TÉCNICO LISBOA

Super-Resolution of Biomedical Images with Generative Adversarial Networks and posterior Tumor Segmentation

João Luís Carrilho Guerreiro

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisors: Prof. Pedro Filipe Zeferino Aidos Tomás
Prof. Nuno Ricardo da Cruz Garcia

Examination Committee

Chairperson: Prof. Diogo Manuel Ribeiro Ferreira
Supervisor: Prof. Pedro Filipe Zeferino Aidos Tomás
Member of the Committee: Prof. Rita Homem de Gouveia Costanzo Nunes

October 2022

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

I am eternally grateful to my parents and grandparents for providing me the opportunity to study out of my hometown, despite the expenses and difficulties inherent to it. They gave me far more than I could have asked, and I hope I will continue to make them proud. Additionally, thank you brothers for handling my bothersome (have strength because I will not change a bit).

At the start of this 5-year challenge, I was afraid and scared of the upcoming difficulties related to the university. My godfather reached out and gave me the courage to carry on this adventure. A special thanks to him for being an inspiration to me since a very young age.

I remember, as a kid going to my godmother's home and having a great time with her family. We spent new year's eve together several times. I am far from Algarve nowadays, but I still remember all her family with care. Thank you, godmother and her whole family.

I want to thank someone that entirely changed my future path by exposing me to the beauty of mathematics. I remember joining the high school with an evident lack of knowledge. I undervalued maths, and I was not inclined to learn it further. Prof. Paulo Ferreira exposed me into a new perspective, gave me unprecedented knowledge, and encouraged me to follow a career related to this science. Thank you for passing great values for the youngsters, the best teacher I have ever had.

A massive thanks to Mariana, someone I appreciate and love very much. Having her by my side gave me the strength to smile during the most challenging times. She supports me unconditionally and is constantly worried about meaningless things (I believe she suffers from mother hen syndrome). I could not have asked for better. Furthermore, her family has always been wonderful, adopting me as one of their own. So again, a deep thank you to Mariana and all her family.

Additionally, I want to thank two fellows I consider as brothers. David and Dário, thank you for providing me the best childhood possible. I will be forever grateful to you.

During my time at the high school, I made a few friends for life. I had the pleasure of meeting and having a great time with Gonçalo, Luís, and Vasco. Despite their limitations and inability to handle my rhythm, they will never let me run alone. I also want to thank those who can not run but will always be by my side. Thank you, Mariana A., Mariana P., Mariana R., Olex, Roxane, and Sara.

I also want to thank those who have been with me since a very young age. Thank you, Inês L., Pedro

P., Tomás M., Alexandra R., Ricardo R., and David C.

Besides the knowledge acquired, the best part of these past 5 years was the people I met in the student dorm of IST. Thank you, Cruz, Miguel, and Pedro, for the best dinners and midnight studies.

Furthermore, thank you, Prof. Pedro Tomás, Prof. Nuno Garcia, and Prof. Helena Aidos for your unconditional feedback and alignment with methodologies. I appreciate all the knowledge you gave me.

Finally, a massive thank you to everyone who grew up with me in Loulé.

Abstract

Magnetic Resonance Imaging (MRI) is an expensive medical imaging technique typically associated with long scanning times. MRI acquisition can be potentially accelerated by decreasing the spatial coverage and reducing the number of measured slices. However, this results in a lower MRI resolution and can eventually lead to misleading medical interpretations. An alternative solution comes from recent breakthroughs in Machine Learning, which have shown that high-resolution images can be recovered via super-resolution, particularly through Generative Adversarial Networks. This thesis conducts a review on GAN-based SR methods, exhibiting the immersive ability of GANs on upscaling MRIs by a $\times 4$ scale factor while at the same time maintaining trustworthy and high-frequency details. Despite quantitative results suggesting SRResCycGAN outperforms other popular deep learning methods in recovering $\times 4$ downgraded images, qualitative results show Beby-GAN holds the best perceptual quality and proves GAN-based methods hold the capacity to reduce medical costs and enable MRI applications where it is currently too slow or expensive. Additionally, Tumor Segmentation is utilized to validate the proficiency of GANs in the MRI reconstruction task. Tumor Segmentation of the synthesized images advocates marginal dissimilarities, thus there is a window for improvement. Furthermore, this thesis suggests that a chain of processes for a faster diagnosis can be conceived by merging both Super-Resolution and Tumor Segmentation. Essentially, tumor segmentation algorithms benefit from the improved spatial resolution derived from super-resolution. The diagnosis process is accelerated by acquiring low-resolution MRIs and subsequently upscaling them (via super-resolution) to detect tumors.

Keywords

Computer Vision; Medical Imaging; MRI Acceleration; Super-Resolution; Tumor Segmentation; Generative Adversarial Networks.

Resumo

Imagiologia por Ressonância Magnética (IRM) é uma técnica dispendiosa que tipicamente está associada a longos temos de aquisição. Este processo pode ser acelerado ao reduzir a cobertura espacial. Porém, isto resulta numa baixa resolução e pode eventualmente levar a diagnósticos errôneos. Proveniente de recentes descobertas no campo da Inteligência Artificial, Redes Adversarias Generativas manifestaram-se como uma alternativa para recuperar RMs de alta resolução via super-resolução. Esta tese conduz uma revisão sobre métodos de SR baseados em GANs, exibindo a capacidade destas em melhor a resolução por um factor de $\times 4$, mantendo, simultaneamente, detalhes fiáveis e de alta frequência. Apesar dos resultados quantitativos sugerirem que o SRResCycGAN supera outros métodos populares na recuperação de imagens degradadas, os resultados qualitativos mostram que o Beby-GAN detém a melhor qualidade percetiva. É assim provado que os métodos baseados em GANs têm a capacidade para reduzir custos médicos e permitem aplicações de IRM onde é actualmente demasiado lento ou caro. Além disso, a Segmentação Tumoral é utilizada para validar a proficiência das GANs na tarefa de reconstrução de RMs. A Segmentação Tumoral das RMs sintetizadas expõe diferenças marginais, havendo assim uma janela para melhorias. Ademais, esta tese sugere uma cadeia de processos para um diagnóstico mais rápido onde se fundem Super-Resolução e Segmentação Tumoral. Essencialmente, os algoritmos de segmentação tumoral beneficiam de uma melhor resolução espacial derivada da super-resolução. O processo de diagnóstico é acelerado pela aquisição de RMs de baixa resolução e pela, subsequente, deteção automática dos tumores.

Palavras Chave

Visão Computacional; Imagem Médica; Aceleração de IRM; Super-Resolução; Segmentação de Tumores; Redes Adversariais Generativas.

Contents

List of Figures

List of Tables

1

Introduction

Contents

1.1 Motivation

Magnetic resonance imaging (MRI) is a medical imaging technique [1] that is predominantly necessary across patient 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 raises costs and leaves many patients on hold. Moreover, a slight movement from the patient can ruin the scan, requiring retesting. Hence, patients have to lie still in the scanners and even hold their breath for thoracic or abdominal imaging [2] since even the slightest movement of breathing can ruin the results. 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 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. For instance, 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. Besides, the duration of the whole process may increase predominately, depending on the pulse sequence type performed [3], 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 often used in emergencies when quick results are needed, such as when there is a severe 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 [4] (excluding scanning preparation and/or pre-scanning 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. An alternative is low-field MRI scanners [5], 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 (see Figure 1.1). 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.

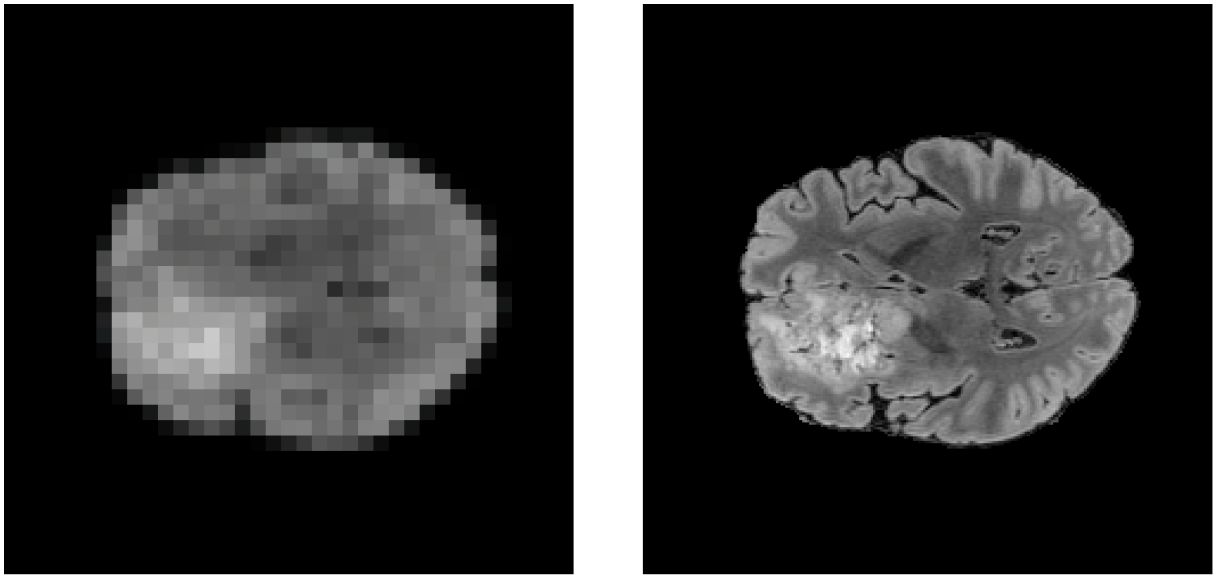


Figure 1.1: Super-Resolution of a Magnetic Resonance Image.

In general, SR methods are based on Generative Adversarial Networks (GANs), which were introduced in 2014 by Goodfellow et al. [6] 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.

To complement Super-Resolution and provide a more sophisticated and fast diagnoses, automated tumor segmentation can be considered. MRIs have been widely utilized to detect and evaluate brain tumors. However, the amount of detailed information present in MRIs poses a significant problem, as it prevents manual segmentation in a reasonable time.

The anatomical structure of a brain tumor is complex, thus intensifying the difficulty of differentiating cancer from the remaining healthy brain tissue. Additionally, not only is handcrafted segmentation time-consuming, but it also can lead to human errors. Therefore, exploiting a method that sustains accurate and reproducible identification of tumors can help physicians to rapidly detect glioblastomas, thus increasing the survival rate.

For instance, with current medical treatments, most people diagnosed with glioblastoma live on average less than two years after the initial diagnosis [7]. An earlier diagnosis extends the suitable period to deal with the disease, thus alleviating patients and physicians. In addition, accelerating the tumor detection process through a computerized medical diagnosis could inherently reduce the average wait time among medical facilities. Besides, an AI that outperforms trained radiologists on diagnoses leads to substantial survival rate improvements. Accordingly, segmenting tumors on MRIs with reduced spa-

tial resolution and attaining accurate segmentations with performance levels equal to those performed with high-resolution MRIs expresses profound worth. However, low-resolution MRIs lack high-level details, thus jeopardizing the tumor detection process. Therefore, by merging both Super-Resolution and Tumor Segmentation concepts, it is possible to conceive a chain of processes that suggests an enhanced pipeline for a faster diagnosis. Essentially, tumor segmentation algorithms benefit from the improved spatial resolution derived from super-resolution. The whole diagnosis process is accelerated by acquiring low-resolution MRIs and then super-resolving those MRIs to be utilized for tumor detection. Consequently, the costs can be reduced due to lower resolution requirements.

Distinct tumoral subregions can be perceived, and accurately detecting these regions within the MRI is consequential. Similarly to super-resolution, there is a concept in Machine Learning entitled Semantic Segmentation (SS). It is the process dedicated to associating each pixel of an image with a class label. Semantic segmentation is analogous to classification, except that in semantic segmentation, the intention is to classify every pixel rather than classify the image as a whole. In essence, semantic segmentation is still a classification problem but with a higher granularity as it performs classification on the pixel level. Following the aforementioned, it is easy to comprehend the benefits of semantic segmentation in the medical context, as it provides the ability to extract regions of interest (ROIs), like tumors and lesions, from 3D image data, such as MRI (see Figure 1.2).

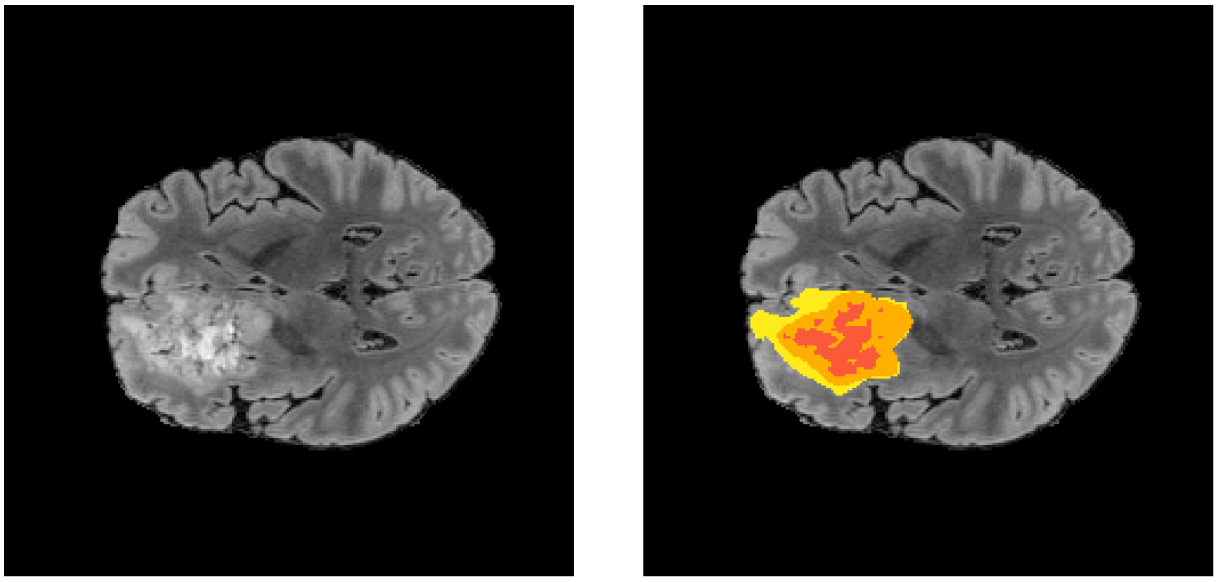


Figure 1.2: Semantic Segmentation of a Magnetic Resonance Image.

1.2 Work Goals & Contributions

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 [8–12] have addressed the subject of Super-Resolution. 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. Its importance around super-resolution manifests relevant contributions to MRI acceleration and real-time spatial quality improvements. Furthermore, it reflects the immersive capacity of super-resolution in sustaining an economical alternative to acquire high-quality imagery that is accurate enough to measure and visualize structures in biological tissues. 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 several 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. Inadvertently, the majority was inspired by the traditional SRGAN model architecture (see Section 3.1) introduced in [13]. The performance of these models is evaluated over FastMRI [14]. Meanwhile, SRGAN is not considered in the experiments due to the lack of performance, as succeeding models have already surpassed him by some margin.

Hypothetically, super-resolved MRIs having similar results as ground-truth MRIs in the tumor segmentation task is suggestive of an accurate recovery of the details inherent to high-resolution MRIs. Therefore, tumor segmentation algorithms and techniques were considered to employ a task-based evaluation intended to assess the GAN-based super-resolution performance. Essentially, the super-resolution performance is estimated by assessing the tumor segmentation performance of the super-resolved brain MRIs. Additionally, through the application of tumor segmentation methods, not only the reconstruction quality of GANs is exposed, but it also supports the idealized pipeline for a faster diagnosis, where Super-Resolution and Tumor Segmentation are consolidated.

Besides the task-based evaluation on Super-Resolution, this works conducts a review on tumor segmentation. In essence, it intends to validate the proficiency of deep learning in providing extremely detailed anatomical information appropriate to be used in reliable automated diagnosis. Ultimately, if proven that tumor segmentation operates properly over super-resolved magnetic resonance images, then it is suggestive that it will work accordingly over other super-resolved biomedical images.

Distinctively from other studies, this work exploits the human visual system and intends to enhance

the comprehension across visualization by employing several rigorous diagrams. Moreover, to ease the comprehension of the concepts addressed (Super-Resolution and Tumor Segmentation), the work adopts a progressive complexity strategy while instructing proper techniques to tackle these tasks. It covers meaningful information and conceives a critical review with high-level detail, delving deep and addressing these vast themes of computer vision. Nonetheless, it supplies detailed and intuitive descriptions of the techniques and evaluation metrics utilized in super-resolution and tumor segmentation.

1.3 Thesis Outline

Regarding the outline of this work, Chapter 3 reviews state-of-the-art GANs for super-resolution. Subsequently, a discussion about optimization strategies is carried, intended to minimize error when fitting super-resolution algorithms. Afterward, Chapter 4 describes experiments performed over FastMRI to exhibit the effectiveness of GANs in medical image reconstruction and processing. Moreover, it mentions GAN-based MRI reconstruction problems and quality assessment metrics. Ultimately, it holds an extensive discussion with quantitative and qualitative results. Chapter 5 contends tumor segmentation techniques, learning strategies to optimize training, and the implementation details employed in the following experiments. Similarly to Chapter 4, tumor segmentation experiments over BraTS [15] are addressed in Chapter 6 jointly with a discussion about the results and respective evaluation metrics. To finish, conclusions are deduced, and future work is proposed in Chapter 7.

2

Background & Related Work

Contents

This chapter describes Super-Resolution and Tumor Segmentation concepts, serving as a background for the following chapters. These machine learning concepts manifest considerable relevance in medical image processing, potentially leading to substantial survival rate improvements. They complement each other by conceiving a chain of processes that suggests an enhanced pipeline for a faster diagnosis. Super-Resolution has the potential to accelerate the acquisition of biomedical images detained with proper quality conditions for medical use. Meanwhile, Tumor Segmentation can automate the diagnoses and ideally outperform trained radiologists. Additionally, Tumor Segmentation can be conceived as a strategy to validate the Super-Resolution performance since the higher the detailed information recovered by Super-Resolution, the greater the success of Tumor Segmentation. This work comprehensively reviews state-of-the-art Generative Adversarial Networks for Super-Resolution, and Tumor Segmentation is pragmatically considered as an evaluation metric.

2.1 Super-Resolution

2.1.1 Problem Definition

Super-Resolution (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:

$$y = \Phi\left(\hat{x}_r; \Omega\right), \tag{2.1}$$

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

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

$$x_g = \mathcal{F}\left(y; \Theta\right) = \Phi^{-1}\left(y; \Theta\right) \approx \hat{x}_r, \tag{2.2}$$

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

$$\hat{\Theta} = \arg\min_{\Theta} \mathcal{L}\left(x_g, \hat{x}_r\right), \tag{2.3}$$

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 for the trained model $\mathcal{F}$.

The degradation process is complex and affected by multiple factors, such as stochastic noise, blur,

compression and variable artifacts. Therefore, a preferable equation to define the degradation model is:

$$y = \Psi\left(\left(\hat{x}_r \otimes k\right) \downarrow_s + N\left(\mu, \sigma^2\right)\right), \tag{2.4}$$

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.

2.1.2 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 [16, 17], can be very efficient and easy to implement (see Figure 2.1). Furthermore, Claude E. Duchon [18] 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 [19].

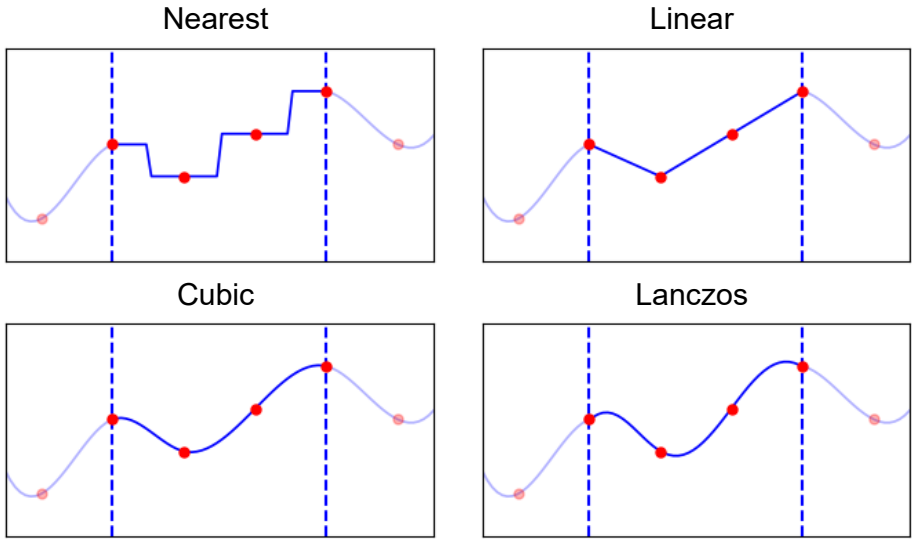


Figure 2.1: Interpolation Algorithms applied in 1-dimension.

2.1.3 Deep Learning Methods

In practice, super-resolution 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 sample. Therefore, they can perform super-resolution 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 super-resolution 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 believing they are real ground truth 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 Figure 2.2.

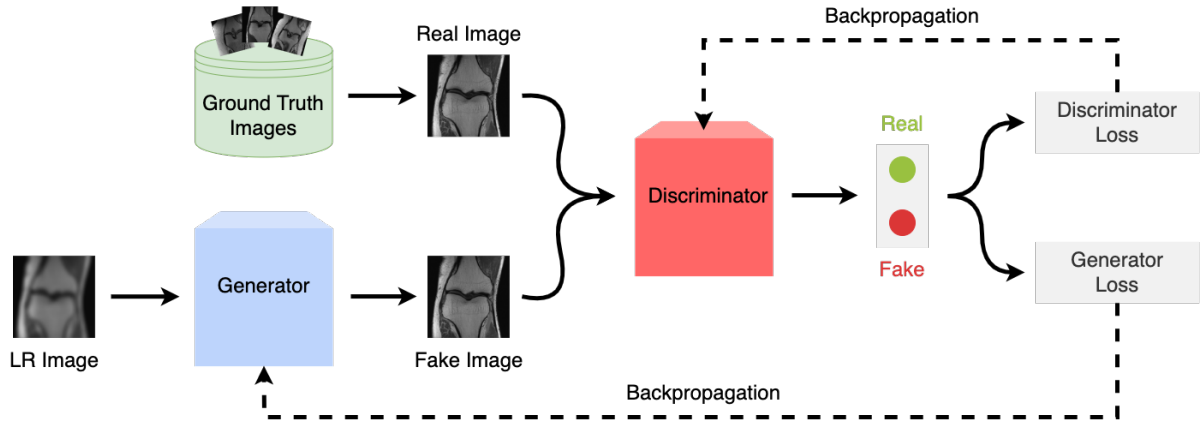


Figure 2.2: Main concept behind GANs.

2.2 Tumor Segmentation

2.2.1 Problem Definition

Tumor Segmentation is a particular case of Semantic Segmentation (SS). Semantic Segmentation intends to assign a well-defined class label to each pixel, thus expressing what the pixel represents. Additionally, semantic segmentation is a classification problem, but with a higher granularity as it performs classification on the pixel level rather than on the image level (volume level if 3D medical images are

used). Evidently, tumor segmentation is expected to segment tumors within medical images or volumes. Essentially, from a medical image, x, semantic segmentation tries to estimate a segmentation mask, m_p, of the ground truth segmentation mask, $\hat{m}_t$ (see Figure 2.3). Thus, the segmentation process can be given as:

$$m_p = \mathcal{T}(x; \Theta) \approx \hat{m}_t, \tag{2.5}$$

where x is any input image intended to be segmented by the segmentation process, $\mathcal{T}$. The segmentation results in a predicted segmentation mask, m_p, which is an estimation of the true mask (ground truth), $\hat{m}_t$. Moreover, the optimization of Θ is expressed by the following equation:

$$\hat{\Theta} = \arg\min_{\Theta} \mathcal{L}(m_p, \hat{m}_t), \tag{2.6}$$

where $\mathcal{L}$ denotes the function that estimates the similarity between the two segmentation masks.

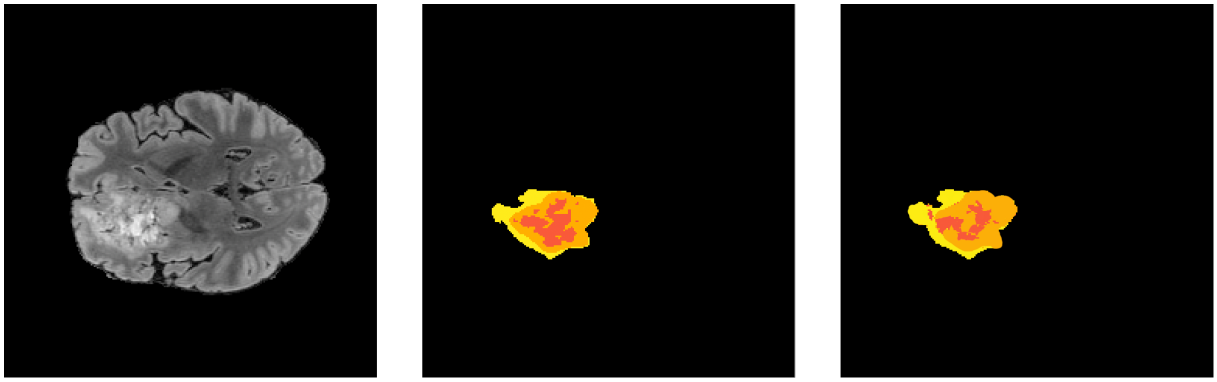


Figure 2.3: Input Magnetic Resonance Image on the left. On the middle and right, the ground truth and predicted segmentation masks are displayed, respectively.

2.2.2 Conventional Methods for Semantic Segmentation

Classical methods were usually based on pixel value comparisons between regions. These methods perceive image features locally while considering variations and gradients of pixel values. They are divided into three main categories: threshold-based [20], edge-based [21, 22] and region-based [23].

Threshold-based techniques are one of the easiest and most rudimentary segmentation methods. A threshold is set to divide each pixel into 2 classes. Pixels that have values greater than the threshold are set to 1 while pixels with values lesser than the threshold value are set to 0. Therefore, the image is converted into a binary map. Evidently, these methods are more effective over images manifesting high levels of contrast due to the comparison performed for each pixel intensity value with respect to the threshold. More sophisticated approaches select a threshold value for each pixel according to the local

image characteristics, thus providing a more robust and adaptive segmentation (see Figure 2.4).

Figure 2.4: Practical view of threshold-based segmentation. Raw Magnetic Resonance Image on the left. On the middle and right, it is shown the product of applying global and local thresholding, respectively.

Edge-based methods intend to detect region boundaries. Edges characterize the physical extent of regions, i.e., they are the boundaries between regions with different properties. Edges are detected by local derivatives (variations) of an image, for instance the image gradient as explained in Section 3.8.7. Optimal edge detection algorithms have several benefits in medical imaging since they can detect the outline of tumors and organs (see Figure 2.5). Nonetheless, the goal of edge-based segmentation is to provide an intermediate segmentation that afterwards region-based or any other type of segmentation techniques can utilize to get the ultimate segmentation result.

