. Interpolation Algorithms applied in 1-dimension.

2. Super-resolution overview

2.1. Problem definition

SR is the process responsible to reconstruct an image that manifests a reduced spatial resolution. Considering a low-resolution image, y, and the corresponding high-resolution ground truth counterpart, $\hat{x}_r$, then the degradation process can be mathematically given as:

$$y = \Phi\left(\hat{x}_r; \Omega\right), \tag{1}$$

where Φ is the degradation function and Ω the respective parameters.

In many real-world scenarios, the understanding of both Φ and Ω is complex and often unknown, thus SR tries to revert the undefined degradation by estimating a high-resolution approximation, x_g, of the ground truth image, $\hat{x}_r$. Essentially, SR is the inverse process of the degradation model, given as:

$$x_g = \mathcal{F}(y; \Theta) = \Phi^{-1}(y; \Theta) \approx \hat{x}_r, \tag{2}$$

where $\mathcal{F}$ is the SR process and Θ the model parameters. The optimization of Θ can be defined as:

$$\hat{\Theta} = \arg\min_{\Theta} \mathcal{L}\left(x_g, \hat{x}_r\right), \tag{3}$$

where $\mathcal{L}$ is a function that estimates the difference error between x_g and $\hat{x}_r$. Moreover, $\hat{\Theta}$ denotes the optimal parameters of the trained model $\mathcal{F}$.

The degradation process is complex and affected by multiple factors (Greenspan et al., 2002), such as stochastic noise, blur, compression and variable artifacts. Therefore, a preferable equation to define the degradation model is:

$$y = \Psi\left(\left(\hat{x}_r \otimes k\right)\downarrow_s + N\left(\mu, \sigma^2\right)\right), \tag{4}$$

where k is the blurring kernel, $\otimes$ the convolution operation and $\downarrow_s$ the downsampling operation with a scale factor of s. In addition, N corresponds to the Gaussian noise with a mean μ and standard deviation σ, and Ψ is the compression operation.

2.2. Interpolation-based upsampling methods

Image Interpolation is the task of resizing images from one pixel grid to another by estimating the pixel intensities of the interpolated points. Interpolation algorithms, such as the Nearest Neighbor, Bilinear, and Bicubic Interpolation (Han, 2013; Rahim et al., 2015), can be very efficient and easy to implement. Furthermore, Duchon (1979) proposed a more sophisticated approach, derived from the Lanczos Filtering. However, despite being the simplest way to upscale an image, these interpolation methods oversimplify the SR problem and in most cases attain solutions with excessively smooth textures (Carey et al., 1999). An illustration of point interpolations in a one-dimensional space is given in Fig. 1.

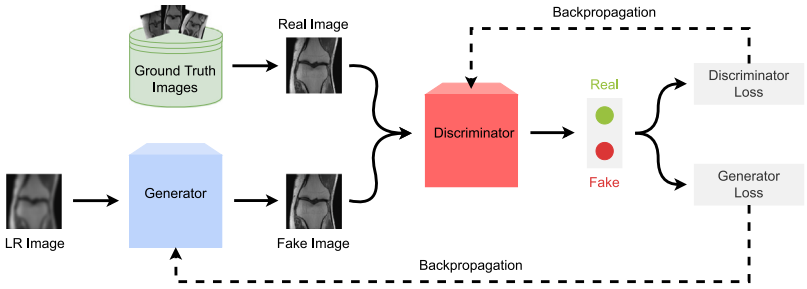


Fig. 2. Main concept behind GANs.

2.3. Deep learning methods

In practice, SR is a problem of missing data. Lost data cannot be recovered by further processing, *i.e.*, information that is not present cannot be inferred. This is where neural networks manifest significant value, considering they can learn to conceive details based on some prior information they have extracted from a large training set. Therefore, they can perform SR by adding details onto an LR image, because even if the information is not on the input LR image, it is somewhere in the training sample.

GANs employ a clever strategy to train a generative model by posing the SR task as a supervised learning problem. They consist of two adversarial neural networks that compete with each other. The first network, denoted as Generator, captures the data distribution, while the second one, named Discriminator, estimates the probabilities of samples being real or fake. In other words, given a sample of LR images, the Generator will produce fake HR images that can fool the Discriminator into classifying them as real images. Meanwhile, the Discriminator intends to accurately label images either as real or fake. The predicted labels will help to train both Neural Networks through backpropagation, where the Discriminator loss function penalizes the Discriminator for wrongly predicted labels, while the Generator loss function penalizes the Generator whenever HR generated images do not deceive the Discriminator and are labeled correctly as fake. Once the training has finished, only the Generator part is needed to upscale the LR images, and ideally, the Generator is capable of generating HR images exceptionally similar to the ground truth ones. A generalized application of GANs applied on the SR task is shown in Fig. 2.

3. GAN-based methods for super-resolution

This section conducts a review on GAN methods that reached state-of-the-art performance. Every method is discussed in order considering its publication date.

3.1. SRGAN

Most methods reviewed in this work were inspired by SRGAN (Ledig et al., 2017), which was a novel SR approach using the GAN concept.

The optimization of SR methods is predominately driven by the choice of the target function. Before SRGAN, most relevant work had largely focused on minimizing the mean squared reconstruction error (MSE), however the resulting estimates failed to match the fidelity present at the high resolution domain (see Sections 4.3 and 7). To cope with this issue, SRGAN introduces a new GAN architecture and diverges from MSE as the single target for optimization. The proposed GAN-based network uses a novel loss intended to optimize the generator network while exploiting high-level feature maps of the Visual Geometry Group (VGG) network (Simonyan and Zisserman, 2014). Moreover, the generator employs a deep residual network (He et al., 2016) with skip connections as depicted in Fig. 3.

The ultimate intention of SRGAN is to train a function $\mathcal{G}$ that estimates HR images from its LR counterparts. Therefore, a generator network is trained as a feed-forward convolutional neural network (CNN). To optimize the generator, the proposed loss function is employed, consisting in a weighted sum of a perceptual loss and an adversarial loss component (see Section 4).

The perceptual loss is regarded as the euclidean distance between the feature representations of a reconstructed image x_g and the ground truth $\hat{x}_r$. Considering the adversarial loss $\mathcal{L}_{\mathcal{G}}$, it favors solutions that reside on the manifold of natural images and is given by Eq. (17) in Section 4.2.

This network is parameterized by $\Theta_{\mathcal{G}}$, representing the weights and biases. Reasoning, these parameters can be obtained by optimizing the generator loss function $\mathcal{L}_{HR}$:

$$\hat{\Theta} = \arg\min_{\Theta_{\mathcal{G}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{HR}\left(\mathcal{G}\left(y_i; \Theta_{\mathcal{G}}\right), \hat{x}_{r_i}\right), \tag{5}$$

where y_i represents an input LR image that will be super-resolved by the inference function $\mathcal{G}(\cdot)$. Moreover, $\hat{x}_{r_i}$ is the corresponding ground truth and N denotes the number of pair-wise training images. Additionally, $\mathcal{L}_{HR}$ is defined in Table 1.

The effectiveness of SRGAN in inferring high-resolution (HR) images, with an upscale factor of ×4, was measured by PSNR and SSIM metrics (see Section 6.2). Its proficiency was further validated through an extensive evaluation process, which conducted a Mean Opinion Score (MOS) test on images sourced from three public benchmark datasets. The results of these tests evidence that, absent from an optimization solely around MSE, SRGAN is able to recover textures and details from heavily downsampled images.

3.2. ESRGAN

Based on the SRGAN pioneer work (Ledig et al., 2017), a new model named Enhanced SRGAN (ESRGAN) (Wang et al., 2018b) was introduced to reduce unpleasant artifacts present in the SRGAN generated data (see Fig. 4). ESRGAN revisits three key components to improve the previous approach: network architecture, adversarial loss and perceptual loss.

The original SRGAN model is built with residual blocks (He et al., 2016) and optimized using a perceptual loss in a GAN framework. Meanwhile, ESRGAN improves the generator structure by removing Batch Normalization (BN) layers and introducing the Residual-in -Residual Dense Block (RRDB), which is of higher capacity and easier to train.

The rationale behind the BN removal is that although Batch Normalization does help a lot on numerous computer vision tasks, concerning SR or image restoration tasks in general, Batch Normalization can create some artifacts as depicted in Fig. 4. BN layers normalize the features using mean and variance in a batch during training and afterwards use the estimated mean and variance of the whole training dataset during testing. When the statistics of training and testing datasets substantially differ, BN layers tend to introduce unpleasant artifacts and limit the generalization ability (Lim et al., 2017).

The high-level architecture design of SRGAN (Ledig et al., 2017), as depicted in Fig. 3, is employed and the replacement of the original

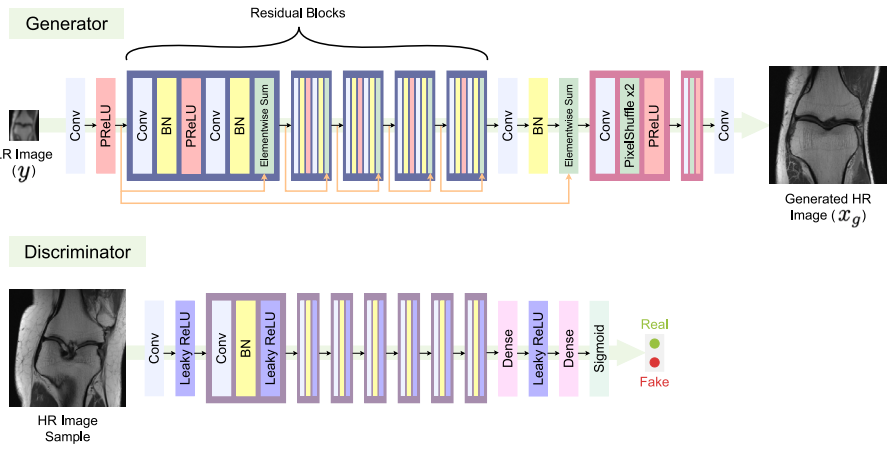


Fig. 3. Basic architecture of SRResNet (SRGAN) (Ledig et al., 2017).

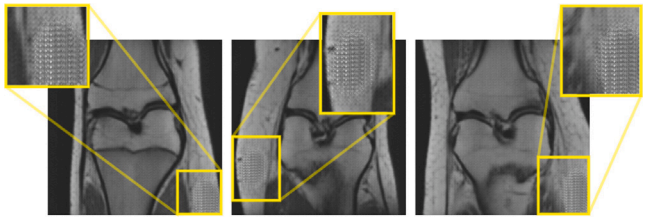


Fig. 4. Batch Normalization artifacts under SRGAN on fastMRI images. These artifacts include visual distortions such as checkerboard patterns, which are undesirable because they can lead to misinterpretation of medical images, hinder accurate diagnosis, and potentially affect treatment planning.

basic block with the proposed RRDB boosts performance and improves the perceptual quality. Deeper models with the proposed RRDB can further improve the recovered textures and reduce unpleasing noises, since the deep model has a strong representation capacity to capture semantic information. The following RRDB and its dense connections are illustrated in Fig. 5 next to the SRGAN Residual Block.

Furthermore, ESRGAN attains sharper edges and more visually pleasing results by proposing an improved perceptual loss that uses the VGG features before activation instead of after activation as in traditional SRGAN. Regarding adversarial loss, the discriminator is refined by shifting to the idea that it learns to judge "whether one image is more realistic than a fake one" rather than "whether one image is real or fake". Essentially, a relativistic discriminator (Jolicoeur-Martineau, 2018) is employed to increase the probability that generated data is perceived as real. Distinctively from the standard discriminator in SRGAN, which estimates the probability that an image instance I is real and natural, the relativistic discriminator tries to estimate the probability that a real image x_r is relatively more real than a randomly sampled set of synthesized images $\mathcal{X}_g$. Accordingly, the dissimilarities between the traditional and the relativistic discriminators are given as follow:

$$\underbrace{D(x_r) = \sigma\left(\hat{D}(x_r)\right) \to 1 \implies D(x_g) = \sigma\left(\hat{D}(x_g)\right) \to 0}_{Standard\,discriminator} \quad \underbrace{D(x_r, \mathcal{X}_g) = \sigma\left(\hat{D}(x_r) - \mathbb{E}_{x_g}\left[\hat{D}(x_g)\right]\right) \to 1, \implies D(x_g, \mathcal{X}_r) = \sigma\left(\hat{D}(x_g) - \mathbb{E}_{x_r}\left[\hat{D}(x_r)\right]\right) \to 0,}_{Relativistic discriminator} \tag{6}$$

where σ denotes the sigmoid activation function, $\hat{D}$ represents the non-transformed discriminator output and $\mathcal{X}$ corresponds to a sampled set of either real or fake images. Moreover, $\mathbb{E}_{x_r}$ and $\mathbb{E}_{x_g}$ correspond to the operation of taking the average over all real and generated sampled data, respectively. Subsequently, the adversarial losses for the generator and discriminator are defined as the following equations:

$$\mathcal{L}_G = -\mathbb{E}_{x_r}\left[\log\left(1 - D(x_r, \mathcal{X}_g)\right)\right] - \mathbb{E}_{x_g}\left[\log\left(D(x_g, \mathcal{X}_r)\right)\right], \tag{7}$$

$$\mathcal{L}_D = -\mathbb{E}_{x_r}\left[\log\left(D(x_r, \mathcal{X}_g)\right)\right] - \mathbb{E}_{x_g}\left[\log\left(1 - D(x_g, \mathcal{X}_r)\right)\right], \tag{8}$$

where D denotes the relativistic discriminator function. Thus, $D(x_g, \mathcal{X}_r)$ represents the probability that a generated image x_g is relatively less realistic than randomly sampled real ones and $D(x_r, \mathcal{X}_g)$ the probability that a real image x_r is more realistic than sampled generated ones. Fundamentally, every generated image realism is relatively compared against the average realism of all real images and vice-versa. Contrarily to the generator adversarial loss in SRGAN (see Section 4.2), Eq. (7) in ESRGAN is simultaneously conditioned by real and generated data. Therefore, the generator benefits from the gradients of both generated and real data during the adversarial training rather than depending solely on the feedback derived from fake data. This discriminator improvement helps the generator to recover more realistic texture details. The generator loss can be given in terms of the adversarial loss $\mathcal{L}_G$, as shown in Table 1. Meanwhile, the discriminator loss can be directly inferred from $\mathcal{L}_D$, without any further computation, as it is solely defined by it, $\mathcal{L}_D = \mathcal{L}_{Discriminator}$.

3.3. RankSRGAN

Perceptual quality can be assessed by perceptual metrics, such as Perceptual Index (PI) (Blau et al., 2018), Natural Image Quality Evaluator (NIQE) (Mittal et al., 2012), and Ma et al. (2017), which are highly correlated with human perception. However, existing methods cannot directly optimize these metrics. Therefore, to optimize a network in the direction of these perceptual metrics a new approach was proposed consisting of a GAN with a Ranker, named RankSRGAN (Zhang et al., 2021b).

