國立臺灣大學醫學院醫療器材與醫學影像研究所

碩士論文

Graduate Institute of Medical Device and Imaging

College of Medicine

National Taiwan University

Master Thesis

超解析度大腦磁振造影技術

The Development of Super-resolution for Brain MRI Images

吳蒨樺

Qian-Hua Wu

指導教授：吳文超 博士

Advisor：Wen-Chau Wu, Ph.D.

中華民國 112 年 8 月

Aug. 2023

doi:10.6342/NTU202303956

誌謝

完成碩士論文的過程中，必須感謝許多人的幫助。首先，我要感謝指導老師吳文超老師，他在醫學領域的專業知識豐富了我的學識，研究上的建議更讓我受益匪淺。從大學時期就開始指導我的趙一平老師和郭立威老師不僅給予我研究方向，更教會我如何從不同角度思考問題，探究問題的本質。老師們的耐心指導使我得以克服研究過程中的種種困難，而他們的建議與意見則使我的研究更加完整，充滿深度。這些經驗將成為我未來成長的重要基石。

在研究所期間，家中的貓咪胖虎陪伴著我度過許多重要時刻。每當我在研究上迷惘時，他獨特的陪伴方式總能給予我情感上的支持。胖虎的存在不僅提醒我生活中的美好，更讓我能夠以輕鬆的心情投入研究之中。

最後，感謝我的家人給予我生活上的一切支持，讓我得以專心投入研究。總之，感謝所有在我學術和生活道路上出現的人們。期盼我們的未來充滿更多的成長與收穫。

doi:10.6342/NTU202303956

摘要

磁振造影具有無輻射傷害以及軟組織高解析度等優點，但其成像時間長，病患可能會因運動雜訊造成影像品質降低，因此，若是能夠縮短掃描時間，便能減輕病患的負擔並降低成本。超解析度成像是一種提高影像解析度的技術，其優點在於不需額外硬體設備的支援，且近年來基於圖形運算單元發展，應用於磁振造影上的相關研究持續增加。而本研究結合超解析度成像與深度學習技術，將低解析度影像重建為高解析度影像，即使在短時間內進行掃描，獲得較低解析度的影像，也能透過超解析深度學習技術重建，從而實現造影流程的加速，並提升影像品質。

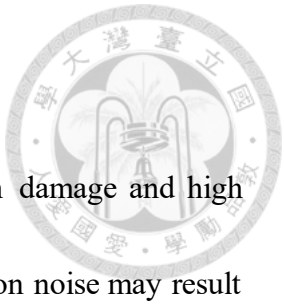
研究重點著重於模型架構的調整、殘差學習、位置感知、邊緣損失、感知損失以及低解析度影像的上採樣方法。為了驗證超解析度影像是否能夠在臨床上使用，我們將多發性硬化症的影像重建，並對高解析度影像與超解析度影像進行病灶分割，比較兩者的分割結果以證實病灶不會因超解析而消失。除了使用常見的結構相似性、峰值訊噪比及均方誤差評估影像，我們也透過數據的交叉訓練，將所有模型重建的超解析度影像映射至同一標準空間以生成不確定性圖，確認大腦各個區域的正確率與穩定性。

所有實驗中，使用位置感知的模型產生了最佳結果，結構相似性、峰值訊噪比與均方誤差分別為 0.97、36.19 以及 0.1179，病灶分割的戴斯係數和真陽性率也都達到 0.96，代表確實可以較短時間進行掃描，取得較低解析度的影像，再使用超解析度技術提高影像解析度，在可維持病灶資訊的前提下，提供臨床診斷所需的影像品質。

本研究專注於對大腦 T1 權重影像進行重建，若未來可針對其他大腦成像序列進行分析，將大幅提升研究價值，有望為醫療領域帶來更重要的實用價值。

關鍵詞：磁振造影、超解析度成像、生成對抗網路、殘差學習、位置感知、重複抽樣不確定性

doi:10.6342/NTU202303956



Abstract

The advantages of magnetic resonance imaging include no radiation damage and high resolution of soft tissues. However, the imaging time is long and motion noise may result in degradation of image quality. Therefore, shortening the scanning process can reduce the burden on the patient and reduce costs. Super-resolution imaging is intended to improve the quality of images. One of its advantages is that it does not require additional hardware equipment to function. Due to the development of graphics processing units, magnetic resonance imaging research has increased in recent years. In this study, low-resolution images were reconstructed into high-resolution images using super-resolution imaging and deep learning technologies. Despite a scan being performed quickly to obtain a low-resolution image, it can be reconstructed with super-resolution deep learning technology, thereby improving image quality and accelerating imaging operations.

This research focuses on the adjustment of the model architecture, residual learning, position awareness, edge loss, perceptual loss, as well as upsampling methods for low-resolution images. For the purpose of verifying that super-resolution images can be used clinically, we will reconstruct the multiple sclerosis image and segment the lesion on the high-resolution image and the super-resolution image, and compare their segmentation results in order to confirm that super-resolution will not result in the disappearance of the lesion. Besides using the common structural similarity, peak signal-to-noise ratio and

doi:10.6342/NTU202303956

mean square error as criteria for evaluating images, we also map the super-resolution images reconstructed by all models into the same standard space through data cross-training in order to verify the accuracy and stability of each brain area.

The model using position awareness produced the best results in all experiments. In terms of structural similarity, peak signal-to-noise ratio, and mean square error, the results were 0.97, 36.19, and 0.1179, respectively. Furthermore, both the true positive rate and the dice coefficient reach 0.96 for lesion segmentation. As a result, it is indeed possible to scan in shorter times, obtain images of lower resolution, and then use super-resolution technology to improve image resolution and provide the image quality required for clinical diagnosis while maintaining lesion information.

The purpose of this study is to reconstruct brain T1-weighted images. Research value will be greatly improved if other brain imaging sequences are analyzed in the future, and it is anticipated to have more practical value in the medical field.

Keywords: magnetic resonance imaging, super-resolution imaging, generative adversarial networks, residual learning, position-aware, bootstrap uncertainty

doi:10.6342/NTU202303956

Contents

doi:10.6342/NTU202303956

doi:10.6342/NTU202303956

List of Figures

doi:10.6342/NTU202303956

List of Tables

doi:10.6342/NTU202303956

Chapter 1 Introduction

1.1 Background

In recent years, Magnetic Resonance Imaging (MRI) has gradually become an important clinical diagnostic tool because of its non-invasive, and high-resolution imaging. An advantage of MRI over Computed Tomography (CT) is that it does not require ionizing radiation to image, so there is no radiation dose issue. In addition, MRI can resolve soft tissues and provide high-resolution tissue images of brain, heart, abdomen, muscles, and joints. Also, it can use pulse and signal measurement timing to generate different imaging sequences to obtain images with various parameters, such as Functional MRI (fMRI), Diffusion-weighted MRI (DWI), Perfusion-weighted MRI (PWI) and Susceptibility weighted imaging (SWI), etc.

However, CT is still the most widely used diagnostic method in clinical applications, the biggest reason comes from the limitation of imaging time except for the high cost of MRI equipment. CT only takes a few seconds to reconstruct an image with same resolution, while MRI may take a long time and easy to cause artifacts or blurred images due to patient movement. Also, the instrument used times reduce by excessively long imaging time and require more labor costs, so MRI is much more expensive than other detection methods.

doi:10.6342/NTU202303956

1.2 Purpose

The limitation of MRI imaging time is related to its principle that the magnetic field gradient is used to perform phase encoding and frequency encoding in signal space (k-space) to distinguish signal sources. Among this, phase encoding requires different magnetic field gradients each time an RF pulse is emitted. For example, a 320x320 image needs 320 times encoding, this is one of the factors contributing to the long imaging time. Although the speed can be accelerated by reducing phase encoding times, the image resolution will also decrease. And according to Nyquist Sampling Theorem, the image will get artifacts when the sampling frequency is lower than twice the maximum frequency of signal.

Therefore, to solve the hardware limitation, this study uses a neural network with powerful image reconstruction capability for post-processing and improves the blurred boundary problem of neural networks in the past.

1.3 Super-resolution

High-resolution images provide detailed information, but obtaining them is often difficult or time-consuming. Hence, improving image resolution has become a major concern in the past few years. In most cases, super-resolution is used to increase resolution. Super-resolution is the process of constructing high-resolution images from obtained low-

doi:10.6342/NTU202303956

resolution images. One method of super-resolution uses multiple low-resolution images to reconstruct a high-resolution image (multi-image super-resolution, MISR), while the other uses a single low-resolution image (single-image super-resolution, SISR). In MRI super-resolution, MISR provides higher accuracy, but it is less efficient than SISR. In addition, MRI scans take a long time to complete, making it hard to capture multiple images. As a result, SISR is a potential approach for solving the problem of super-resolution.