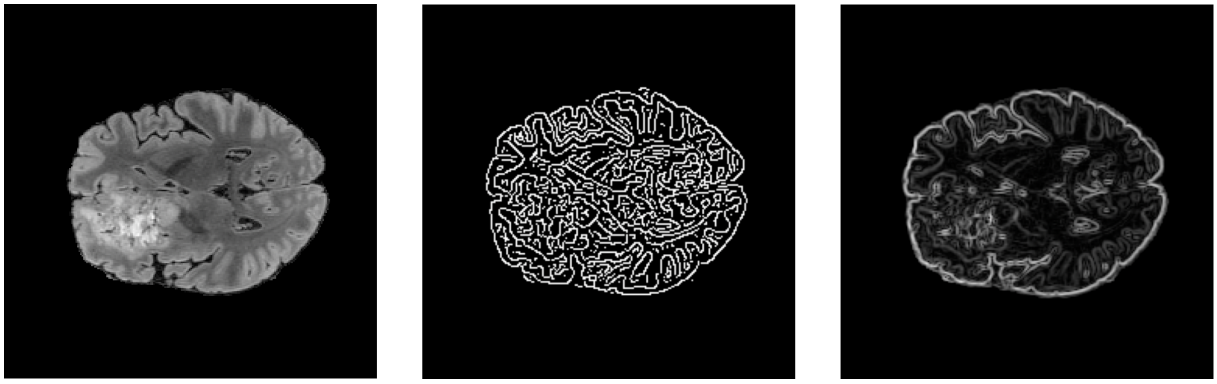


Figure 2.5: Pragmatic view of edge-based segmentation. Raw Magnetic Resonance Image on the left. On the middle and right, it is displayed the product of applying Canny and Prewitt edge detection techniques, respectively.

Region-based methods work by searching for similarities between adjacent pixels and eventually grouping them under a defined class. Essentially, they operate iteratively by grouping together neighboring pixels that have similar properties, such as pixel intensity values. Accordingly, they split groups of pixels that are dissimilar in value. The algorithms grow regions by adding more pixels, and additionally shrinks and merges regions with each other. For instance, watershed segmentation is a region-based

technique that perceives images as topographic maps, where pixel intensity determines elevation. It detects lines forming ridges and marks the areas between the watershed lines. The watershed technique has diverse relevant use cases, including medical image processing. For example, it can help to detect variations of intensity in MRI scans, i.e., differences between lighter and darker regions, thus potentially assisting with medical diagnosis.

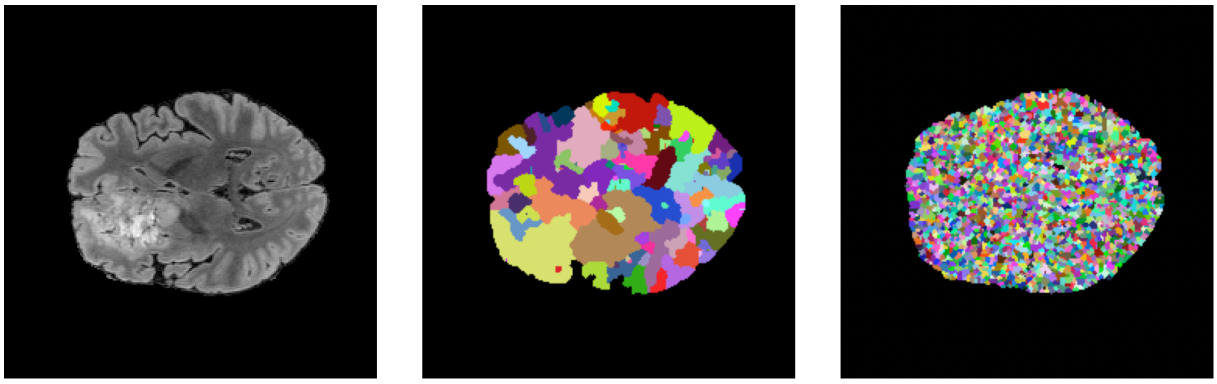


Figure 2.6: Empirical view of region-based segmentation in biomedical images. Raw Magnetic Resonance Image on the left. On the middle and right, it is displayed the result of applying a watershed segmentation with Image Gradient and Sobel (edge detection techniques) to compute the elevation maps, respectively.

Looking at Figure 2.6, it is noticeable that watershed segmentation with Sobel detects a lot more regions, which results from the Sobel higher noise sensitivity. Accordingly, the edge detection technique is tricked into regarding noise as edges, thus affecting considerably the elevation map.

Reasoning, these conventional techniques can be exploited to extract a set of features. Subsequently, traditional machine learning algorithms can be utilized to perform an ultimate semantic segmentation (see Section 5.1).

2.2.3 Deep Learning Methods for Semantic Segmentation

Prior to the dissemination of deep learning, tree-based techniques and other machine learning algorithms were vastly employed to tackle the challenging semantic segmentation problem. Traditional methods are exceptional over limited data, however with more data available deep learning excels traditional techniques and attains better performances within semantic segmentation and many other computer vision tasks (see Figure 2.7).

Semantic Segmentation based on Neural Networks (NN) is feasible due to the unfolding of large medical datasets and the reduction of computing requirements necessary to process them. Furthermore, developments in the deep learning field have greatly advanced the performance of these state-of-the-art visual recognition systems, thus leading neural networks to surpass the hard work of traditional machine learning models. Accordingly, a kind of neural networks designed to process multi-dimensional

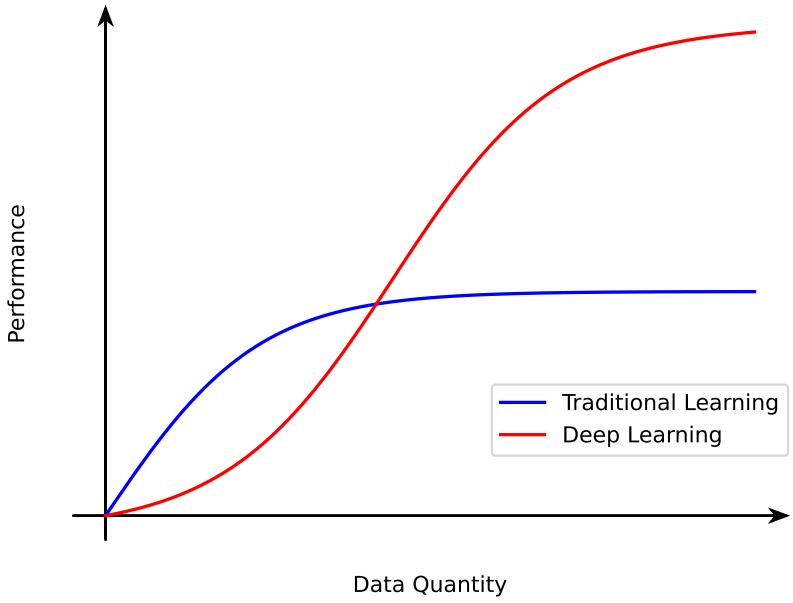


Figure 2.7: Traditional Machine Learning versus Deep Learning.

data, such as images/volumes, are entitled Convolutional Neural Networks (CNN). They are the most frequently used technique to solve image-based problems, including semantic segmentation.

Among the diverse CNN-based models, Fully Convolutional Networks (FCNs) [24] introduced a novel strategy to solve semantic segmentation (see Section 5.2). They received a lot of attention by exhibiting that convolutional networks can be trained to accommodate pixel-level classification in an end-to-end manner. FCNs incited a trend among semantic segmentation models, serving as an inspiration for several other approaches. Almost all the subsequent state-of-the-art methods on semantic segmentation adopted the encoder-decoder paradigm introduced by FCNs. Although, at the time FCNs were introduced, there were already many other CNN-based use cases employed for semantic segmentation. For instance, preceding the early FCN-based state-of-the-art architectures, Zikic et al. [25] had already investigated the application of CNNs to segment brain tumor tissues. Additionally, Prasoon et al. [26] proposed a novel system to segment tibial cartilage in low field knee MRI scans.

An example of an FCN is the DeepLab network proposed by Chen et al. [27]. It consist in a state-of-the-art semantic segmentation model having an encoder-decoder architecture. The main improvements suggested by DeepLab were the aggregation of the à trous algorithm [28] and the usage of Conditional Random Fields (CRF) [29] in semantic segmentation. Later, to robustly segment objects at multiple scales, the authors proposed DeepLabv2 [30] by introducing the Atrous Spatial Pyramid Pooling (ASPP), motivated by the works of Lazebnik et al. [31] and He et al. [32]. Additionally, the authors revisited some concepts, introducing cascaded modules to capture multi-scale context and a refined version of the ASPP module, thus resulting in DeepLabv3 [33]. Ultimately, DeepLabv3+ [34] extends DeepLabv3

by adding an effective decoder module to improve the segmentation results, especially along object boundaries. All versions exhibited good results.

Following the FCN idea of an encoder-decoder architecure, DeconvNet was proposed by Noh et al. [35]. DeconvNet mitigates the limitations of the existing methods based on fully convolutional networks (FCNs) by integrating a deep deconvolution network. Essentially, it used a convolutional network followed by an hierarchically opposite deconvolutional network for semantic segmentation. Furthermore, built upon the concept of FCNs, U-Net was proposed by Ronneberger et al. [36]. It was designed for biomedical image segmentation. However, it has proven to be generalizable for practically any semantic segmentation task. Additionally, it employed skip-connections to tackle the information loss problem inherent to FCNs (problem described in Section 5.2.2). The name U-Net comes from the adopted U-shaped architecture, which consists in a contracting and an expanding path way regarded as the encoder and decoder, respectively (Section 5.2.2). SegNet proposed by Badrinarayanan et al. [37], is another FCN architecture, which also followed the encoder-decoder architecture. The architecture core consists of an encoder followed by a topologically identical decoder. Moreover, the encoder network is equal to the first 13 convolutional layers in the VGG16 network [38] designed for object classification.

The basic architectural intuition of DeconvNet, U-Net and SegNet are similar except some individual modifications. The main differences compared to FCN are that these networks are symmetric, i.e, the second half of those architectures, regarded as the decoder, is the mirror version of the first half, the encoder. Reasoning, the popularity of encoder-decoder architectures for semantic segmentation was solidified with the onset of works like Deconvnet, U-Net and SegNet.

Additionally, motivated by the idea that FCN cannot represent global context information, Liu et al. [39] proposed ParseNet. It manifested improvements by merging the essence of global average pooling and L2 normalization layer in an FCN architecture. Accordingly, ParseNet was followed by other approaches that also intended to integrate global context on deep convolutional networks for semantic segmentation. Afterwards, PSPNet was proposed by Zhao et al. [40], consisting in a pyramid scene parsing network to embed complex scenery context features in an FCN-based architecture. Essentially, the authors incorporated global contextual information by adding a Pyramid Pooling Module on top of the last extracted feature map. Furthermore, Peng et al. [41] proposed Global Convolutional Network (GCN) to address classification and localization issues in semantic segmentation. To take advantage of both local and global features simultaneously, the FCN architecture was borrowed as their basic framework to retain the localization performance and large kernel were employed to make global convolution practical. Additionally, the authors introduced boundary refinement blocks, which further improved the performance near the object boundaries. All ParseNet, PSPNet and GCN have used global context information along with local feature to improve segmentation.

Furthermore, Myronenko et al. [42] followed the encoder-decoder structure of FCN and added a

variational auto-encoder (VAE) branch to reconstruct the input images jointly with segmentation, thus regularizing the shared encoder. The approach proposed won the BraTS 2018 challenge [43, 44]. Jiang et al. [45] proposed a novel two-stage cascaded U-Net, which refines the prediction in a progressive way, to segment the substructures of brain tumors. The approach took first place in BraTS 2019 challenge segmentation task. Additionally, an optimized U-Net for biomedical image segmentation was introduced by Isensee et al. [46]. It consists in a framework directly built around the original U-Net architecture. Further, the authors employed several minor modifications to the nnU-Net pipeline [47] and made the framework compatible with the BraTS-specific processing, thus allowing the framework to be applied on the segmentation task of BraTS intended to segment brain tumors. Subsequently, the method won the first place in the BraTS 2020 challenge.

2.3 Summary

This chapter discusses Super-Resolution and Tumor Segmentation, providing additional information and background about these concepts.

From this chapter, it is convenient to retain the importance and intuition of generative adversarial networks, as the next chapter 3 will intensively address several approaches that adopt this architecture strategy for Super-Resolution. Additionally, classical methods for image interpolation were addressed.

Moreover, conventional methods for semantic segmentation were described, since they sustain a practical way to extract features that can be used to train traditional machine learning algorithms. This work intends to propose a tree-based approach that exploits these classical methods to extract relevant features 5.4.1.

Ultimately, it was described the evolution of CNN-based models for semantic segmentation, from the early FCN network, that lead to the popularity of encoder-decoder networks, until the recent nnU-Net that won the BraTS 2020 challenge.

3

Generative Models for Super-Resolution

Contents

This section conducts a review on GAN methods for the Super-Resolution problem 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 [13], which was a novel super-resolution approach using the GAN concept.

The optimization of SR methods is predominately driven by the choice of the target function. Before SRGAN, the 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 3.8.3 and 4.4.3). 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 loss intended to optimize the generator network while exploiting high-level feature maps of the VGG network [38]. Moreover, the generator employs a deep residual network [48] with skip connections as depicted in Figure 3.1.

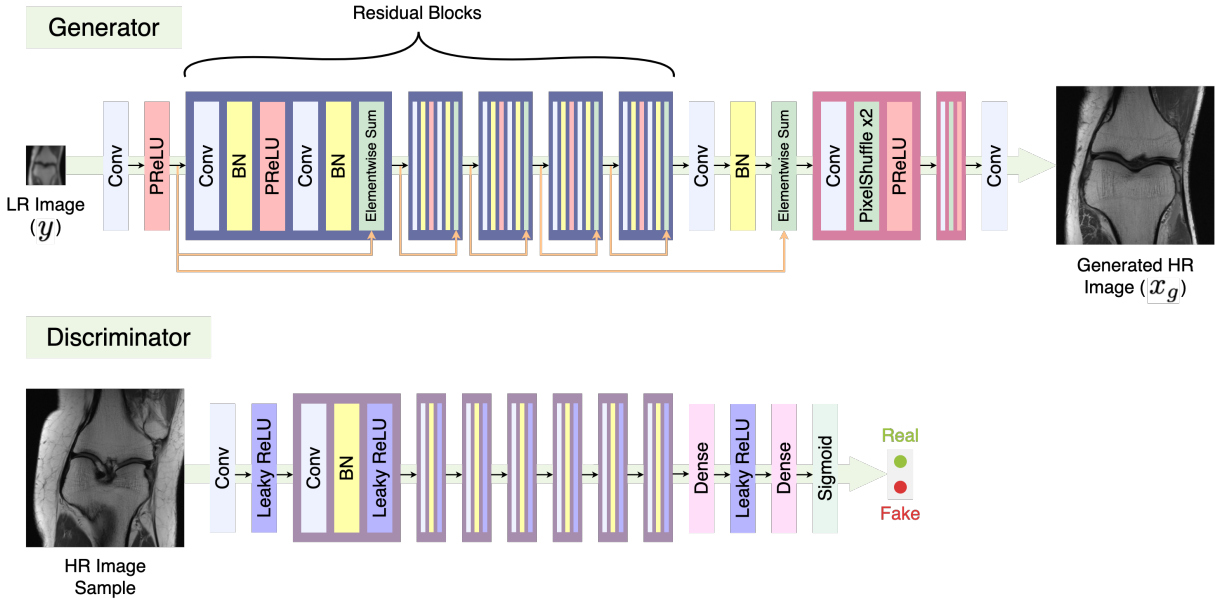


Figure 3.1: Basic architecture of SRResNet (SRGAN). Figure adapted from [13].

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 3.8).

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 favours solutions that reside on the manifold of natural images and is given by the equation in Section 3.8.2.

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 $\mathcal{L}_{\mathcal{HR}}$:

$$\hat{\Theta} = \arg\min_{\Theta_{\mathcal{G}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\mathcal{HR}} \left(\mathcal{G} \left(y_i; \Theta_{\mathcal{G}} \right), \hat{x}_{r_i} \right), \tag{3.1}$$

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, $\mathcal{L}_{\mathcal{HR}}$ is defined in Table 3.1.

Results evidence that, absent from an optimization solely around MSE, SRGAN is able to infer HR images by an upscale factor of $\times 4$, recovering textures and details from heavily downsampled images.

3.2 ESRGAN

Based on the SRGAN pioneer work [13], a model named Enhanced SRGAN (ESRGAN) [49] was introduced to reduce unpleasant artifacts present in the SRGAN generated data. 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 [48] 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 super-resolution or image restoration tasks in general, Batch Normalization can create some artifacts as depicted in Figure 3.2. 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 [50].

The high-level architecture design of SRGAN [13], as depicted in Figure 3.1, is employed and the replacement of the original 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 Figure 3.3 next to the SRGAN Residual Block.

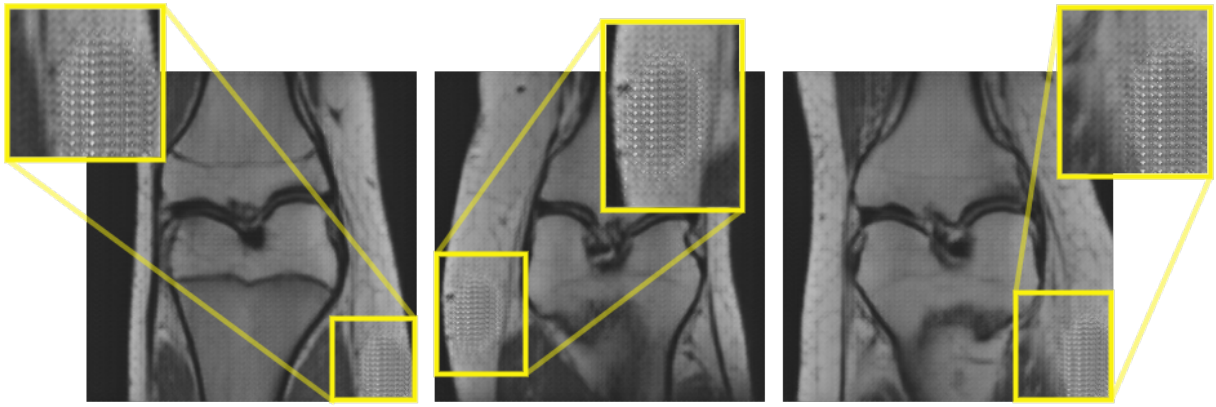


Figure 3.2: Batch Normalization artifacts under SRGAN on fastMRI images.

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". This improvement helps the generator to recover more realistic texture details. The adversarial losses for the generator and discriminator are defined as the following equations:

$$\mathcal{L}_{\mathcal{G}} = -\mathbb{E}_{x_r}\left[\log\left(1 - \mathcal{D}\left(x_r, x_g\right)\right)\right] - \mathbb{E}_{x_g}\left[\log\left(\mathcal{D}\left(x_g, x_r\right)\right)\right], \tag{3.2}$$

$$\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{x_r}\left[\log\left(\mathcal{D}\left(x_r, x_g\right)\right)\right] - \mathbb{E}_{x_g}\left[\log\left(1 - \mathcal{D}\left(x_g, x_r\right)\right)\right], \tag{3.3}$$

where $\mathbb{E}_{x_r}$ and $\mathbb{E}_{x_g}$ correspond to the operation of taking the average over all real and generated fake data, respectively. Moreover, $\mathcal{D}\left(x_g, x_r\right)$ represents the probability that a generated image x_g is relatively less realistic than a real one x_r and $\mathcal{D}\left(x_r, x_g\right)$ the probability that a real image x_r is more realistic than a generated one x_g. The generator loss can be given in terms of the adversarial loss $\mathcal{L}_{\mathcal{G}}$, as shown in Table 3.1. Meanwhile, the discriminator loss can be directly inferred from $\mathcal{L}_{\mathcal{D}}$, without any further computation, as it is solely defined by it, $\mathcal{L}_{\mathcal{D}} = \mathcal{L}_{Discriminator}$.

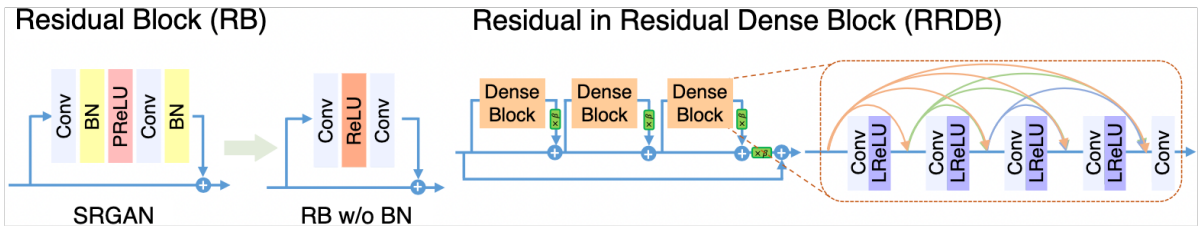


Figure 3.3: Batch normalization removal on the left. On the right, Residual in Residual Dense Block is embedded in the model. Figure adapted from [49].

3.3 RankSRGAN

Perceptual quality can be assessed by perceptual metrics, such as Perceptual Index (PI) [51], Natural Image Quality Evaluator (NIQE) [52], and Ma [53], 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 an approach was proposed consisting of a GAN with a Ranker, named RankSRGAN [54].

RankSRGAN employs the standard architecture design of SRGAN [13]. In addition to SRGAN a 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 behaviour of perceptual quality metrics. The ranker is trained by optimizing a margin-ranking loss [55] and eventually learns to rank images according to their perceptual scores.

The Ranker adopts a Siamese architecture to learn the behaviour of perceptual metrics as depicted in the middle section of Figure 3.4. Primarily, different SR models are used to generate images. Then, these generated images are put together two by two to form pair-wise images. Subsequently, these pairs are ranked/labeled according to the quality score calculated by the perceptual metric, as expressed in (3.4). Afterwards, the Siamese-like Ranker network is trained over the rank dataset consisting of the 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.

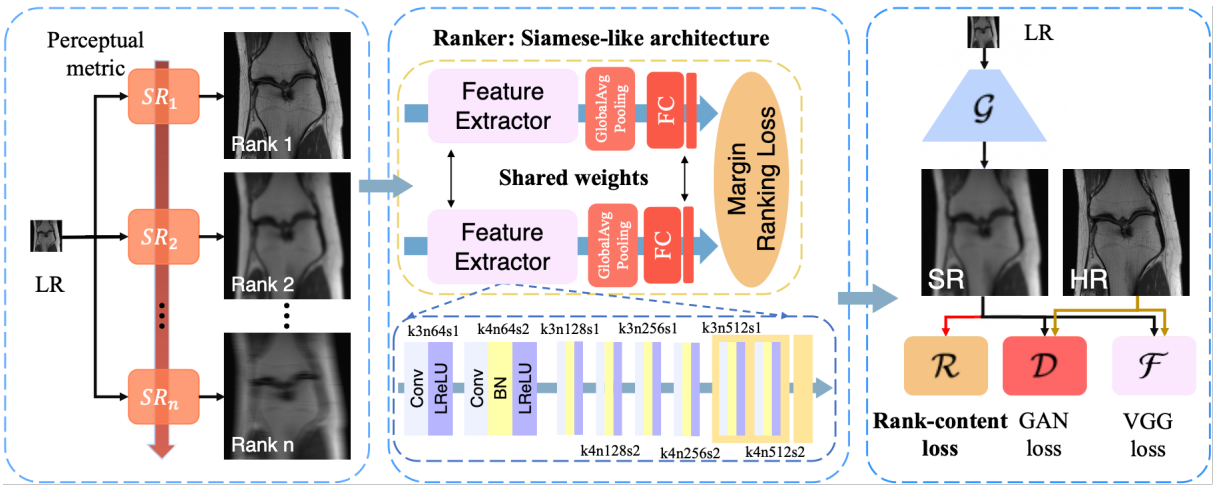


Figure 3.4: Overview of RankSRGAN. Essentially, RankSRGAN consists of a generator ($\mathcal{G}$), a discriminator ($\mathcal{D}$), a fixed feature extractor ($\mathcal{F}$) and a ranker ($\mathcal{R}$). Figure adapted from [54].

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 Figure 3.4, 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:

$$\begin{aligned} \mathcal{L}\left(s_1, s_2, \gamma\right) &= \max\left(0, \left(s_1 - s_2\right) \cdot \gamma + \epsilon\right), \\ &\begin{cases} \gamma = 1 & \text{if} \quad m_1 > m_2 \\ \gamma = -1 & \text{if} \quad m_1 < m_2 \end{cases}, \end{aligned} \tag{3.4}$$

where s_1 and s_2 correspond to the ranking scores of the generated images x_{g_1} and x_{g_2}, respectively. Moreover, m_1 and m_2 represent the quality scores of the pair-wise images x_{g_1} and x_{g_2}, while γ is the rank label of the pair-wise training images. A lower ranking score indicates better perceptual quality. Additionally, as a means to ease comprehension the following can be conjectured:

$$\begin{cases} s_1 < s_2 & \text{if} \quad m_1 > m_2 \\ s_1 > s_2 & \text{if} \quad m_1 < m_2 \end{cases}. \tag{3.5}$$

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

$$\hat{\Theta} = \arg\min_{\Theta_{\mathcal{R}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(s_1^{(i)}, s_2^{(i)}, \gamma^{(i)}\right) = \arg\min_{\Theta_R} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\mathcal{R}\left(x_{g_1}^{(i)}; \Theta_{\mathcal{R}}\right), \mathcal{R}\left(x_{g_2}^{(i)}; \Theta_{\mathcal{R}}\right), \gamma^{(i)}\right), \tag{3.6}$$

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 (3.5).

Compared to SRGAN, this method simply introduces a well-trained Ranker that is used by the rank-content loss (defined in Section 3.8) 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 favour 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 [56] in image-to-image translation applications, a new deep cyclic network structure was proposed, named SRResCycGAN [57]. 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 [58]. SRResCycGAN overcomes this challenge and maintains the domain consistency between the LR and HR data distributions by following the CycleGAN structure, as shown in Figure 3.5.

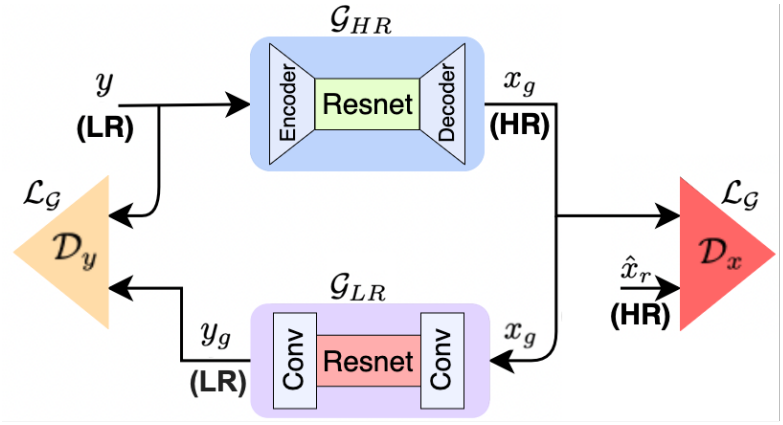


Figure 3.5: SRResCycGAN structure. Figure adapted from [57].