RankSRGAN employs the standard architecture design of SRGAN (Ledig et al., 2017). In addition to SRGAN, a novel rank-content loss is introduced to optimize the perceptual quality. In essence, this Rank Loss, $\mathcal{L}_{\mathcal{R}}$, uses a well-trained Ranker, which can measure the output image quality by learning the behavior of perceptual quality metrics. The ranker is trained by optimizing a margin-ranking loss (Burges et al., 2005) and eventually learns to rank images according to the perceptual scores measured by a given indifferentiable perceptual metric (e.g. PI or NIQE).

The Ranker adopts a Siamese architecture to learn the behavior of perceptual metrics as depicted in the middle section of Fig. 6. Primarily, different SR models are used to generate images. Then, these generated images are put together two by two to form pairwise images. Subsequently, these pairs are ranked/labeled according

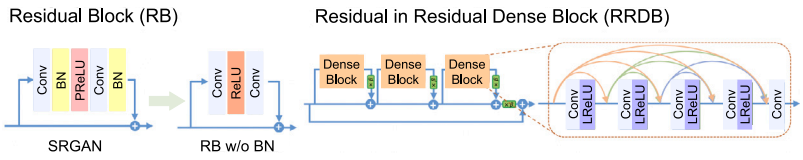


Fig. 5. On the left, theoretical batch normalization removal in SRGAN basic block. On the right, structure of the introduced Residual in Residual Dense Block, which replaces the basic SRGAN residual block.
Source: Figure adapted from Wang et al. (2018b).

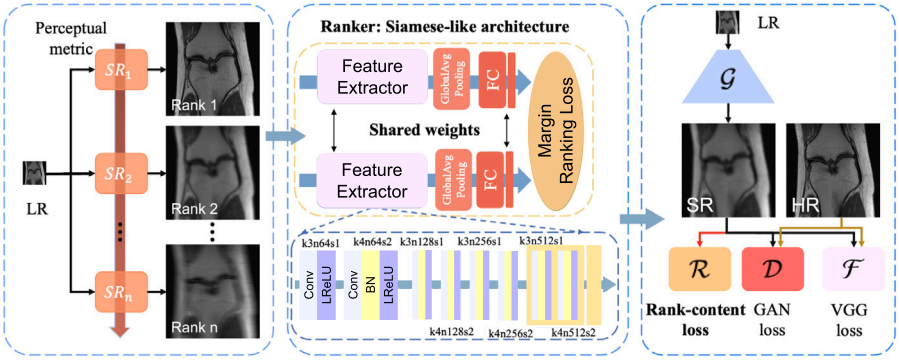


Fig. 6. Overview of RankSRGAN. Essentially, RankSRGAN consists of a generator ($\mathcal{G}$), a discriminator (D), a fixed feature extractor ($\mathcal{F}$) and a ranker ($\mathcal{R}$).
Source: Figure adapted from Zhang et al. (2021b).

to the quality score calculated by the perceptual metric, as expressed in (9). Afterwards, the Siamese-like Ranker network is trained over the rank dataset consisting of the aforementioned pair-wise images and its associated ranking labels. Ultimately, the rank-content loss derived from the well-trained Ranker is introduced to guide the GAN training.

The Siamese architecture manifests effectiveness over pair-wise inputs and is designed to simulate the behavior of perceptual metrics through the learning to rank approach. As shown in Fig. 6, the Ranker has two identical network branches with shared weights, which contain a series of convolutional, Leaky ReLU, pooling and fully-connected layers to attain the ranking information. Each one of these network branches processes an image and produces a ranking score s_i. Afterwards, the outputs of both branches are passed to the margin-ranking loss. Subsequently, the gradients can be computed and back-propagation is applied to update the parameters of the whole Ranker network. To train the Ranker, the margin-ranking loss is employed, such that the ranking score difference between generated images with equally good perceptual quality is small, and the ranking score difference between generated images with dissimilar quality is large:

$$\mathcal{L}\left(s_1, s_2, \gamma\right) = \max\left(0, \left(s_1 - s_2\right) \cdot \gamma + \epsilon\right), \quad \begin{cases} \gamma = 1 & \text{if} \quad m_1 > m_2 \\ \gamma = -1 & \text{if} \quad m_1 < m_2 \end{cases}, \tag{9}$$

where s_1 and s_2, correspond to the ranking scores, derived from the Ranker, of the generated images x_{g_1} and x_{g_2}, respectively. Moreover, estimated by a perceptual quality metric, m_1 and m_2 represent the perceptual quality scores of the pair-wise images x_{g_1} and x_{g_2}. Meanwhile, γ is the rank label of the pair-wise training images contained in the rank dataset and ϵ is a constant used to control the distance between ranking scores. A lower ranking score indicates better perceptual quality. Additionally, as a means to ease comprehension the following can be conjectured:

$$\begin{cases} s_1 < s_2 & \text{if} \quad m_1 > m_2 \\ s_1 > s_2 & \text{if} \quad m_1 < m_2 \end{cases}. \tag{10}$$

Optimally the Ranker outputs similar ranking orders as the perceptual metric. Therefore, the optimization process is carried out through the minimization of the function:

$$\begin{aligned} \hat{\Theta} &= \arg\min_{\Theta_{\mathcal{R}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(s_1^{(i)}, s_2^{(i)}, \gamma^{(i)}\right) = \\ &= \arg\min_{\Theta_{\mathcal{R}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\mathcal{R}\left(x_{g_1}^{(i)}; \Theta_{\mathcal{R}}\right), \mathcal{R}\left(x_{g_2}^{(i)}; \Theta_{\mathcal{R}}\right), \gamma^{(i)}\right), \end{aligned} \tag{11}$$

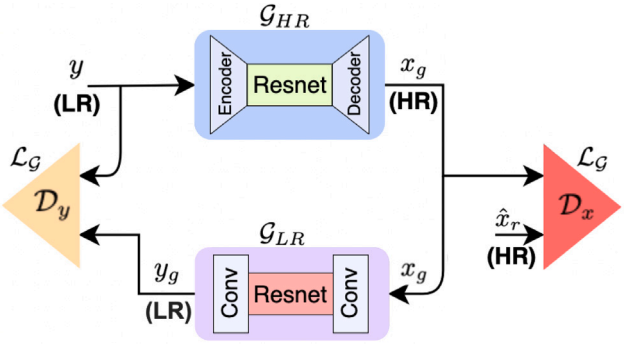
where N represents the number of pair-wise training images, $\Theta_{\mathcal{R}}$ represents the Ranker network weights and $\mathcal{R}(\cdot)$ is the mapping function of the Ranker, which optimally intends to satisfy (10).

Compared to SRGAN, this method simply introduces a well-trained Ranker that is used by the rank-content loss (defined in Section 4) to constrain the generator in the SR space. However, RankSRGAN uses multiple SR models to build the rank dataset since in general a single SR model does not outperform all other SR models on all images. Therefore, mixed orders are obtained within models while evaluating with some perceptual metric. Consequently, the Ranker will favor different algorithms on different images, thus concurrently optimizing the SR network in the direction of multiple SR algorithms. Inherently, RankSRGAN combines the best parts of different SR methods and achieves superior performance both in perceptual metrics and visual quality.

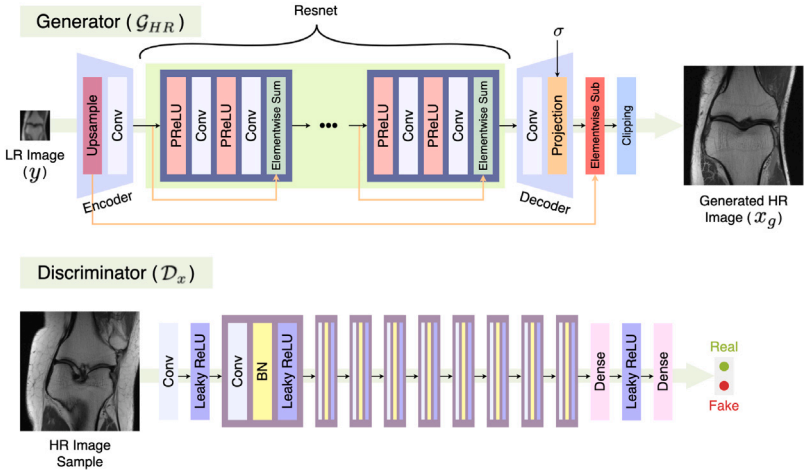
3.4. SRResCycGAN

Inspired by the success of CycleGAN (Zhu et al., 2017) in image-to-image translation applications, a new deep cyclic network structure was proposed, named SRResCycGAN (Umer and Micheloni, 2020). In essence, a GAN is trained to achieve LR to HR translation in an end-to-end manner.

In real-world settings, the LR image endures multiple possible errors during the image acquisition process, such as the inherent sensor noise, stochastic noise, compression artifacts, and possible discrepancies between the forward observation model and the camera device. MRI acquisition is no exception as it can contain a significant amount of noise caused by operator performance, patient motion, equipment or environment, leading to unpleasant results (Vaishali et al., 2015). SRResCycGAN overcomes this challenge and maintains the domain



(a) Global structure, adapted from Umer and Micheloni (2020).



(b) Architecture

Fig. 7. SRResCycGAN structure and architecture. The network consists of a GAN framework, which follows a cyclic learning strategy to enforce domain consistency between LR and HR images.

consistency between the LR and HR data distributions by following the CycleGAN structure, as shown in Fig. 7(a).

The generator $\mathcal{G}_{HR}$ takes the input LR image y and generates the HR image x_g with the supervision of the discriminator network $\mathcal{D}_x$, which tries to estimate the probabilities of HR samples being real or fake. Then, to maintain the domain consistency between the LR and HR data distributions, the $\mathcal{G}_{LR}$ takes as input the fake generated HR image x_g and transforms it back into a LR image y_g. Likewise, the $\mathcal{G}_{LR}$ is under supervision of the discriminator network $\mathcal{D}_y$, which estimates the probabilities of LR samples being real or fake, analogous to $\mathcal{G}_{HR}$ with HR images.

Using exclusively adversarial loss, the $\mathcal{G}_{HR}$ network can map the same set of LR input images to any random permutation of images in the HR target domain. This network behavior favors results that are the "best possible" rather than "perfect". Reasoning, in the context of MRI the generated images should be as close as possible to the ground truths, therefore results would not fulfill the requirements to assist medical applications. To overcome this challenging ill-posed problem, the referred cyclic process introduces a cycle consistency loss to enforce that $\mathcal{G}_{LR}(\mathcal{G}_{HR}(y)) \approx y$, thus reducing the number of possible mappings.

Regarding network architecture, the HR generator network $\mathcal{G}_{HR}$ is borrowed from SRResCGAN (Umer et al., 2020). The generator consists of a Encoder-Resnet-Decoder structure as shown in Fig. 7. Inside the Encoder, the LR image y is upsampled and afterwards is subtracted from the output of the Decoder. The Resnet consists of 5 residual blocks and the projection layer in the Decoder handles the data fidelity and prior terms by computing the proximal map with the estimated noise standard deviation σ.

The innovation disclosed by this approach comes from the proposed cyclic loss component directed to maintain the domain consistency between LR and HR images. This cyclic loss, along with other components, is used to optimize the SRResCycGAN network through the following equation:

$$\mathcal{L}_{HR} = \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\mathcal{TV}} + \lambda \cdot \mathcal{L}_1 + \eta \cdot \mathcal{L}_{\mathcal{CYC}}, \tag{12}$$

where $\mathcal{L}_{\mathcal{P}}$ is the perceptual loss, $\mathcal{L}_{\mathcal{G}}$ the adversarial loss, $\mathcal{L}_{\mathcal{TV}}$ the total-variation loss, $\mathcal{L}_1$ content loss and $\mathcal{L}_{\mathcal{CYC}}$ the cyclic loss. Additionally, λ and η are coefficients intended to balance the different loss components, and both take the value of 10 in Umer and Micheloni (2020). These losses are defined in the Learning Strategies Section 4.

3.5. BSRGAN

Single Image Super-Resolution (SISR) methods would not perform well if the assumed degradation model deviates from those in real images. Therefore, a model named BSRGAN (Zhang et al., 2021a) was proposed along with a new degradation model.

Although several degradation models take additional factors into consideration, such as blur, they are still not effective enough to cover the diverse degradations of real images. Therefore, a deep blind ESRGAN is trained based on the new degradation model, which consists of randomly shuffled blur, downsampling and noise degradations as shown in Fig. 8. With the random shuffle strategy, the degradation space can be expanded substantially. Consequently, the SR model is able to super-resolve LR images under unknown and diverse degradations.

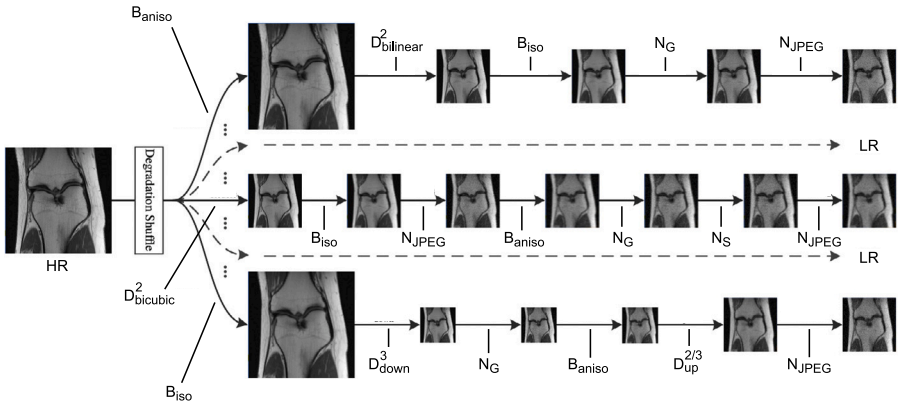


Fig. 8. Proposed BSRGAN degradation model for a scale factor of 2. For a scale factor of 4, an additional bilinear or bicubic downscaling is applied. The type of blur employed is denoted by B_{type} and N_{type} is the type of noise. Meanwhile, D^{scale}_{type} stands for the downsampling applied under a defined scale.
Source: Figure adapted from Zhang et al. (2021a).