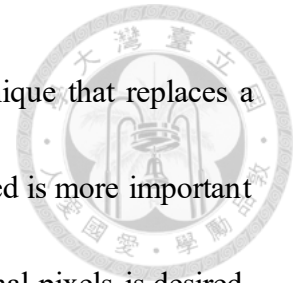
By using SISR, one low-resolution input may correspond to many possible high-resolution outputs, and it is usually difficult to map low-resolution input to high-resolution space. Thus, SISR is a notoriously challenging and ill-posed issue. At present, mainstream algorithms for SISR methods can be categorized into four categories: interpolation-based methods, reconstruction-based methods, and example-based methods.

1.4 Approaches for super-resolution

1.4.1 Interpolation-based methods

Interpolation is a method of estimating data points based on existing data points. It involves using a mathematical formula to fill in gaps between existing data points. Nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation are some of the most common interpolation methods.

doi:10.6342/NTU202303956



Nearest neighbor interpolation is a simple image interpolation technique that replaces a pixel with the value of its nearest neighbor. It is often used when speed is more important than accuracy. It is also used when an exact reproduction of the original pixels is desired.

Bilinear interpolation works by taking the weighted average of the four pixels surrounding the unknown pixel. The weights are based on the distance of the unknown pixel to each of the four surrounding pixels. The closer the unknown pixel is to a specific pixel, the higher the weight assigned to that pixel. The result is an interpolated pixel value that accurately reflects the color of the unknown pixel.

Bicubic interpolation uses a cubic polynomial to interpolate between pixels. The method works by fitting a curve to the four closest pixels to the desired one and then calculating the value of the desired pixel from the curve. It produces smoother results than nearest-neighbor interpolation, but is more computationally expensive.

1.4.2 Reconstruction-based methods

Super-resolution methods based on reconstruction often require complex prior knowledge, but they can generate high-resolution images with vivid and precise details. While reconstruction-based methods offer high performance, their performance declines rapidly when the scaling factor is increased, and these methods are often time-consuming.

doi:10.6342/NTU202303956

1.4.3 Example-based methods

Example-based super-resolution technique, also known as learning-based super-resolution, boosts the resolution of low-resolution images. For generating high-resolution images, it uses the relationship between low- and high-resolution pairs as a prior constraint. Compared to reconstruction-based methods, when enlarging amplification factors, it can achieve better results based on the information from the large number of training sets.

Traditional example-based super-resolution algorithms are generally divided into two categories based on how their priors are obtained. The first one is an implicit priori algorithm where the priori is directly reflected by the training set. Almost all K-nearest neighbor algorithms fall into this category. Clearly, implicit priors omit the learning process, but K-nearest neighbor searching makes it more expensive to recover high-resolution estimations. The second is explicit priori-based algorithms. To represent the learned priori, dictionary and regression functions are common methods. Dictionary-based algorithms use low- and high-resolution dictionary pairs to represent the priori between low- and high-resolution training images. Likewise, regression functions learned from supervised or semi-supervised methods are used to represent the priori relationship between low- and high-resolution images. Generally, when the training set is large, these explicit priori-based algorithms take a very long time to train. As a result, neither of these

doi:10.6342/NTU202303956

traditional example-based super-resolution algorithms is suitable for applying on large training sets.

doi:10.6342/NTU202303956

Chapter 2 Methods of Super-Resolution

2.1 Super-resolution Neural Network

With the development of deep learning methods and their successful application in the computer vision field, super-resolution convolutional neural networks (SRCNN) were introduced [1]. By utilizing only three layers of networks, it is possible to reconstruct natural images with high quality. SRCNN is a feed-forward network that consists of three steps, namely patch extraction and representation, non-linear mapping, and reconstruction. The patch extraction and representation step extracts small patches from the input image, and then maps these patches to a high-dimensional feature space, which captures the underlying characteristics of the input. The second step is a non-linear mapping step, which further enhances the feature representation to generate a more complicated representation that can be used to reconstruct the input image. Finally, the reconstruction step combines the features from the previous steps to generate a high-resolution image. The architecture of the SRCNN is shown in Figure 1.

doi:10.6342/NTU202303956

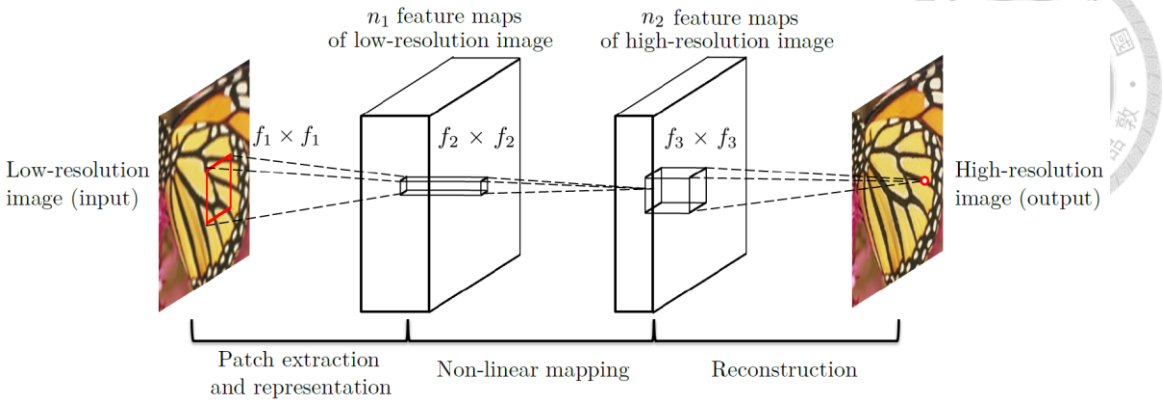


Figure 1. Architecture of SRCNN. [1]

2.2 Super-resolution Generative Adversarial Network

Based on Ian Goodfellow's Generative Adversarial Network (GAN) [2], [3] proposed a SRGAN for solving super-resolution problems. GAN consists of two models, a generative and a discriminative model. In the generative model, the goal is to create natural-looking images that closely resemble the original data distribution. In the discriminative model, a given image is analyzed to determine whether it looks natural or artificial. In the original paper, the generative model is compared to a team of counterfeiters, while the discriminative model is compared to police, trying to detect counterfeits. Both methods improve as alternating optimization is performed, until the discriminative model is unable to detect counterfeit currency at all. In SRGAN, the generative model generates HR images, whereas the discriminative model determines whether the image comes from the generative model or the original image from the database. If the discriminative model

doi:10.6342/NTU202303956

believes that the image generated by the generative model is the image from the database, then the generative model is considered as having successfully generated a high-resolution image. The process is then repeated until the discriminative model is unable to detect any differences between the generated and original images.

There are multiple residual blocks in the generator network (SRResNet). Residual blocks are the basic building blocks of residual networks [4], which provide solutions to problems of vanishing gradients. The discriminator is a binary classifier commonly used. The architecture of the SRGAN is illustrated in Figure 2.

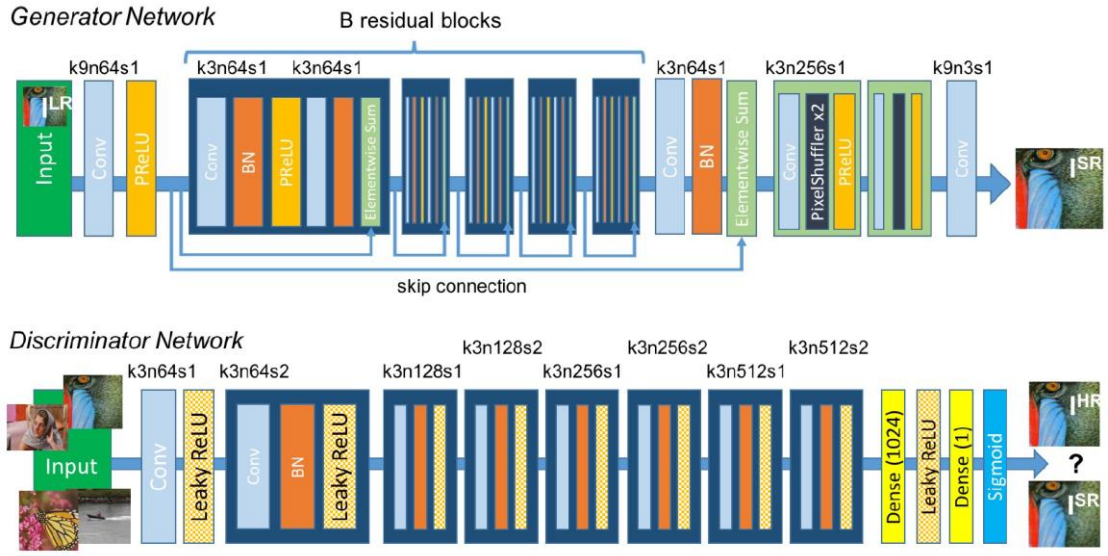


Figure 2. Architecture of SRGAN. [3]

doi:10.6342/NTU202303956

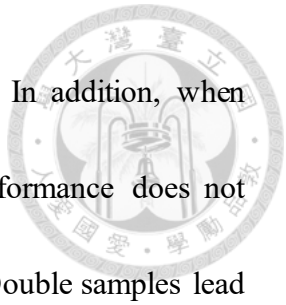
2.3 3D Deep Neural Network for Multimodal Brain MRI Super-resolution