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 behaviour favours 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 [59]. The generator consists of a Encoder-Resnet-Decoder structure as shown in Figures 3.5 and 3.6. Inside

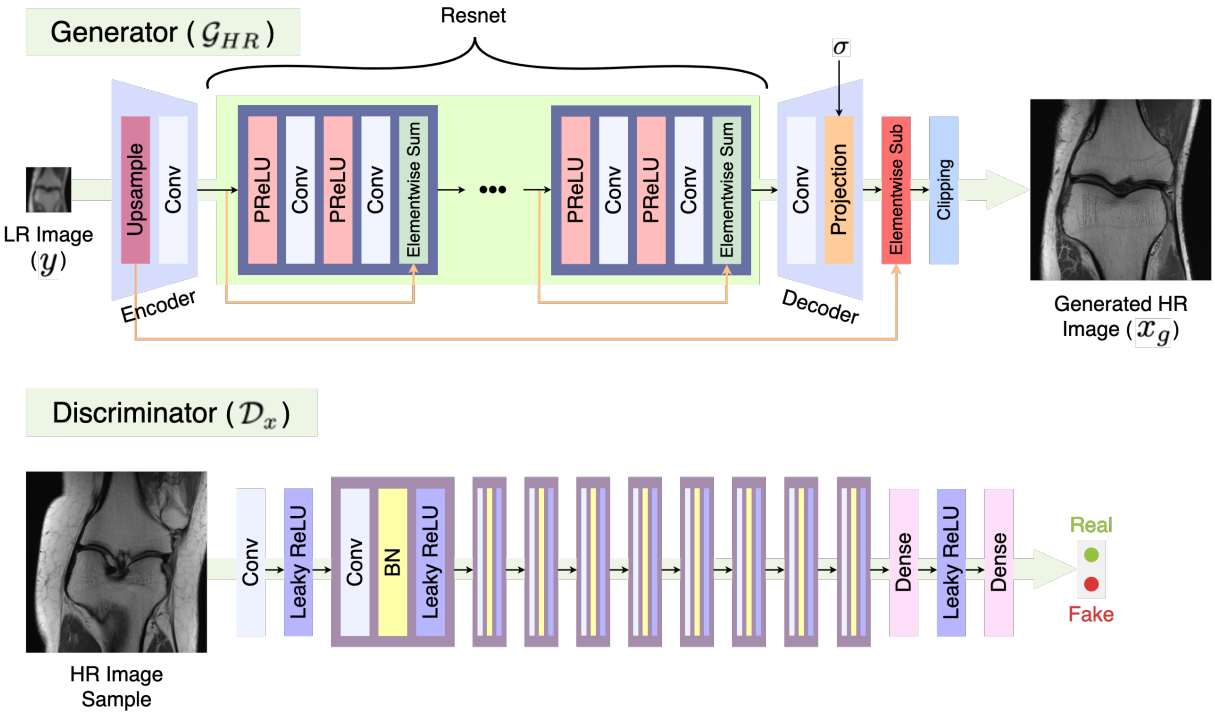


Figure 3.6: SRResCGAN architecture.

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:

$$\mathcal{L}_{\mathcal{HR}} = \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\mathcal{TV}} + \lambda \cdot \mathcal{L}_1 + \eta \cdot \mathcal{L}_{\mathcal{CYC}}, \tag{3.7}$$

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 [57]. These losses are defined in the Learning Strategies Section 3.8.

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 [60] was proposed along with a 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 degradation model, which consists of randomly shuffled blur, downsampling and noise degradations as shown in Figure 3.7. With the random shuffle strategy, the degradation space

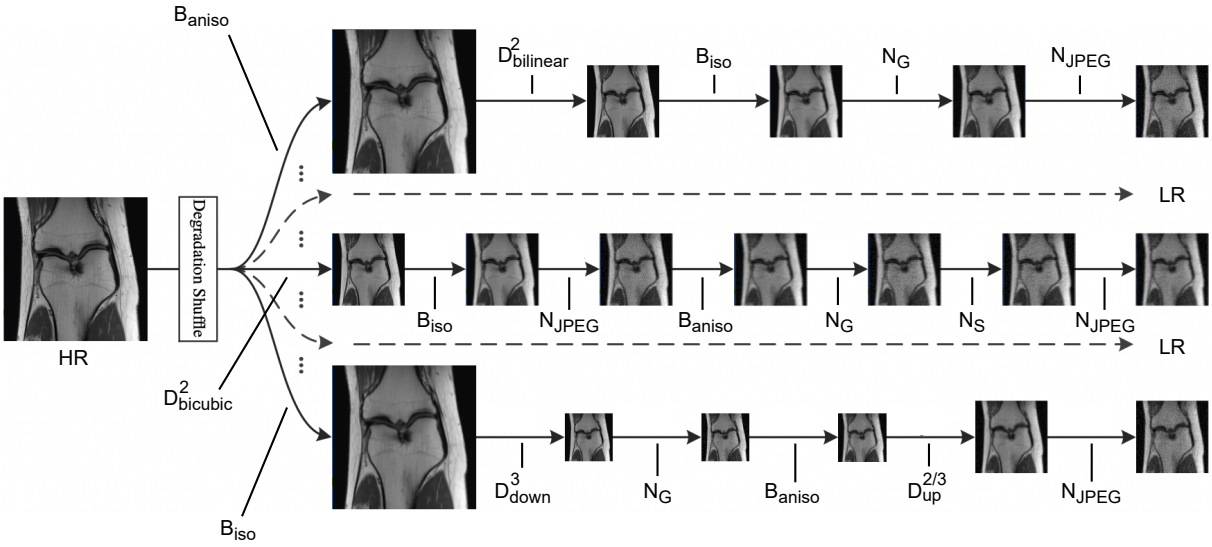


Figure 3.7: Proposed BSRGAN degradation model for a scale factor of 2. For scale a factor of 4, an additional bilinear or bicubic downscaling is applied. The type of blur employed is denoted by B_{type} and N_{type} is the type of noise. Meanwhile, D^{scale}_{type} stands for the downsampling applied under a defined scale. Figure adapted from [60].

can be expanded substantially. Consequently, the SR model is able to super-resolve LR images under unknown and diverse degradations.

The novelty of this approach lies in the 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 [61].

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 behaviour often undermines the realism of the whole image. To address these issues Beby-GAN [62] 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 favouring 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 Figure 3.10). 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 Figure 3.8.

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}, must meet two constraints:

Constraint 1

It is mandatory that the best-buddy patch p_{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} 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

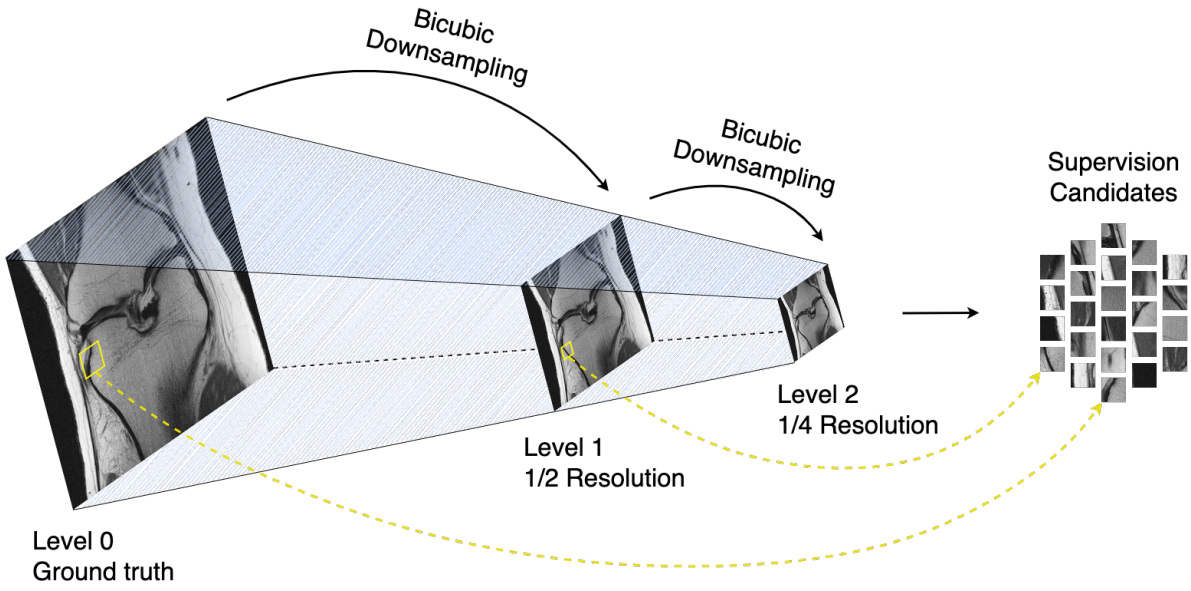


Figure 3.8: Beby-GAN 3-level image pyramid obtained with bicubic downsampling. Subsequent images are subject to repeated downsampling. Additionally, other types of degradation can be introduced in each subsampling level.

learning strategy, where the most appropriate supervision patch is used in every iteration.

Following these two objectives, the selected best-buddy patch p_{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} 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:

$$p_{BB_i} = \arg\min_{p \in \mathcal{S}} \alpha \left\| p - \hat{p_{r_i}} \right\|_2^2 + \beta \left\| p - p_{g_i} \right\|_2^2, \tag{3.8}$$

where p represents a patch contained in $\mathcal{S}$, which is the supervision candidate dataset of the generated image. Essentially, $\mathcal{S}$ 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}) is given as the distance between the estimated patch p_g and the best-buddy patch p_{BB}, i.e, the 1-norm of the difference, as defined in Section 3.8. Reasoning, when $\alpha \gg \beta$, the best-budy 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 [57], 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:

$$\mathcal{L}_{\mathcal{BP}} = \|Z(x_g, s) - y\|_1 , \tag{3.9}$$

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 [62] 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 Figure 3.9.

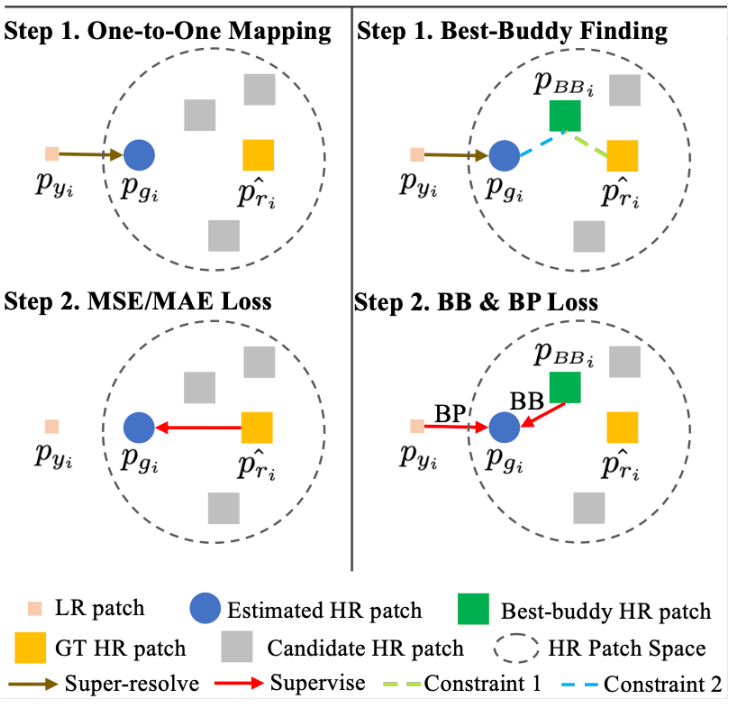


Figure 3.9: Comparison between MSE/MAE and best-buddy (BB) loss with a back-projection (BP) constraint. Figure adapted from [62].

As shown in Figure 3.2, previous GAN-based methods were prone to undesirable artifacts, specially in flat regions. Consequently, a strategy is introduced in [62], named Region-Aware Adversarial Learning, 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 deviation is computed. Subsequently, a binary mask can be formulated as:

$$M_{i,j} = \begin{cases} 1 & \text{if} \quad \sigma\left(p_{i,j}\right) \geq \delta \\ 0 & \text{if} \quad \sigma\left(p_{i,j}\right) < \delta \end{cases}, \tag{3.10}$$

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 Figure 3.10.

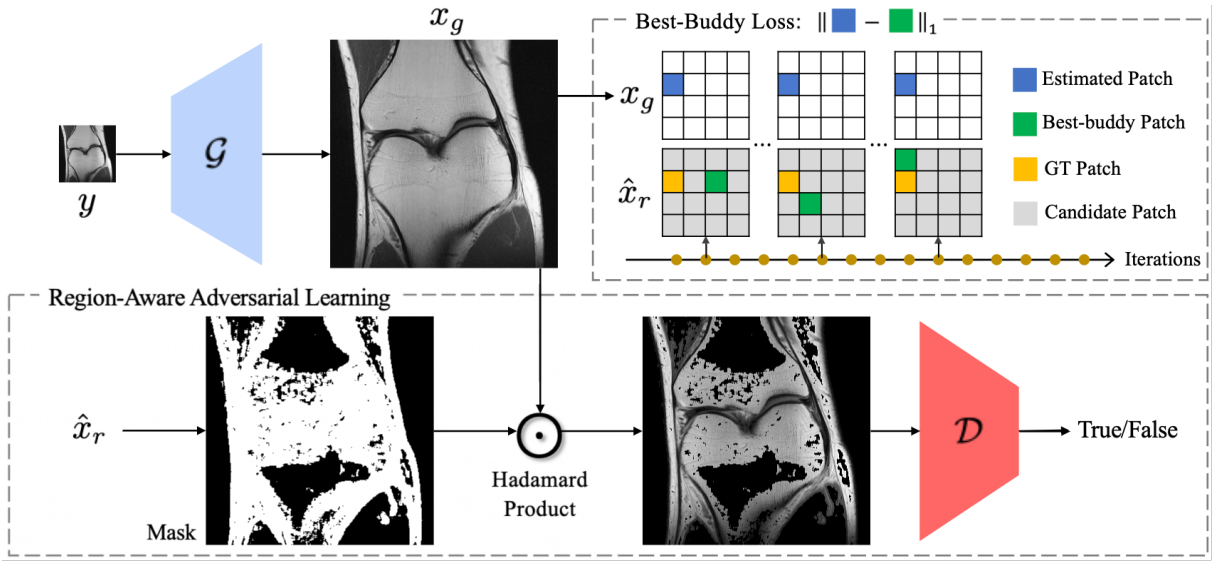


Figure 3.10: Scheme of the Beby-GAN framework. Figure adapted from [62].

Ultimately, Beby-GAN borrows a pre-trained ESRGAN generator architecture [49] 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 3.1. 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 [63] 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 [64]. Then, employs the Spectral Normalization (SN) regularization [65] 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, 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 Figure 3.11. 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.

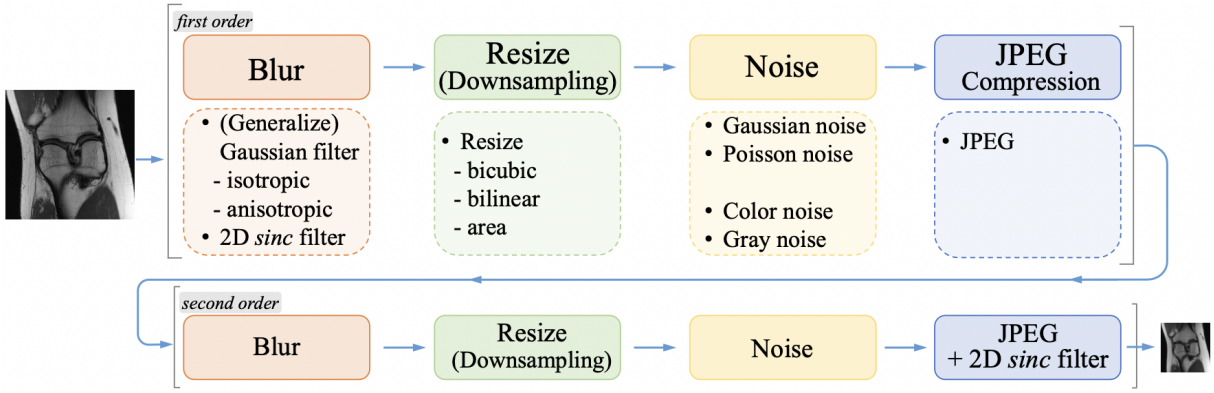


Figure 3.11: High-order Degradation Model. Figure adapted from [63].

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 Figures 3.1 and 3.3. 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 [64], the VGG-style discriminator in ESRGAN is improved to a U-Net design with skip connections as depicted in Figure 3.12, 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

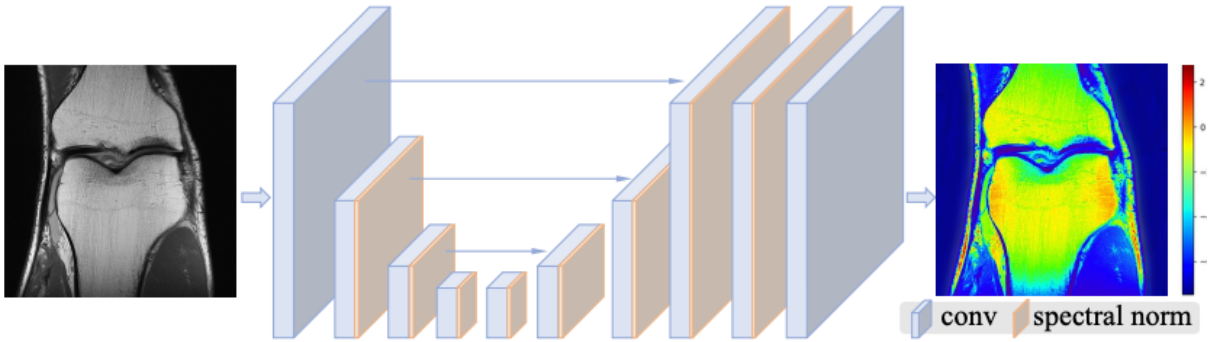


Figure 3.12: U-Net discriminator architecture with Spectral Normalization. Figure adapted from [63].

alleviate the over-sharp and unpleasant artifacts introduced by GAN training.

Real-ESRGAN outperforms previous approaches (e.g. ESRGAN [49] and BSRGAN [60]) in both artifacts suppression and restoring texture details by local detail enhancement.

3.8 Learning Strategies

This section discusses learning strategies utilized in super-resolution. Furthermore, a concise comparison of the numbers of parameters and generator losses from each GAN model regarded in this work is given in Table 3.1.

Table 3.1: Comparison of GAN-based SR models. $\mathcal{L}_{\mathcal{P}}$ represents the perceptual loss, $\mathcal{L}_{\mathcal{G}}$ the adversarial loss, $\mathcal{L}_{\mathcal{R}}$ the rank-content loss, $\mathcal{L}_{\mathcal{CYC}}$ the cyclic loss, $\mathcal{L}_{\mathcal{BB}}$ the best-buddy loss, $\mathcal{L}_{\mathcal{TV}}$ 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	$\mathcal{L}_{\mathcal{P}} + \lambda\mathcal{L}_{\mathcal{G}}$
ESRGAN	16.7M	$\mathcal{L}_{\mathcal{P}} + \lambda\mathcal{L}_{\mathcal{G}} + \eta\mathcal{L}_1$
RankSRGAN	1.55M	$\mathcal{L}_{\mathcal{P}} + \lambda\mathcal{L}_{\mathcal{G}} + \eta\mathcal{L}_{\mathcal{R}}$
SRResCycGAN	380k	$\mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\mathcal{TV}} + \lambda\mathcal{L}_1 + \eta\mathcal{L}_{\mathcal{CYC}}$
BSRGAN	16.7M	$\mathcal{L}_{\mathcal{P}} + \lambda\mathcal{L}_{\mathcal{G}} + \eta\mathcal{L}_1$
Beby-GAN	16.7M	$\lambda\mathcal{L}_{\mathcal{BB}} + \eta\mathcal{L}_{\mathcal{BP}} + \theta\mathcal{L}_{\mathcal{P}} + \phi\mathcal{L}_{\mathcal{G}}$
Real-ESRGAN	16.7M	$\mathcal{L}_{\mathcal{P}} + \lambda\mathcal{L}_{\mathcal{G}} + \eta\mathcal{L}_1$

3.8.1 Perceptual Loss $(\mathcal{L}_{\mathcal{P}})$

Proposed by Johnson et al. [66] 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, favours 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:

$$\mathcal{L}_{\mathcal{P}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\mathcal{VGG}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\phi\left(\hat{x}_{r_i}\right) - \phi\left(x_{g_i}\right)\right\|_2^2, \tag{3.11}$$

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 $\phi(\cdot)$ denotes the image feature maps obtained by some convolution layer within the VGG19 network [38].

3.8.2 Adversarial Loss $(\mathcal{L}_{\mathcal{G}})$

The standard GAN loss function introduced by Goodfellow et al. [6] 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:

$$\min_{\theta_{\mathcal{G}}}\max_{\theta_{\mathcal{D}}}\mathbb{E}_{x_r}[\log(\mathcal{D}_{\theta_{\mathcal{D}}}(x_r))] + \mathbb{E}_{y}[\log(1-\mathcal{D}_{\theta_{\mathcal{D}}}(\mathcal{G}_{\theta_{\mathcal{G}}}(y)))], \tag{3.12}$$

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 $\mathcal{D}_{\theta_{\mathcal{D}}}(x_r)$ 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, $\mathcal{D}_{\theta_{\mathcal{D}}}(\mathcal{G}_{\theta_{\mathcal{G}}}(y)))$ is the discriminator's estimate of the probability that a generated image is real. Moreover, $\theta_{\mathcal{G}}$ and $\theta_{\mathcal{D}}$ denote the weights and biases that parameterize the generator network $\mathcal{G}$ and discriminator network $\mathcal{D}$, respectively.

The generator and discriminator are jointly optimized with the objective given in function (3.12). 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 (3.12), 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:

$$\mathcal{L}_{\mathcal{G}} = \frac{1}{N}\sum_{i=1}^{N} -\log(\mathcal{D}_{\theta_{\mathcal{D}}}(\mathcal{G}_{\theta_{\mathcal{G}}}(y_i))), \tag{3.13}$$

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 equations (3.2) and (3.3).

3.8.3 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. 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:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} \left\| \mathcal{G}(y_i) - \hat{x}_{r_i} \right\|_1 , \tag{3.14}$$

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:

$$\mathcal{L}_2 = \frac{1}{N}\sum_{i=1}^{N} (\mathcal{G}(y_i) - \hat{x}_{r_i})^2, \tag{3.15}$$

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 favoured over $\mathcal{L}_2$ in image-to-image translations. Looking at Table 3.1 it is evident the preferable usage of $\mathcal{L}_1$ over $\mathcal{L}_2$, for instance in ESRGAN [49], SRResCycGAN [57], BSRGAN [60] and Real-ESRGAN [63]. 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.

3.8.4 Rank-content Loss ($\mathcal{L}_{\mathcal{R}}$)

After the Ranker $\mathcal{R}$ is trained through the learning to rank approach [67], a ranking score s of a generated image x_g can be estimated. Therefore, the rank-content loss can be formulated as:

$$\mathcal{L}_{\mathcal{R}} = \frac{1}{N}\sum_{i=1}^{N}\sigma(\mathcal{R}(\mathcal{G}(y_i))), \tag{3.16}$$

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.

3.8.5 Cyclic Loss ($\mathcal{L}_{\mathcal{CYC}}$)

Used with generative adversarial networks that perform unpaired image-to-image translation. Cyclic loss 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.

$$\mathcal{L}_{\mathcal{CYC}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{G}_{LR}(\mathcal{G}_{HR}(y_i)) - y_i\right\|_1. \tag{3.17}$$

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:

$$\begin{aligned}\mathcal{G}_{LR}(\mathcal{G}_{HR}(y)) &\approx y\\ \mathcal{G}_{HR}(\mathcal{G}_{LR}(x_g)) &\approx x_g.\end{aligned} \tag{3.18}$$

3.8.6 Best-Buddy Loss ($\mathcal{L}_{\mathcal{BB}}$)

Employed to alleviate the immutable one-to-one constraint and take into account the inherent uncertainty of SISR. Best-buddy loss 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:

$$\mathcal{L}_{\mathcal{BB}} = \frac{1}{NP}\sum_{i=1}^{N}\sum_{j=1}^{P}\left\|p_{g_{i,j}} - p_{BB_{i,j}}\right\|_1, \tag{3.19}$$

where $p_{g_{i,j}}$ represents a fake generated patch from the estimated image x_{g_i} and $p_{BB_{i,j}}$ 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.

3.8.7 Total-variation Loss $(\mathcal{L}_{\mathcal{TV}})$

Occasionally, MRI images can contain a significant amount of noise due to radiofrequency pulses and coils, field strength, or receiver bandwidth. Super-resolving a noisy LR image results in noisy HR image, as super-resolution 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 [68]. 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:

$$\begin{aligned}\mathcal{L}_{\mathcal{TV}} = \frac{1}{N}\sum_{i=1}^{N}(\|\nabla_h \mathcal{G}\left(y_i\right) - \nabla_h\left(\hat{x}_{r_i}\right)\|_1 + \\ \|\nabla_v \mathcal{G}\left(y_i\right) - \nabla_v\left(\hat{x}_{r_i}\right)\|_1),\end{aligned} \tag{3.20}$$

where ∇_h and ∇_v represent the horizontal and vertical gradients of the images, respectively. An image gradient is a directional change in the intensity or color of an image, as shown in Figure 3.13.