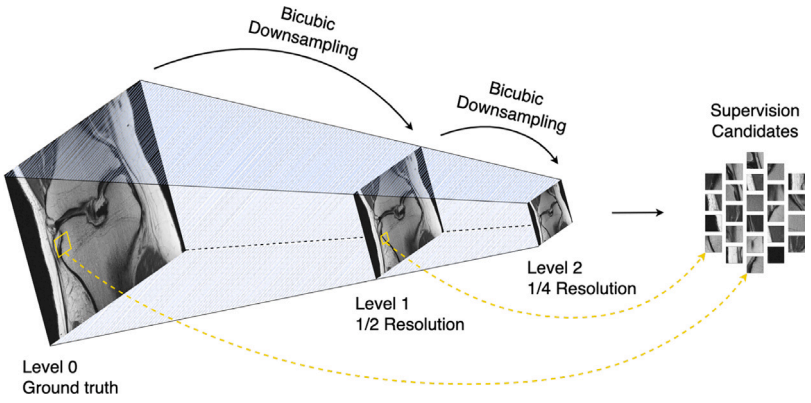


Fig. 9. Beby-GAN 3-level image pyramid obtained with bicubic downsampling. Subsequent images are subject to repeated downsampling. Additionally, other types of degradation can be introduced in each subsampling level.

The novelty of this approach lies in the new degradation model and the possibility of existing network structures such as ESRGAN to be borrowed to train a deep blind SR model with paired LR-HR images. Following ESRGAN, a perceptual quality-oriented model is trained, named BSRGAN, by minimizing a weighted combination of L1 loss, VGG perceptual loss and spectral norm-based least square PatchGAN loss (Isola et al., 2017).

3.6. Beby-GAN

Most SR methods rely on one-to-one mappings, which is not flexible enough to solve the ill-posed SR challenge. Also, to recover spatial resolution, GANs generate fake details. However, this behavior often undermines the realism of the whole image. To address these issues Beby-GAN (Li et al., 2021) is proposed. It consists in the idea of relaxing the immutable one-to-one constraint and allow estimated patches to dynamically seek the best supervision during training, thus attaining photo-realistic high-frequency details.

Commonly used loss functions tend to enforce a rigid mapping between the given LR and HR images, thus constraining the HR space and eventually jeopardizing the network. To relax this one-to-one constraint a novel best-buddy loss is introduced. In essence, the best-buddy loss consists in an improved one-to-many MAE loss, that uses HR supervision signals to flexibly exploit the ubiquitous self-similarity existent in natural images, *i.e.*, an HR patch is supervised by different but close to its corresponding ground truth patches, hence favoring trustworthy and rich details through a more flexible supervision.

A single LR patch may correspond to multiple HR solutions. The key idea is that the generated HR patch can be supervised by different HR targets in different iterations, *i.e.*, gradient updates (see Fig. 11). These close to ground truth patches are sourced from multiple scales of the corresponding ground truth image. Essentially, the ground truth is downsampled with multiple scale factors. This results in a multi-scale ground truth image pyramid, which is subsequently split to generate all candidates, resulting in multiple patches with diverse resolutions, as depicted in Fig. 9.

During training, to supervise an estimated HR patch p_g and thus optimize the network, Beby-GAN looks for its corresponding supervision patch in the current iteration. The supervision patch, also named best-buddy patch p_{BB} BB, must meet two constraints:

Constraint 1

It is mandatory that the best-buddy patch p_{BB} BB is similar to the predefined ground truth patch $\hat{p_r}$. Relying on the multi-scale self-similarity present in natural images it is expected to find an HR patch consonant with $\hat{p_r}$.

Constraint 2

To alleviate the optimization process, the best-buddy patch p_{BB} BB is required to be close to the generated HR patch p_g. Accordingly, it is vital that p_g is a decent estimation and thus the generator needs to be well initialized to avoid bad early predictions. Also, the diverse resolution patches, resulting from the multi-scale pyramid, ensure that there is always some patch close enough to supervise p_g, even when the network is warming up and estimations are not very good. This results in a scalable and flexible learning strategy, where the most appropriate supervision patch is used in every iteration.

Following these two objectives, the selected best-buddy patch p_{BB} BB is perceived as a plausible SR target. During training, in every iteration, the multi-scale ground truth image pyramid and the generated image are split in patches. Each estimated patch p_g from the fake HR image is supervised with the best-buddy patch p_{BB} BB in the current iteration rather than supervised with the immutable ground truth patch $\hat{p}_r$. The best-buddy patch for some LR patch p_{y_i} in the current iteration is given as:

$$p_{BB_i} = \arg\min_{p \in S} \alpha \left\| p - \hat{p}_{ri} \right\|_2^2 + \beta \left\| p - p_{g_i} \right\|_2^2, \tag{13}$$

where p represents a patch contained in S, which is the supervision candidate dataset of the generated image. Essentially, S consists of patches from the multi-scale image pyramid. Moreover, α and β denote scaling parameters. Furthermore, to update the gradients of the generator network, the best-buddy loss for this patch pair (p_g, p_{BB}) BB is given as the distance between the estimated patch p_g and the best-buddy patch p_{BB} BB, *i.e.*, the 1-norm of the difference, as defined in Section 4. Reasoning, when $\alpha \gg \beta$, the best-buddy loss corresponds to the traditional MAE loss.

Reasonably, this relaxation in the one-to-one constraint, may encourage results that slightly diverge from the real ground truths, which is not optimal in the MRI context. However, as previously mentioned, SISR is an ill-posed challenge, where it is theoretically impossible to estimate the ground truth, because from one LR image there can be multiple plausible solutions. This non determinism comes from the fact that different ground truth images can have equal LR images even if they went through different degradation processes. Furthermore, it is reasonable to consider that this relaxation can help the training phase to jump out of a bad local minimum and have more chances of finding either a better local minimum or even the global minimum, *i.e.*, even though this idea makes the plausible HR space bigger, the optimal solution is not discarded and may become easier to reach as a result of the flexible and scalable supervision. Additionally, ignoring the inherent uncertainty of SISR can lead to never recover the ground truth nor even a good solution.

Therefore, to avoid substantial deviations from reality and without breaking the concept of relaxing the one-to-one mapping, a back-projection constraint is enforced on the generated image x_g. Analogous in some extent to the cyclic loss from SRResCycGAN (Umer and Micheloni, 2020), an HR-to-LR operation is introduced to ensure the validity of the estimated HR images. Thus, a back-projection loss can be defined to ensure that the projections of the generated images onto the LR space are still consistent with the corresponding input LR images:

$$\mathcal{L}_{BP} = \left\| Z\left(x_g, s\right) - y \right\|_1, \tag{14}$$

where $Z(I, s) : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{\frac{H}{s} \times \frac{W}{s}}$ represents a downscaling operation with a downscale factor s. The operation adopted in Li et al. (2021) is bicubic downsampling. Additionally, y denotes a LR image and x_g a generated HR one. The supervision patch selection process and loss inference are illustrated in Fig. 10.

As shown in Fig. 4, previous GAN-based methods were prone to undesirable artifacts, specially in flat regions. Consequently, Li et al. (2021) introduced a Region-Aware Adversarial Learning strategy, which directs the model to focus on generating details for textured areas adaptively. In essence, the network treats smooth and well-textured areas differently, and only performs the adversarial training on rich-texture areas. This separation encourages the network to focus more on regions with rich details while avoiding generating unnecessary texture on flat regions. Therefore, less undesirable artifacts are introduced in the reconstructed HR images.

This separation is conducted according to local pixel statistics. In detail, the ground truth $\hat{x}_r \in \mathbb{R}^{H \times W}$ of the current iteration is split into patches $p \in \mathbb{R}^{k \times k}$ with size k^2. Then, for each patch, the standard

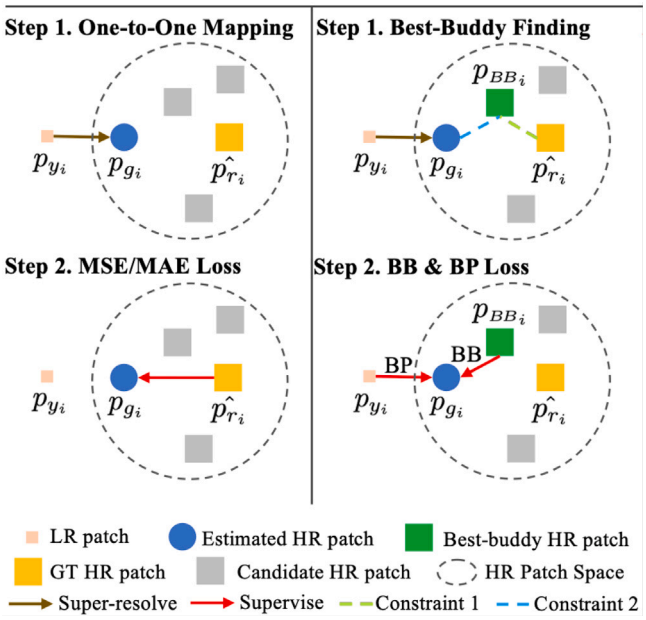


Fig. 10. Comparison between MSE/MAE and best-buddy (BB) loss with a back-projection (BP) constraint. On the left, representation of the rigid mapping between LR and HR patches present in traditional optimization processes. On the right, the LR-HR mapping is relaxed, enabling diverse targets to supervise the estimated HR patch, rather than being restricted to the immutable predefined ground truth patch.
Source: Figure adapted from Li et al. (2021).

deviation is computed. Subsequently, a binary mask can be formulated as:

$$M_{i,j} = \begin{cases} 1 & \text{if} \quad \sigma\left(p_{i,j}\right) \geq \delta \\ 0 & \text{if} \quad \sigma\left(p_{i,j}\right) < \delta \end{cases}, \tag{15}$$

where the pair (i, j) denotes the patch location and δ is a predefined threshold. Moreover, σ corresponds to the standard deviation. This results in highly textured regions marked as 1 while flat regions as 0. Afterwards, the mask M is applied on both the generated HR image x_g and the ground truth $\hat{x}_r$, thus yielding x_g^M and $\hat{x}_r^M$, respectively. Then, the resulting masked images are fed into the Discriminator. In essence, only the textured content is fed into the Discriminator, considering that smooth regions can be easily recovered without adversarial training. The whole process can be seen in Fig. 11.

Ultimately, Beby-GAN borrows a pre-trained ESRGAN generator architecture (Wang et al., 2018b) due to its proven state-of-the-art performance. Hence, both models have the same number of parameters in the generator, as show in Table 1. Essentially, Beby-GAN exploits the example-based methods idea of searching for one-to-many LR-HR mappings to produce visually pleasing results. Also, a significant drawback when implementing a multi-scale SR task is that more computations and memory space are required for model training and storage.

3.7. Real-ESRGAN

The previous ESRGAN approach is extended to achieve superior visual performance on various datasets. Real-ESRGAN (Wang et al., 2021) aims to restore general real-world LR images by synthesizing training pairs with a more practical degradation process. In essence, starts by improving the VGG-style discriminator in ESRGAN to a U-Net design (Schonfeld et al., 2020). Then, employs the Spectral Normalization (SN) regularization (Miyato et al., 2018) to stabilize the training process, since the U-Net structure and complicate degradations also increase the training instability.

Even after intensive efforts like BSRGAN, synthetic LR images still have evident differences from realistic degraded images. Moreover,

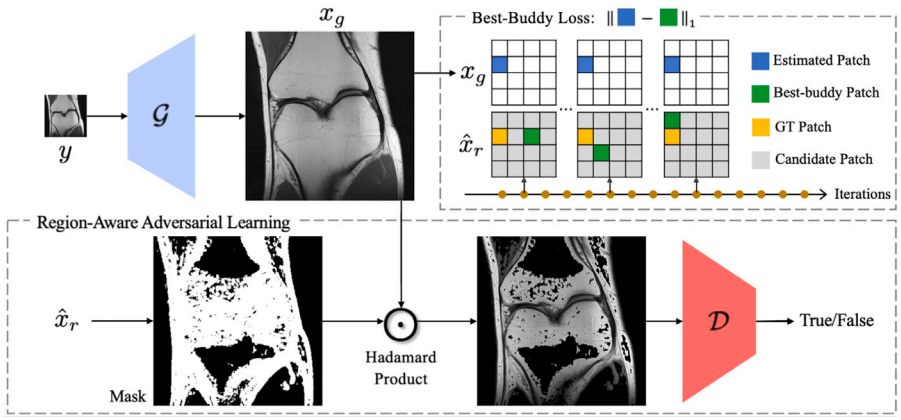


Fig. 11. Scheme of the Beby-GAN framework. The region-aware adversarial learning is intended to make the adversarial training focus on rich-texture areas, thereby only the textured content is fed into the Discriminator.
Source: Figure adapted from Li et al. (2021).

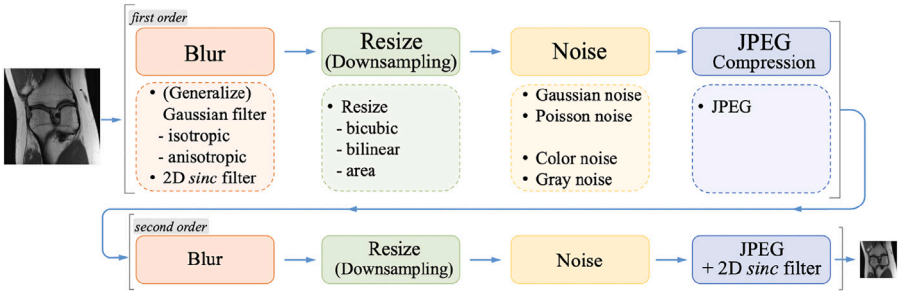


Fig. 12. High-order Degradation Model. Empirically, Real-ESRGAN adopts a second-order degradation model, where the degradation process is repeated at most twice for a balance between simplicity and effectiveness.
Source: Figure adapted from Wang et al. (2021).

real-life degradation processes are quite diverse. Therefore, to better mimic the real-world degradation process Real-ESRGAN uses a synthetic data generation process as depicted in Fig. 12. Consequently, Real-ESRGAN robustness is improved and is capable of restoring more realistic textures for real-world samples, while other methods either fail to remove degradations or add unnatural textures.

Real-ESRGAN adopts the same generator as ESRGAN, which follows the basic architecture of SRGAN with several Residual-in-Residual Dense Blocks (RRDB), as shown in Figs. 3 and 5. Regarding the discriminator, as Real-ESRGAN aims to address a larger degradation space than ESRGAN, the original discriminator design is no longer suitable. Requiring a greater discriminative power and inspired by Schonfeld et al. (2020), the VGG-style discriminator in ESRGAN is improved to a U-Net design with skip connections as depicted in Fig. 13, which provides detailed per-pixel feedback to the generator by outputting realness values for each pixel. Ultimately, the SN regularization is employed to stabilize training and alleviate the over-sharp and unpleasant artifacts introduced by GAN training.

Real-ESRGAN outperforms previous approaches (e.g. ESRGAN Wang et al., 2018b and BSRGAN Zhang et al., 2021a) in both artifact suppression and restoring texture details by local detail enhancement.