In this study, MRI super-resolution was achieved by deep three-dimensional convolutional neural networks. In contrast with previous studies, this study analyzes eight factors that influence convolutional neural networks' performance in 3D brain MRI. Besides model-related adjustments and optimizations, residual learning and multiscale training are also used.

Residual learning aims to generate the difference between input and target. Final results are obtained by adding the model output to the input. Directly learning to generate the target from input may result in input information loss during convolution. Residual learning preserves input information and makes learning easier. It also reduces the learning process complexity. Additionally, residual learning can reduce generalization errors and improve generalization performance. This makes it a desirable approach to many machine learning tasks.

Since brain MR imaging can be acquired in a variety of settings, this study explores multiscale settings. Datasets for learning are scaled with factors 2 and 3. In the first training case, there are two single scale factors and one multiple scale factor, both with the same number of subjects. The second training case doubles the learning dataset for multiple scale factors. According to the results, reconstruction performance decreases

doi:10.6342/NTU202303956



significantly when scale factors differ from those used in training. In addition, when training on multi-scale data within the same training samples, performance does not change significantly compared to training on a single-scale dataset. Double samples lead to better performance in the training dataset. This indicates that multi-scale data in training benefits the model, and highlights the importance of using multiple scales for training. Furthermore, it shows that training on multiple scales is more beneficial than training on a single scale.

To improve the resolution of the low-resolution T1-weighted image, the proposed method in this study uses information from an image with a different contrast, such as a T2-weighted image or FLAIR image. However, the use of images with different contrasts can also lead to difficulties in registration, which can in turn lead to errors in the final image. The disadvantage of this method is that it takes extra time to acquire another contrast image, and the technique isn't always possible.

doi:10.6342/NTU202303956

2.4 Multi-Contrast Super-Resolution MRI Through a Progressive Network

The study proposes a one-level non-progressive neural network for low-upsampling multi-contrast super-resolution, and a two-level progressive neural network for high-upsampling multi-contrast super-resolution. In these networks, multi-contrast information is integrated into a high-level feature space. In addition, it minimizes mean-square error, adversarial loss, perceptual loss, and textural loss in order to optimize image quality.

The one-level non-progressive network is based on Wasserstein's generative adversarial network with gradient penalty (WGAN-GP) [7] and consists of a generator and discriminator. An encoder-decoder network and a reference feature extraction network make up its generator. Through an encoder-decoder network [8], image quality can be improved. Between encoder and decoder, there are three skip connections that transmit feature maps. Through the similarity between input and output images, skip connections can improve training processes. The reference feature extraction network extracts feature maps from reference images. After the feature maps have been extracted, they are fed into the encoder-decoder network. As with the encoder in the encoder-decoder network, the reference feature extraction network uses the same architecture.

In contrast, the two-level progressive network up-samples at a factor of 4 for multi-

doi:10.6342/NTU202303956

contrast super-resolution (MCSR). This model combines two one-level non-progressive networks. The generator consists of two encoder-decoder networks and one reference feature extractor network. It uses the same discriminator as the one-level non-progressive network. The first level increases the resolution by twofold. In the second level, images with a 4-fold resolution are obtained.

Besides the MSE loss and perceptual loss, this study also includes texture matching loss. By statistically matching extracted features, the texture matching loss proposed by [9] generates an image with high similarity between the generator output and the ground truth. This loss was then applied to natural images for super-resolution by [10]. MCSR imaging involves a style transfer between images of different contrasts. By applying this loss, all features are appropriately matched and fully utilized.

As in 2.3, it also requires another contrast image, which in this case is a proton density weighted image. However, this technique is not without its drawbacks. For one, it requires the use of two contrast images, which can add complexity to the imaging process. Additionally, it is not always possible to obtain a proton density weighted image, which can limit the usefulness of this technique.

doi:10.6342/NTU202303956

2.5 Cascade Neural-network Model for Brain MRI Super-resolution

A two-step deep learning architecture called DeepVolume is proposed in this study to reconstruct accurate thin-section MR images. A multitask 3-D U-net is used in the first stage to fuse thick-section MR images in axial and sagittal planes based on prior knowledge of brain volume segmentation. This is to ensure the reconstruction result has the correct brain structure, thus the name brain structure-aware network. The second stage is a spatial connection-aware network, which is a recurrent convolutional network embedded with long short-term memory (LSTM). By utilizing previously unassessed sagittal information, the preliminary reconstruction results are adjusted slice-by-slice to enhance the precision of the reconstruction.

In terms of its network architecture, the brain structure-aware network can be called a 3D U-net. Recently, many medical imaging studies ranked this network architecture as the top choice for medical image analysis [12]. The 3D U-net architecture is a type of convolutional neural network (CNN) that uses a 3D convolutional layer to capture a 3D volume of the image. It also uses a U-shaped structure with skip connections between the layers to capture more detailed information from the image. This makes it well-suited for medical image analysis, as it can capture subtle details that may be missed by other types of networks. The study transfers the segmentation network to the reconstruction network using a linear convolutional block as the last layer of the 3-D U-net. The contracting-

doi:10.6342/NTU202303956

expanding network architecture facilitates the deep fusion of thick-sections in two planes of the brain.

The brain structure-aware network utilizes only 50% of the sagittal information from thick sagittal MRIs. Since thick-section MR images in the sagittal plane are co-registered with axial thick-section images, the resolution in the sagittal plane is lower. Consequently, the second stage recovers more realistic and accurate thin-section images by utilizing previously unused sagittal information. Image details are extracted using an encoding–decoding network [13]. In order to strengthen the spatial connection of the reconstruction results, a convolutional LSTM block is inserted between the spatial connection-aware network. In a convolutional LSTM, the hidden state may carry the most important information about neighboring sagittal thick-section MR images, allowing accurate reconstruction of image details. It may be possible to generate strong spatially continuous reconstruction results in this way.

Although each network serves a distinct purpose, the two-step architecture requires twice as much time for reconstruction. This makes the two-step architecture less efficient than a single-step approach. Furthermore, the two-step architecture requires more resources, as it needs to store two sets of weights compared to one. This can lead to increased memory and storage costs.

doi:10.6342/NTU202303956

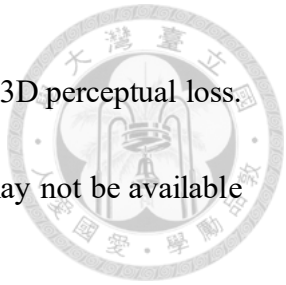
2.6 Super-resolution Optimized Using a Perceptual-tuned Generative Adversarial Network

This study proposes Super-resolution Optimized Using Perceptual-tuned Generative Adversarial Networks (SOUP-GAN) in order to produce thinner slices with anti-aliasing and deblurring. This system uses a scale-attention SR architecture that works for arbitrarily selected sampling factors. Further, the perceptual loss defined in the pretrained 2D VGG applies to 3D medical images. With GAN training, the method with 3D perceptual loss significantly improves perceived image quality.

The scale-attention architecture utilizes pre- and post-attention modules to deal with differential information between scales. The model is constructed using a sampling factor scale of 2 to 6. In practice, it can provide predictions with sampling factors ranging from 1.5 to 6.5 for most cases. Across all scales, most of the model parameters are shared. This allows the model to learn features at different scales, while maintaining a reasonable number of parameters. By calculating alignment weights, the study integrates selected sampling factors into its attention model. In contrast to conventional interpolation methods, this method allows for higher resolution image interpolation.