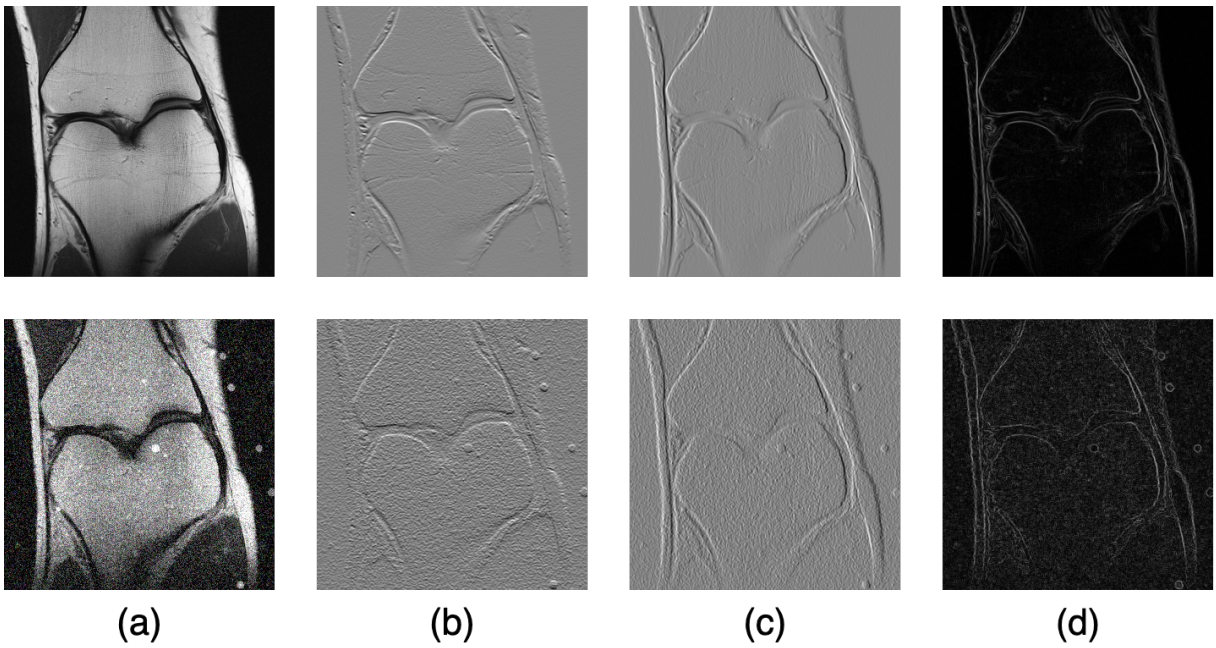


Figure 3.13: (a) MRI images, (b) image vertical gradients, (c) image horizontal gradients and (d) image gradient magnitudes.

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:

$$\begin{aligned} \nabla_h I &= I(i, j+1) - I(i, j-1) \\ \nabla_v I &= I(i+1, j) - I(i-1, j), \end{aligned} \tag{3.21}$$

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 favour the generator to optimize in the direction of results with reduced noise level. As shown in Figure 3.13, image gradients will as well detect noise, thus heavily penalizing noisy images that introduce artifacts that are not present in the ground truths.

3.8.8 Batch Normalization $(\mathcal{BN})$

Training deep neural networks is challenging. These networks suffer from gradient vanishing [69], 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.

$$\begin{aligned} \mathcal{BN}(z) &= \gamma \cdot \frac{z - \hat{\mu}_{\mathcal{B}}}{\hat{\sigma}_{\mathcal{B}}} + \beta, \\ \hat{\mu}_{\mathcal{B}} &= \frac{1}{N} \sum_{z \in \mathcal{B}} z, \\ \hat{\sigma}^2_{\mathcal{B}} &= \frac{1}{N} \sum_{z \in \mathcal{B}} (z - \hat{\mu}_{\mathcal{B}})^2 + \epsilon, \end{aligned} \tag{3.22}$$

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 Figure 3.2), thus hampering a stable performance over training. Furthermore, BN layers also enlarges computational complexity and memory usage.

3.8.9 Spectral Normalization $(\mathcal{SN})$

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 $L : h_{in} \rightarrow h_{out}$ and limiting the ability of weight matrices W_i to amplify inputs in any direction. By definition, the Lipschitz norm is defined as:

$$\|L\|_{Lip} = \sup_{h} \sigma(\nabla L(h)), \tag{3.23}$$

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

$$\sigma(W) = \max_{h:h\neq 0} \frac{\|Wh\|_2}{\|h\|_2} = \max_{\|h\|_2 \leq 1} \|Wh\|_2, \tag{3.24}$$

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:

$$\|L\|_{Lip} = \sup_{h} \sigma(\nabla L(h)) = \sup_{h} \sigma(W). \tag{3.25}$$

If the Lipschitz constant of activation functions $\|a_i\|_{Lip} = 1$, then equation (3.25) 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 equation (3.25) to be rewritten as:

$$\|L\|_{Lip} = \sup_{h} \sigma(W) = \sigma(W). \tag{3.26}$$

Spectral normalization normalizes the spectral norm of the weight matrix so it satisfies the Lipschitz

constraint $\sigma(W) = 1$:

$$\bar{W}_{\mathcal{SN}}(W) = \frac{W}{\sigma(W)}. \tag{3.27}$$

Accordingly, normalizing parameters of each layer with equation (3.24), 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:

$$\sigma(\bar{W}_{\mathcal{SN}}(W_i)) = 1. \tag{3.28}$$

3.9 Implementation Details

3.9.1 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 equation (3.2) with $\lambda = 5 \times 10^{-3}$ and $\eta = 1 \times 10^{-2}$. The optimizer employed is Adam [70] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Implemented with PyTorch framework and trained over DIV2K [71] and Flickr2K [72] 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.

3.9.2 RankSRGAN

Regarding the Ranker network, the small constant ϵ present in the margin-ranking loss function (3.4) is set to 0.5. The weights are initialized with a method described in He et al. [73]. Moreover, the ranker is trained over DIV2K [71] and Flickr2K [72] 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 [70] is used with a weight decay of 1×10^{-4}. The learning rate is set to 1×10^{-3} and is decayed with a factor of 2 every 1×10^5 iterations.

Concerning the pre-trained RankSRGAN network, the training is carried out with a mini-batch size of 8. The optimization target is defined in Table 3.1, where $\lambda = 5 \times 10^{-3}$ and $\eta = 3 \times 10^{-2}$. To optimize the network, the Adam optimizer [70] 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 [71] dataset.

3.9.3 SRResCycGAN

The training phase is carried out with a batch size of 16 over 51×10^3 iterations. For optimization, the Adam optimizer [70] is employed with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and no weight decay. The optimization target is the loss function in equation (3.7). 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 [74]. Ultimately, the estimated noise standard deviation σ (projection layer parameter) is computed according to [75].

3.9.4 BSRGAN

Pre-trained with batch size of 48 over a unified dataset including DIV2K [71], Flick2K [72], WED [76] and FFHQ [77]. BSRGAN is implemented with PyTorch and trained by minimizing a weighted combination of losses, as shown in Table 3.1, where $\lambda = 0.1$ and $\eta = 1$. For optimization, the Adam optimizer [70] is employed with a fixed learning rate of 1×10^{-5}.

3.9.5 Beby-GAN

Training performed over DIV2K [71] and Flickr2K [72] datasets. Batch size of 8 for 6×10^5 iterations. The pre-trained model is optimized via Adam [70] 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, the equation (3.8) 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 3.1, where $\lambda = 0.1$, $\eta = 1$, $\theta = 1$ and $\phi = 5 \times 10^{-3}$. Additionally, it is implemented with PyTorch.

3.9.6 Real-ESRGAN

Since the same generator architecture from ESRGAN [49] 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 [70] 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 [71], Flickr2K [72] and OutdoorSceneTraining [78] datasets. For optimization, the equation in Table 3.1 is minimized, where $\lambda = 0.1$ and $\eta = 1$.

3.10 Summary

This chapter comprehensively reviewed state-of-the-art GAN methods in chronological order, from the original SRGAN to the recent Beby-GAN and Real-ESRGAN, which manifest improvements over many datasets. Additionally, some approaches novelty lied on the proposal of new loss functions or degradation models, while borrowing the architecture of other approaches as their basic framework. For instance, the SRGAN architecture was massively adopted by many approaches, manifesting substantial value as a backbone for Super-Resolution.

Furthermore, learning strategies employed to optimize Super-Resolution GANs were described. It is important to apprehend and retain Table 3.1, where the generator losses are exhibited for each approach. From the table it is evident the high usage of the perceptual loss in Super-Resolution GAN-based methods, which favours the generation of images with natural image statistics.

Ultimately, it was reported the implementation details employed for every approach in the MRI Super-Resolution experiments. The following chapter 4 will describe them and present the results.

4

Super-Resolution Experiments

Contents

This chapter defines the basic methodology used to perform and evaluate Super-Resolution. To proceed on experiments the first step is to acquire a dataset with LR-HR image pairs. Afterwards, these pairs are fed into SR networks, which are trained or tested over the data. An alternative to attain LR-HR pairs is described in the next section 4.1.1. Subsequently, after super-resolving every LR image, it is essential to evaluate the SR performance, thus popular image quality metrics are discussed in Section 4.2.

4.1 Data

4.1.1 FastMRI Dataset

To test every GAN method mentioned in chapter 3 the FastMRI dataset [14] was employed. 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. In this thesis only the knee collection is considered, for instance, 973 volumes were used from the single-coil knee training set.

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 Figure 4.1. 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 see. Consequently, having such heterogeneous types of scans mixed can have an impact in the generalization capability of GANs preventing them from converging into an optimal solution, hence robustness during training and testing is a significant factor.

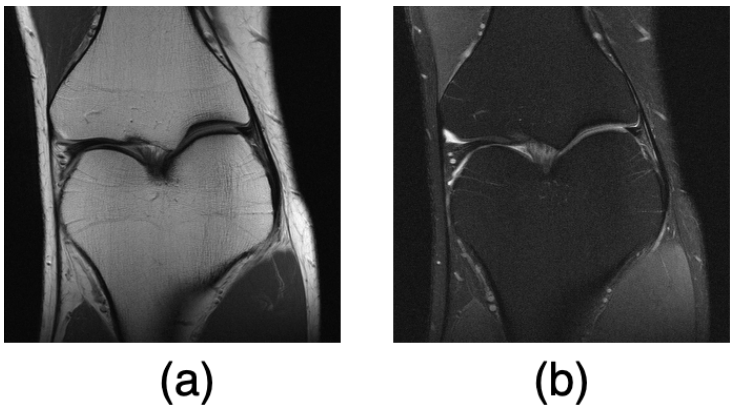


Figure 4.1: A proton-density weighted image without fat suppression (a) and with fat suppression (b).

4.1.2 Image Preprocessing

Ground truth images from the FastMRI single-coil knee test set are not publicly released to ensure that models do not overfit the data from this set. Therefore, only the training set was used during this work. Pre-trained models were used directly, hence the overfitting issue was not considered, since exclusively the test phase was performed. These models were trained on several datasets, such as DIV2K and Flickr2K (see Table 4.2).

The training set yields, for each slice, the k-space data and the corresponding ground truth. To evaluate the super-resolution 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. At the beginning of the test phase each k-space data of every MRI slice is downsampled through bicubic interpolation with a downscale factor of $\times 4$, resulting in LR-HR pairs holding the downsampled k-space data and the ground truth (reconstructed from fully-sampled multi-coil acquisitions using the simple root-sum-of-squares method [79]).

4.2 Image Quality Metrics

Several Image Quality Metrics (IQMs) 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 favour specific image characteristics, such as the losses discussed in Section 3.8.

4.2.1 Mean Squared Error (MSE)

Among the many IQM used to evaluate the HR image quality, Mean Squared Error (MSE) is the most popular metric. 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:

$$MSE = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} (\hat{x}_r(i,j) - x_g(i,j))^2, \tag{4.1}$$

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

4.2.2 Peak Signal-to-Noise Ratio (PSNR)

It 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 in the following equation:

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX_I}{RMSE(\hat{x}_r, x_g)} \right), \tag{4.2}$$

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 have visually unsatisfying dissimilarities but hold a high PSNR score (see Figure 4.2). 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.

Figure 4.2: Misleading PSNR values.

4.2.3 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. [80] 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 [81]. Evidently, from a human visual perspective, SSIM is comparatively better than MSE and PSNR. SSIM can be defined as

follows:

$$SSIM = \frac{\left(2\mu_{\hat{x}_r}\mu_{x_g} + c_1\right)\left(2\sigma_{\hat{x}_r x_g} + c_2\right)}{\left(\mu_{\hat{x}_r}^2 + \mu_{x_g}^2 + c_1\right)\left(\sigma_{\hat{x}_r}^2 + \sigma_{x_g}^2 + c_2\right)}, \tag{4.3}$$

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 [80].

4.2.4 Other Relevant Metrics

A few other metrics used to assess image quality are Mean SSIM (MSSIM) [80], Natural Image Quality Evaluator (NIQE) [52], Universal Image Quality Index (UQI) [82], Feature Similarity Index Matrix (FSIM) [83], Gradient Similarity measure (GSM) [84], and Task-based Evaluation.

4.3 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 [85], 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 super-resolution 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, most of the pre-trained models available were trained over diverse data from exhaustive datasets, thus they learn to estimate the distribution of real-world images holding photo-realistic details. Therefore, in this work, pre-trained models are applied directly in the reconstruction task. In essence, the training phase with FastMRI is skipped with the idea that the robustness of each pre-trained model will be evaluated by performing the target task. The training details for each pre-trained model present in this work are outlined in Section 3.9.

Using pre-trained models in real-world applications can be substantially good, especially with real MRIs, since its degradation is usually unknown. Evidently, a robust model would serve the best value in MRI acceleration.

4.4 Model Issues

4.4.1 GAN Noise

Noisy results were a significant problem 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 3.8.7). 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 [86] and Block Matching 3D [87].

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:

$$\begin{aligned} NLM(p, c\,; I) &= \frac{1}{K(p)} \sum_{b \in B(p,r)} I_c(b) w(p, b), \\ K(p) &= \sum_{b \in B(p,r)} w(p, b), \end{aligned} \tag{4.4}$$

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 image I and it defines the width and height of the search zone. 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:

$$\begin{aligned} d^2 &= d^2(B(p, f), B(b, f)) = \\ &= \frac{1}{3} \sum_{c=1}^{3} \frac{1}{(2f+1)^2} \left\| B(p, f) - B(b, f) \right\|_2^2 = \\ &= \frac{1}{12f^2 + 12f + 3} \sum_{c=1}^{3} \sum_{i=1}^{(2f+1)^2} (I_c[B(p, f)](i) - I_c[B(b, f)](i))^2, \end{aligned} \tag{4.5}$$

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:

$$w(p, b) = e^{-\frac{\max\left(0,\, d^2 - 2\sigma^2\right)}{h^2}}, \tag{4.6}$$

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

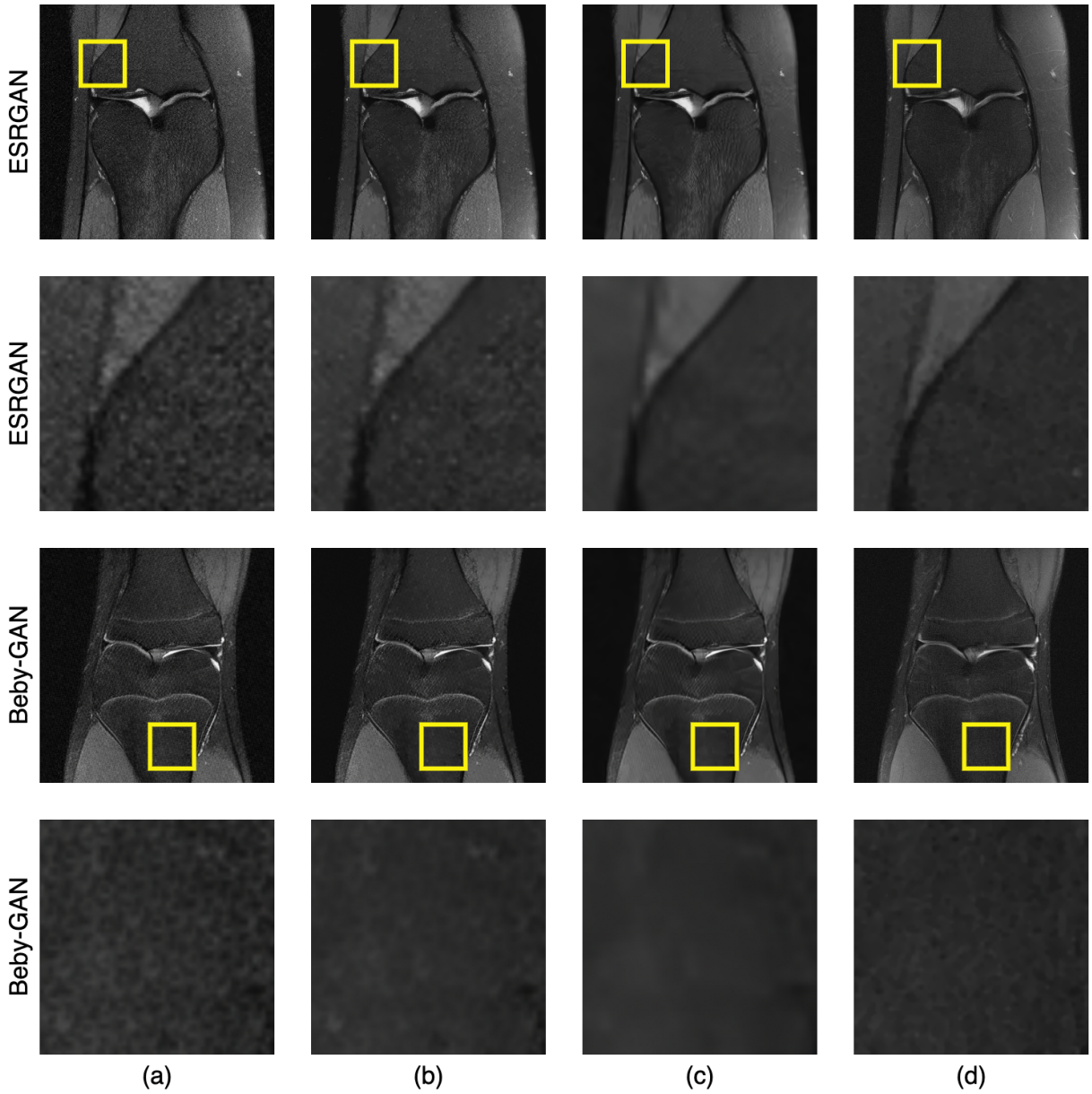


Figure 4.3: (a) Generated Images w/o denoising, (b) Generated Images w/ NLM, (c) Generated Images w/ BM3D and (d) Ground Truth Images.

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.

Generally, the denoising step improved the performance of the super-resolution task by $\approx 20\%$ over SSIM metric (results can be seen in Table 4.2). From Figure 4.3 it is possible to observe the noise correction and the reduction of the checkerboard artifact pattern, common in GANs. This pattern results from upsampling and downsampling layers [88]. For instance, deconvolution layers reverse the convolution operation, however they may introduce upsampling artifacts [89]. 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 consider them as fake sustains substantial value by further aligning the generator in the direction of photo-realistic textures.

In this work NLM was conducted with filter strength $h = 4$, search zone size equal to 51×51 ($r = 10$) and with color patches' size of 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 two methods.

Moreover, as explained in Section 3.8.7, a total variation loss component is an alternative solution to prevent noisy results.

4.4.2 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 the mean pixel value (MPV) of the image is computed along with the mean pixel value difference (MPVD) between ground truths and generated images. MPV and MPVD were calculated for every LR-HR pair and afterwards averaged. For every pair the computations were performed over the entire image (Globally) or over a central section (Locally), for instance with a center crop factor of 0.2. Values obtained can be seen in Table 4.1.

The noise step manifests improvements concerning pixel value differences, thus suggesting that generated images are closer to their corresponding ground truths. Also, the MPV can be misleading

Table 4.1: Pixel value statistics. Red color indicates the largest deviation and Green color the smallest.

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
ESRGAN w/ NLM	50.08	11.82	46.78	10.75
ESRGAN w/ BM3D	49.62	11.05	46.23	9.62
RankSRGAN	51.29	11.50	47.88	10.29
RankSRGAN w/ NLM	51.19	10.86	47.87	9.62
RankSRGAN w/ BM3D	50.77	10.52	47.35	9.11
SRResCycGAN	51.60	10.27	48.29	8.67
SRResCycGAN w/ NLM	51.56	10.19	48.37	8.56
SRResCycGAN w/ 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
Beby-GAN w/ NLM	50.75	10.96	47.59	9.58
Beby-GAN w/ BM3D	50.32	10.64	47.05	9.04
Real-ESRGAN	49.02	11.52	44.93	9.93

despite its ability to detect slight pixel value deviations whenever the means between fake images and ground truths do not match. The rationale behind this is that the MPVs between two dissimilar images can be the same while pixel values are not, i.e., completely different images can have the same mean pixel value. Therefore having a Global MPV closer to the Global MPV of the ground truths does not necessarily mean the generated images are better. Therefore, MPVD is evidently a better evaluation measure, especially because it is analogous to RMSE. Both measures can be calculated by the following equations:

$$MPV = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} I(i,j), \tag{4.7}$$

$$MPVD = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} \left|\hat{x}_r(i,j) - x_g(i,j)\right|. \tag{4.8}$$

Moreover, the background of the fake images was exhibiting some dissimilarities, thus images computed from the difference between the generated image and the corresponding ground truth were plotted to further analysis. From the resulting images it was evident there is no pixel value deviation present, since every image was following the MRI outline/shape. This suggests that the MPVD is a consequence of a non optimal reconstruction and not of pixel value anomalies. In the presence of pixel value deviation the resulted difference would not show any shapes. If the deviations are exclusively in the sharp edges, then it is a problem related with high-frequency details reconstruction. Additionally, in case of pixel misalignment the MRI outline would be doubled when the differences are displayed, however in this circumstance there is no manifestation of pixel misalignment, as shown in Figure 4.4. Therefore, the

background differences are due to the LR downsampling, which inherently changes the background luminance and introduces noise. Furthermore, to correct some minor color issues in ESRGAN, a grayscale step was carried out before conducting the denoise step.

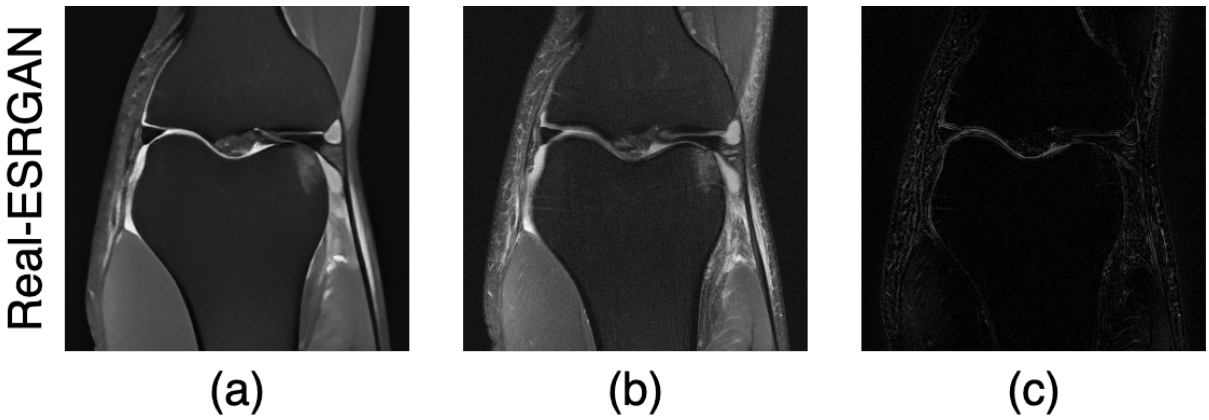


Figure 4.4: (a) Generated Image w/o denoising, (b) Ground Truth Image and (c) Difference.

4.4.3 Over Smoothing

BSRGAN and Real-ESRGAN showed excessive smoothing, consequently a few ground truth texture information was unrecovered. Looking at Table 3.1, ESRGAN, BSRGAN and Real-ESRGAN have equal generator loss functions, however ESRGAN did not exhibit overly smooth results. The reason is that ESRGAN was trained with less weight given to the content loss component (see Section 4.3). As mentioned in Section 3.8, optimizing the network with content loss as a main optimization target can lead to overly smooth results, because content loss is highly correlated with the pixel-to-pixel differences and these differences are poorly correlated with perceptual quality.

Additionally, an inappropriate degradation level can cause over smoothing, as models may expect the same level of degradation as the one used during training [60]. Models' robustness is a significant factor to avoid this phenomenon [60, 90–92].

4.5 Quantitative Results

All experiments were conducted on Google Colab using an Intel Xeon CPU with 2.20GHz and 13GB of RAM. Results can be seen in Table 4.2. For every method the LR images were obtained with bicubic downsampling and a scaling factor of $\times 4$. 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. Moreover, the scale column denotes the upscaling factor.

Table 4.2: Results Comparison. Red color indicates the worst performance overall and Green color the best. Gray color stands for the additional time derived from the denoise step.

Method	Input	Scale	Optimizer	Datasets	MSE	PSNR	SSIM	Time (ms)
ESRGAN	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	297.46	24.47	0.5939	4417
ESRGAN w/ NLM	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	283.30	24.82	0.6585	6574 (+2157)
ESRGAN w/ BM3D	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	252.28	25.58	0.7286	10600 (+6183)
RankSRGAN	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	266.94	24.99	0.6319	651
RankSRGAN w/ NLM	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	250.93	25.44	0.7057	2731 (+2080)
RankSRGAN w/ BM3D	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	235.78	25.89	0.7392	7371 (+6720)
SRResCycGAN	Bicubic	$\times 4$	Adam	AIM2020 RISR	228.00	25.94	0.7456	2602
SRResCycGAN w/ NLM	Bicubic	$\times 4$	Adam	AIM2020 RISR	227.32	25.98	0.7459	4780 (+2178)
SRResCycGAN w/ BM3D	Bicubic	$\times 4$	Adam	AIM2020 RISR	231.48	25.92	0.7442	8983 (+6381)
BSRGAN	Bicubic	$\times 4$	Adam	DIV2K, Flick2K, WED, FFHQ	254.11	25.33	0.7157	3652
Beby-GAN	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	264.76	25.11	0.6493	3819
Beby-GAN w/ NLM	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	251.02	25.50	0.7140	5853 (+2134)
Beby-GAN w/ BM3D	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K	236.78	25.91	0.7439	10113 (+6294)
Real-ESRGAN	Bicubic	$\times 4$	Adam	DIV2K, Flickr2K, OutdoorSceneTraining	274.40	24.99	0.7137	3715

As can be seen, MSE, PSNR, and SSIM suggest SRResCycGAN outperforms every other GAN-based method in recovering $\times 4$ downgraded images. Meanwhile, ESRGAN obtained the worst results in terms of performance metrics and image generation time. RankSRGAN holds the fastest reconstruction time followed by SRResCycGAN. Looking at Table 3.1 it is evident that methods with less parameters in the generator have as well a faster reconstruction time. The aforementioned is illustrated in Figure 4.5.