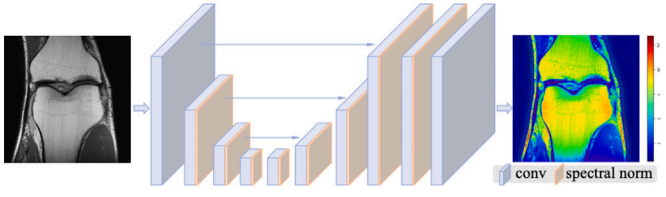


Fig. 13. U-Net discriminator architecture with Spectral Normalization.
Source: Figure adapted from Wang et al. (2021).

Table 1
Comparison of GAN-based SR models. $\mathcal{L}_P$ represents the perceptual loss, $\mathcal{L}_G$ the adversarial loss, $\mathcal{L}_R$ the rank-content loss, $\mathcal{L}_{CYC}$ the cyclic loss, $\mathcal{L}_{BB}$ the best-buddy loss, $\mathcal{L}_{TV}$ the total-variation loss and $\mathcal{L}_1$ the content loss. Moreover, λ, η, θ and ϕ are coefficients to balance the different loss components.

Method	Parameters	Loss
SRGAN	16.7M	$\mathcal{L}_P + \lambda\mathcal{L}_G$
ESRGAN	16.7M	$\mathcal{L}_P + \lambda\mathcal{L}_G + \eta\mathcal{L}_1$
RankSRGAN	1.55M	$\mathcal{L}_P + \lambda\mathcal{L}_G + \eta\mathcal{L}_R$
SRResCycGAN	380k	$\mathcal{L}_P + \mathcal{L}_G + \mathcal{L}_{TV} + \lambda\mathcal{L}_1 + \eta\mathcal{L}_{CYC}$
BSRGAN	16.7M	$\mathcal{L}_P + \lambda\mathcal{L}_G + \eta\mathcal{L}_1$
Beby-GAN	16.7M	$\lambda\mathcal{L}_{BB} + \eta\mathcal{L}_{BP} + \theta\mathcal{L}_P + \phi\mathcal{L}_G$
Real-ESRGAN	16.7M	$\mathcal{L}_P + \lambda\mathcal{L}_G + \eta\mathcal{L}_1$

4. Loss functions and regularization strategies

This section discusses learning strategies utilized in SR. Furthermore, a concise comparison of the numbers of parameters and generator losses from each GAN model regarded in this work is given in Table 1.

4.1. Perceptual loss ($\mathcal{L}_P$)

The perceptual Loss was proposed by Johnson et al. (2016) to measure the perceptual similarity between two images and enhance the visual quality by minimizing the error in a feature space rather than pixel space. Fundamentally, instead of computing distances in the image pixel space, the images are first mapped into the feature space. Therefore, it enforces rich textures and favors the generation of images with natural image statistics by using an objective that focuses on

the feature distribution rather than merely comparing the appearance. Perceptual loss can be expressed in the equation below:

$$\mathcal{L}_{\mathcal{P}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\mathcal{VGG}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\phi\left(\hat{x}_{r_i}\right) - \phi\left(x_{g_i}\right)\right\|_2^2, \tag{16}$$

where x_{g_i} represents the generated HR image and $\hat{x}_{r_i}$ is the corresponding ground truth image. Moreover, N represents the number of training samples, and $\phi(I)$ denotes the feature maps at some convolution layer within the VGG19 network (Simonyan and Zisserman, 2014), succeeding the feeding of an image I as its input.

4.2. Adversarial loss ($\mathcal{L}_{\mathcal{G}}$)

The Adversarial Loss was proposed to impose the generated images to lie in the natural image space. The standard GAN loss function introduced by Goodfellow et al. (2014) corresponds to a min–max game approach, therefore it is also known as the min–max loss. The generator tries to minimize the following function while the discriminator tries to maximize it:

$$\min_{\theta_{\mathcal{G}}}\max_{\theta_D}\ \mathbb{E}_{x_r}[\log(\mathcal{D}_{\theta_D}(x_r))] + \mathbb{E}_y[\log(1-\mathcal{D}_{\theta_D}(\mathcal{G}_{\theta_{\mathcal{G}}}(y)))], \tag{17}$$

where x_r denotes a real image and $x_g = \mathcal{G}_{\theta_{\mathcal{G}}}(y)$ represents a generated HR image when given input LR image y. Additionally, $\mathbb{E}_{x_r}$ corresponds to the expected value over all real data instances and $\mathcal{D}_{\theta_D}(x_r)$ is the discriminator's estimate of the probability that a real data instance x_r is real. Meanwhile, $\mathbb{E}_y$ is the expected value over all input LR instances y and, in consequence, the expected value over all generated fake instances x_g. In addition, $\mathcal{D}_{\theta_D}(\mathcal{G}_{\theta_{\mathcal{G}}}(y))$ is the discriminator's estimate of the probability that a generated image is real. Moreover, $\theta_{\mathcal{G}}$ and θ_D denote the weights and biases that parameterize the generator network $\mathcal{G}$ and discriminator network D, respectively. The generator and discriminator are jointly optimized with the objective given in function (17). Looking at it as a min–max game, this formulation of the loss enables the function above to be categorized into two equations formulating the Discriminator and Generator losses. Accordingly, the generator loss $\mathcal{L}_{\mathcal{G}}$ is defined based on the discriminator's output and only affects the right term of the expression (17), the term that reflects the distribution of the generated data. Therefore, during the generator's training the left term is dropped, since it only reflects the distribution of the real data. In essence, the adversarial loss for the generator can be represented as follows:

$$\mathcal{L}_{\mathcal{G}} = \frac{1}{N}\sum_{i=1}^{N} -\log(\mathcal{D}_{\theta_D}(\mathcal{G}_{\theta_{\mathcal{G}}}(y_i))), \tag{18}$$

where N represents the number of LR training samples and y_i is a input LR image.

GAN models try to replicate a probability distribution. Therefore, GANs use loss functions that reflect the distance between the distribution of the data generated by the GAN and the distribution of the real/desired data. Consequently, in order to address other challenges, several different variations of the original GAN loss have been proposed, such as Eqs. (7) and (8).

4.3. Content loss ($\mathcal{L}_1$ and $\mathcal{L}_2$)

Reasonably, the most used optimization target in SR applications due to its simplicity and decent results is the Content Loss. It is computed by averaging the pixel-wise differences between the generated HR images and the corresponding ground truths, *i.e.*, each pixel value in a x_g is directly compared with each pixel value in the corresponding $\hat{x}_r$. In essence, estimates the quality of the reconstruction by calculating how different the generated images are from the real images. Therefore, it is also called reconstruction loss.

From this class of loss functions many variants are formulated, such as $\mathcal{L}_1$ and $\mathcal{L}_2$. These loss functions are in charge of optimizing the error between pixel values corresponding to the generated and ground truth images. Reducing the distance between pixels can effectively ensure the quality of the reconstructed image and therefore hold a higher peak signal to noise ratio value.

Regarding $\mathcal{L}_1$, also known as Mean Absolute Error (MAE), it is computed by averaging the sum of the absolute differences between predictions and actual observations:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{G}(y_i) - \hat{x}_{r_i}\right\|_1, \tag{19}$$

where $\mathcal{G}(y_i)$ represents a generated HR image x_{g_i} when given an LR image y_i and $\hat{x}_{r_i}$ is the corresponding ground truth image.

Concerning $\mathcal{L}_2$, also known as Mean Square Error (MSE) or quadratic loss, it is computed by averaging the sum of the squared differences between generated and real images:

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{G}(y_i) - \hat{x}_{r_i})^2, \tag{20}$$

Due to the squaring operation, the predictions that are far away from the actual values are heavily penalized in comparison to those less deviated.

Generally, $\mathcal{L}_2$ loss converges faster than $\mathcal{L}_1$, but in image processing applications it is prone to over smoothing. Hence, $\mathcal{L}_1$ and its variants are favored over $\mathcal{L}_2$ in image-to-image translations. Looking at Table 1 it is evident the preferable usage of $\mathcal{L}_1$ over $\mathcal{L}_2$, for instance in ESRGAN (Wang et al., 2018b), SRResCycGAN (Umer and Micheloni, 2020), BSRGAN (Zhang et al., 2021a) and Real-ESRGAN (Wang et al., 2021). Nonetheless, $\mathcal{L}_1$ is not immune to over smoothing and optimizing the SR network with content loss as the sole optimization target usually leads to unnatural blurry reconstructions, because these losses measure the error magnitude without considering its direction.

4.4. Rank-content loss ($\mathcal{L}_{\mathcal{R}}$)

After the Ranker $\mathcal{R}$ is trained through the learning to rank approach (Liu et al., 2009), a ranking score s of a generated image x_g can be estimated. Therefore, the rank-content loss can be formulated as:

$$\mathcal{L}_{\mathcal{R}} = \frac{1}{N}\sum_{i=1}^{N}\sigma(\mathcal{R}(\mathcal{G}(y_i))), \tag{21}$$

where y_i is an input LR image, $\mathcal{R}(\mathcal{G}(y_i))$ is the ranking score of the generated image $x_{g_i} = \mathcal{G}(y_i)$ and σ denotes the sigmoid function. Note that lower ranking scores imply better perceptual quality and yield the loss closer to 0.

4.5. Cyclic loss ($\mathcal{L}_{\mathcal{CYC}}$)

The Cyclic loss is used with generative adversarial networks that perform unpaired image-to-image translation. It intends to maintain the domain consistency between the LR and HR domains by enforcing forward and backwards consistency, thus reducing the space of possible HR mapping functions.

$$\mathcal{L}_{\mathcal{CYC}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{G}_{LR}(\mathcal{G}_{HR}(y_i)) - y_i\right\|_1. \tag{22}$$

Fundamentally, $\mathcal{L}_{\mathcal{CYC}}$ enforces the intuition that $\mathcal{G}_{HR}$ and $\mathcal{G}_{LR}$ mappings should reverse each other, *i.e.*, they are inverse functions:

$$\begin{aligned}&\mathcal{G}_{LR}(\mathcal{G}_{HR}(y)) \approx y\\ &\mathcal{G}_{HR}(\mathcal{G}_{LR}(x_g)) \approx x_g.\end{aligned} \tag{23}$$

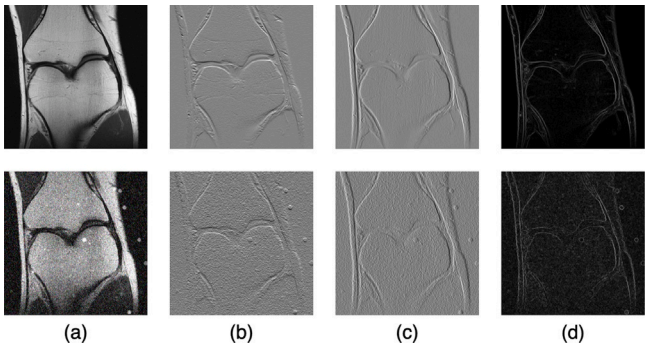


Fig. 14. Image gradients provide information about edges and boundaries within an image. (a) original MRI, (b) vertical gradients, (c) horizontal gradients and (d) gradient magnitudes.

4.6. Best-Buddy loss ($\mathcal{L}_{BB}$)

The Best-buddy loss is employed to alleviate the immutable one-to-one constraint and take into account the inherent uncertainty of SISR. It enables a trustworthy and much more flexible supervision. As a result, generated images do not lack several high-frequency structures unlike images estimated by SR methods that focus on learning the single-LR-single-HR mapping with MSE/MAE loss. It is defined as follows:

$$\mathcal{L}_{BB} = \frac{1}{NP}\sum_{i=1}^{N}\sum_{j=1}^{P}\left\| p_{g_{i,j}} - p_{BB_{i,j}} \right\|_1, \tag{24}$$

where $p_{g_{i,j}}$ represents a fake generated patch from the estimated image x_{g_i} and $p_{BB_{i,j}}$ BB is the corresponding best-buddy patch (the most suitable supervision patch for $p_{g_{i,j}}$ in the current iteration). Moreover, N represents the number of training images and P the number of patches in each image. Essentially, best-buddy loss corresponds to the overall distance between the generated patches and the corresponding selected best-buddy patches.

4.7. Total-variation loss ($\mathcal{L}_{TV}$)

MRI images often suffer from a low signal-to-noise ratio. Super-resolving a noisy LR image results in noisy HR image, as SR leads to spatial noise correlations, *i.e.*, the SR network cannot distinguish noise from useful features, and consequently, the noise is amplified in the generated HR images, hence degrading the resulting image quality. Accordingly, MRIs need to be denoised beforehand or the SR models should manifest rigorous robustness to noise.

Additionally, optimizing the generator network with adversarial and perceptual losses as the main targets can lead to noisy and highly pixelated outputs (Park et al., 2018). Therefore, total-variation loss is introduced to minimize the gradient discrepancy and ensure the spatial continuity and smoothness, thus avoiding noisy and overly pixelated results, while also preserving the sharpness in the generated HR images. It is defined as follows:

$$\mathcal{L}_{TV} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\| \nabla_h G\left(y_i\right) - \nabla_h\left(\hat{x}_{r_i}\right)\right\|_1 + \left\| \nabla_v G\left(y_i\right) - \nabla_v\left(\hat{x}_{r_i}\right)\right\|_1\right), \tag{25}$$

where ∇_h and ∇_v represent the horizontal and vertical gradients, respectively. An image gradient is a directional change in the intensity or color of an image, as shown in Fig. 14.

Horizontal edges can be detected by calculating the vertical gradient and likewise vertical edges can be detected with the horizontal gradient. These gradients can be computed through the following equations:

$$\begin{aligned} \nabla_h I &= I(i, j+1) - I(i, j-1), \\ \nabla_v I &= I(i+1, j) - I(i-1, j), \end{aligned} \tag{26}$$

where $I(i, j)$ represents the pixel value of the grayscale image I in row i and column j. The horizontal gradient ∇_h is calculated by taking the differences between column values and, equivalently, ∇_v is computed by taking the differences between row values. In RGB images, gradients are calculated for each channel separately.

Whether the generator network is fed with noisy LR images or the generator itself introduces noise and artifacts, using noise free ground truth images and total-variation loss will favor the generator to optimize in the direction of results with reduced noise level. As shown in Fig. 14, image gradients will as well detect noise, thus heavily penalizing noisy images that introduce artifacts that are not present in the ground truths.