The 3D perceptual loss is calculated using a pre-trained VGG by integrating all the axial, sagittal, and coronal surfaces of the 3D data. In other words, for the loss to be calculated, all the 3D patch planes must be input into the VGG. One significant drawback of this

doi:10.6342/NTU202303956



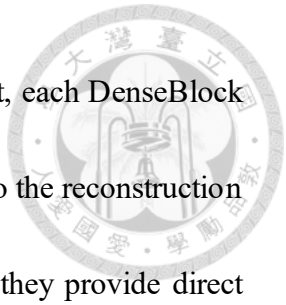
method is that it is much more computationally expensive to calculate 3D perceptual loss. In addition, this method requires a pre-trained VGG model, which may not be available in all cases.

2.7 3D Multi-level Densely Connected Super-resolution Network with Generative Adversarial Network

In this study, a multi-level densely connected super-resolution network (mDCSRN) is proposed using a generative adversarial network (GAN). With the mDCSRN, training and inferences take place quickly, and the GAN facilitates realistic output that is hardly distinguishable from original HR images. In the experiment, the architecture recovers 4x resolution-downgraded images and runs 6x faster than other popular deep learning methods.

According to a recent study [16], a densely connected super-resolution network (DCSRN) with a single DenseBlock can capture image features and restore super-resolution images, outperforming other state-of-the-art methods. However, the network performance needs to be improved so that a deeper model can capture more complex information. As the number of layers increases, the memory consumption of a DenseNet increases dramatically, making deeper DCSRNs difficult to train or deploy. To solve this problem, the study proposes a multi-level densely connected structure, in which a single deep

doi:10.6342/NTU202303956



DenseBlock is divided into several shallow blocks. As with DenseNet, each DenseBlock takes the output of all previous DenseBlocks and connects it directly to the reconstruction layer. The skip connections enable uninterrupted gradient flow since they provide direct access to all previous layers. There is less overfitting and more efficiency with this method. As opposed to DenseNet, mDCSRN does not have a pooling layer, allowing it to use information at full resolution. A further improvement is to add a 1x1x1 convolutional layer as a compressor before all the following DenseBlocks. Information compression is one of the key attributes that enables deep learning models to generalize so well. As a result, the model learns universal features to avoid overfitting.

Due to the efficiency of the model and the information sharing, we chose this model as our basic model. The architecture of mDCSRN-GAN is shown in Figure 3.

doi:10.6342/NTU202303956

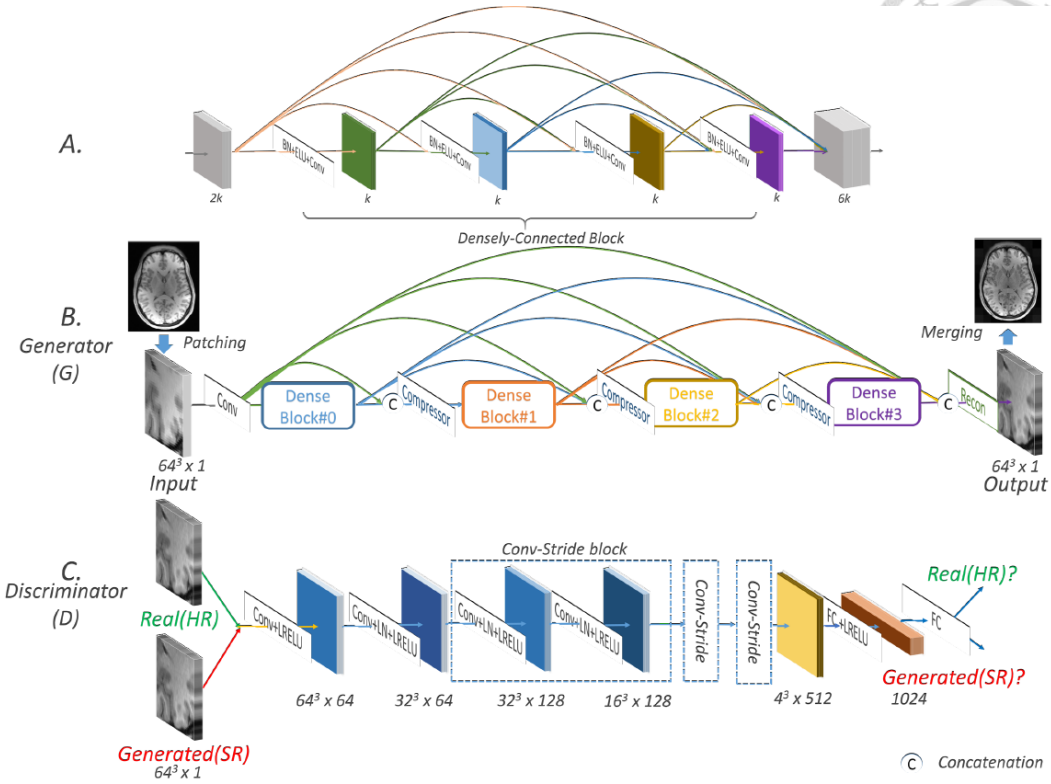


Figure 3. Architecture of mDCSRN-GAN. [15]

doi:10.6342/NTU202303956

Chapter 3 Experiments and Evaluation methods

3.1 Dataset

In this study, we use two datasets. The first applies to model training and testing, while the second is for evaluating images with lesions.

3.1.1 WU-Minn HCP 1200 Subjects Data

Sixteen components of the National Institutes of Health (NIH) have contributed to the Human Connectome Project (HCP), which will last five years and be divided into two consortia of research institutions. A key objective of HCP is to build a "network map" (connectome) that will shed light on the anatomical and functional connectivity of the healthy human brain. In addition, the data collected will facilitate research into brain disorders like dyslexia, autism, Alzheimer's disease, and schizophrenia.

As part of the WU-Minn-Oxford consortium, improved MRI instrumentation, image acquisition techniques, and image analysis techniques have been developed for mapping human brain connectivity with spatial resolutions that are significantly better than previously possible. Through these methods, the WU-Minn-Oxford consortium used a special 3 Tesla MRI scanner to acquire MRI images (1113 with structural MR scans) and behavioral data on 1,206 healthy young adult participants, including twins and their siblings from 300 families collected between 2012 and 2015. Additionally, 184 subjects

doi:10.6342/NTU202303956

had multimodal 7T MRI scans and 95 subjects had resting-state MEG (rMEG) and/or task MEG (tMEG) data. For the first time, 3T MRI and behavioral retest data for 46 subjects were also available.

In all cases, the structural images are 320x320x256 with a resolution of 0.7mm x 0.7mm x 0.7mm. We use T1-weighted images. 990 subjects were selected as training data, 111 subjects as validation data, and 11 subjects as testing data.

3.1.2 2008 MICCAI MS Lesion Segmentation Challenge

Medic Image Computing & Computer-Assisted Interventions Society (MICCAI) is a professional organization for scientists. As these fields are multidisciplinary, members of the society come from several scientific disciplines, including computer science, robotics, physics, and medicine. Among the most well-known events of the society is its flagship event, The MICCAI Conference. It facilitates the publication and presentation of original research related to MICCAI.

In 2008, the competition in conjunction with MICCAI Conference aimed to compare algorithms for segmenting Multiple Sclerosis (MS) lesions. Among its datasets, there are 20 labeled cases used for training and 31 unlabeled cases, which originated from two sources: Children's Hospital Boston (CHB) provided 28 datasets (10 for training and 18 for testing) and University of North Carolina (UNC) provided 23 datasets (10 for training

doi:10.6342/NTU202303956

and 13 for testing). Cases at UNC were acquired on Siemens 3T Allegra MRI scanners with 1mm slice thicknesses and 0.5mm in-plane resolutions. CHB cases did not have scanner information available. MRI images for all subjects are re-sliced to 512x512x512 with 0.5mm x 0.5mm x 0.5mm resolution.

However, neither site has standardized its protocol, hence we only use data from CHB. Furthermore, the ground truth for lesion labels is only included in training data, which is why we use 10 subjects to test the method.

3.2 Preprocess

Because of the complex nature of MRI imaging, many factors may affect image quality and cause noise. These factors may include patient motion, choice of coil, and radiofrequency pulse parameters. Poor image quality can lead to inaccurate diagnosis and treatment, so it is important to optimize MRI images to reduce noise and improve image quality.