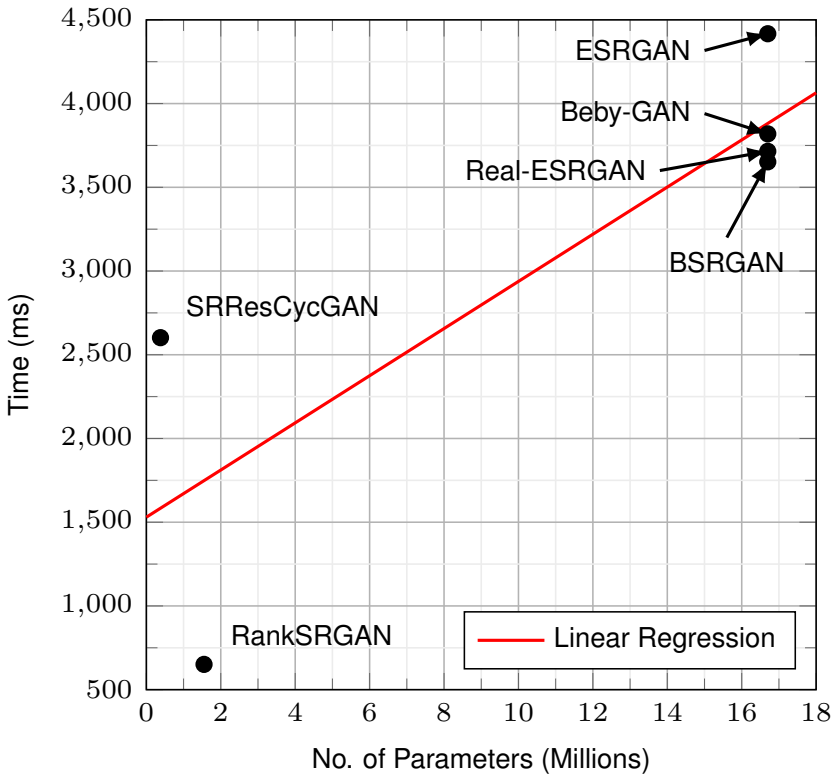


Figure 4.5: Number of Parameters vs Reconstruction Time.

Moreover, the denoising step yields high impact specially over SSIM. For instance in ESRGAN, the denoising holds improvements of $\approx 20\%$.

4.6 Qualitative Results

A qualitative analysis is conducted to further evaluate the MRI reconstruction quality of GANs. Figure 4.6 shows a comparative illustration of an MRI slice reconstructed over each super-resolution 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. Reasoning, standard quantitative measures, for instance, MSE, PSNR, and SSIM, fail to capture and accurately evaluate image 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. 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.

Figures A.1 and A.2 comparatively illustrate the reconstruction quality, exposing that generated images have sharper edges and richer textures. Looking at the 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 Figures 4.7 and 4.8, it is shown a comparative reconstruction evaluation between LR patches and the corresponding generated ones from every method with a denoising step using NLM (except for Real-ESRGAN and BSRGAN since the denoising step was not employed).

As stated before, SISR methods are usually sensitive to errors in the blur kernel. This is possibly the main reason Real-ESRGAN and BSRGAN are producing overly smooth results, as they are assuming a higher level of degradation in the LR images which is not present.

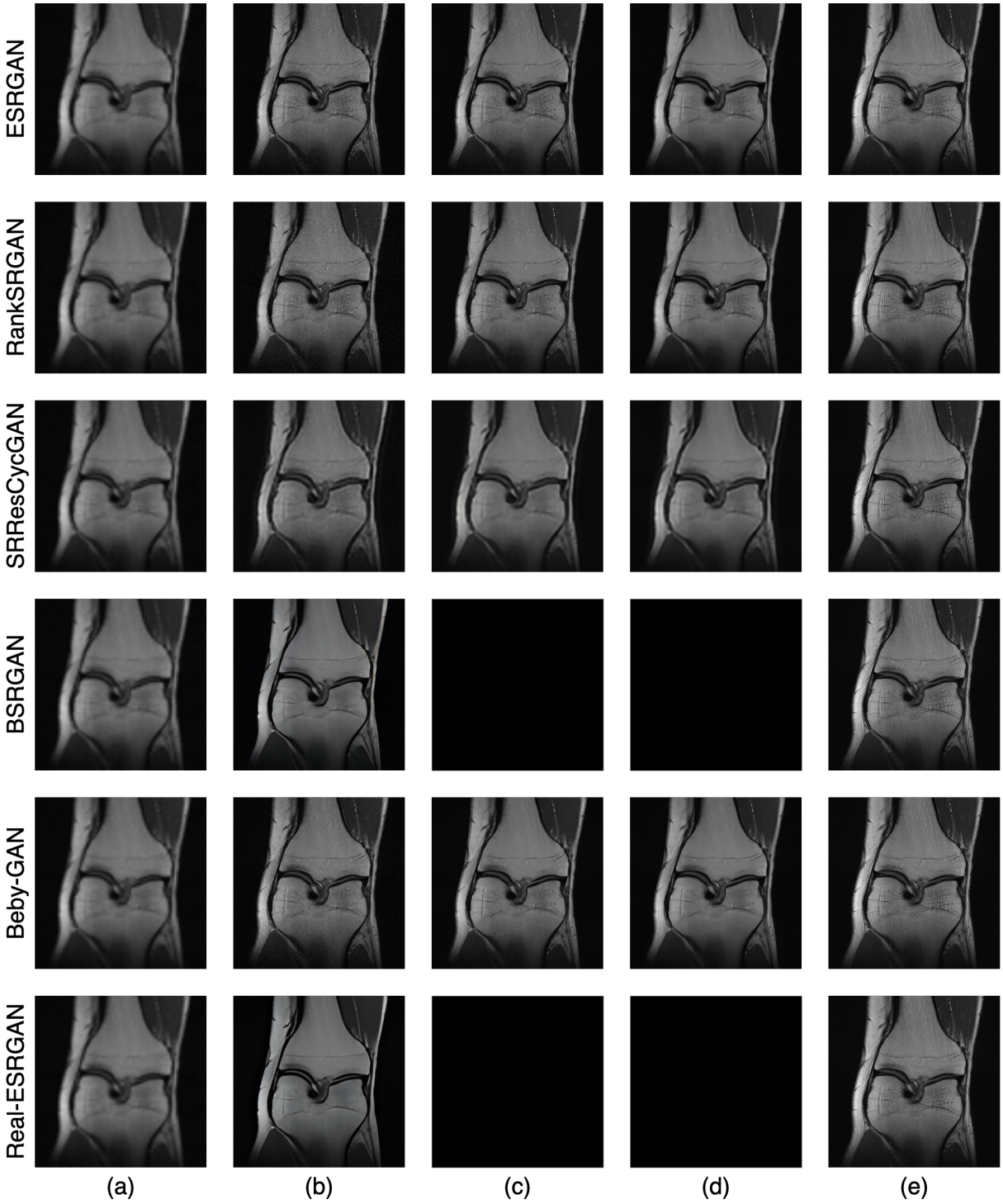


Figure 4.6: (a) Input LR Images, (b) Generated Images w/o denoising, (c) Generated Images w/ NLM, (d) Generated Images w/ BM3D and (e) Ground Truth Images.

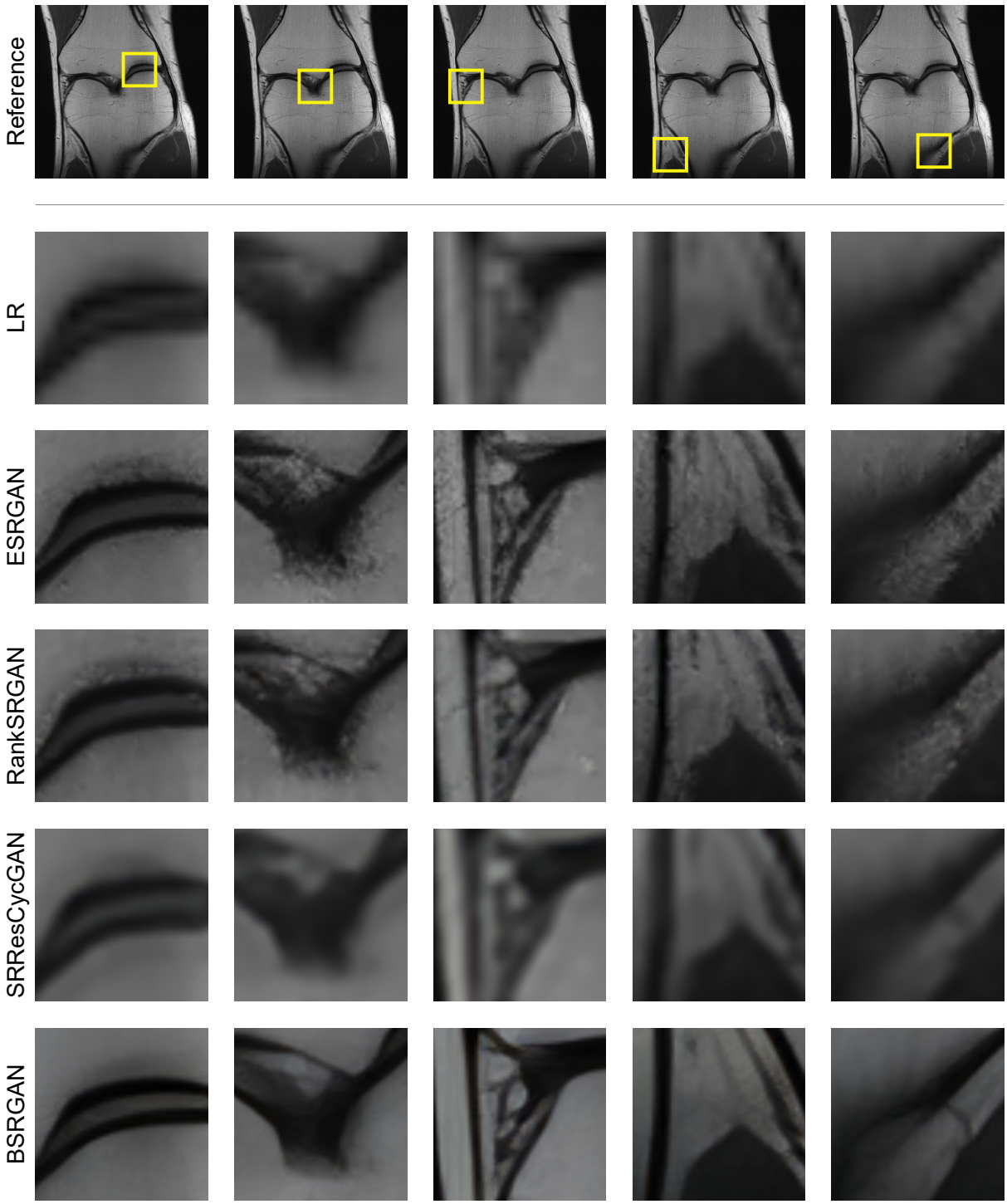


Figure 4.7: (Part 1) Comparison of Generated Images from every approach w/ NLM (except for BSRGAN and Real-ESRGAN).

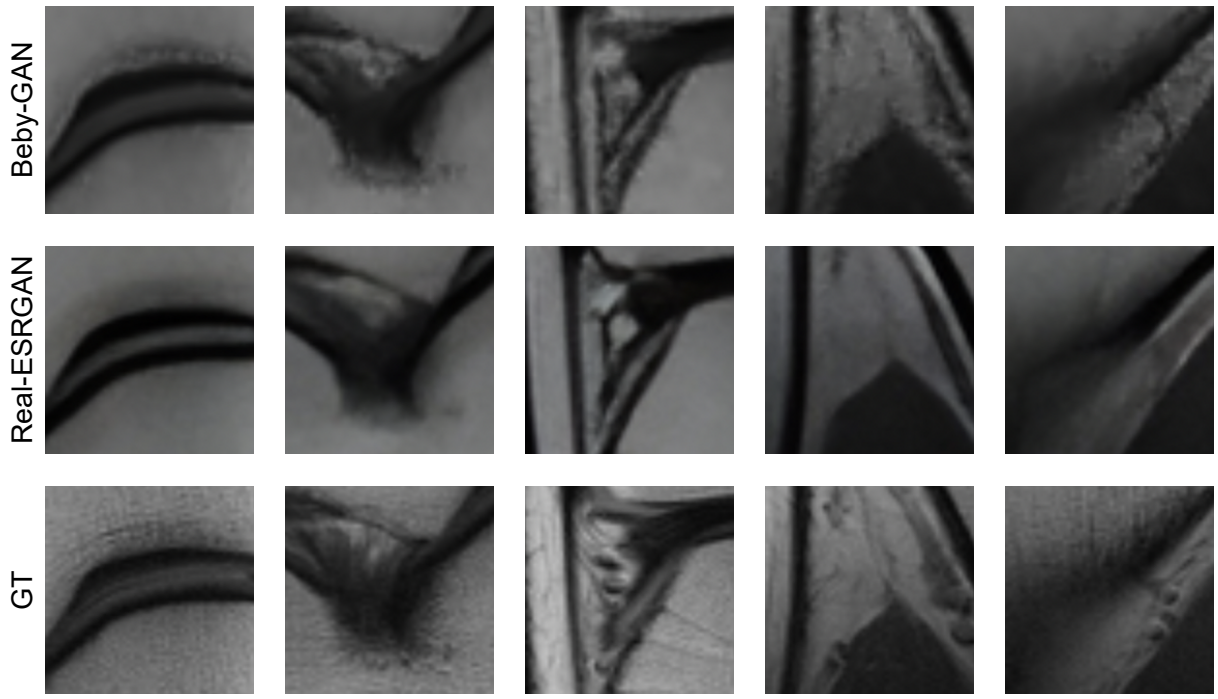


Figure 4.8: (Part 2) Comparison of Generated Images from every approach w/ NLM (except for BSRGAN and Real-ESRGAN).

4.7 Discussion

From the last chapter it was evident the high usage of the perceptual loss. 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 natural and perceptually pleasing results. However, this can be misleading in the medical imaging context, for instance MRI, since the reconstructed MRI may look natural and real, but not equal to the ground truth. This dissimilarity due to artifacts inclusion or omissions of relevant details can lead to erroneous conclusions. The same occurs in adversarial training with GANs, usually used to attain photo-realism. The discriminator predicts relative realness instead of the absolute value. Consequently, realistic fake patterns can be wrongly conjectured as real even if they are far from the ground truth. However, the function that perfectly recovers the target image might be impossible to estimate, since the reconstruction problem is inherently ill-posed, i.e., for any distorted image there can be multiple plausible solutions that would be perceptually pleasing. Therefore, GANs remain a solid candidate to spatially resolve MRIs and accelerate their acquisition.

Additionally, looking at Table 3.1, ESRGAN, BSRGAN and Real-ESRGAN have equal generator loss functions, however ESRGAN did not exhibit overly smooth results. The reason is that ESRGAN was trained with less weight given to the content loss component. Optimizing to the content loss usu-

ally 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 3.8). 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 super-resolution is employed. For example, approaches that hallucinate finer detail might be less suited for medical applications or surveillance.

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 a 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.

Differences on training and testing data domains have impact on the results. For instance, considering the image preprocessing adopted in this work, the models would produce worse and visually unpleasant results if the pre-trained models used in the testing experiments were trained with LR images computed by either simple or complex degradations far from bicubic downsampling.

Ultimately, despite quantitative results suggesting SRResCycGAN outperforms other popular deep learning methods in recovering $\times 4$ 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.

4.8 Summary

This chapter reviewed Super-Resolution experiments over an MRI dataset, described image quality metrics used to evaluate SR performance and discussed inherent problems faced during GAN-based Super-Resolution. Ultimately, provided an analysis on the results and exhibited the quantitative and qualitative experimental results.

In these experiments, pre-trained models were applied directly in the reconstruction task, thus skipping the training phase. The rationale, is that most pre-trained models available were intensively trained over diverse data from exhaustive datasets. They have learned to estimate the distribution of real-world images holding photo-realistic details.

Strategies to combat GAN noise problems were provided, since noisy results were a significant problem present in the majority of the methods considered in this work. Generally, the denoising solution employed improved the performance of the super-resolution task by $\approx 20\%$ over SSIM metric.

Furthermore, the perceptual loss can be misleading in the medical imaging context, for instance MRI reconstruction, since the generated MRI may look natural and real, but not equal to the ground truth. Additionally, optimizing to the content loss usually leads to unnatural and blurry reconstructions with low perceptual quality. Finding a balance between both optimization targets is the best option.

Ultimately, despite quantitative results suggesting SRResCycGAN outperforms other popular deep learning methods in recovering $\times 4$ downgraded images, qualitative results show Beby-GAN holds the best perceptual quality.

5

Tumor Segmentation Methods

Contents

5.1 Traditional Machine Learning Methods

Deep Learning has made tremendous progress in unstructured data, however on tabular data (structured), tree-based models are exceptional, especially if the amount of data available is limited (see Figure 2.7). The choice of features is very important. In Machine Learning (ML), particularly computer vision, some features are computed using refined filters that extract information of images/volumes upon convolution. These filters are calculated by dedicated algorithms, such as edge-based methods. Therefore, the classical segmentation techniques mentioned in Section 2.2.2 sustain a practical way to extract features and can be primarily regarded as feature extractors. Reasoning, machine learning algorithms can be employed and utilize the segmentation results as features.

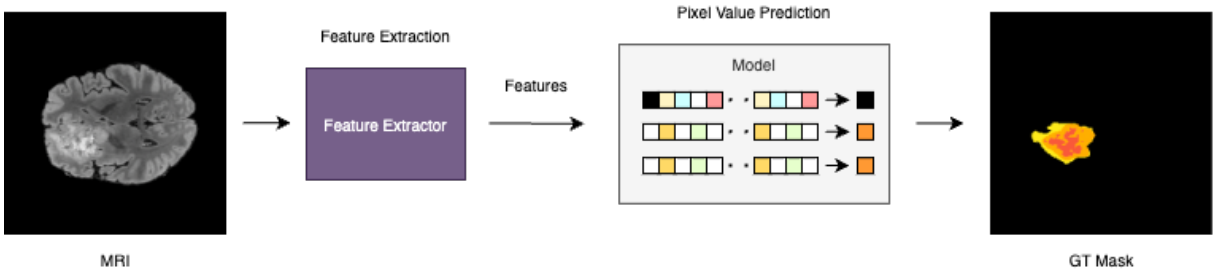


Figure 5.1: Main concept behind Semantic Segmentation with Traditional Machine Learning algorithms.

In essence, a clever strategy to perform semantic segmentation is by merging all the conventional techniques to extract a set of features. Afterwards, a traditional machine learning algorithm, such as Random Forest [93], Support Vector Machine [94], or XGBoost [95], can be trained with these features to recognize patterns and make pixel-level predictions.

Accordingly, a pixel-level dataset is built, where each pixel value of an image/volume is mapped into a feature vector. These vectors compromise a set of feature maps, of which the number equals to the length of the vectors as well as the total number of kernels (filters) utilized. Evidently, feature maps are generated by the application of filters (feature extractors) to the input image/volume. Mapping pixel data into the feature space alleviates the subsequent learning and generalization steps of the segmentation algorithms, as this higher dimensional space contains additional, unique, and refined information. This process consists in feature extraction, which is briefly discussed in Section 6.2. Essentially, the original pixel value intensity and the subsequent features generated from it are associated with the label of the target class corresponding to that pixel.

Feature vectors can be obtained by performing a transverse slice across the MRI stacked with the feature maps (see Figure 5.2). Subsequently, feature vectors are fed into the model, which will predict the pixel label based on the feature values of each pixel. During training, the model receives the feature representations consisting of the original pixel and the features extracted. The true/target pixel label

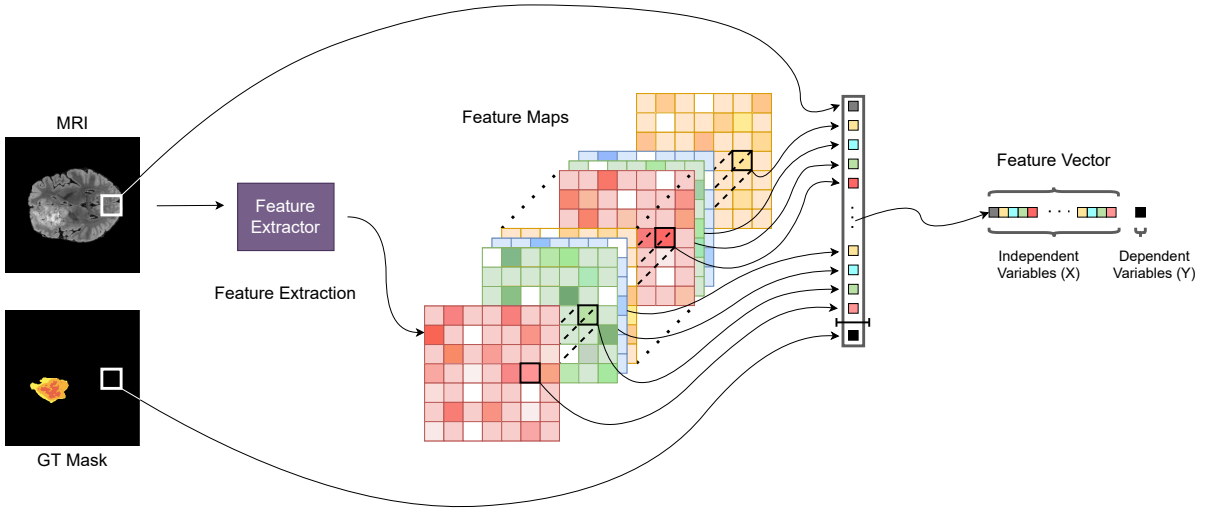


Figure 5.2: Tumor Segmentation methodology using Traditional Machine Learning methods. The resolution of the feature maps is reduced for visualization purposes. In reality, the feature maps resolution should match the original MRI resolution.

is also provided to compute the loss and optimize the model. Evidently, during inference and testing, the model is fed with the feature vectors. However, this time the true label is absent since the task of the model consists in predicting the label for every pixel. Reasoning, this aligns with the conventional methodology of supervised learning.

Additionally, flattening the ground truth segmentation mask and every feature, including the MRI, manifests an alternative to ease comprehension and alleviate the complexity around the transverse slice.

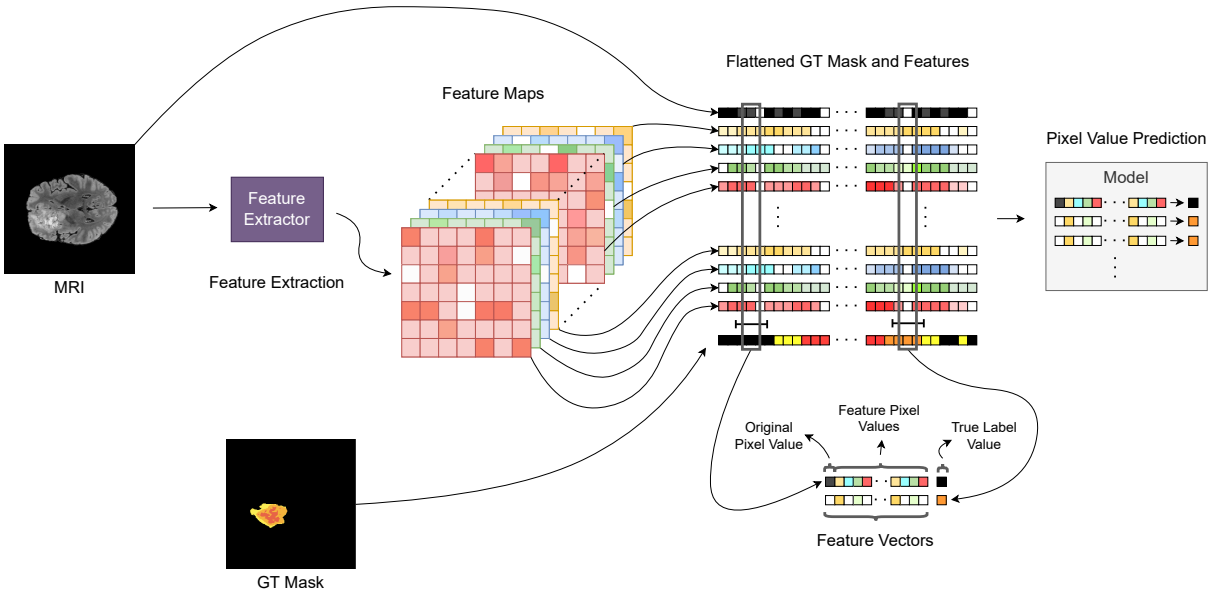


Figure 5.3: Alternative representation of the Tumor Segmentation methodology using Traditional Machine Learning methods.

Figure 5.3 depicts a flattening of the original MRI, the ground truth segmentation mask, and each feature map resultant from the convolution with feature extractor filters. Reasoning, each feature map can be regarded as a column confined within tabular data. Therefore, having all features stacked vertically, similar to Figure 5.3, induces the concept of a transposed tabular data. Thus it is evident that a row, which includes the feature vector, can be extracted with a simple vertical slice. Accordingly, by reverting the transposition (not essential), the tabular data is consummated, and traditional machine learning algorithms that work properly with structured data can be exploited.

Table 5.1: Example of tabular data utilized by Traditional Machine Learning methods for Tumor Segmentation

Feature Vectors					
Original Pixel Value	Feature 1	Feature 2	...	Feature N	Label
45	0	1	...	255	0
100	2	0	...	0	1
180	1	0	...	0	0
240	2	1	...	0	4
120	1	0	...	255	2
...	...	...	...	...	...

Nonetheless, although the prediction is computed pixel by pixel, independently of the neighboring pixels, since features are theoretically extracted through sliding kernels (filters) with sizes often larger than 1×1 (for 2D use cases), then the context of the surrounding pixels is marginally taken into consideration, i.e., each pixel will partially have its local spatial context information encoded in the feature maps. Additionally, to extract feature maps, three-dimensional kernels can be preferred over the traditional two-dimensional kernels, as these 3D filters can extract supplementary information by considering a wider perimetral neighborhood of pixels to encode the local context. These higher-dimensional filters concede a means to take advantage of volumetric data, thus manifesting improvements in the processing of biomedical images, such as magnetic resonance images. Additional contextual information about feature maps generation is given in Section 6.2.

Ultimately, the methodology described in this section can take substantial advantage of transfer learning, which corresponds to the use of previously acquired information to help solve an equal or related problem. Analogous to conventional segmentation techniques, CNNs make use of convolutional layers that utilize filters to help learn patterns and recognize important features in images. After training a CNN on a large dataset, a set of weights is learned internally, which fundamentally consist in filters that can be employed to extract additional features. Therefore, pre-trained deep learning models are suited to be reused in the processing and extraction of relevant features. As described in the following section, layers dedicated to extract features can be isolated and used conveniently.

5.2 Deep Learning Methods

From the previous section, it is evident that semantic segmentation often requires the extraction of features to represent sample images/volumes in a richer and more convenient manner, thus improving the model learning and pattern recognition processes. These extracted features compromise meaningful information derived from raw input data and can be inherently correlated with each other or even with the target variable. Accordingly, they consist in a fundamental element to uncover relationships and learn patterns present in the data.

Traditional Machine learning algorithms lie on external techniques to extract features. In contrast, Neural networks perform the extraction of features internally and can be conveyed as having two distinct internal parts, one for feature extraction and another for classification, thus these two processes are jointly handled by the same structure.