4.8. Batch normalization ($\mathcal{BN}$)

Training deep neural networks is challenging. These networks suffer from gradient vanishing (Hochreiter, 1998), which happens when the number of layers is increased in the neural network. Thus, the gradient becomes too small, preventing the network from improving. Therefore, BN layers are used to accelerate the training and reduce generalization error by standardizing, for each mini-batch, the inputs fed into a layer. This regularization has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep neural networks.

$$\begin{aligned} \mathcal{BN}(z) &= \gamma \cdot \frac{z - \hat{\mu}_{\mathcal{B}}}{\hat{\sigma}_{\mathcal{B}}} + \beta, \\ \hat{\mu}_{\mathcal{B}} &= \frac{1}{N}\sum_{z\in\mathcal{B}} z, \\ \hat{\sigma}^2_{\mathcal{B}} &= \frac{1}{N}\sum_{z\in\mathcal{B}}\left(z - \hat{\mu}_{\mathcal{B}}\right)^2 + \epsilon, \end{aligned} \tag{27}$$

where N represents the number of inputs in the minibatch $\mathcal{B}$ and $z \in \mathcal{B}$ denotes the input of the batch normalization layer throughout the sample minibatch $\mathcal{B}$. Moreover, $\hat{\mu}_{\mathcal{B}}$ is the sample mean and $\hat{\sigma}_{\mathcal{B}}$ is the sample standard deviation of the minibatch $\mathcal{B}$. The resulting minibatch has zero mean and unit variance and consequently the variable magnitudes for intermediate layers cannot diverge during training, because BN actively centers and rescales them back. Furthermore, γ denotes a elementwise scale parameter and β a shift parameter that have the same shape as input z. Also, a small constant $\epsilon > 0$ is added to the variance estimate to avoid division by zero attempts, for instance when the empirical variance estimate vanishes.

However, BN occasionally introduces artifacts that appear among iterations and different settings (see Fig. 4), thus hampering a stable performance over training. Furthermore, BN layers also enlarges computational complexity and memory usage.

4.9. Spectral normalization ($\mathcal{SN}$)

Spectral Normalization is a regularization technique used to improve the stability and generative quality of GANs, in particular to stabilize the training of the discriminator and consequently improve the generator sample quality. If the discriminator quickly learns to distinguish the real and fake data distributions, then the gradients of the discriminator vanishes and thus it fail to update the generator any further.

To address this problem, SN controls the Lipschitz constant of the discriminator to mitigate exploding and vanishing gradient problems. In essence, SN is added to every hidden layer of the discriminator, thus constraining the spectral norm of each layer $L : h_{in} \to h_{out}$ and limiting the ability of weight matrices W_i to amplify inputs in any direction. By definition, the Lipschitz norm is defined as:

$$\|L\|_{Lip} = \sup_h \sigma(\nabla L(h)), \tag{28}$$

where h represents the input vector h_{in} fed to the layer L and σ denotes the spectral norm given as:

$$\sigma(W) = \max_{h:h\neq 0} \frac{\|Wh\|_2}{\|h\|_2} = \max_{\|h\|_2\leq 1} \|Wh\|_2, \tag{29}$$

which is equivalent to the largest singular value of the matrix W. Therefore, for a linear layer $L_i = W_i \cdot h_{in_i}$, the Lipschitz norm can be written as:

$$\|L\|_{Lip} = \sup_h \sigma(\nabla L(h)) = \sup_h \sigma(W). \tag{30}$$

If the Lipschitz constant of activation functions $\|a_i\|_{Lip} = 1$, then Eq. (30) can be further simplified. Functions commonly used in neural networks, such as ReLU, Leaky ReLU, Sigmoid or Softmax, have Lipschitz norm = 1. Therefore, using them as activation functions in the discriminator architecture allows Eq. (30) to be rewritten as:

$$\|L\|_{Lip} = \sup_h \sigma(W) = \sigma(W). \tag{31}$$

Spectral normalization normalizes the spectral norm of the weight matrix so it satisfies the Lipschitz constraint $\sigma(W) = 1$:

$$\bar{W}_{SN}(W) = \frac{W}{\sigma(W)}. \tag{32}$$

Accordingly, normalizing parameters of each layer with Eq. (29), defined as spectral normalization, will upper bound the Lipschitz constant of the discriminator function by 1. This results from the fact that, for every layer, the following is satisfied:

$$\sigma(\bar{W}_{SN}(W_i)) = 1. \tag{33}$$

5. Implementation details

5.1. ESRGAN

Initially, a PSNR-oriented method is trained with $\mathcal{L}_1$ loss. The learning rate is set to 2×10^{-4} and decayed with a factor of 2 every 2×10^5 iterations. Afterwards, this PSNR-oriented model is employed as the starting point for the ESRGAN generator. The ESRGAN model training is performed with mini-batch size set to 16. The generator is trained with a learning rate of 1×10^{-4} and decayed every 1×10^5 mini-batch updates by a rate of 2. The optimization target of the generator is the loss function in Eq. (7) with $\lambda = 5 \times 10^{-3}$ and $\eta = 1 \times 10^{-2}$. The optimizer employed is Adam (Kingma and Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Implemented with PyTorch framework and trained over DIV2K (Agustsson and Timofte, 2017) and Flickr2K (Timofte et al., 2017) datasets. The testing phase is consummated with MRI image pairs holding an HR image spatial size of 320 × 320 and an LR size of 80 × 80.

5.2. RankSRGAN

Regarding the Ranker network, the small constant ϵ present in the margin-ranking loss function (9) is set to 0.5. The weights are initialized with a method described in He et al. (2015). Moreover, the ranker is trained over DIV2K (Agustsson and Timofte, 2017) and Flickr2K (Timofte et al., 2017) datasets. In detail, an Holdout is employed to split all image samples. This cross validation technique assigns 90% of the data to training and the remaining 10% to validation. For optimization, the Adam optimizer (Kingma and Ba, 2014) is used. The learning rate is set to 1×10^{-3} and is decayed with a factor of 2 every 1×10^5 iterations.

Concerning the RankSRGAN network, the training is carried out with a mini-batch size of 8. The optimization target is defined in Table 1, where $\lambda = 5 \times 10^{-3}$ and $\eta = 3 \times 10^{-2}$. To optimize the network, the Adam optimizer (Kingma and Ba, 2014) is employed with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Both generator and discriminator learning rates are initialized to 1×10^{-4} and halved every 1×10^5 iterations. Implemented with Pytorch and used DIV2K (Agustsson and Timofte, 2017) dataset.

5.3. SRResCycGAN

The training phase is carried out with a batch size of 16 over 51×10^3 iterations. For optimization, the Adam optimizer (Kingma and Ba, 2014) is employed with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and no weight decay. The optimization target is the loss function in Eq. (12). The learning rate is initialized to 1×10^{-4} and decayed with a factor of 2 every 10^4 iterations. Moreover, the network is implemented with Pytorch and it is used the training data provided in the AIM2020 Real Image Super-Resolution (Wei et al., 2020). Ultimately, the estimated noise standard deviation σ (projection layer parameter) is computed according to Liu et al. (2013).

5.4. BSRGAN

Trained with batch size of 48 over a unified dataset including DIV2K (Agustsson and Timofte, 2017), Flick2K (Timofte et al., 2017), WED (Ma et al., 2016) and FFHQ (Karras et al., 2019). BSRGAN is implemented with PyTorch and trained by minimizing a weighted combination of losses, as shown in Table 1, where $\lambda = 0.1$ and $\eta = 1$. For optimization, the Adam optimizer (Kingma and Ba, 2014) is employed with a fixed learning rate of 1×10^{-5}.

5.5. Beby-GAN

Training performed over DIV2K (Agustsson and Timofte, 2017) and Flickr2K (Timofte et al., 2017) datasets. Batch size of 8 for 6×10^5 iterations. The model is optimized via Adam (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rate is initialized to 1×10^{-4} and holds a cosine decay. In every iteration, Eq. (13) to find the best-buddy patch has $\alpha = 1$ and $\beta = 1$. Regarding the binary mask, the threshold σ is fixed to 0.025 and the kernel size is 11 (11 × 11 patch size). The optimization target is the loss function in Table 1, where $\lambda = 0.1$, $\eta = 1$, $\theta = 1$ and $\phi = 5 \times 10^{-3}$. Additionally, it is implemented with PyTorch.

5.6. Real-ESRGAN

Since the same generator architecture from ESRGAN (Wang et al., 2018b) is adopted, then initially a network from ESRGAN is finetuned for faster convergence. Both the generator and discriminator of Real-ESRGAN model are trained for 4×10^5 iterations with Adam (Kingma and Ba, 2014) as optimizer. The learning rate is set to 1×10^{-4} with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and no weight decay. Implemented with PyTorch and trained with images from DIV2K (Agustsson and Timofte, 2017), Flickr2K (Timofte et al., 2017) and OutdoorSceneTraining (Wang et al., 2018a) datasets. For optimization, the equation in Table 1 is minimized, where $\lambda = 0.1$ and $\eta = 1$.

5.7. Correcting GAN noise

Noisy results were a significant problem present in the majority of the methods considered in this work. Besides the noise inherent in the LR images, GANs are prone to introduce noise themselves or even amplify it (see Section 4.7). Therefore, to address this issue a final denoising step was conducted to gently smooth out the generated images. Two approaches were considered: Non-Local Means (Buades et al., 2011) and Block Matching 3D (Dabov et al., 2007).

Non-Local Means (NLM) algorithm replaces the value of a pixel by an average of values from neighbor pixels. Given a generic noisy image I with 3 channels (colored), the estimated value for a pixel p in channel c is computed as a weighted average of all the pixels in a square neighborhood from channel c and centered at p:

$$\begin{aligned} NLM(p,c;I) &= \frac{1}{K(p)} \sum_{b\in B(p,r)} I_c(b) w(p,b), \\ K(p) &= \sum_{b\in B(p,r)} w(p,b), \end{aligned} \tag{34}$$

where $K(p)$ is a normalizing factor and b denotes a pixel from the neighborhood centered at p and with size $(2r+1)\times(2r+1)$. The constant r depends on the standard deviation σ of the noise and it defines the width and height of the search zone. Considering that the pixels most resembling p may not necessarily be spatially close to it and having the intention of denoising p, it is therefore licit to scan a vast portion of the image in search of all the pixels in the channel c that really resemble the reference pixel p. Accordingly, the size of the search window grows (r is increased) for larger values of σ due to the necessity of finding more similar pixels to reduce the noise. Moreover, $c \in [1,2,3]$ and $I_c(p)$ is the value of the pixel p in image I and channel c. Additionally, $w(p,b)$ represents a weight that depends on the similarity between the pixels p and b. The similarity between these two pixels relies on the resemblance between the two square neighborhoods of fixed size and centered at the corresponding pixels, *i.e.*, results from how closely related the image at the point p is to the image at the point b. This resemblance is measured by the squared Euclidean distance of the $(2f+1)\times(2f+1)$ color patches (square neighborhoods) centered respectively at p and b, given as:

$$\begin{aligned} d^2 &= d^2(B(p,f), B(b,f)) = \\ &= \frac{1}{3}\sum_{c=1}^{3}\frac{1}{(2f+1)^2}\,\|B(p,f) - B(b,f)\|_2^2 = \\ &= \frac{1}{12f^2+12f+3}\sum_{c=1}^{3}\sum_{i=1}^{(2f+1)^2}(I_c[B(p,f)](i) - I_c[B(b,f)](i))^2, \end{aligned} \tag{35}$$

where $B(p,f)$ represents a neighborhood centered at pixel p and with size $(2f+1)\times(2f+1)$. Furthermore, $i \in [0,(2f+1)^2]$ and $I_c[B(p,f)](i)$ corresponds to the i-th pixel value of the neighborhood centered at p in image I and channel c. In essence, each pixel value is restored as an average of the most resembling pixels, where this resemblance is computed in the color image. Therefore, for each pixel, each channel value is the result of the average of the same pixels. Ultimately, to compute the weights an exponential kernel is used:

$$w(p,b) = e^{-\frac{\max\left(0,\, d^2-2\sigma^2\right)}{h^2}}, \tag{36}$$

where h is a parameter set depending on the value of σ. It controls the decay of the exponential function and thus the decay of the weights. Neighborhoods with square distances smaller than $2\sigma^2$ have $w(p,b)$ set to 1 and thus the pixel b has a higher influence on the estimated pixel value of p. Meanwhile larger distances decrease rapidly due to the exponential kernel.

Regarding the Block Matching 3D (BM3D) algorithm, it consists of an expansion of the NLM technique and is the current state-of-the-art for image denoising. BM3D is based on the fact that an image has a locally sparse representation in transform domain. This sparsity is enhanced by grouping similar 2D image patches into 3D groups. In detail, blocks are processed within the image in a sliding manner and similar blocks to the currently processed one are searched. The matched blocks are stacked together to form a 3D array and due to the similarity between them, the data in the array exhibit high level of correlation. This correlation is exploited by applying a 3D decorrelation unitary transform and effectively attenuating the noise by shrinkage of the transform coefficients. The subsequent inverse 3D transform yields estimations of all matched blocks. After iteratively repeating this procedure for all image blocks, the final estimate is computed as weighted average of all overlapping block-estimates.

In this work NLM was conducted with filter strength $h = 4$, search zone size equal to 51×51 ($r = 25$) and with color patches of size 5×5 ($f = 2$). BM3D performed both hard-thresholding and wiener filtering with noise standard deviation $\sigma = \frac{1}{28}$.

Real-ESRGAN and BSRGAN did not have any problem with noise as their results were already excessively smooth. Therefore, the denoising step was discarded in these methods. Furthermore, to correct some minor color issues in ESRGAN, a grayscale step was carried out before conducting the denoising step.

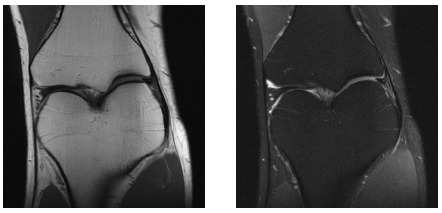


Fig. 15. A proton-density weighted image without fat suppression (a) and with fat suppression (b).

6. Experimental setup

6.1. Dataset

The FastMRI dataset (Zbontar et al., 2018; Knoll et al., 2020) was employed to test the GAN methods discussed in the previous section. FastMRI is a large-scale release of raw MRI data that can be used to train and evaluate machine learning approaches for MRI reconstruction and acceleration. It consists of two collections: knee MRIs and brain MRIs. Each collection is split into training, validation, and downsampled/masked test sets. Considering both collections and all splits, FastMRI contains a total of 8344 MRI volumes, corresponding to 167.375 slices, where each slice corresponds to one 2D image and is represented by the k-space data and the corresponding ground truth.

The dataset includes data from multiple modalities with different contrasts. Additionally, two pulse sequences were used, yielding coronal proton-density weighting with and without fat suppression, as shown in Fig. 15. Fat suppression is commonly used in MRI to suppress the signal from adipose tissue (body fat) and make details in regions covered by fat easier to perceive.