Figure 4 shows an image of the HCP dataset with noise in the axial view. Despite noise almost being confined to peripherals of the brain and not affecting interpretation, it will be classified as an edge in the following steps of edge detection. Thus, if the brain mask is labeled, it can be used to eliminate outside noise.

doi:10.6342/NTU202303956

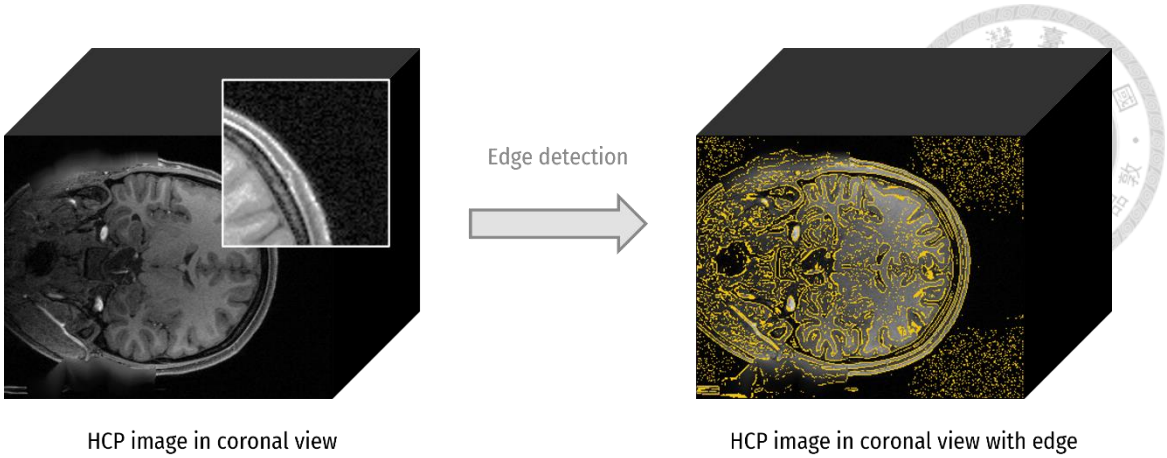


Figure 4. Image with noise in the HCP dataset.

The labeling process for masks is shown in Figure 5. First, binarize the original high-resolution images. In experiments, 0.03 threshold has shown to be the best setting. Next, perform the closing operation in morphology on the binary image using a disk with a radius of 10 to obtain the brain mask. Last but not least, set the pixel value outside the mask to zero to reduce peripheral brain noise. This will help to reduce the distraction while processing each image, allowing the neural network to focus on the important information. It also makes the data easier to interpret and analyze.

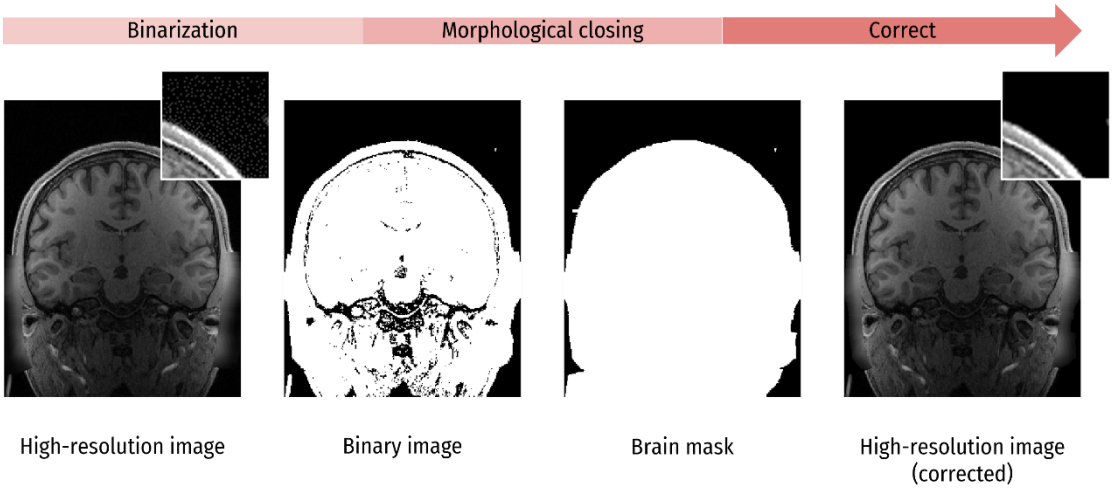


Figure 5. The labeling process for brain masks.

doi:10.6342/NTU202303956

3.3 Down-sampling

Figure 6 illustrates the steps involved in downsampling. In order to generate low-resolution images, we maxpool high-resolution images with 2x2x2 kernels and stride 2 for downsampling. By the end of this process, we should have 160x160x128 pixels of images.

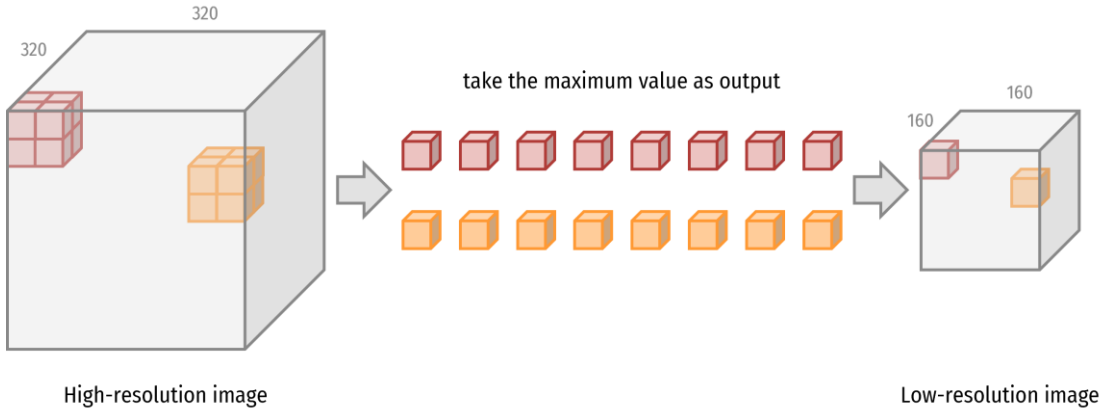


Figure 6. Steps for downsampling high-resolution images.

3.4 Up-sampling

To enter low-resolution images into the model, the resolution must first be increased. For this, we try two different methods, upsampling from the spatial domain and frequency domain, respectively. The following describes the implementation process and results of the two methods.

doi:10.6342/NTU202303956

3.4.1 Upsampling based on the spatial domain

Here, we directly perform cubic interpolation on the downsampled images. This helps to reduce the computational cost while still preserving the overall image quality. Furthermore, this technique is also beneficial for producing sharper edges and more accurate results.

3.4.2 Upsampling based on the frequency domain

Figure 7 illustrates the steps involved in upsampling based on frequency domain. First, perform a Fourier transform on the downsampled image generated in step 3.2. The resulting image is 160x160x128 in the frequency domain. Then fill the surrounding area with zeros to 320x320x256. For the final step, perform the Fourier transform again in order to obtain the image in spatial domain.

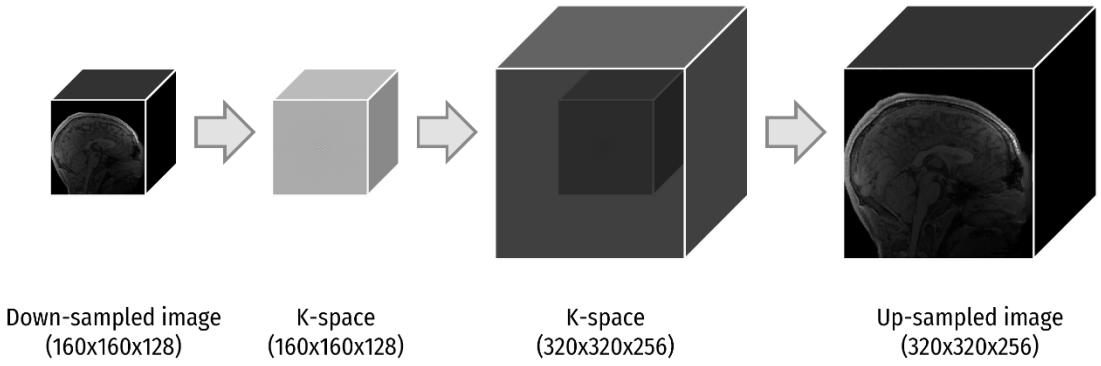


Figure 7. Pre-upsampling steps for low-resolution images in the frequency domain.

Despite the fact that upsampling in the frequency domain can provide more detail, Nyquist sampling theorem indicates that aliasing may occur when the sample frequency

doi:10.6342/NTU202303956

is less than the effective frequency. In our experiment, the brain area does not generate serious aliasing, but the surrounding area does. Figure 8 illustrates how aliasing appears in surrounding areas after frequency domain processing.