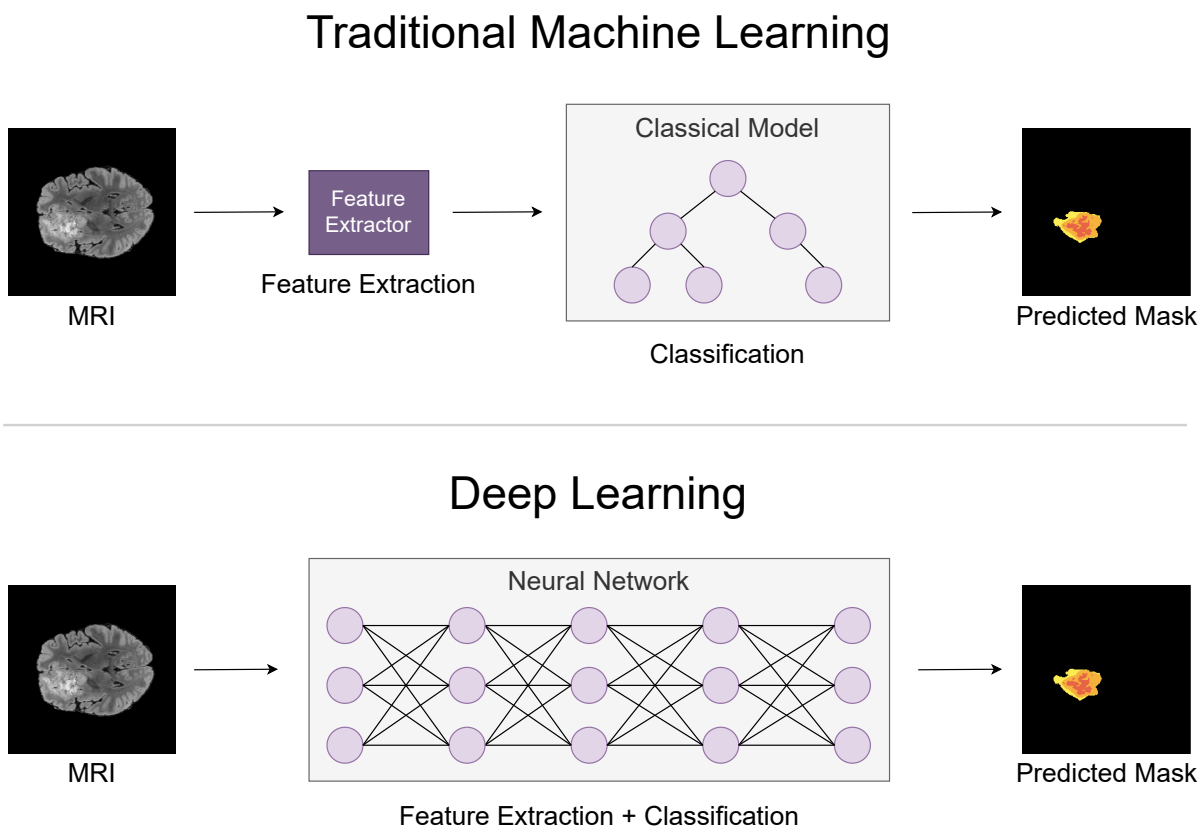


Figure 5.4: Traditional versus Deep Learning feature extraction.

After the great success of the AlexNet proposed by Krizhevsky et al. [96], the usage of CNNs in computer vision rose. Fully Convolutional Networks (FCNs) proposed by Long et al. [24] were among the first to solidify the exceptional capability of neural networks to perform semantic segmentation. Accordingly, this section reviews relevant CNN-based semantic segmentation architectures that manifested

state-of-the-art performance.

5.2.1 Fully Convolutional Network (FCN)

Fully Convolutional Networks (FCNs) were a pioneer CNN architecture designed to solve spatially dense prediction tasks. Before FCN, the general convolutional neural network primarily consisted of convolutional and pooling layers followed by fully connected layers at the end. Concerning semantic segmentation, the fully connected layers are not required at the end of the network since the main goal is to generate segmentation maps rather than predict the class label of the image. Based on this assumption, Long et al. [24] suggested the removal of the fully connected layers, as these layers can be thought of as doing 1×1 convolutions.

Removing the fully connected layers allows the input size to be dynamic. The rationale is that the usage of dense layers (fully connected layers) constrains the input size to be static, thus non-compatible sized inputs have to be resized prior to feeding them into the network. Replacing dense layers with convolutions mitigates this constraint. Therefore, transfiguring fully connected layers into convolutional layers with kernels that cover their entire input image will transform the network into a fully convolutional network that takes inputs of arbitrary size and outputs segmentation maps.

Evidently, feature maps obtained at the output layers are heavily downsampled due to the convolutions performed. To tackle this inherent problem of low-resolution, the authors considered upsampling the final layer using an interpolation technique. Bilinear upsampling (see Section 2.1.2) is an alternative, however it was not enough to attain a fine-grained segmentation, thus the authors proposed a more sophisticated strategy following the idea of filters that learn to upsample using deconvolution. Essentially, the outputs are upsampled to the input dimensions by deconvolution layers within the network, also known as transposed convolutional layers. The last upsampling layer is simply the traditional bilinear interpolation, while intermediate deconvolutional layers are initialized to bilinear upsampling and are iteratively refined during training. Subsequently, this stack of deconvolution layers and activation functions can learn a nonlinear upsampling.

The loss of information at the final feature layer due to the downsampling using convolution layers is still evident. The segmentation output of the network is unpleasantly coarse and rough as it is difficult for the network to upsample with the deconvolution layers by using little information. Therefore, the upsampling strategy goes further with the addition of skip connections. In essence, combining fine intermediate layers with coarse layers allows the model to make local predictions that respect the global structure of the input image. Accordingly, the addition of skip connections results in a set of 3 networks called FCN-8s, FCN-16s, and FCN-32s. The regular network that does not employ skip connections is FCN-32s. In FCN-16s, the information from the second to last pooling layer is combined with the final feature map before upsampling, i.e., the fourth pooling layer is used along with the non-upsampled

output of FCN-32s. FCN-8s replicates the same principle but now including information from an even earlier pooling layer (see Figure 5.5). FCN-8s manifested the best performance overall, with marginal improvement in the high-level detail and smoothness of the segmentation map output.

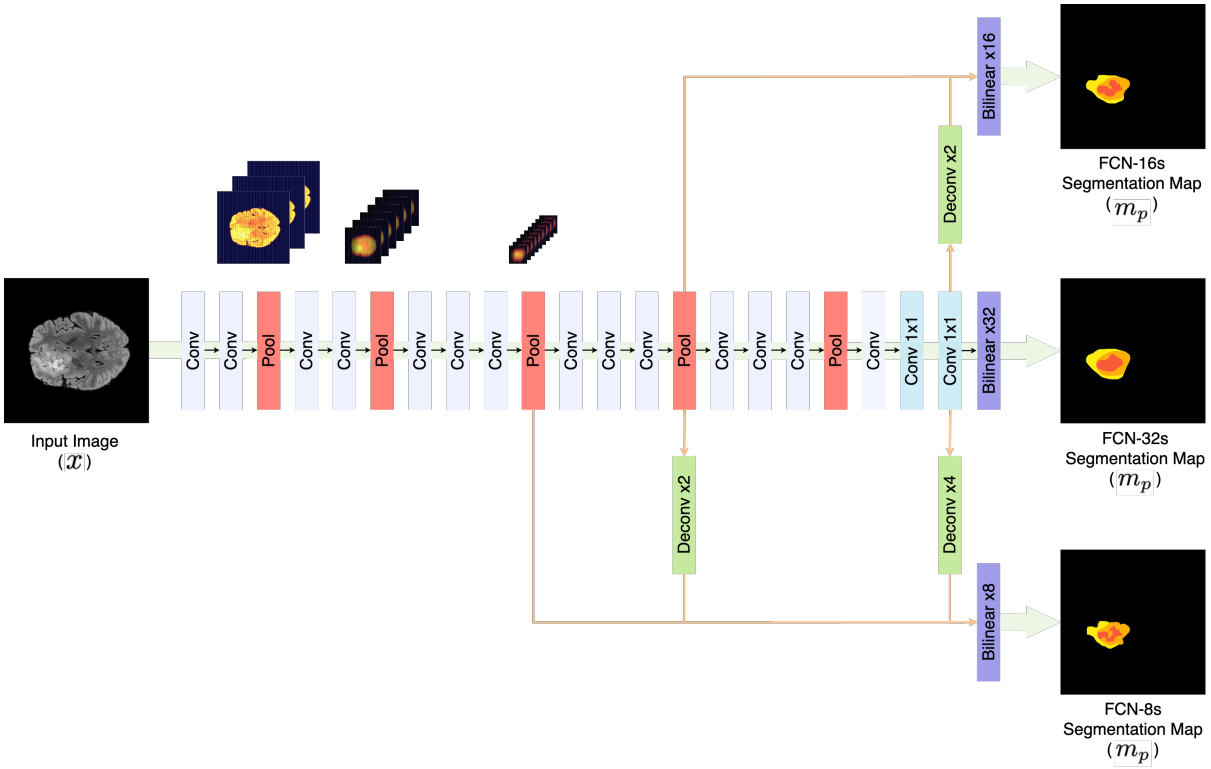


Figure 5.5: Architecture of FCN-32s, FCN-16s and FCN-8s. Feature maps are illustrated in a distinct colormap (not grayscale) for visualization purposes and ease of comprehension. Activation functions are omitted for the same reason.

The first part of the network containing convolutional and max pooling layers is called the encoder. It fundamentally encodes information while also downsampling the image as a repercussion. Meanwhile, the part used for upsampling is entitled the decoder. This pattern of reducing the size with an encoder and subsequently upsampling with the decoder is seen in many architectures due to the success of FCNs.

Following the upsampling strategy aforementioned with the usage of skip connections, additional architectures can be conceived. For instance, FCN-4s are obtained by combining the second pooling layer with the $\times 2$ upsampled third layer, plus the $\times 4$ upsampled fourth layer and the $\times 8$ upsampled final convolutional layer (last feature before upsampling). The last layer is a bilinear interpolation intended to do a $\times 4$ upsample. The same procedure applies to FCN-2s.

Furthermore, CNNs extract features automatically, however this process takes reasonable time to be refined during training. An alternative to short-cut this process and converge faster is to re-use model weights from pre-trained models that were developed for standard computer vision tasks and trained

along benchmark datasets, such as ImageNet [97, 98]. This technique is named transfer learning. For instance, the work of Long et al. [24] exploited ImageNet-trained weights of a VGG16 network. The FCN-32s model was initialized from the VGG16 model and trained for one hundred thousand iterations. Additionally, this technique may improve performance substantially. Reasoning, CNNs can be used not only for classification or segmentation but also for feature extraction. As aforementioned, the whole network can be interpreted as holding two distinct and separable components, the encoder and the decoder. Therefore, after intensive training, the decoder can be disregarded, and the encoder is used for transfer learning. The encoder can be utilized to extract features, which in turn can be used by any classical classifier, such as three-based models. Essentially, the set of convolutional layers from the encoder will learn a set of filters that can be used to assist the training of other methods, either neural networks or traditional machine learning methods. The CNN models trained for image classification contain meaningful information, thus the convolutional layers can be re-used for feature extraction (see Section 6.2).

5.2.2 U-Net

The FCN architecture is modified and extended by Ronneberger et al. [36] in order to excel with very few data and yield precise segmentations. From this work, an FCN-based semantic segmentation architecture entitled U-net was proposed. The name U-net comes from its peculiar U-shaped architecture and consists of an encoder that downsamples the input image to a feature map and a decoder that adversely upsamples back the feature map to the input image size using learned upsampling layers.

U-Net was designed to resolve the information loss problem inherent to FCNs. The main contribution of the U-Net architecture is the high exploitation of skip connections. As seen above in FCN, images are downsampled as part of the encoder. However, this leads to high information loss, which can not be easily recovered by the decoder. FCN tries to address this forfeit by taking information from earlier pooling layers and combining them with the last feature map. U-Net follows a similar approach by proposing to send information from a downsampling layer in the encoder to the hierarchical-correspondent upsampling layer in the decoder (see Figure 5.6). Since layers at the start of the encoder have more information, they would improve the upsampling operation of the decoder by providing finer details of the input image. Therefore, employing skip connections overcomes the FCN bottleneck issue in the middle of the encoder-decoder architecture. Information is transferred between the encoder and decoder layers, i.e., feature representations pass through the bottleneck by skipping it. An additional modification to FCN is the large number of feature channels yielded in the decoder of the U-Net. This allows the network to propagate contextual information to higher resolution layers. Accordingly, the expansive and contracting paths are partially symmetrical.

The encoder follows the typical architecture of CNNs, consisting of several convolutional and max

pooling layers. At every downsampling step, the number of channels of the feature maps is doubled, i.e., the number of features extracted by the encoder duplicates. Regarding the decoder, it consists of upsampling and convolutional layers. At every upsampling step, the number of features is halved, and in addition, it is employed a concatenation of the upsampled feature maps with the correspondingly cropped feature maps from the contracting path (encoder). The upsampling operation used was nearest neighbor interpolation (see Section 2.1.2). Moreover, convolutional layers are always followed by the ReLU activation function. The cropping of feature maps from the encoder in advance of the combination of them with feature maps from the decoder is required due to the loss of border pixels in every convolution operation. Additionally, if cropping is discarded, then when the encoder layers are merged with the decoder layers, the resolution of the two feature maps will not match. Essentially, cropping is utilized to deal with the smaller output size of convolutional layers. Furthermore, simply resizing the feature maps is not an alternative because that will jeopardize the skip connections since the feature maps would not align. Upsampled feature maps do not correspond to low-resolution versions of the corresponding encoder feature maps. They rather represent partially cropped feature maps. Evidently, this would introduce several artifacts and additional loss of information due to issues concerning arbitrary resizing, which is not optimal in medical image processing. At last, a 1×1 convolution is used to map each feature vector to the desired dimensionality equal to the number of classes. Similarly to FCN, fully connected layers are not present in the U-Net architecture.

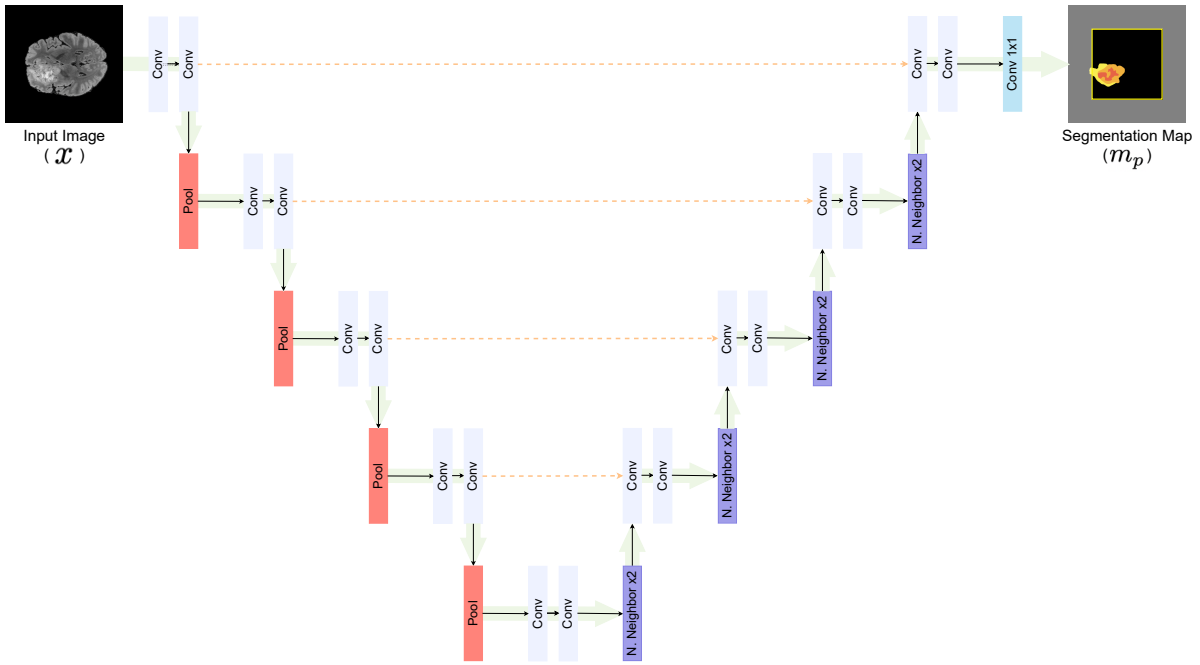
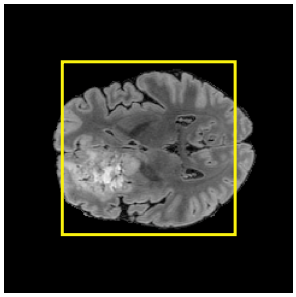


Figure 5.6: U-Net Architecture. Activation functions are omitted for visualization purposes and ease of comprehension. Feature map shape variations after every downsample and upsample step were not showcased for the same reason. Dashed orange lines denote skip connection and concatenation operations.

Reasoning, due to the unpadded convolutions, the resulting segmentation map is of smaller size compared to the input by a constant border width. Alternatives to attain equal sizes between the input and the output are padding-zero or mirroring at the image borders in every convolutional and pooling layer. Discarding these alternatives the output segmentation map considers only a central region of the original image. The output is smaller than the input to ensure sufficient classification of each pixel in the segmentation map.



Input Image

(x)

Figure 5.7: U-net output size (yellow) compared to input size.

Ultimately, U-Net allied with data augmentation, such as elastic deformations (see Section 6.1.3), required only few annotated data and has a reasonable training time, thus manifesting exceptional performance and state-of-the-art segmentation scores.

5.2.3 Open BraTS Solution

Henry et al. [99] trained multiple U-Net based neural networks to automate and standardize brain tumor segmentation. Two independent ensembles of models from different training pipelines were trained. In each pipeline, the execution of a model was repeated several times and at the end the saved weights were averaged, effectively creating a new self-ensembled model. Afterwards, both pipeline segmentation maps were merged, taking into account the performance of each ensemble for each specific tumor subregion.

The network employed follows a 3D U-Net architecture with convolutional and max pooling layers in the encoder part. Each convolutional layers is followed by a normalization layer and a nonlinear activation function ReLU. Similarly to U-Net, the number of filters is doubled after each downsampling step. Moreover, two dilated convolutions are employed, following the last step of downsampling. Their output is subsequently concatenated with the feature maps of the last convolutional layer before the dilated convolutions. Regarding the decoder, is consists of convolutional and upsampling layers. After each upsampling the number of features is halved, identical to U-Net. The upsampling operation is a

trilinear interpolation. Padding is employed in every convolutional layer. As a result, convolutions do not change the spatial dimensions of the feature maps, thus no cropping is required like in the classical U-Net. Accordingly, during skip connection with concatenation between encoder and decoder layers, equal hierarchical-level feature maps share the same size. Regarding max pooling layers no padding is employed, thus changing the spatial dimensions of the feature maps. A $1 \times 1 \times 1$ convolution follows the last step of the upsampling to map each feature vector to the desired dimensionality.

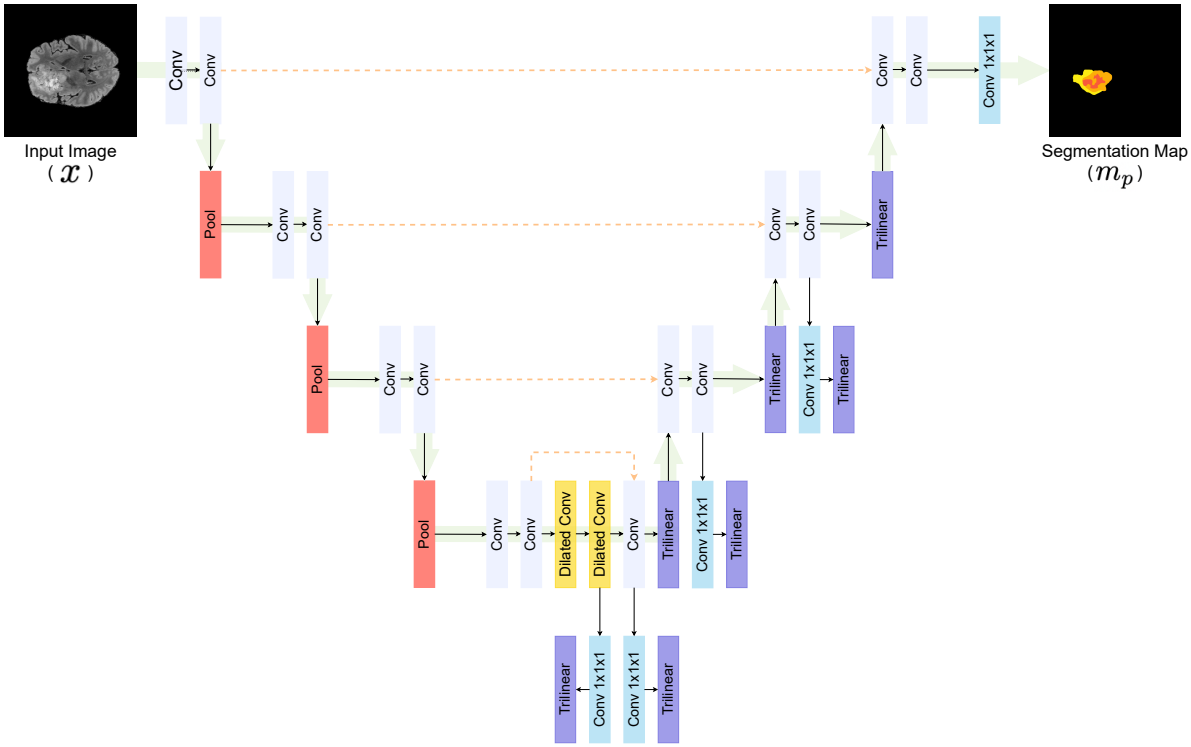


Figure 5.8: Open BraTS Solution Architecture. Dashed orange lines denote skip connection and concatenation operations.

The network is optimized using the Dice Loss (see Section 5.3.1). Additionally, deep supervision was employed after the dilated convolutions and in every decoder step (except the last). Deep supervision intends to add companion optimization targets at each layer of the decoder network. Afterwards, these companion losses are summed with the loss of the output (main loss) to compute the final loss. Deep supervision was implemented by adding a convolutional layer with a sigmoid activation followed by a trilinear upsampling with a factor depending on the depth of the feature maps.

5.3 Learning Strategies

5.3.1 Dice Loss ($\mathcal{L}_{Dice}$)

This loss derives from the Dice Similarity Coefficient (see Section 6.3.1), a widely used metric in computer vision to estimate the similarity between two images or volumes, for instance, segmentation masks. High Dice scores translate to high-fidelity segmentations. Therefore, this loss can be defined, directly in terms of the Dice coefficient, as follows:

$$\mathcal{L}_{Dice} = \frac{1}{N}\sum_{n=1}^{N} 1 - Dice(\hat{m}_t, m_p), \tag{5.1}$$

where N denotes the number of training samples, $\hat{m}_t$ is the ground truth segmentation mask and m_p the predicted mask. Reasoning, the smaller the Dice score, the greater the loss.

Many variations of this loss were formulated, such as, Generalized Dice Loss [100] and Log-Cosh Dice Loss [101]. Generalized Dice Loss is a multi-class extension of Dice Loss that controls the contribution of each class by setting class weights that are inversely proportional to the label frequencies. Although, giving higher weights to low-frequency classes seems counter-intuitive, this initiative effectively tackles class imbalance. Meanwhile, Log-Cosh Dice Loss tries to tackle the non-convexity nature of the Dice Loss function, thus alleviating the optimization process since convex functions are easier to optimize.

5.3.2 Jaccard Loss ($\mathcal{L}_{Jaccard}$)

The Jaccard loss is frequently referred to as the intersection-over-union (IoU) loss. Similar to Dice Loss and derived from the Jaccard Index (see Section 6.3.2), this loss is employed to optimize the segmentation task through minimization of the given equation:

$$\mathcal{L}_{Jaccard} = \frac{1}{N}\sum_{n=1}^{N} 1 - Jaccard(\hat{m}_t, m_p), \tag{5.2}$$

where N denotes the number of training samples, $\hat{m}_t$ is the ground truth segmentation mask and m_p the predicted mask. Implicitly, minimizing equation (5.2) implies maximizing the Jaccard Index, which consequently advocates a superior segmentation quality. Analogous to Dice loss, many variations can be formulated, such as the Generalized IoU (GIoU) [102] and Distance-IoU (DIoU) [103] losses.

5.3.3 Cross-Entropy Loss ($\mathcal{L}_{CE}$)

Also known as Logarithmic loss or Logistic loss. Cross-Entropy (CE) is derived from Kullback-Leibler divergence [104], a measure of dissimilarity between two distributions. It intends to measure the differences in information content between the ground truth and predicted segmentation maps. The segmentation output is required to be a distribution of probabilities. Essentially, each pixel has an estimated probability distribution representing the predicted probability for each class. CE punishes how close to zero is the predicted probability of the actual ground truth class, i.e., penalizes how far from one (maximum probability) it is. The penalty is logarithmic, yielding larger absolute values for probability estimations that are close to zero (the model wrongly predicted a low probability for the correct class) and smaller values for estimations tending to one (the model correctly estimates a distribution of probabilities for the pixel where the true class has a high probability). Cross-Entropy loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{CE} &= \frac{1}{N}\sum_{n=1}^{N} CE\left(\hat{m}_{t_n}, m_{p_n}\right) = \\ &= -\frac{1}{NWH}\sum_{n=1}^{N}\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{c=1}^{C} P^c_{t_n}(i,j)\cdot\log(P^c_{p_n}(i,j)), \end{aligned} \tag{5.3}$$

where N denotes the number of training samples, C is the number of classes present, W is the width of the segmentation maps, and H is the height. Moreover, $\hat{m}_t$ is the ground truth segmentation mask, m_p is the predicted mask, and $P^c_t(i,j)$ is a binary signal that is equal to one if the pixel in position (i,j) of the ground truth segmentation map n has class c. Essentially, it emulates the true class probability distribution of a pixel from the mask $\hat{m}_t$. Furthermore, $P^c_p(i,j)$ represents the predicted probability of a pixel (i,j) being of class c. The formula provided was extended to work simultaneously with binary and multiclass problems. Nonetheless, it is evident that for each class, a Binary Cross-Entropy (BCE) is computed, where solely the classes c ($label = c$) and $\neg c$ ($label \neq c$) are considered.

Similarly to Dice and Jaccard losses, Cross-Entropy loss works best with balanced data. Under segmentation maps with heavy class imbalance, this loss may not be adequate, for instance, in segmentation maps of MRI with labeled small tumors where the background consists of the majority class. Since there is an unequal distribution of pixels that represents an object and the rest of the image, the $\mathcal{L}_{CE}$ may not be appropriate to effectively evaluate the performance of a segmentation model. The inaccuracies of the minority classes are overshadowed by the accuracy of the majority class. An adaptation of this loss can be conceived, entitled Weighted Cross-Entropy loss (WCE), and it is widely used in medical imaging. WCE extends CE by assigning different weights to each class, thus some pixels can be considered more important to classify correctly.

5.4 Implementation Details

5.4.1 Tree-based Method

The training was performed over the BraTS dataset (see Section 6.1.1). A five-fold cross-validation technique was employed with a Random Forest classifier, training 175 trees in total for each fold. The maximum depth of each tree was fixed at 60. The training was not performed per-batch, thus only 20 volumes were considered due to computation constraints. The function selected to measure the quality of a split was Gini impurity. Meanwhile, the optimization target was a multiclass Dice Score, which takes label imbalance into account. Prior to training, a feature selection process was conducted on a five-fold cross-validation pipeline, where a Random Forest Classifier was trained with 100 trees per-fold. Implemented in Python 3.7 with Scikit-Learn and Keras libraries.