This paper only considers the knee collection, where 973 volumes were used from the single-coil knee training set. At the beginning of the test phase the k-space data of every MRI slice is converted into the final MRI slice. An Inverse Fast Fourier Transform (Cooley and Tukey, 1965) is applied on the k-space data resulting in an image of complex numbers. Following, the absolute values of the complex image are computed to derive the final ground truth MRI slice. To evaluate the SR performance it is necessary to formulate LR-HR image pairs. Consequently, a preprocessing step is employed to simulate the degradation inherent to MRI acquisition under few measurements. Accordingly, each ground truth MRI slice is downsampled through bicubic interpolation with a downscale factor of ×4.

6.2. Image quality metrics

Several metrics are used to evaluate models' performances quantitatively. Additionally, inspired by these metrics, alternative loss functions can be formulated to encourage results that yield higher metric scores or favor specific image characteristics.

6.2.1. Mean Squared Error (MSE)

Among the many metrics used to evaluate the HR image quality, Mean Squared Error (MSE) is the most popular one. It is computed by averaging the pixel-wise squared differences between the generated HR image and the corresponding ground truth. The MSE between two images is given as follows:

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(\hat{x}_r(i,j) - x_g(i,j))^2, \tag{37}$$

where W denotes the image width and H the image height. Moreover, $(i,\ j)$ define the pixel position, while $\hat{x}_r$ and x_g represent the ground truth and generated HR images, respectively. Evidently, both images

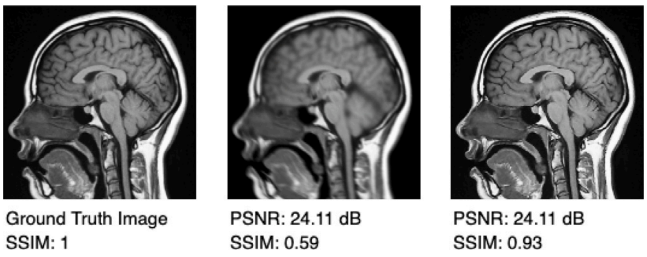


Fig. 16. PSNR values can be misleading as they do not consider an image structural composition, which is, adversely, well perceived by human vision.
Source: The ground truth MRI image in this illustration is sourced from Gaillard.

must share the same size. A few variants can be derived from MSE, such as the Root MSE (RMSE), which is simply the square root of the MSE, and is measured in the same units as the pixel values of the images. Therefore, the interpretation of RMSE is more straightforward than MSE.

6.2.2. Peak Signal-to-Noise Ratio (PSNR)

The Peak Signal-to-Noise Ratio (PSNR) is commonly used to measure the reconstruction quality, and is inversely proportional to the logarithm of the MSE between the ground truth and the HR generated image. PSNR is expressed as:

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX_I}{RMSE(\hat{x}_r, x_g)}\right), \tag{38}$$

where MAX_I corresponds to the maximum possible pixel value, for instance, 255 regarding 8-bit images. Generally, a higher PSNR value suggests a better reconstruction quality. However, PSNR can sometimes be misleading, as images with visually unsatisfying dissimilarities sometimes hold a high PSNR score (see example in Fig. 16).

This results from the poor correlation between pixel-wise differences and human perception of image quality. Both MSE and PSNR are highly correlated with the pixel-to-pixel differences, thus occasionally leading to blurry, overly smooth, and unnatural images due to loss of high-frequency information.

6.2.3. Structural similarity index measure (SSIM)

MSE and PSNR do not consider the image structural composition, which is, adversely, well perceived by human vision. Therefore, to quantify the structural similarity between two images (Wang et al., 2004) introduced the Structural Similarity Index Measure (SSIM). SSIM is based on luminance, contrast, and changes in structural information. The key idea behind considering structural information changes is that pixels are strongly correlated especially when they are spatially close. Additionally, MSE and PSNR estimate absolute errors, while SSIM gives perception and saliency-based errors (Sara et al., 2019). Evidently, from a human visual perspective, SSIM is comparatively better than MSE and PSNR. SSIM can be defined as follows:

$$SSIM = \frac{\left(2\mu_{\hat{x}_r}\mu_{x_g} + c_1\right)\left(2\sigma_{\hat{x}_r x_g} + c_2\right)}{\left(\mu^2_{\hat{x}_r} + \mu^2_{x_g} + c_1\right)\left(\sigma^2_{\hat{x}_r} + \sigma^2_{x_g} + c_2\right)}, \tag{39}$$

where $\mu_{\hat{x}_r}$ and μ_{x_g} represent the means of the ground truth and the generated HR image, respectively. Accordingly, $\sigma_{\hat{x}_r}$ and σ_{x_g} are the standard deviations of $\hat{x}_r$ and x_g. Moreover, $\sigma_{\hat{x}_r x_g}$ denotes the covariance between both images, while c_1 and c_2 are constants set to avoid instability (Wang et al., 2004).

6.2.4. Pixel value deviation

If every pixel value or the majority of them in the generated image are equally shifted by a constant, then the reconstruction quality is substantially affected. Therefore, it is meaningful to check for deviations in pixel values such as a fixed constant added to every pixel or pixel values irregularities within a section of the generated image.

To detect anomalies in the fake images two measures can be computed, namely the mean pixel value (MPV) of the real/generated image sets ($\mathcal{X}_r = \{I \mid I \text{ is a real image}\}$ and $\mathcal{X}_g = \{I \mid I \text{ is a fake image}\}$) and the mean pixel value difference (MPVD) between the generated images $\mathcal{X}_g$ and their corresponding ground truths $\mathcal{X}_r$:

$$MPV(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} I(i,j), \tag{40}$$

$$MPVD(\mathcal{X}_r, \mathcal{X}_g) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\left|\hat{x}_r(i,j) - x_g(i,j)\right|. \tag{41}$$

6.3. Pre-trained models

A common practice when training GANs is to use pre-trained models to initialize the optimization process. This typically results in higher performance compared to training from scratch (Grigoryev et al., 2022), especially in limited-data regimes like medical applications. For instance, the field of MRI reconstruction still lacks large public datasets. Accordingly, a pre-trained image SR network, that has already learned to extract powerful and informative features from natural images, can be used as a starting point or even borrowed to carry out the whole task.

Reasoning, the majority of pre-trained models were trained over diverse data from exhaustive datasets, thus they learn to estimate the distribution of real-world images holding photo-realistic details. Therefore, pre-trained models are herein applied directly in the reconstruction task. In essence, the training phase is skipped with the idea that the robustness of each pre-trained model will be evaluated by performing the target task. A robust model trained over diverse data would be able to generalize effectively in the SR task of medical images. The training details for each pre-trained model present in this work are outlined in Section 5. Afterwards, the architecture of the pre-trained model that manifests the best results, overall, will be trained over FastMRI, where the pre-trained model is used to initialize the optimization process. Essentially, it is employed a strategy that consummates a selection criteria by following the idea that through pre-trained models it is possible to estimate the architecture that better fits the problem.

7. Experimental results

7.1. Quantitative results

All experiments were conducted on Google Colab using an Intel Xeon CPU with 2.20 GHz and 13 GB of RAM. For every method the input LR images were obtained with bicubic downsampling and a scaling factor of ×4. Upscaling results for the different methods are presented in Table 2, where the Time (ms) column shows the average time in milliseconds spent to reconstruct an 80 × 80 degraded MRI slice into a HR one with size 320 × 320.

MSE, PSNR, and SSIM suggest SRResCycGAN outperforms every other GAN-based method in recovering ×4 downgraded images. Meanwhile, ESRGAN obtained the worst results both in metrics and image generation time. RankSRGAN holds the fastest reconstruction time followed by SRResCycGAN. Looking at Table 1 it is evident that methods with less parameters in the generator have as well a faster reconstruction time. The aforementioned is also illustrated in Fig. 17.

Table 2
Comparison of different SR methods coupled with denoising. Since Real-ESRGAN and BSRGAN results were already excessively smooth, denoising was not applied. The input images are downscaled 4×with a bicubic interpolation and different methods are used to recover the original upscaled image. Red color indicates the worst performance overall and Green color the best. Gray color stands for the additional time (regarding original SR method) derived from the denoise step.

Method	MSE	PSNR	SSIM	Time (ms)
ESRGAN	297.46	24.47	0.5939	4417
↳ with NLM	283.30	24.82	0.6585	6574 (+2157)
↳ with BM3D	252.28	25.58	0.7286	10600 (+6183)
RankSRGAN	266.94	24.99	0.6319	651
↳ with NLM	250.93	25.44	0.7057	2731 (+2080)
↳ with BM3D	235.78	25.89	0.7392	7371 (+6720)
SRResCycGAN	228.00	25.94	0.7456	2602
↳ with NLM	227.32	25.98	0.7459	4780 (+2178)
↳ with BM3D	231.48	25.92	0.7442	8983 (+6381)
BSRGAN	254.11	25.33	0.7157	3652
Beby-GAN	264.76	25.11	0.6493	3819
↳ with NLM	251.02	25.50	0.7140	5853 (+2134)
↳ with BM3D	236.78	25.91	0.7439	10113 (+6294)
Real-ESRGAN	274.40	24.99	0.7137	3715

Table 3
Pixel value statistics for different SR and denoising methods. Since Real-ESRGAN and BSRGAN results were already excessively smooth, denoising was not applied. Red color indicates the largest deviation (worst case) and Green color the smallest (best case). Input images were downsampled with a ×4 scale factor.

Method	Global MPV †	Global MPVD ‡	Local MPV †	Local MPVD ‡
GROUND TRUTH	**55.90**	**0.00**	**47.53**	**0.00**
ESRGAN	50.18	12.36	46.77	11.30
↳ with NLM	50.08	11.82	46.78	10.75
↳ with BM3D	49.62	11.05	46.23	9.62
RankSRGAN	51.29	11.50	47.88	10.29
↳ with NLM	51.19	10.86	47.87	9.62
↳ with BM3D	50.77	10.52	47.35	9.11
SRResCycGAN	51.60	10.27	48.29	8.67
↳ with NLM	51.56	10.19	48.37	8.56
↳ with BM3D	51.10	10.32	47.78	8.55
BSRGAN	49.95	11.20	46.70	9.63
Beby-GAN	50.84	11.49	47.58	10.23
↳ with NLM	50.75	10.96	47.59	9.58
↳ with BM3D	50.32	10.64	47.05	9.04
Real-ESRGAN	49.02	11.52	44.93	9.93

† MPV — Mean pixel value of generated/ground truth image.
‡ MPVD — Mean pixel value difference between generated and ground truth images.

7.1.1. Denoising improvements

Table 2 and Fig. 18 evaluate the impact of the denoising step. In particular, Table 2 shows that the denoising step improved the performance of the SR task by ≈ 20% on the SSIM metric. This result is complemented with Fig. 18, which allows visualizing the noise correction and the reduction of the checkerboard artifact pattern, common in GANs, which results from the upsampling and downsampling layers (Kinoshita and Kiya, 2020). For instance, deconvolution layers reverse the convolution operation, however they may introduce upsampling artifacts (Odena et al., 2016). Additionally, decreasing the spatial resolution of an image can result in a checkerboard pattern since details are lost. Reasoning, the discriminator ability to detect images containing checkerboard artifacts and to consider them as fake sustains substantial value by further aligning the generator in the direction of photo-realistic textures.

To further analyze the obtained results, the pixel value deviation metrics (see also Section 6.2.4) were computed over the entire image (Globally) or over a central section (Locally) of such image with a crop factor of 0.2, *i.e.*, for an image of size $W \times H$ (here with $W = H = 320$), cropping was performed across the center section to obtain a patch of size $0.2W \times 0.2H$.

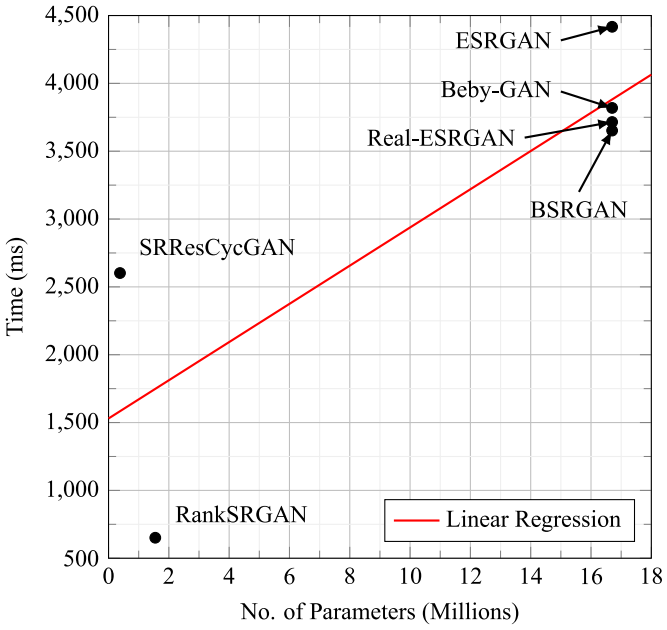


Fig. 17. Correlation between the number of parameters and the time required to reconstruct MRIs with a ×4 scale factor.

The results presented in Table 3 show that the denoising step reduces pixel value differences, suggesting that images become closer to their corresponding ground truths. This is particularly evident when observing the MPVD metric, which provides a more accurate assessment since it is analogous to the MSE, replacing the l^2-norm by an l^1-norm.

7.1.2. Reconstruction errors in high-frequency details

An additional evaluation was considered by visualizing pixel value differences, particularly by plotting the differences between the generated images and the corresponding ground truths. One of such plots is presented in Fig. 19 for the Real-ESRGAN.

Such figures allowed to discard problems related to pixel value shifting (bias) across the whole image. Such plots also discard errors due to pixel position misalignment, as this would correspond to a salient outline on the image edges when such differences are displayed. Hence, this suggests that the MPVD observed in Table 3 is a consequence of a non optimal reconstruction, particularly associated with high-frequency details.

7.2. Qualitative results

A qualitative analysis was also conducted to further evaluate the MRI reconstruction quality of GANs. Fig. 20 shows a comparative illustration of an MRI slice reconstructed over each SR method. Within each line, it is employed the super-resolved MRI version under different denoising scenarios, as well as the corresponding LR and Ground Truth images, as a means to ease side-by-side comparison. Unlike quantitative measures, visual examples advocate that Beby-GAN and RankSRGAN present the best perceptual quality. Furthermore, Beby-GAN has slightly less noise and fewer checkerboard artifacts, thus outmatching the performance of RankSRGAN. Fig. 21 presents a visual comparison between an LR MRI, its reconstruction through an MRI-specific trained Beby-GAN model, and the corresponding ground truth (see Section 7.3).