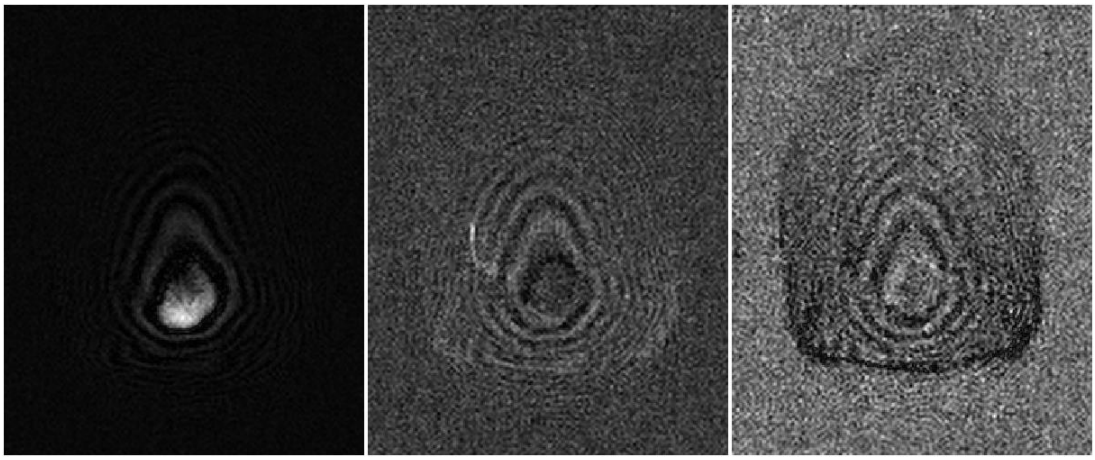


Figure 8. The surrounding areas of the image have aliasing after pre-upsampling in frequency domain.

According to [17], we use compressed sensing. This will restore the original image that is free from noise and aliasing. The compressed sensing approach uses advanced algorithms to reconstruct the original image with fewer measurements than the previous methods. This reduces the computational time and cost of the procedure. The compressed sensing steps are shown in Figure 9. The first step is to input the downsampled k-space (①). In order to obtain the aliased image (②), apply the Fourier transform. After that, use wavelet transformations and thresholding to denoise the image (③-⑤). Denoised images are Fourier transformed to obtain their k-space (⑥). Mask the denoised k-space (⑦) with the mask in 1, and then subtract 1 from it (⑧). As a final step, perform the Fourier

doi:10.6342/NTU202303956

transform again (⑨) and add it to the aliased image (⑩). From ② to ⑩, we repeat until the threshold no longer increases.

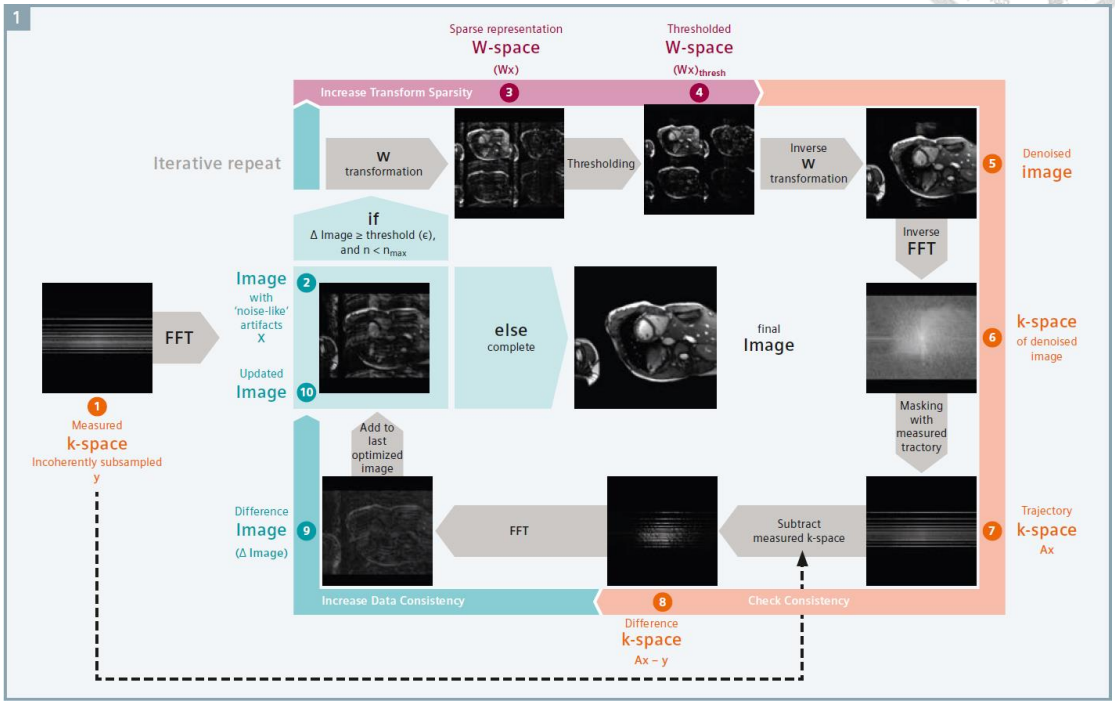


Figure 9. Compressed sensing steps. [17]

Using compressed sensing, images go from a normal histogram to a very extreme histogram (Figure 10(a)). Due to this, training results are poor. Thus, we use histogram matching to match the histograms of the compressed sensing image and the cubic interpolation image (Figure 10(b)(c)). By doing this, we can obtain a better perceptual quality of the reconstructed image. The resulting images are then used to retrain the neural network. This approach helps improve the performance of the network and produces better results.

doi:10.6342/NTU202303956

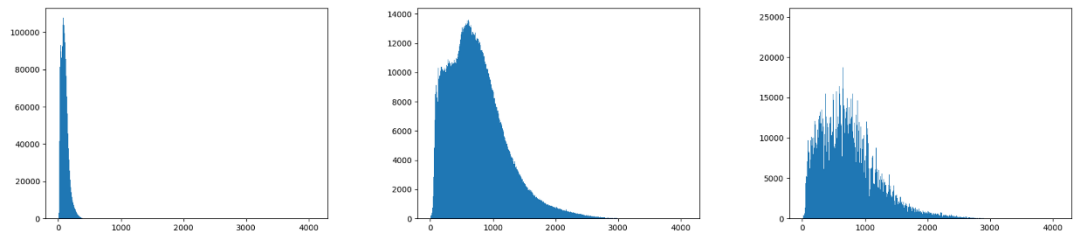


Figure 10. (a) Histogram of the image after compressed sensing. (b) Histogram of the image after cubic interpolation. (c) Result of histogram matching.

3.5 Patching, merging, and data augmentation

Traditionally, super-resolution models are fed images almost in two dimensions. Since MRI images are three-dimensional, it is advantageous to generate images with higher resolution if trained with 3D information. Due to memory limitations, it is not possible to input the entire 3D image into the model. Accordingly, we follow the same patching and data augmentation procedures as described in [15]. The size of each patch is 64x64x64. To augment the data, we used stride 32 instead of 64 as in the reference paper. This allowed us to generate more patches from the same data and increased the amount of training data.

In terms of validation and testing data, there is no overlap at the beginning. Directly input the patches generated by stride 64 into the model. After model conversion, piece them together to reconstruct the image. However, image input in the form of patches to a model

doi:10.6342/NTU202303956

is more likely to produce noise on the boundary. Each patch's cutting position can be clearly seen on the splicing result. In the same way, we refer to the paper method above in order to solve this issue. As shown in Figure 11, the first step is to pad images with zero. During patching, overlap the patch just a little, which is roughly the size that can remove the boundary noise. In our method, 16 is used. As a final step, remove the overlap portion of patches in the stitching process after reconstructing them through the model.