5.4.2 Open BraTS Solution

A U-Net based model is trained for 60 epochs over the BraTS dataset (see Section 6.1.1). For optimization, the Ranger optimizer [105] is used with a learning rate set to 2×10^{-4} and no rate decay. The batch size selected was 1. Additionally, the optimization target was simply a batch-wise Dice loss without weighting. Implemented with Pytorch v1.12.1+cu113 on Python 3.7. None of the pipelines from [99] were considered, and the ensemble strategy was discarded. The base model from pipeline A of [99] was trained once.

5.5 Summary

This chapter reviews the methodology to perform semantic segmentation with traditional machine learning algorithms. Additionally, describes relevant fully convolutional neural networks for segmentation.

A clever strategy to perform semantic segmentation is by merging all the conventional techniques introduced in chapter 2 to extract a set of features and, afterwards, use a classical machine learning algorithm to make pixel-level predictions. Moreover, this strategy can take substantial advantage of transfer learning by exploiting pre-trained deep learning models to extract relevant and unique features.

Furthermore, FCN is reviewed, a pioneer work that suggested the removal of the fully connected layers from CNNs. Following, U-Net and Open BraTS, an extension work of U-Net, were described, manifesting good performance on the semantic segmentation task.

Ultimately, optimization targets were defined, and the implementation details of the experiments exhibited in the next chapter 6 were reported.

6

Tumor Segmentation Experiments

Contents

This chapter describes the basic methodology for conducting tumor segmentation experiments. The first step is to acquire a dataset consisting of pairs holding raw input images and the corresponding labeled segmentation maps. Prior to training, the dataset requires some processing, and this chapter addresses the preprocessing employed in the experiments performed. Additionally, techniques used to increase the amount of data are described. Afterwards, semantic segmentation models can be trained by feeding the pair-wise data. Subsequently, quality metrics are utilized to evaluate the performance of the segmentation, and this chapter intends to discuss them.

Furthermore, Super-Resolution can be assessed by tumor segmentation, following a task-based evaluation strategy. Accordingly, a dataset consisting of LR-HR image pairs is conjectured beforehand. Afterwards, tumor segmentation models, trained with ground truth images and their corresponding segmentation maps, are employed to segment unseen super-resolved and raw ground truth images.

6.1 Data

6.1.1 BraTS Dataset

Since FastMRI did not hold segmentation maps for each MRI scan, then to perform the semantic segmentation experiments, the BraTS dataset [15, 43, 44, 106, 107] was used. BraTS compromises a collection of volumetric brain MRIs that have tumoral regions. The training split provided for the BraTS2021 challenge included 1251 brain MRIs, along with the segmentation annotations of the tumorous regions. The annotated tumor sub-regions are based upon known observations visible to the trained radiologists. The validation split did not include the annotation, thus it was not used in these experiments. In order to validate the performance of the models, a hold-out technique was employed, as described in Section 6.1.2.

Furthermore, every scan was skull-stripped, meaning that the brain tissue was isolated from the skull and the extracerebral tissues. Volumes have dimensions of $240 \times 240 \times 150$ voxels. Four different modalities were provided for each instance, along with the segmentation map. The four modalities given were: native (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2), and Fluid Attenuated Inversion Recovery (FLAIR). Regions annotated within the volume can be of four distinct classes: necrotic tumor core (NCR — label 1), peritumoral edematous (ED — label 2), enhancing tumor (ET — label 4), and background (non-tumoral voxels — label 0). Label 3 does not hold any representation, thus it is not considered. An illustration is given in Figure 6.1.

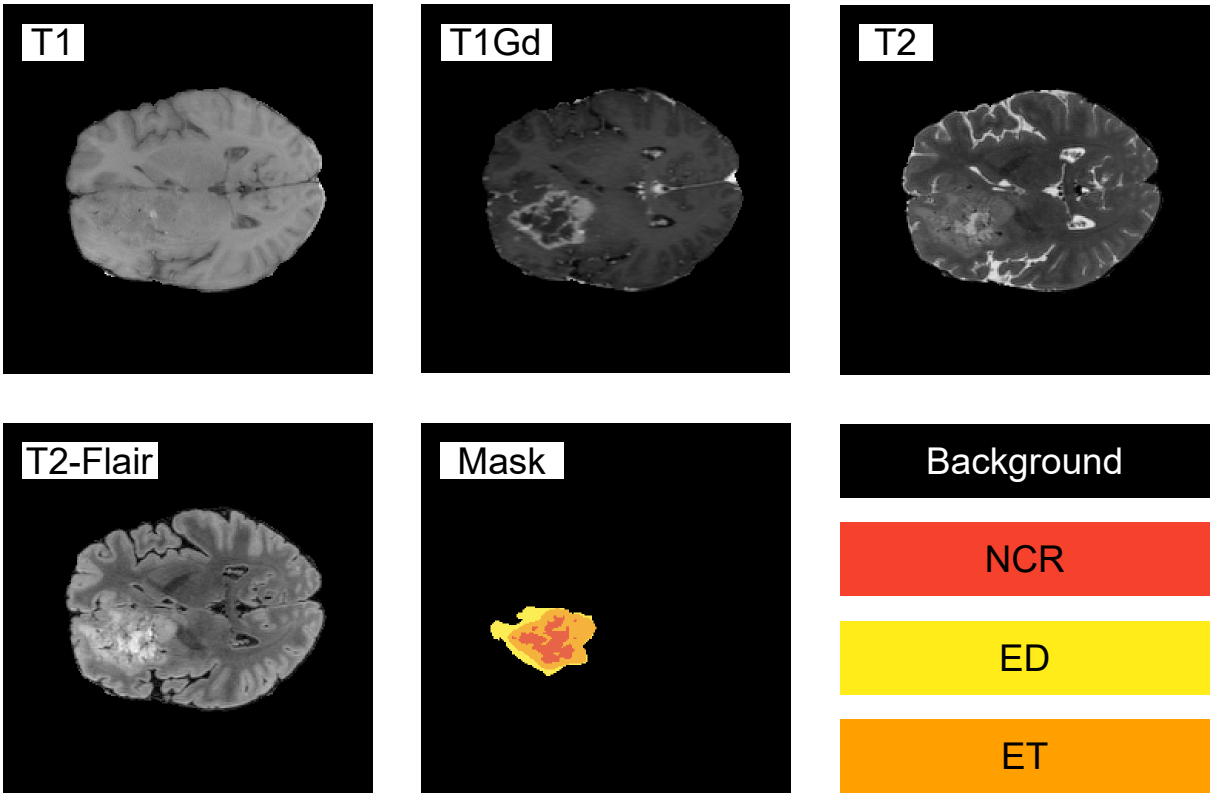


Figure 6.1: Illustration of one instance from the BraTS dataset.

6.1.2 Data Preprocessing

The BraTS 2021 challenge did not include annotations in the validation split. Additionally, the testing split is not publicly available. Therefore, to proceed with the evaluation of the tumor segmentation, a hold-out technique was employed. The training set was shuffled into two splits, training and testing. The new training split contains 80% of the data from the original split, while the testing split has the remaining 20%.

Concerning the preliminary tree-based approach implemented, every volume was cropped to a central $128 \times 128 \times 128$ region, thus improving data balancing and reducing computations required, as redundant background voxels (label 0) on the borders of each volume are cropped. Furthermore, the cropping did discard most of the dark/void region from the raw MRI scans, however the background voxels remain the majority class since the non-tumoral brain tissue also has label 0. To further balance the data, 85% of the remaining background pixels were additionally dropped.

Considering the OpenBraTS Solution, the first step taken was data standardization since MRI intensities vary depending on manufacturers, acquisition parameters, and sequences. In order to discard outliers, the volumes were clipped to all intensity values to the 1 and 99 percentiles of the non-zero voxels distribution of the volume. Afterwards, a per-volume min-max scaling was performed. Subsequently,

cropping was employed to discard background voxels, as they do not provide helpful information and can be ignored by the neural network. Similarly to the tree-based approach, the volumes were cropped to a fixed region size of $128 \times 128 \times 128$. However, in this method, the cropping selects a random region rather than always selecting the central voxels.

Furthermore, two new datasets were consummated to validate the reconstruction quality of Super-Resolution GANs. From the original BraTS, a tricubic interpolation was used to downsample the whole training and testing splits generated by the hold-out technique. The downscaling factor adopted was 2. Thus a new dataset, entitled Low-Resolution BraTS (LRBraTS), was formulated. Simultaneously holding BraTS and LRBraTS means LR-HR pairs are present, thus the conditions to perform Super-Resolution are met. The Real-ESRGAN model, discussed in Section 3.7, was used to super-resolve the LRBraTS by an upscale factor of $\times 2$, resulting in a new dataset, SRBraTS. The implementation details of Real-ESRGAN are described in Section 3.9.

6.1.3 Data Augmentation

Deep neural networks require large amounts of data to excel the traditional techniques and attain better performances within semantic segmentation and many other computer vision tasks. More data usually improves the overall performance since it exposes the algorithms to more features and information. To satisfy this requirement, data augmentation is employed to increase the amount of labeled data in the training phase. Additionally, data augmentation techniques alleviate the overfitting problem by artificially extending the datasets. Essentially, this allows the models to learn invariance to such deformations without the need to have to manually annotate additional data. Data Augmentation is particularly important in biomedical segmentation since deformation is the most common variation in tissue, and realistic deformations can be simulated accurately. Therefore, to make our methods more robust, the following data augmentations were employed in the training split with an augmentation factor of $\times 2$:

Table 6.1: Augmentation operations employed.

Methods	Probability	Range	Directions
Flip	$\approx 65\%$	-	frontal, sagittal, vertical
Rotation	$\approx 65\%$	$[-\pi, \pi]$	frontal, sagittal, vertical
Shift	$\approx 65\%$	$\pm 10\%$	frontal, sagittal, vertical
Zoom	$\approx 65\%$	$\pm 10\%$	frontal, sagittal, vertical
Elastic Distortion	30%	$\sigma = (2, 2, 2)$ $g = (6, 6, 4)$	frontal, sagittal, vertical

From table 6.1, g denotes the shape of the deformation grid. Each element in g corresponds to the number of points, along one direction, in the deformation grid. For instance, if $g = (5, 5, 5)$, the volume is deformed with a $5 \times 5 \times 5$ deformation grid. Moreover, σ is the standard deviation of a normal distribution used to sample the displacements of the deformation grid.

The tree-based approach did not take advantage of the data augmentation pipeline formulated since only 20 volumes were used during training. Regarding the Open Brats Solution approach, the on-the-fly data augmentation techniques described in [99] were discarded.

6.2 Feature Extraction

As aforementioned, to improve the image recognition performance, we need to get feature maps that express unique features instead of lying solely on the raw pixel value intensities. This section briefly showcases features that were extracted using chopped encoders from deep learning models and classing semantic segmentation techniques. Figure 6.2 exhibits features that the tree-based method will use to train. The current implementation is preliminary, thus the results in Section 6.4 do not reflect the improvements these features provide.

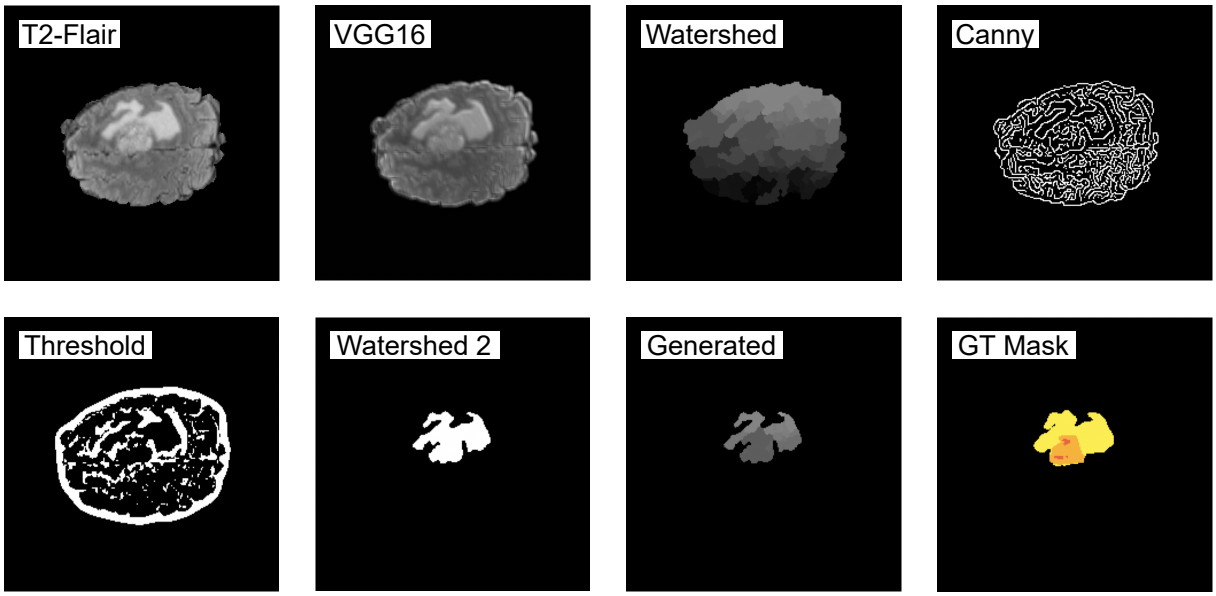


Figure 6.2: Visualization of the features extracted from a 2D slice of a BraTS MRI scan. Each feature was obtained by convolving the T2-flair input MRI with a certain filter. VGG16 figure used the filters from the encoder of VGG16. The Watershed figure in the first row was obtained by employing a watershed with an Image Gradient to detect the edges. The watershed in the second row was obtained with Sobel and an additional threshold. The generated feature is a combination of both watersheds resulting from a pixel-wise multiplication.

6.3 Evaluation Metrics

6.3.1 Dice Similarity Coefficient

A successful prediction intends to maximize the overlap between true and predicted labels. Dice similarity coefficient (DSC), also known as F1-score or Sørensen-Dice index, is a metric that aims to mathematically quantify how good this overlapping is. DSC is defined as:

$$\text{Dice}(A, B) = \frac{2 \cdot \|A \cap B\|}{\|A\| + \|B\|}, \tag{6.1}$$

where A and B denote two binary segmentation masks for a given class, $\|A\|$ represents the norm of A, and $\|A \cap B\|$ corresponds to the overlap given by the intersection between both masks. Essentially, the numerator represents two times the area of the intersection, while the denominator represents the area of union summed with the area of the intersection.

6.3.2 Jaccard Index

Similarly to DSC, it can be used to measure the similarity between two segmentation maps. It is also known as the Intersection over Union (IoU), as is defined as follows:

$$\text{IoU}(A, B) = \frac{\|A \cap B\|}{\|A \cup B\|}, \tag{6.2}$$

where A and B denote two binary segmentation masks for a given class. Essentially, the numerator corresponds to the number of matching pixels, while the denominator represents the total number of matching and mismatching pixels. Regarding binary or multiclass segmentation, DSC and IoU are calculated by computing the scores of each class and afterwards averaging them.

6.3.3 Hausdorff Distance (95%)

The Hausdorff Distance (HD) is the maximum perpendicular distance between the closest points from the contours of two regions. Essentially, it is complementary to the DSC, as it measures the maximum distance between the margin of the two regions. It is computed as follows:

$$\text{H}(A, B) = \max\left(\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(b, a)\right), \tag{6.3}$$

where $d(a, b)$ denotes the distance between two pixels, a and b, in the border of two region included in the segmentation masks, A and B, respectively. A segmentation map could exhibit almost voxel-perfect overlap with the ground truth segmentation map, but if a single voxel is far away from the ground truth,

then the Hausdorff distance will be high. Accordingly, Hausdorff distance heavily penalizes outliers and can be judged noisier compared to DSC and IoU. However, this metric is very convenient to evaluate segmentation predictions in the medical context. Therefore, only the 95th percentile of the distances between the boundaries of the two regions is usually considered. Due to Hausdorff distance being sensitive to noise, this can help significantly by avoiding potential outliers.

6.4 Quantitative Results

All metrics suggest that tumor segmentation with the ground truth MRIs outperforms tumor segmentation performed over the super-resolved MRIs. Despite Super-Resolved MRIs exhibiting photo-realistic details, they did not manifest the best results for the segmentation of tumors. However, the algorithms were intensively trained with the ground truth images, which evidently may marginally benefit the tumor segmentation of the ground truth images. Although super-resolved images are reconstructions of the ground truths, the distributions between them can have minor dissimilarities. For instance, Super-Resolution algorithms can marginally change pixel value intensities in some regions, which can subsequently lead to an inferior segmentation. Furthermore, the Super-Resolution algorithm used (Real-ESRGAN) was not trained over the BraTS dataset, thus the Super-Resolution has the potential to be improved further. Nonetheless, super-resolving medical images is a complex task, and despite having all these constraints, the tumor segmentation still manifested satisfactory results over the super-resolved dataset.

Table 6.2: Tumor segmentation results comparison between the super-resolved and ground truth brain MRIs. Red color indicates the worst performance overall and Green color the best.

Method	Input	Scale	Optimization Target	DSC	IoU	HD95
Tree-based	SRBraTS	$\times 2$	Multiclass DSC	0.26	0.26	28.76
Tree-based	BraTS	-	Multiclass DSC	0.29	0.28	57.81
Open BraTS	SRBraTS	$\times 2$	Multiclass DSC	0.61	0.52	21.4
Open BraTS	BraTS	-	Multiclass DSC	0.82	0.75	8.35

Jaccard Index (IoU) metric was the most affected by the Super-Resolution. However, this is expected due to its nature. It is more sensitive to changes in the overlap of the regions compared to the Dice Similarity Coefficient.

Table 6.3: Tumor segmentation results for each tumoral region. NCR is the necrotic tumor core, ED is the peritumoral edematous, and ET is the enhancing tumor. Green color indicates the best performance overall.

Methods	Input	DSC			IoU			HD95		
		NCR	ED	ET	NCR	ED	ET	NCR	ED	ET
Open BraTS	SRBraTS	0.43	0.36	0.68	0.34	0.26	0.58	18.2	42.1	12.7
Open BraTS	BraTS	0.69	0.79	0.83	0.59	0.68	0.75	9.5	10.2	5.3

6.5 Qualitative Results

Qualitative results advocate an adequate Super-Resolution and a non-optimal but decent tumor segmentation of the super-resolved MRIs. Looking at Figure 6.3 it is possible to see a few dissimilarities between the predicted segmentations of the super-resolved and the ground truth MRIs. The difference is not large despite IoU suggesting that the tumor segmentation over SRBraTS is inferior by some margin.

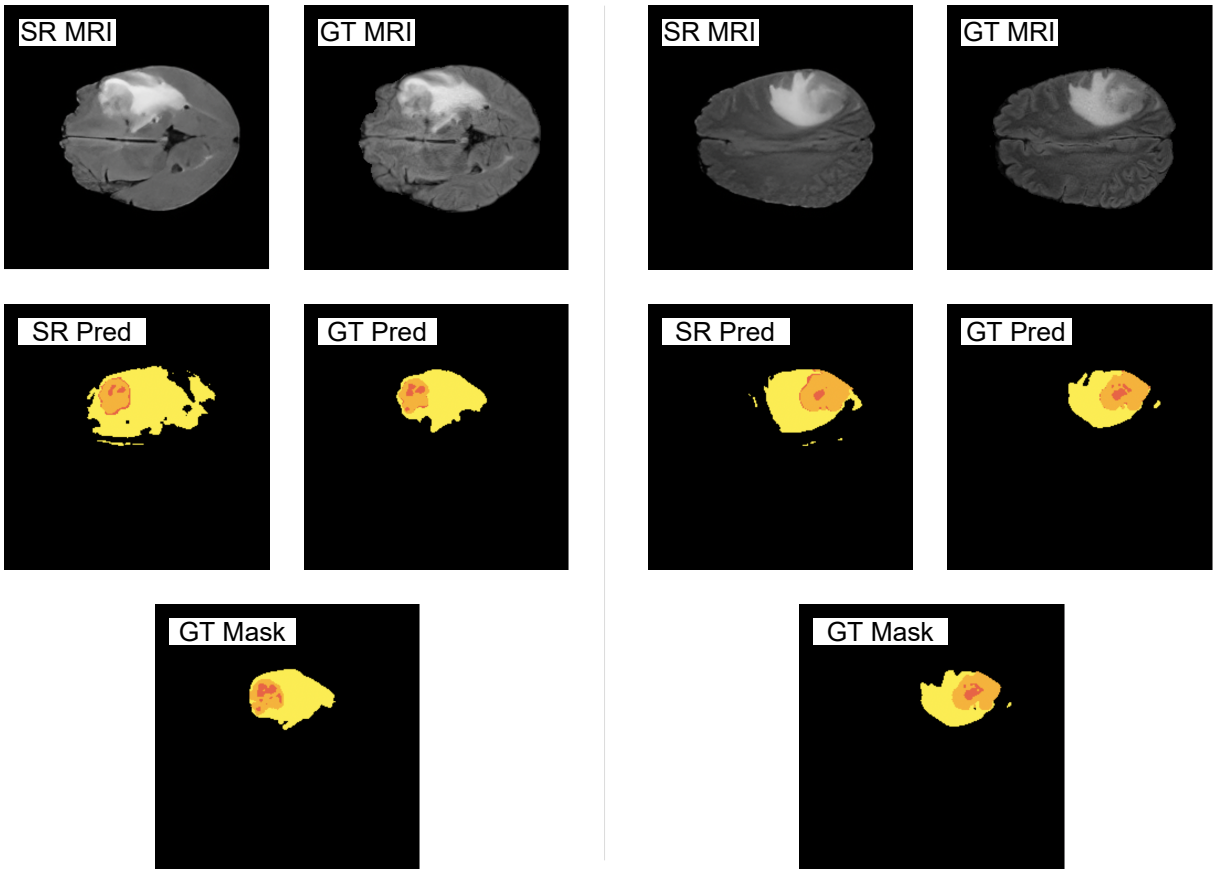


Figure 6.3: Tumor Segmentation results with BraTS and SRBraTS. The first row exhibits the super-resolved MRIs and the corresponding ground truths. Below each MRI is the predicted segmentation map that was obtained from it.

Figures B.2 and B.3 show that the tumor segmentation over BraTS achieved solid results. Furthermore, looking at Figure B.1, it is evident that despite metrics suggesting that the segmentation was not optimal, it was still reasonably satisfactory.

6.6 Discussion

Dice score has manifested to be a good target for optimization, enhancing the performance of models substantially. However, it can be further improved by considering a more mathematically convex

alternative that optimizes faster and easier, such as Log-Cosh Dice Loss [101].

Matching the tumor segmentation performance over SRBraTS with the performance over BraTS can be suggestive that the super-resolution was reliable. Looking at Tables 6.2 and 6.3, it is possible to acknowledge that a few dissimilarities were present. A reason can be the usage of content loss as an optimization target. Accordingly, this leads to overly smooth results, as discussed in section 4.7, or marginal changes in the pixel value intensities of some areas, consequently confusing the segmentation model into interpreting some regions as tumors, which can cascade to a larger region and jeopardize the prediction. This explains the impact on the prediction of peritumoral edematous (ED — label 2), which was the region that suffered the highest impact. Since it contains the tumor border, the SR algorithm can get confused due to the transitions of tissue inherent to that region. Additionally, the Super-Resolution model employed was not the best model from the experiments of Chapter 4. This suggests an additional extent for improvements besides other alternatives discussed in Section 6.4. For instance, if the Super-Resolution algorithm was trained intensively with MRIs holding tumoral regions, then the SR algorithm could have learned patterns to better mimic the data distribution of the high-resolution images and reconstruct tumoral regions accordingly instead of interpreting them as downsampling artifacts. Furthermore, if the models for tumor segmentation were trained with the super-resolved images and their corresponding annotations, then the segmentation performance with SRBraTS would possibly be substantially higher.

Ultimately, after evaluating the proficiency of GANs in reconstructing medical images, this work is looking forward to develop a tree-based method for brain tumor segmentation that utilizes features extracted from deep learning approaches. The method is still in progress, lacking several optimization techniques, thus his results were not analyzed intensively. For these experiments, the tree-based method was trained with only a subset of the features extracted, as it only sustains a preliminary approach that is intended to be further enhanced.

6.7 Summary

This chapter intensively reviewed Semantic Segmentation algorithms and applied them in the Tumor Segmentation task. Afterwards, pragmatically evaluated the Super-Resolution performance from Real-ESRGAN by segmenting tumors over its super-resolved images. Additionally, two new datasets were conceived from these experiments that can be utilized to evaluate Super-Resolution in future works.

Results advocate that FCNs are solid approaches for tackling the tumor segmentation concept and other medical image applications.

The tumor segmentation over SRBraTS (Super-Resolved BraTS) manifested satisfactory results while also exhibiting a margin for improvement.

7

Conclusion

Contents

A knee MRI scan usually takes 30 to 60 minutes but can take as long as 2 hours. Acquiring less amount of k-space data will reduce the acquisition time. However this results in MRIs with relatively low spatial resolution. Furthermore, the images obtained from computed tomography (CT), magnetic resonance imaging (MRI), or any other medical imaging technique often have low resolution, inherent noise, and lack of structural information. Therefore, making a correct diagnosis judgment in the medical field becomes a significant challenge. This work has proven that high-frequency details can be recovered from Low-Resolution signals, and GAN-based super-resolution 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, GAN-based techniques are promising CS-MRI reconstruction methods, enabling resolution improvements, zooming into images, and data acquisition acceleration. Additionally, denoising solutions led to performance boosts on the super-resolution task, with manifested reduction of the checkerboard pattern inherent to GAN synthesis.

Although the task-based evaluation showcases space for improvements in the performance of GANs, they still provide good perceptual quality. Tumor Segmentation of Super-Resolved images exhibited an inferior performance relative to tumor segmentation with ground truth images. However, several constraints coexisted that impacted these results. The tumor segmentation still manifested satisfactory results over the SRBraTS dataset. Furthermore, fully convolutional neural networks exhibited solid results in segmenting tumors, thus solidifying the proficiency of Deep Learning in the medical image context. Merging both Super-Resolution and Tumor Segmentation can provide an automatic pipeline for diagnoses that healthcare can substantially benefit from. Ultimately, two new datasets were formulated to use Tumor Segmentation to validate the Super-Resolution quality in medical image reconstruction.

7.1 Future work

This work is intended to proceed into a Ph.D., where the first step is to build an ensemble of several models for tumor segmentation (both traditional and deep learning). Furthermore, the ultimate goal is to design an end-to-end deep neural network that can super-resolve and segment tumors. This will lead to substantial accelerations in the data acquisition pipeline of medical images and sustain massive value due to its automatic diagnosis.

Bibliography

[1] P. C. Lauterbur, “Image formation by induced local interactions: examples employing nuclear magnetic resonance,” *nature*, vol. 242, no. 5394, pp. 190–191, 1973.

[2] E. Funk, P. Thunberg, and A. Anderzen-Carlsson, “Patients’ experiences in magnetic resonance imaging (mri) and their experiences of breath holding techniques,” *Journal of advanced nursing*, vol. 70, no. 8, pp. 1880–1890, 2014.