These results also show that standard quantitative measures (MSE, PSNR, and SSIM) fail to truly capture and accurately evaluate image

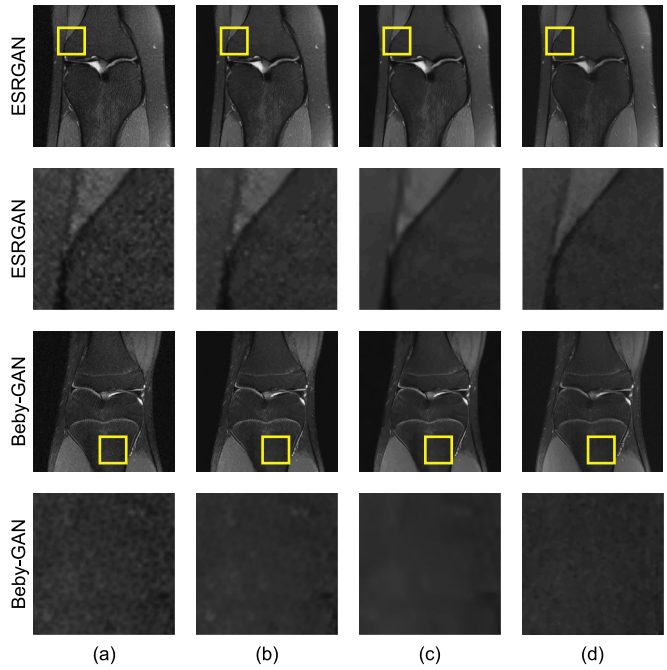


Fig. 18. MRI noise correction and checkerboard artifact mitigation as a result of the denoising process. Images in lines 2 and 4 represent zoomed-in sections of the image found in the corresponding column of the row above. (a) Generated Images without denoising, (b) Generated Images with NLM, (c) Generated Images with BM3D and (d) Ground Truth Images.

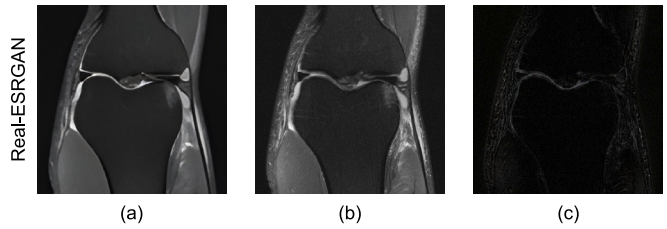


Fig. 19. Image generated by Real-ESRGAN without denoising (a), ground truth (b), and corresponding difference (c). Pixel value shifting and position misalignment are absent since the pixel value differences between the generated image and the corresponding ground truth are negligible.

quality with respect to the human visual perception. Although SRResCycGAN has better scores over quantitative metrics, it is evident that the method still manifests some blur and lack of high-frequency details (see Figs. 22 and 24). Additionally, in these comparative experiments, Real-ESRGAN and BSRGAN exhibit overly smooth results, where high-frequency information and rich textures are missing. Nonetheless, the generated images hold sharp edges and an overall good quality.

Figs. 22 and 23 comparatively illustrate the reconstruction quality, exposing that generated images have sharper edges and richer textures. Looking at results, NLM looks slightly better than BM3D as it is perceptually closer to the ground truth. The reason is that BM3D is excessive for the current noise level, thus it over smooths details. Therefore, in Fig. 24, it is shown a comparative reconstruction evaluation between LR, ground truth and generated patches when applying NLM for denoising (except for Real-ESRGAN and BSRGAN, since the generated images are already overly smooth). The Figure suggests that SRResCycGAN patches are relatively closer to the LR patches compared to other methods. Accordingly, Beby-GAN, RankSRGAN, and ESRGAN manifest adequate results and allude that high-frequency details are recovered. Moreover, SISR methods are usually sensitive to errors in the blur kernel. This is possibly the main reason Real-ESRGAN and BSRGAN

Table 4
Results of Beby-GAN trained over FastMRI and comparison with a standard model trained on ImageNet. The input images are downscaled 4×with a bicubic interpolation. The Green line indicates the best results and the last column indicates the total reconstruction time (inference), with the value under parenthesis specifying the time of the denoising algorithm.

Method	MSE	PSNR	SSIM	Time (ms)
Original network (trained on ImageNet)				
Beby-GAN	264.76	25.11	0.6493	3819
↳ with NLM	251.02	25.50	0.7140	5853 (+2134)
↳ with BM3D	236.78	25.91	0.7439	10113 (+6294)
With MRI-specific training				
Beby-GAN	183.51	26.61	0.7134	3407
↳ with NLM	172.99	27.02	0.7628	5382 (+1975)
↳ with BM3D	**166.70**	**27.30**	**0.7711**	22212 (+18805)

are producing overly smooth results, as they are assuming a higher level of degradation in the LR images, which is not present.

It should be noticed that the radio frequency (RF) spiking artifacts present in MRI ground truths of Fig. 23 were diminished by the degradation process. Consequently, these artifacts are not perceptible in the LR images, nor are they fully recovered when applying SR, particularly in foreground voxels, thereby showing that the SR techniques can avoid recovering specific information that should not be reinstated.

7.3. Trained Beby-GAN

Taking the previous discussion under consideration, Beby-GAN was selected as the best model for further evaluation. Hence, in the follow-up experiment it is fully trained during 10.000 epochs (with weights initialized from an ImageNet pre-trained model) to assess whether MRI-specific training can further improve the results. To assess the performance, for every MRI volume the 20-th slice is put aside to formulate a testing split. All the remaining slices were used for training.

Fig. 21 presents a visual comparison between a low-resolution MRI and the corresponding reconstruction from the trained Beby-GAN model. It is noticeable the recovery of high-frequency details in conjunction with an accurate inference of anatomical information.

Additionally, Table 4 presents the quantitative assessment of the Beby-GAN model, considering two cases: trained with standard ImageNet (as in Table 2) and trained on FastMRI. Despite the models having been trained intensively over diversified data, it is evident that the generalization capability of SR GANs over wide data domains has some drawbacks. Beby-GAN manifests greater performance in the MRI reconstruction task when the training data domain space is narrowed, i.e., when it is trained exclusively with MRIs. This is an example that reducing the variance and domain of the dataset can sometimes be beneficial. Evidently, it depends highly on the target task, for instance, in exclusive-MRI processing it is preferable to discard non-MRI training samples. However, for tougher tasks that require a finer generalization capability might be appropriate to consider domains with distinct medical image types/modalities.

Furthermore, the parameters of both denoising algorithms were identical to the previous experiments, despite of the noise present in the generated images being significantly reduced. In particular, on the SSIM metric, an improvement of $\approx$ 8% is attained. Intriguingly, the BM3D processing time increased threefold over the images generated by the trained Beby-GAN. Nonetheless, this denoising processing time remains negligible when compared with the extensive data acquisition times required by MRI scans. Regarding NLM processing time, there was no discernible difference.

Fig. 20. Qualitative analysis of several GAN architectures on MRI reconstruction. Within each line, it is employed the super-resolved MRI version under different denoising scenarios, as well as the corresponding LR and Ground Truth images. (a) Input LR Images, (b) Generated Images without denoising, (c) Generated Images with NLM, (d) Generated Images with BM3D and (e) Ground Truth Images.

7.4. Discussion

Perceptual-driven approaches focus on feature distribution and high-level representations rather than merely comparing pixel values. Using perceptual loss as a term in the loss function will encourage the generation of complex textures or accurate structures depending on the depth of the layers in the VGG network considered. Early layers capture low-level spatial information, such as edges, blobs and textures. Proceeding deeper in the network, layers start to learn less about fine-grained spatial details and more about features with global semantic meanings and abstract object information. Using a feature reconstruction loss encourages the output to be perceptually similar to the ground truth, but does not force them to match exactly. Accordingly, this can be misleading in the medical image processing context. For instance, in MRI reconstruction the synthesized MRI may look perceptually pleasing, but not equal to the ground truth. A similar issue occurs in

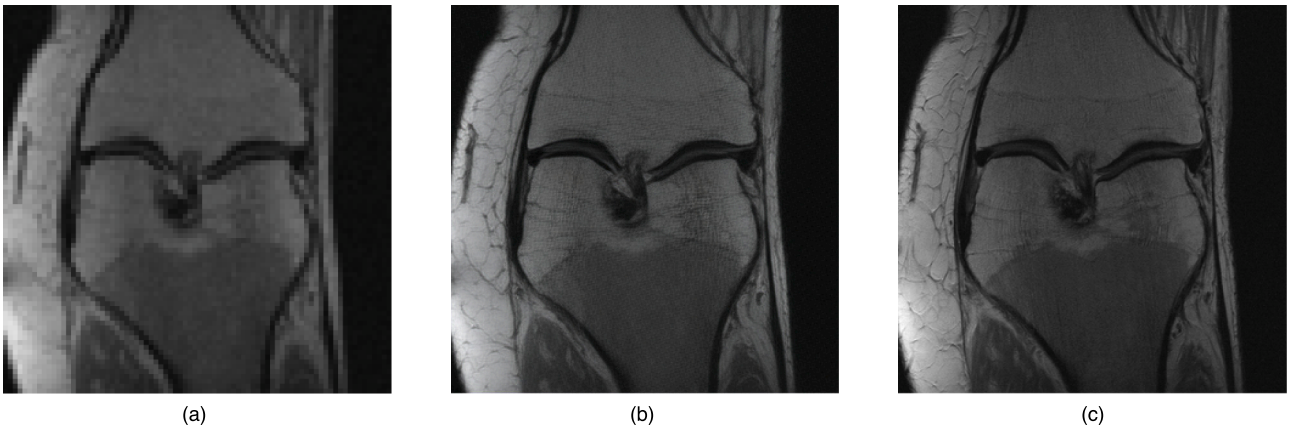


Fig. 21. High-frequency details recovery of Beby-GAN trained on FastMRI with weights initialized from an ImageNet pre-trained model. (a) Input LR Image, (b) Beby-GAN Generated Image without the denoising step and (c) Ground Truth Image.

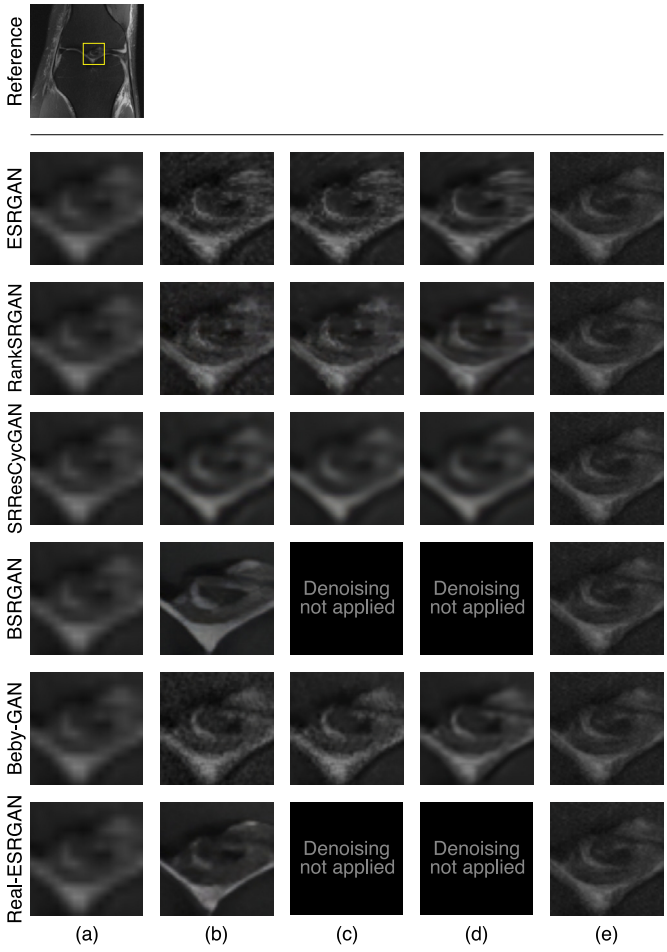


Fig. 22. Within each line, it is employed the super-resolved MRI patch version under different denoising scenarios, as well as the corresponding LR and Ground Truth images. (a) Input LR Images, (b) Generated Images without denoising, (c) Generated Images with NLM, (d) Generated Images with BM3D and (e) Ground Truth Images. Despite SRResCycGAN achieving better scores over MSE and PSNR, it manifests more blur and lacks high-frequency details when compared to Beby-GAN, RankSRGAN and ESRGAN. Real-ESRGAN and BSRGAN exhibit overly smooth results.

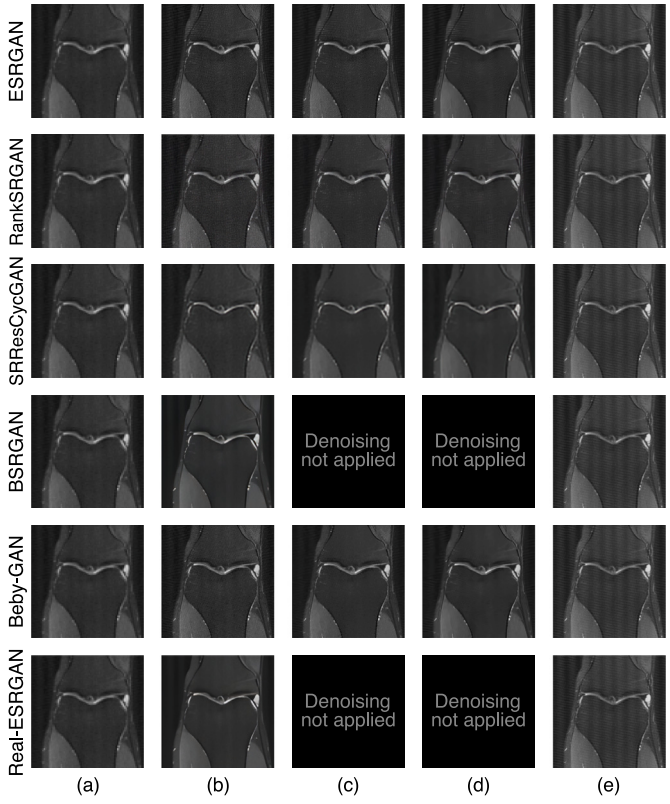


Fig. 23. Within each line, it is employed the super-resolved MRI version under different denoising scenarios, as well as the corresponding LR and Ground Truth images. (a) Input LR Images, (b) Generated Images without denoising, (c) Generated Images with NLM, (d) Generated Images with BM3D and (e) Ground Truth Images. GAN-based reconstructed images manifest sharp edges and rich textures. The radio frequency spiking artifacts present in MRI ground truths were diminished by the degradation process.

adversarial training with GANs, usually used to attain photo-realism. The discriminator predicts relative realness instead of the absolute value, thus favoring results residing on the manifold of natural images. Consequently, realistic fake patterns can be wrongly conjectured as real and authentic even if they are far from the ground truth. This dissimilarity due to artifacts inclusion or omissions of relevant details can lead to erroneous conclusions in healthcare. Nonetheless, the mapping function that perfectly recovers the target image is challenging to estimate, since the reconstruction problem is inherently ill-posed, *i.e.*, for any distorted image there can be multiple plausible solutions