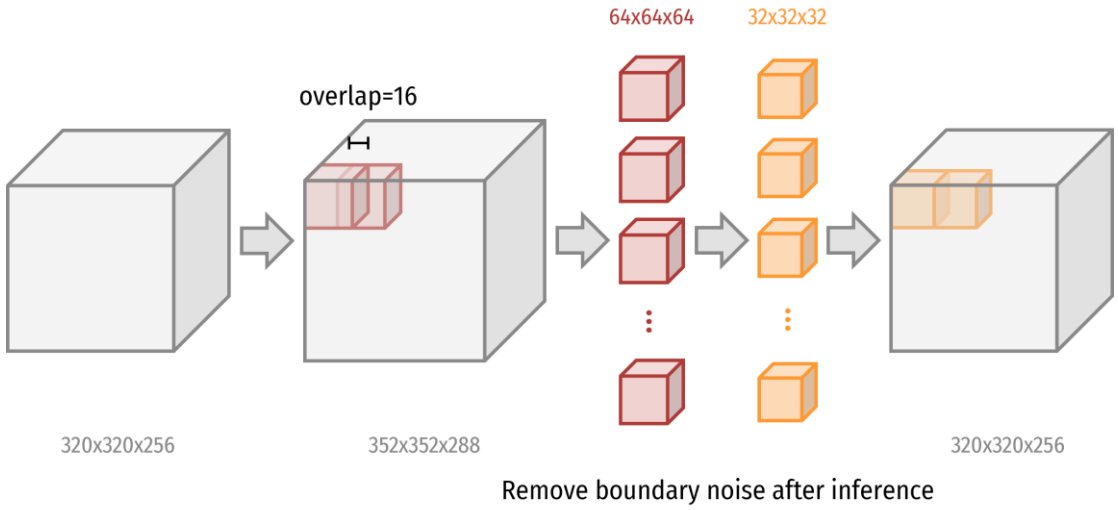


Figure 11. A step-by-step guide to merging patches.

3.6 Training

3.6.1 Experiment 1 (edge loss)

To enhance an image's edge, we design a loss function combined with Canny edge detection [18]. The flow chart for Canny edge detection is shown in Figure 12. First, remove noise from the image with a Gaussian filter. Then, compute the maximum

doi:10.6342/NTU202303956

gradient of each pixel according to the four gradient directions in Figure 13. In non-maximum suppression, each pixel is compared with its two neighbors along the same gradient. If the pixel gradient is not the maximum, it should be removed. Finally, the hysteresis thresholding is applied to find the best edge of the image. The thresholding is based on the low-threshold and high-threshold. Pixels with intensity above the high-threshold are accepted as edge pixels, and pixels with intensity between the low-threshold and high-threshold are accepted when they are connected to any pixel above the high-threshold.

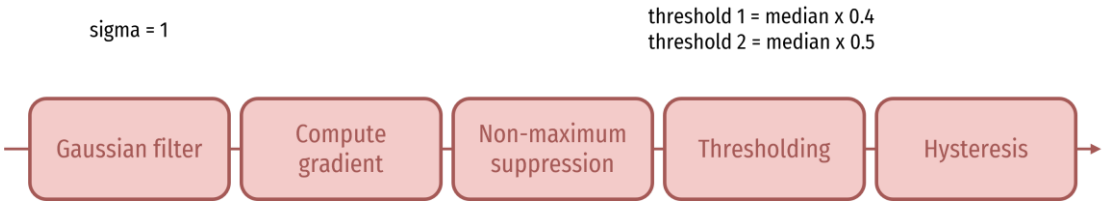


Figure 12. Canny edge detection steps.

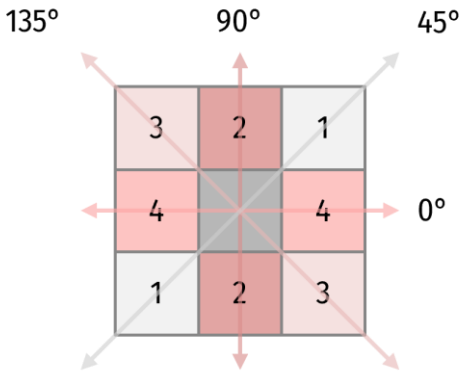


Figure 13. Four gradient directions.

In terms of the loss function, we use Canny edge detection to detect the edge of the image, in hopes that this will enhance high frequency information. The modified loss function is

doi:10.6342/NTU202303956

as follows:

$$loss\ function = \frac{1}{N}\Sigma_{i=1}^{N}\mu \times (SR_i - HR_i)^2 + (1-\mu) \times [C(SR_i) - C(HR_i)]^2$$

$$\mu = \begin{cases} 0.85,\ if\ C(HR_i) = 0 \\ 0.15,\ if\ C(HR_i) = 1 \end{cases}$$

As shown in the formula, N represents the number of pixels in the image, SR represents the super-resolution image generated by the model, HR represents the high-resolution image (ground truth), and C represents the Canny edge detection algorithm. We believe that the computed edge proportion should be higher when pixel HR_i is the edge. Based on [19], the coefficient μ should be 0.85 when HR_i is an edge, and 0.15 otherwise. Thus, compared to calculating only pixel differences, the edge detection method should generate a more detailed image.

Figure 14 illustrates the process of training. Low-resolution images are input to the model with a modified loss function that combines edge detection. High-resolution images are the model's objective. In addition, low-resolution images are upsampled in spatial domain. The model is optimized to produce high-resolution images that are indistinguishable from the original images. This process is repeated until the desired image quality is achieved. The trained model can then be applied to other data.

doi:10.6342/NTU202303956

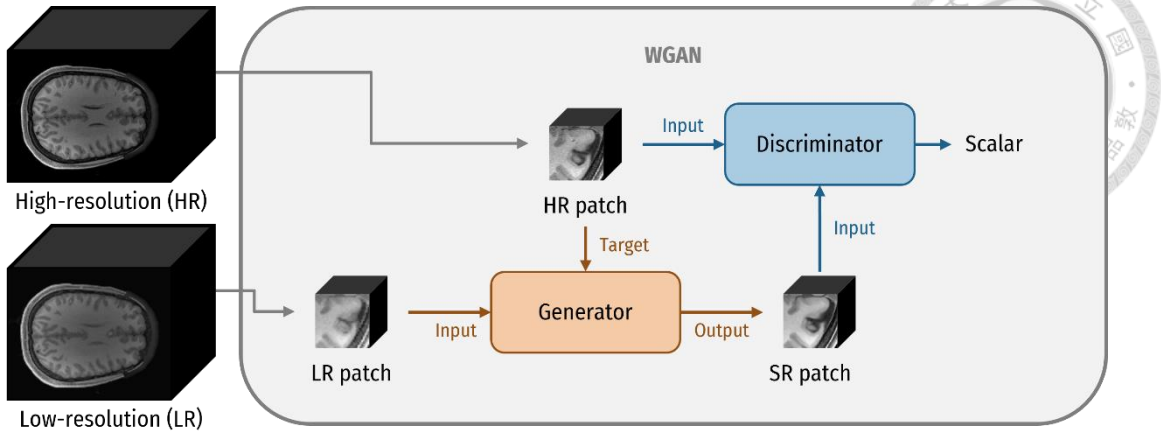


Figure 14. Training process for experiment 1.

3.6.2 Experiment 2 (residual learning with two-step training)

In order to make the edges of the image less smooth, we adjust the model architecture. As seen in experiment 1, the model inputs low-resolution images with the goal of outputting the same image as high-resolution images. However, low-resolution images may lose information when they are convolutioned. To prevent this situation, we use a pretrained model to generate the initial super-resolution image, which contains more detail information than the low-resolution image, but it isn't sufficiently clear. Afterward, input the initial image into the second model whose objective is the difference between the high- and low-resolution images (residual). Lastly, add residual to low-resolution image to obtain super-resolution image. In other words, it does not have to convolution the low-resolution image. Figure 15 shows the flow chart for experiment 2. Additionally, low-resolution images are up-sampled in spatial domain.

doi:10.6342/NTU202303956

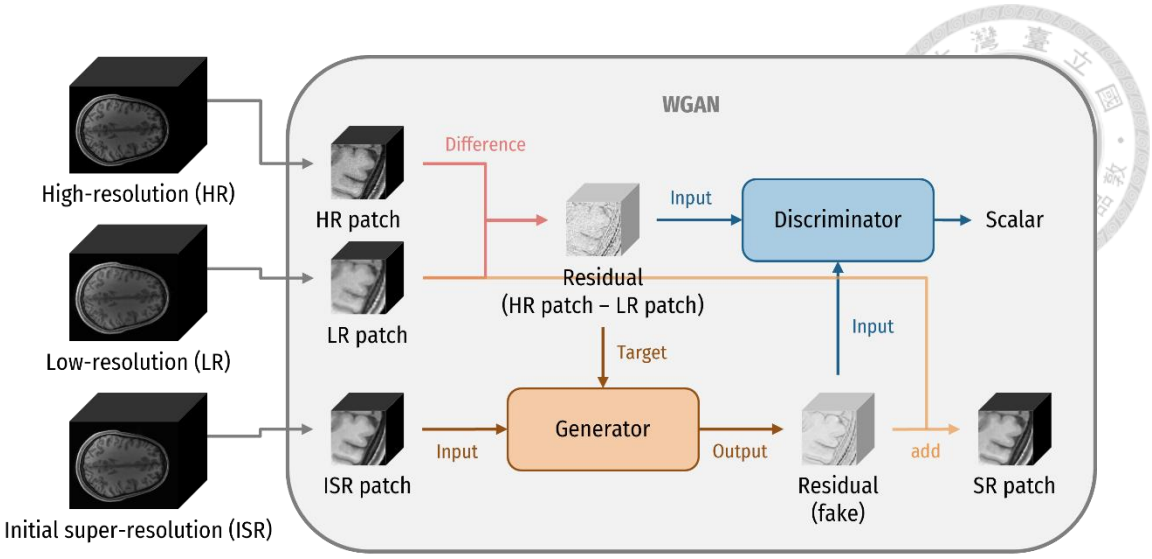


Figure 15. Training process for experiment 2.