[3] E. F. Jackson, L. E. Ginsberg, D. F. Schomer, and N. E. Leeds, “A review of mri pulse sequences and techniques in neuroimaging,” *Surgical Neurology*, vol. 47, no. 2, pp. 185–199, 1997.

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

[5] J. P. Marques, F. F. Simonis, and A. G. Webb, “Low-field mri: An mr physics perspective,” *Journal of magnetic resonance imaging*, vol. 49, no. 6, pp. 1528–1542, 2019.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[7] A. C. Tan, D. M. Ashley, G. Y. López, M. Malinzak, H. S. Friedman, and M. Khasraw, “Management of glioblastoma: State of the art and future directions,” *CA: a cancer journal for clinicians*, vol. 70, no. 4, pp. 299–312, 2020.

[8] D. Mahapatra, B. Bozorgtabar, and R. Garnavi, “Image super-resolution using progressive generative adversarial networks for medical image analysis,” *Computerized Medical Imaging and Graphics*, vol. 71, pp. 30–39, 2019.

[9] X. Li, Y. Wu, W. Zhang, R. Wang, and F. Hou, “Deep learning methods in real-time image super-resolution: a survey,” *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 1885–1909, 2020.

[10] H. HÜSEM and Z. ORMAN, “A survey on image super-resolution with generative adversarial networks,” *Acta Infologica*, vol. 4, no. 2, pp. 139–154, 2020.

[11] S. Anwar, S. Khan, and N. Barnes, “A deep journey into super-resolution: A survey,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[12] C. Tian, X. Zhang, J. C.-W. Lin, W. Zuo, and Y. Zhang, “Generative adversarial networks for image super-resolution: A survey,” *arXiv preprint arXiv:2204.13620*, 2022.

[13] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

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

[15] U. Baid, S. Ghodasara, S. Mohan, M. Bilello, E. Calabrese, E. Colak, K. Farahani, J. Kalpathy-Cramer, F. C. Kitamura, S. Pati *et al.*, “The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification,” *arXiv preprint arXiv:2107.02314*, 2021.

[16] D. Han, “Comparison of commonly used image interpolation methods,” in *Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Atlantis Press, 2013, pp. 1556–1559.

[17] A. N. A. Rahim, S. N. Yaakob, R. Ngadiran, and M. W. Nasruddin, “An analysis of interpolation methods for super resolution images,” in *2015 IEEE Student Conference on Research and Development (SCOReD)*. IEEE, 2015, pp. 72–77.

[18] C. E. Duchon, “Lanczos filtering in one and two dimensions,” *Journal of Applied Meteorology and Climatology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[19] W. K. Carey, D. B. Chuang, and S. S. Hemami, “Regularity-preserving image interpolation,” *IEEE transactions on image processing*, vol. 8, no. 9, pp. 1293–1297, 1999.

[20] J. N. Kapur, P. K. Sahoo, and A. K. Wong, “A new method for gray-level picture thresholding using the entropy of the histogram,” *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 273–285, 1985.

[21] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[22] J. M. Prewitt *et al.*, “Object enhancement and extraction,” *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[23] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 06, pp. 583–598, 1991.

[24] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[25] D. Zikic, Y. Ioannou, M. Brown, and A. Criminisi, “Segmentation of brain tumor tissues with convolutional neural networks,” *Proceedings MICCAI-BRATS*, vol. 36, no. 2014, pp. 36–39, 2014.

[26] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, “Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network,” in *International conference on medical image computing and computer-assisted intervention*. Springer, 2013, pp. 246–253.

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” *arXiv preprint arXiv:1412.7062*, 2014.

[28] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[29] J. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” 2001.

[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[31] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06)*, vol. 2. IEEE, 2006, pp. 2169–2178.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[33] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*, 2017.

[34] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[35] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[41] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[42] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 311–320.

[43] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[44] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki *et al.*, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," *arXiv preprint arXiv:1811.02629*, 2018.

[45] Z. Jiang, C. Ding, M. Liu, and D. Tao, "Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task," in *International MICCAI brainlesion workshop*. Springer, 2019, pp. 231–241.

[46] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert *et al.*, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.

[47] F. Isensee, P. F. Jäger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein, "nnu-net for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2020, pp. 118–132.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[50] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[51] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 pirm challenge on perceptual image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[52] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.

[53] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[54] W. Zhang, Y. Liu, C. Dong, and Y. Qiao, "Ranksrgan: Super resolution generative adversarial networks with learning to rank," *arXiv preprint arXiv:2107.09427*, 2021.

[55] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 89–96.

[56] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[57] R. M. Umer and C. Micheloni, "Deep cyclic generative adversarial residual convolutional networks for real image super-resolution," in *European Conference on Computer Vision*. Springer, 2020, pp. 484–498.

[58] S. Vaishali, K. K. Rao, and G. S. Rao, “A review on noise reduction methods for brain mri images,” in *2015 International Conference on Signal Processing and Communication Engineering Systems*. IEEE, 2015, pp. 363–365.

[59] R. M. Umer, G. L. Foresti, and C. Micheloni, “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*, 2020, pp. 438–439.

[60] K. Zhang, J. Liang, L. Van Gool, and R. Timofte, “Designing a practical degradation model for deep blind image super-resolution,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4791–4800.

[61] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[62] W. Li, K. Zhou, L. Qi, L. Lu, N. Jiang, J. Lu, and J. Jia, “Best-buddy gans for highly detailed image super-resolution,” *arXiv preprint arXiv:2103.15295*, vol. 2, 2021.

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

[64] E. Schonfeld, B. Schiele, and A. Khoreva, “A u-net based discriminator for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8207–8216.

[65] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.

[66] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[67] T.-Y. Liu *et al.*, “Learning to rank for information retrieval,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[68] S.-J. Park, H. Son, S. Cho, K.-S. Hong, and S. Lee, “Srfeat: Single image super-resolution with feature discrimination,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 439–455.

[69] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

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

[71] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[72] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, “Ntire 2017 challenge on single image super-resolution: Methods and results,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.

[73] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

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

[75] X. Liu, M. Tanaka, and M. Okutomi, “Single-image noise level estimation for blind denoising,” *IEEE transactions on image processing*, vol. 22, no. 12, pp. 5226–5237, 2013.

[76] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, “Waterloo exploration database: New challenges for image quality assessment models,” *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2016.

[77] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

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

[79] P. B. Roemer, W. A. Edelstein, C. E. Hayes, S. P. Souza, and O. M. Mueller, “The nmr phased array,” *Magnetic resonance in medicine*, vol. 16, no. 2, pp. 192–225, 1990.

[80] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[81] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through fsim, ssim, mse and psnr—a comparative study,” *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.

[82] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE signal processing letters*, vol. 9, no. 3, pp. 81–84, 2002.

[83] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “Fsim: A feature similarity index for image quality assessment,” *IEEE transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.

[84] A. Liu, W. Lin, and M. Narwaria, “Image quality assessment based on gradient similarity,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1500–1512, 2011.

[85] T. Grigoryev, A. Voynov, and A. Babenko, “When, why, and which pretrained gans are useful?” *arXiv preprint arXiv:2202.08937*, 2022.

[86] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 2. Ieee, 2005, pp. 60–65.

[87] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[88] Y. Kinoshita and H. Kiya, “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, 2020, pp. 1139–1144.

[89] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” *Distill*, vol. 1, no. 10, p. e3, 2016.

[90] A. Zomet, A. Rav-Acha, and S. Peleg, “Robust super-resolution,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[91] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Advances and challenges in super-resolution,” *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[92] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution,” *IEEE transactions on image processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[93] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[94] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[95] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

[97] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[98] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[99] T. Henry, A. Carré, M. Lerousseau, T. Estienne, C. Robert, N. Paragios, and E. Deutsch, “Brain tumor segmentation with self-ensembled, deeply-supervised 3d u-net neural networks: a brats 2020 challenge solution,” in *International MICCAI Brainlesion Workshop*. Springer, 2020, pp. 327–339.

[100] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.

[101] S. Jadon, “A survey of loss functions for semantic segmentation,” in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020, pp. 1–7.

[102] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[103] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-iou loss: Faster and better learning for bounding box regression,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[104] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[105] L. Wright, "Ranger - a synergistic optimizer." https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer, 2019.

[106] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific data*, vol. 4, no. 1, pp. 1–13, 2017.

[107] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection," *The cancer imaging archive*, vol. 286, 2017.

A

Additional Super-Resolution Qualitative Results

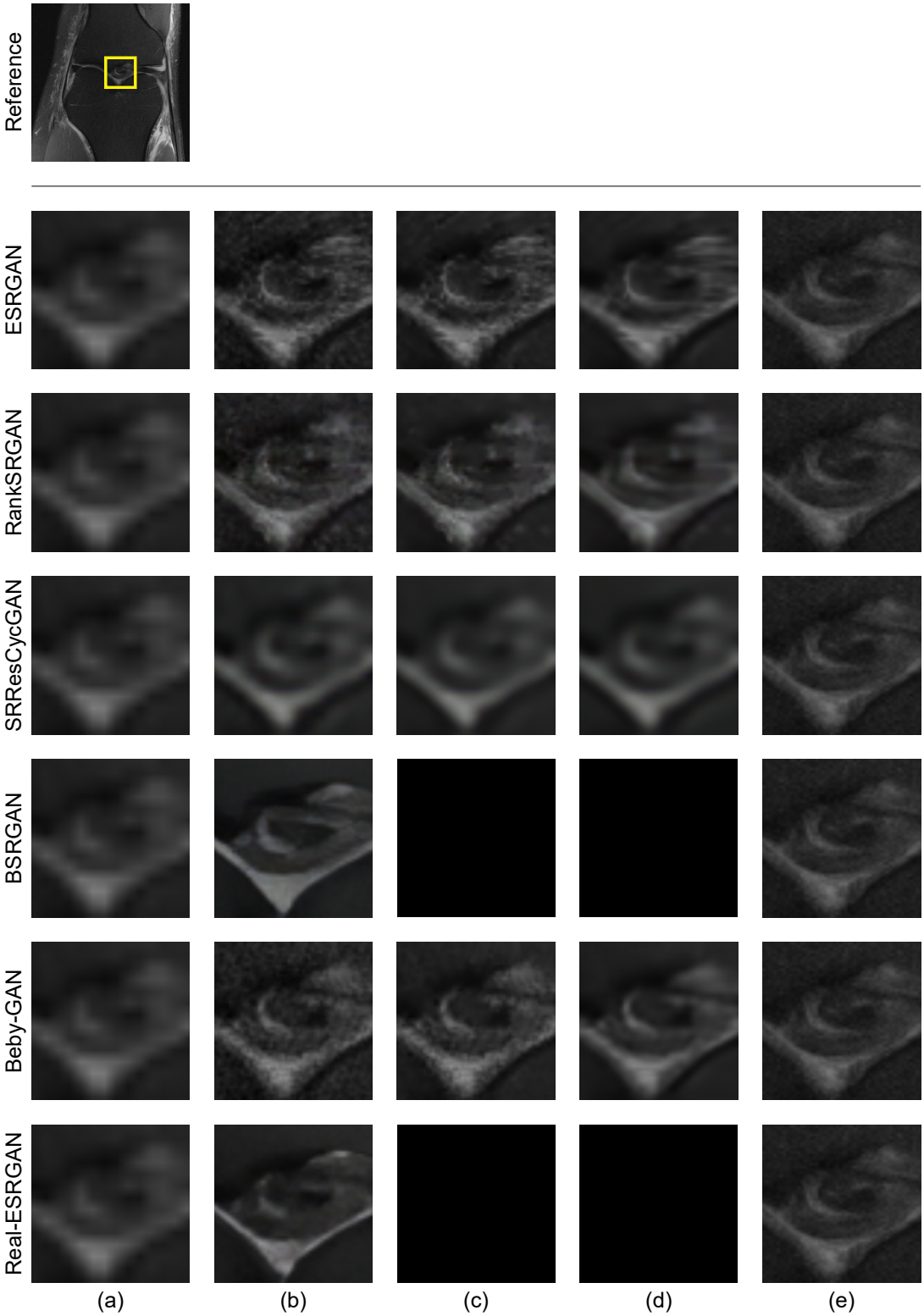


Figure A.1: Patch comparison from (a) Input LR Images, (b) Generated Images w/o denoising, (c) Generated Images w/ NLM, (d) Generated Images w/ BM3D and (e) Ground Truth Images.

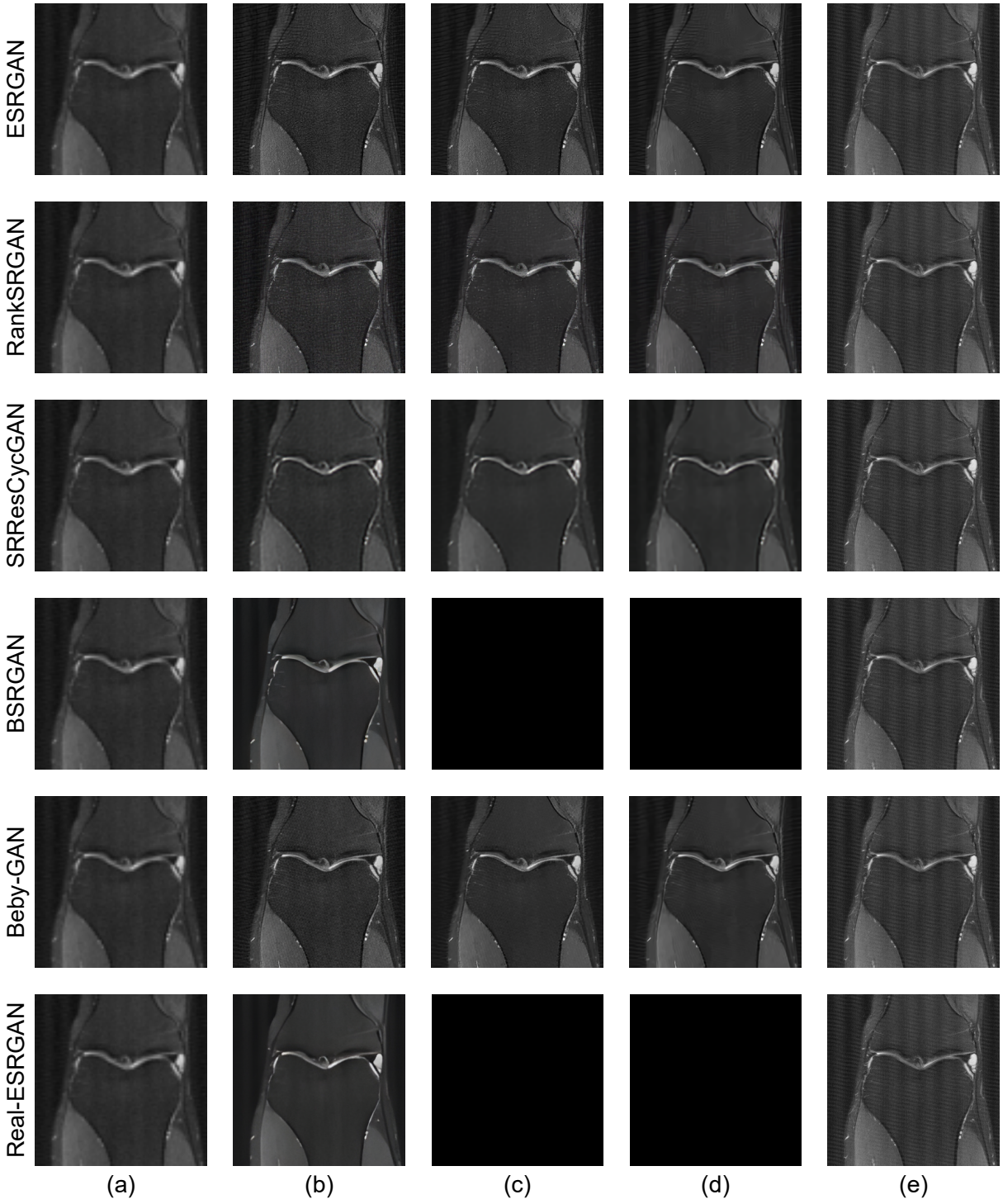


Figure A.2: (a) Input LR Images, (b) Generated Images w/o denoising, (c) Generated Images w/ NLM, (d) Generated Images w/ BM3D and (e) Ground Truth Images.

B

Additional Tumor Segmentation Qualitative Results

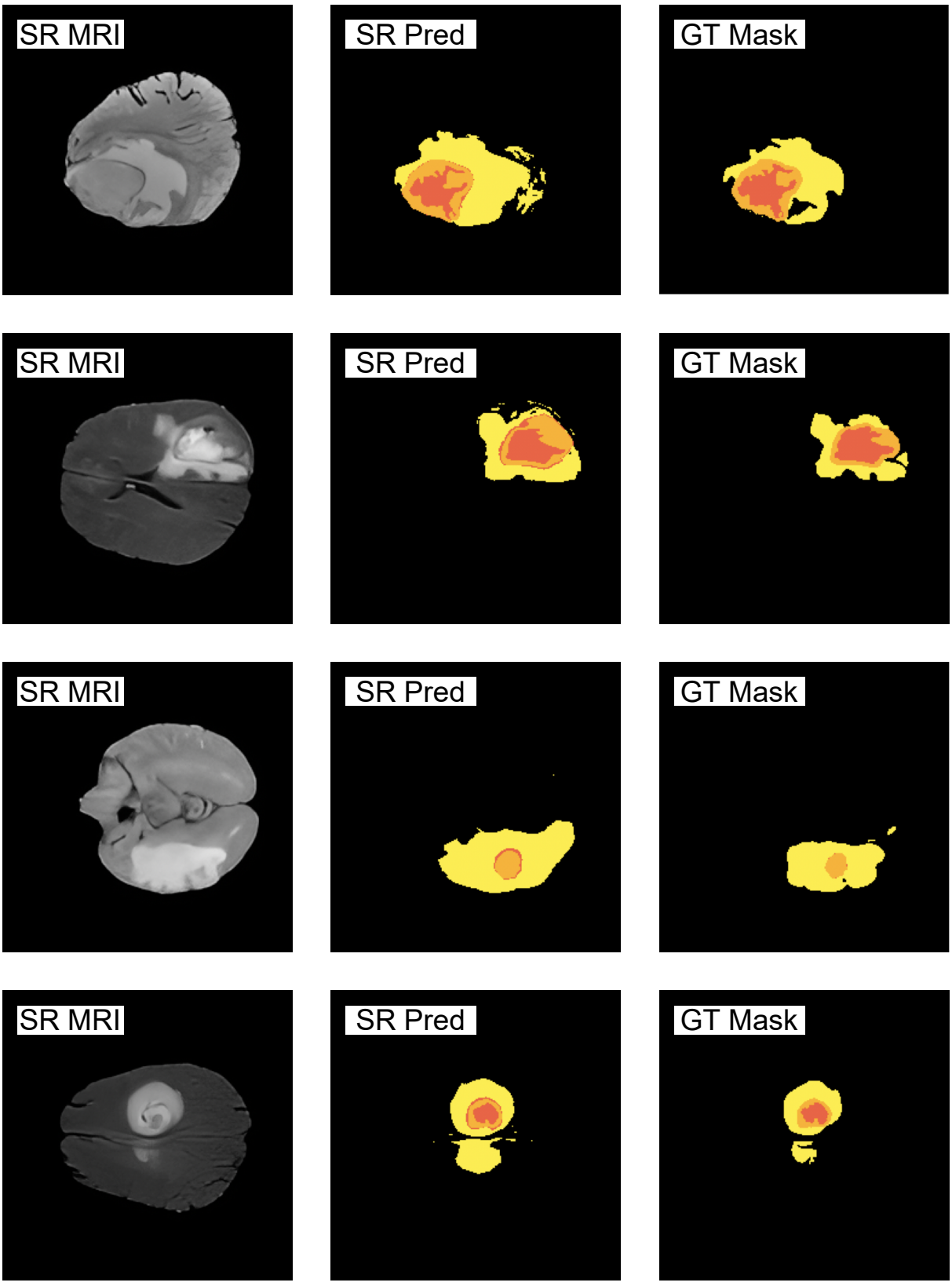


Figure B.1: Results of Tumor Segmentation with SRBraTS (Super-Resolved BraTS).

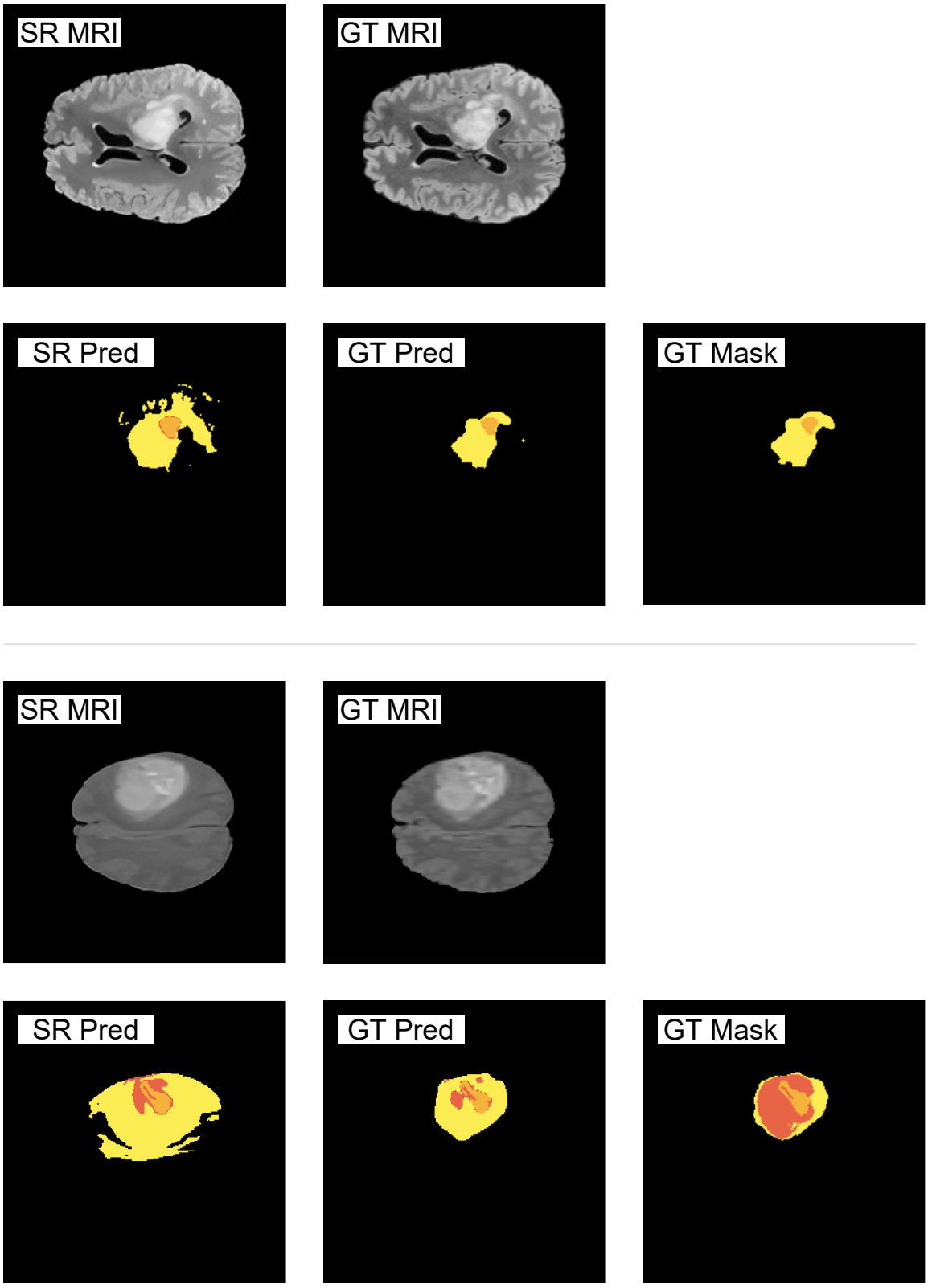


Figure B.2: (Part 1) Comparing the Tumor Segmentation over BraTS and SRBrats.

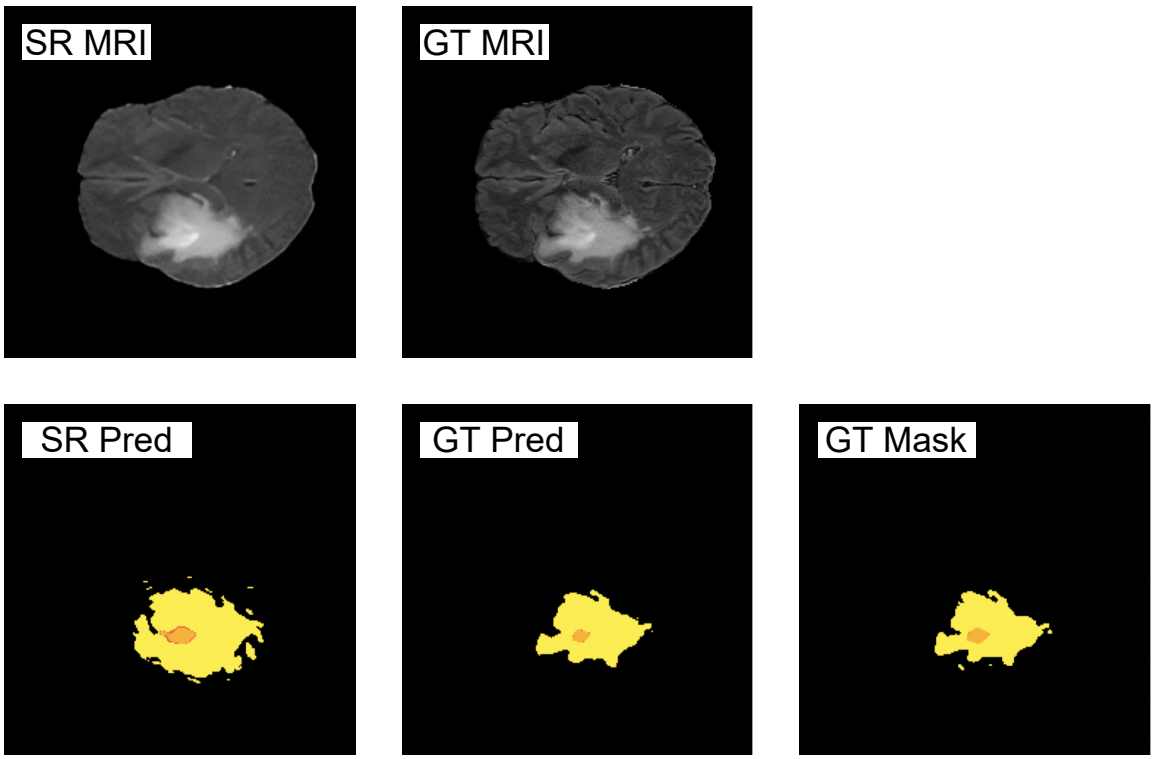


Figure B.3: (Part 2) Comparing the Tumor Segmentation over BraTS and SRBrats.