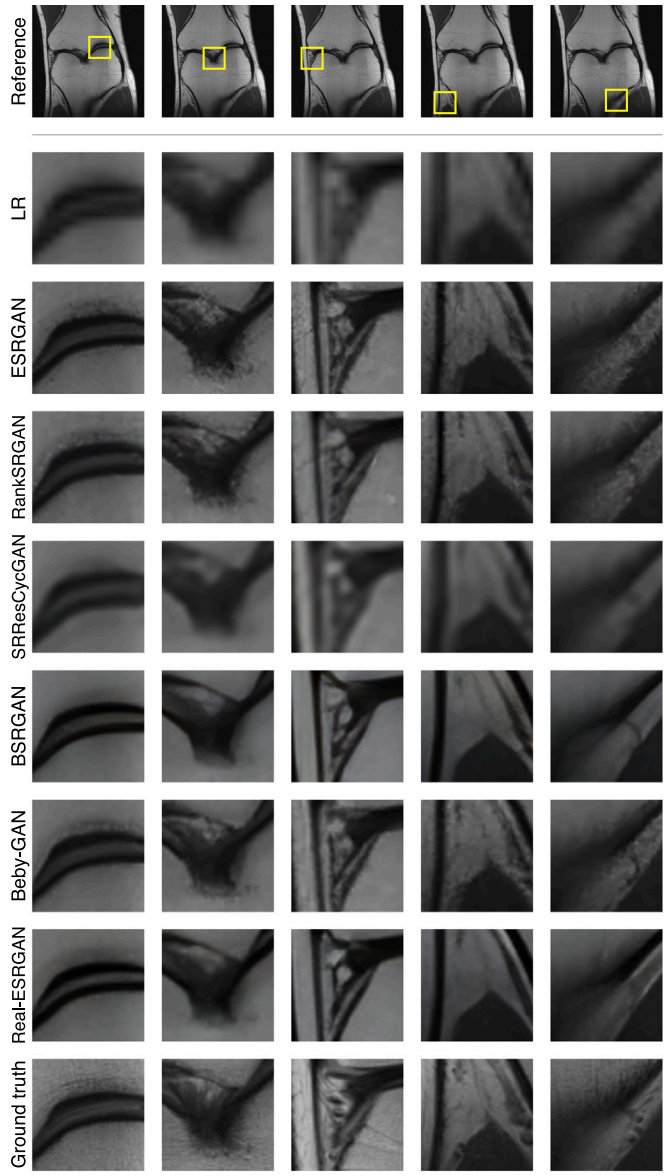


Fig. 24. Comparison of patches from images generated by GAN-based methods with NLM denoising technique (except for BSRGAN and Real-ESRGAN). Within each line, a different reconstruction method is employed. Patches represent zoomed-in sections from reconstructions of the image found in the corresponding column of the first row.

that would be perceptually pleasing. Therefore, GANs remain a solid candidate to spatially resolve MRIs and accelerate their acquisition.

Additionally, optimizing to the content loss usually leads to unnatural and overly smooth reconstructions with low perceptual quality. In contrast, the distortion-based performance is improved, since they focus on minimizing pixel-wise errors (see Section 4). Alternatively, focusing on the adversarial loss leads to a perceptually better reconstruction, but as aforementioned it tends to decrease the distortion-based quality. Therefore, finding a balance between both optimization targets is the best option. Nonetheless, it is evident that the ideal loss function depends on the application where SR is employed. For example, approaches that tend to generate artifacts are less suited for medical applications.

Since most methods assume a bicubic downsampling kernel, they might fail with real degraded images. The reason is that blur kernels play a vital role when used to train SISR methods, however they are way too basic. Inaccurate degradation estimations will inevitably result in artifacts. The real complex degradations usually come from complicate combinations of different degradation processes, therefore an high-order model to mimic the real-world degradation process would sustain significant value. Enlarging the degradation space covered by the degradation model will improve SR practicability. Moreover, SISR Models could see a boost in robustness and performance if they were trained under data degraded by this high-order model rather than degraded by simple synthetic degradations. Even if the super-resolver performs worse for unrealistic bicubic downsampling, it is still a preferable choice for real SISR.

Methods manifest greater performance in the MRI reconstruction task when trained exclusively with MRIs. This suggests that differences on training and testing data domains have impact on the results. Following the aforementioned and considering the image preprocessing adopted in this work, the models used in the experiments would produce worse and visually unpleasant results if the they were trained with LR images computed by either simple or complex degradations far from bicubic downsampling.

Ultimately, the choice of GAN-based models for SR in this study was based on a careful state-of-the-art review, considering the best and most promising methods, as well as their recognition in the literature. As a consequence, most of the GAN models explored in this study drew inspiration from the traditional SRGAN model architecture (see Section 3.1) introduced by Ledig et al. (2017). However, this may cause an unintentional bias, which may be tackled in new approaches resulting from alternative research paths.

8. Conclusions

In routine clinical practice, knee MRI scans typically range from 20 to 40 min, as longer acquisition times can lead to compromised image quality due to motion artifacts. This timeframe can vary greatly depending on the specific region of interest (ROI) and patient circumstances. While it is true that acquiring a reduced amount of k-space data can shorten the acquisition time, it results in MRI images with relatively low spatial resolution. Furthermore, the images obtained from computed tomography (CT), magnetic resonance imaging, or any other medical imaging technique often have low resolution, inherent noise, and lack of structural information in which it becomes a big challenge in the medical field to make a correct diagnosis judgment. This work has proven that high-frequency details can be recovered from Low-Resolution signals and GAN-based SR has the potential to quarter the acquisition time (not considering the negligible period of time to reconstruct the MRI, which does not affect the patient in any manner). Therefore, SISR GAN-based techniques are promising CS-MRI reconstruction methods, enabling resolution improvements, zooming into images and accelerating data acquisition. Additionally, denoising solutions led to performance boosts on the SR task, with manifested reduction of the checkerboard pattern inherent to GAN synthesis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All experiments were conducted using open-data repositories. The models were taken from the authors official repositories and are used as there stated

Acknowledgments

This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under projects PCIF/MPG/0051/2018 and EXPL/CCI-COM/0656/2021 and through the research units INESC-ID (ref. UIDB/50021/2020) and LASIGE (ref. UIDB/00408/2020 and UIDP/00408/2020).

References

Agustsson, E., Timofte, R., 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 126–135.

Anwar, S., Khan, S., Barnes, N., 2020. A deep journey into super-resolution: A survey. ACM Comput. Surv. 53, 1–34.

Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L., 2018. The 2018 pirm challenge on perceptual image super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops.

Buades, A., Coll, B., Morel, J.M., 2011. Non-local means denoising. Image Process. Line 1, 208–212.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 89–96.

Carey, W.K., Chuang, D.B., Hemami, S.S., 1999. Regularity-preserving image interpolation. IEEE Trans. Image Process. 8, 1293–1297.

Chen, Y., Christodoulou, A.G., Zhou, Z., Shi, F., Xie, Y., Li, D., 2020. MRI super-resolution with gan and 3d multi-level densenet: smaller, faster, and better. arXiv preprint arXiv:2003.01217.

Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine calculation of complex fourier series. Math. Comput. 19, 297–301.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process. 16, 2080–2095.

Duchon, C.E., 1979. Lanczos filtering in one and two dimensions. J. Appl. Meteorol. Climatol. 18, 1016–1022.

Funk, E., Thunberg, P., Anderzen-Carlsson, A., 2014. Patients' experiences in magnetic resonance imaging (mri) and their experiences of breath holding techniques. J. Adv. Nurs. 70, 1880–1890.

Gaillard, Frank, Normal brain (MRI) | Radiology Case | Radiopaedia.org, Normal brain (MRI) | Radiology Case | Radiopaedia.org. https://radiopaedia.org/cases/normal-brain-mri-6.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Adv. Neural Inf. Process. Syst. 27.

Greenspan, H., Oz, G., Kiryati, N., Peled, S., 2002. MRI inter-slice reconstruction using super-resolution. Magn. Reson. Imaging 20, 437–446.

Grigoryev, T., Voynov, A., Babenko, A., 2022. When, why, and which pretrained gans are useful? arXiv preprint arXiv:2202.08937.

Gupta, R., Sharma, A., Kumar, A., 2020. Super-resolution using gans for medical imaging. Procedia Comput. Sci. 173, 28–35.

Han, D., 2013. Comparison of commonly used image interpolation methods. In: Conference of the 2nd International Conference on Computer Science and Electronics Engineering. ICCSEE 2013, Atlantis Press, pp. 1556–1559.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hochreiter, S., 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6, 107–116.

Hüsem, H., Orman, Z., 2020. A survey on image super-resolution with generative adversarial networks. Acta Infologica 4, 139–154.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1125–1134.

Jackson, E.F., Ginsberg, L.E., Schomer, D.F., Leeds, N.E., 1997. A review of MRI pulse sequences and techniques in neuroimaging. Surg. Neurol. 47, 185–199.

Johnson, J., Alahi, A., Fei-Fei, L., 2016. Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision. Springer, pp. 694–711.

Jolicoeur-Martineau, A., 2018. The relativistic discriminator: a key element missing from standard gan. arXiv preprint arXiv:1807.00734.

Karras, T., Laine, S., Aila, T., 2019. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4401–4410.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kinoshita, Y., Kiya, H., 2020. Checkerboard-artifact-free image-enhancement network considering local and global features. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. APSIPA ASC, IEEE, pp. 1139–1144.

Knoll, Florian, Zbontar, Jure, Sriram, Anuroop, Muckley, Matthew J, Bruno, Mary, Defazio, Aaron, Parente, Marc, Geras, Krzysztof J, Katsnelson, Joe, Chandarana, Hersh, et al., 2020. fastMRI: A publicly available raw k-space and DICOM dataset of knee images for accelerated MR image reconstruction using machine learning. Radiology: Artificial Intelligence 2 (1), e190007.

Lauterbur, P.C., 1973. Image formation by induced local interactions: examples employing nuclear magnetic resonance. Nature 242, 190–191.

Ledig, C., Theis, L., Huszár, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4681–4690.

Li, X., Wu, Y., Zhang, W., Wang, R., Hou, F., 2020. Deep learning methods in real-time image super-resolution: a survey. J. Real-Time Image Process. 17, 1885–1909.

Li, W., Zhou, K., Qi, L., Lu, L., Jiang, N., Lu, J., Jia, J., 2021. Best-buddy gans for highly detailed image super-resolution. arXiv preprint arXiv:2103.15295 2.

Lim, B., Son, S., Kim, H., Nah, S., Lee, K.Mu., 2017. Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 136–144.

Liu, X., Tanaka, M., Okutomi, M., 2013. Single-image noise level estimation for blind denoising. IEEE Trans. Image Process. 22, 5226–5237.

Liu, T.Y., et al., 2009. Learning to rank for information retrieval. Found. Trends Inf. Retr. 3, 225–331.

Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L., 2016. Waterloo exploration database: New challenges for image quality assessment models. IEEE Trans. Image Process. 26, 1004–1016.

Ma, C., Yang, C.Y., Yang, X., Yang, M.H., 2017. Learning a no-reference quality metric for single-image super-resolution. Comput. Vis. Image Underst. 158, 1–16.

Mahapatra, D., Bozorgtabar, B., Garnavi, R., 2019. Image super-resolution using progressive generative adversarial networks for medical image analysis. Comput. Med. Imaging Graph. 71, 30–39.

Marques, J.P., Simonis, F.F., Webb, A.G., 2019. Low-field MRI: An mr physics perspective. J. Magn. Reson. Imaging 49, 1528–1542.

Mittal, A., Soundararajan, R., Bovik, A.C., 2012. Making a completely blind image quality analyzer. IEEE Signal Process. Lett. 20, 209–212.

Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y., 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. Distill 1, e3.

Park, S.J., Son, H., Cho, S., Hong, K.S., Lee, S., 2018. Srfeat: Single image super-resolution with feature discrimination. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 439–455.

Rahim, A.N.A., Yaakob, S.N., Ngadiran, R., Nasruddin, M.W., 2015. An analysis of interpolation methods for super resolution images. In: 2015 IEEE Student Conference on Research and Development. SCOReD, IEEE., pp. 72–77.

Sara, U., Akter, M., Uddin, M.S., 2019. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. J. Comput. Commun. 7, 8–18.

Schonfeld, E., Schiele, B., Khoreva, A., 2020. A u-net based discriminator for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8207–8216.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Tian, C., Zhang, X., Lin, J.C.W., Zuo, W., Zhang, Y., 2022. Generative adversarial networks for image super-resolution: A survey. arXiv preprint arXiv:2204.13620.

Timofte, R., Agustsson, E., Gool, L.Van., Yang, M.H., Zhang, L., 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 114–125.

Umer, R.M., Foresti, G.L., Micheloni, C., 2020. Deep generative adversarial residual convolutional networks for real-world super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 438–439.

Umer, R.M., Micheloni, C., 2020. Deep cyclic generative adversarial residual convolutional networks for real image super-resolution. In: European Conference on Computer Vision. Springer, pp. 484–498.

Vaishali, S., Rao, K.K., Rao, G.S., 2015. A review on noise reduction methods for brain MRI images. In: 2015 International Conference on Signal Processing and Communication Engineering Systems. IEEE, pp. 363–365.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13, 600–612.

Wang, X., Xie, L., Dong, C., Shan, Y., 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1905–1914.

Wang, X., Yu, K., Dong, C., Loy, C.C., 2018a. Recovering realistic texture in image super-resolution by deep spatial feature transform. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 606–615.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Loy, C.Change., 2018b. Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops.

Wei, P., Lu, H., Timofte, R., Lin, L., Zuo, W., Pan, Z., Li, B., Xi, T., Fan, Y., Zhang, G., et al., 2020. Aim 2020 challenge on real image super-resolution: Methods and results. In: European Conference on Computer Vision. Springer, pp. 392–422.

Zbontar, J., Knoll, F., Sriram, A., Murrell, T., Huang, Z., Muckley, M.J., Defazio, A., Stern, R., Johnson, P., Bruno, M., et al., 2018. Fastmri: An open dataset and benchmarks for accelerated mri. arXiv preprint arXiv:1811.08839.

Zhang, K., Liang, J., Gool, L.Van., Timofte, R., 2021a. Designing a practical degradation model for deep blind image super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4791–4800.

Zhang, W., Liu, Y., Dong, C., Qiao, Y., 2021b. Ranksrgan: Super resolution generative adversarial networks with learning to rank. arXiv preprint arXiv:2107.09427.

Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232.