3.6.3 Experiment 3 (residual learning)

Using residual images as the objective can yield detailed images, but it takes twice as long because two models are involved. In this experiment, we generated residual images directly from low-resolution images instead of using the pretrained model output. Thus, it can use less reconstruction time while maintaining low-resolution image information. Furthermore, this experiment's upsampling is performed in a spatial domain, as indicated by its flow chart (Figure 16).

doi:10.6342/NTU202303956

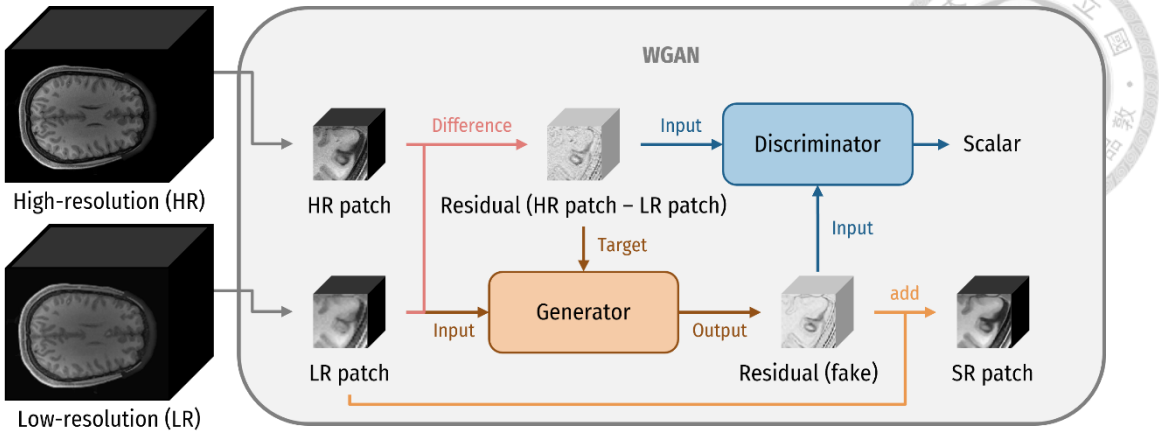


Figure 16. Training process for experiment 3.

3.6.4 Experiment 4 (residual learning with different upsampling method)

Unlike experiment 3, experiment 4 upsamples in the frequency domain instead of the spatial domain. The flow chart is the same as figure 16. Although the upsampled images contain aliasing, they are more detailed. During super-resolution, artifacts can also be reduced. Moreover, frequency domain upsampling allows for greater flexibility in signal filling. This flexibility can be used to fill in missing details, allowing for better image reconstruction.

3.6.5 Experiment 5 (position-aware residual learning)

In experiment 5, the brain atlas is combined. Since the model input is in patch format, it does not know the location of the patch in the image. For this reason, we train the model with atlas information so that it knows the location of the patches. Automated Anatomical

doi:10.6342/NTU202303956

Labeling [20] is the atlas we used and the generate step is shown in Figure 17. The first step is to map the low-resolution image to MNI space to obtain the transform matrix. As a next step, invert the transform matrix. The inverse matrix is then applied to the template in MNI space. As a result, we will be able to get brain atlases for each subject in the subject space.

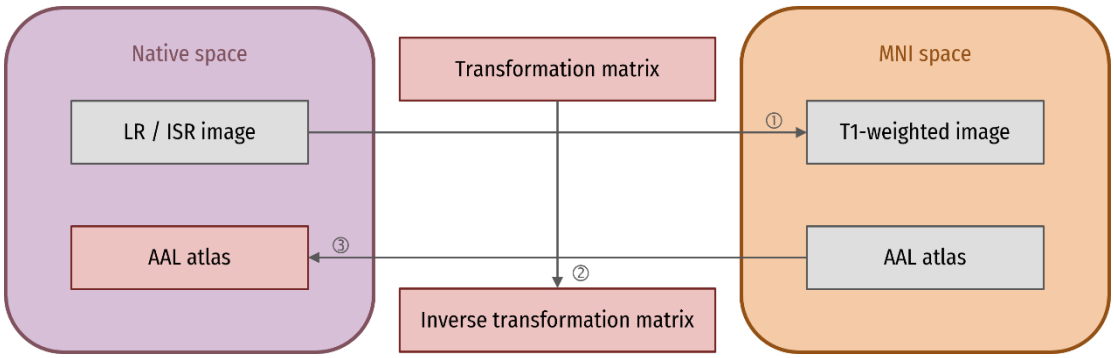


Figure 17. Steps for generating the AAL atlas.

Automated Anatomical Labeling (AAL) is a computer-assisted method for labeling anatomical structures in medical images. It can be used to accurately identify anatomical structures in medical images, allowing for faster and more accurate diagnoses. It also reduces manual labeling time, enabling clinical professionals to spend more time with their patients. In AAL1, there are 90 anatomical volumes of interest. Anatomical regions in each hemisphere are shown in Table 1.

Table 1. Anatomical regions in each hemisphere and their label. [20]

Anatomical description	**Label**
Central region	
Precentral gyrus	PRE

doi:10.6342/NTU202303956

Postcentral gyrus	POST
Rolandic operculum	RO
Frontal lobe	
Lateral surface	
Superior frontal gyrus, dorsolateral	F1
Middle frontal gyrus	F2
Inferior frontal gyrus, opercular part	F3OP
Inferior frontal gyrus, triangular part	F3T
Medial surface	
Superior frontal gyrus, medial	F1M
Supplementary motor area	SMA
Paracentral lobule	PCL
Orbital surface	
Superior frontal gyrus, orbital part	F1O
Superior frontal gyrus, medial orbital	F1MO
Middle frontal gyrus, orbital part	F2O
Inferior frontal gyrus, orbital part	F3O
Gyrus rectus	GR
Olfactory cortex	OC
Temporal lobe	
Lateral surface	
Superior temporal gyrus	T1
Heschl gyrus	HES
Middle temporal gyrus	T2
Inferior temporal gyrus	T3
Parietal lobe	
Lateral surface	
Superior parietal gyrus	P1
Inferior parietal, but supramarginal and angular gyri	P2
Angular gyrus	AG
Supramarginal gyrus	SMG
Medial surface	
Precuneus	PQ
Occipital lobe	
Lateral surface	
Superior occipital gyrus	O1

doi:10.6342/NTU202303956

Middle occipital gyrus	O2
Inferior occipital gyrus	O3
Medial and inferior surfaces	
Cuneus	Q
Calcarine fissure and surrounding cortex	V1
Lingual gyrus	LING
Fusiform gyrus	FUSI
Limbic lobe	
Temporal pole: superior temporal gyrus	T1P
Temporal pole: middle temporal gyrus	T2P
Anterior cingulate and paracingulate gyri	ACIN
Median cingulate and paracingulate gyri	MCIN
Posterior cingulate gyrus	PCIN
Hippocampus	HIP
Parahippocampal gyrus	PHIP
Insula	IN
Sub cortical gray nuclei	
Amygdala	AMYG
Caudate nucleus	CAU
Lenticular nucleus, putamen	PUT
Lenticular nucleus, pallidum	PAL
Thalamus	THA

Experiment 5's model architecture is shown in Figure 18. The atlases are patched and input into the model before each dense block. Atlas patches are concatenated with features and then convoluted. By using this method, the model could be provided with patch location information. In addition, densely inputting atlas patches might prevent lost information during convolution. Depending on the region, the model may learn different characteristics. Hence, the more detailed atlas assists us in improving image quality. This method also enables the model to capture global and local information simultaneously.

doi:10.6342/NTU202303956

The model can then utilize both types of information to generate a more accurate output. Furthermore, this approach improves the stability of the model, making it more robust.

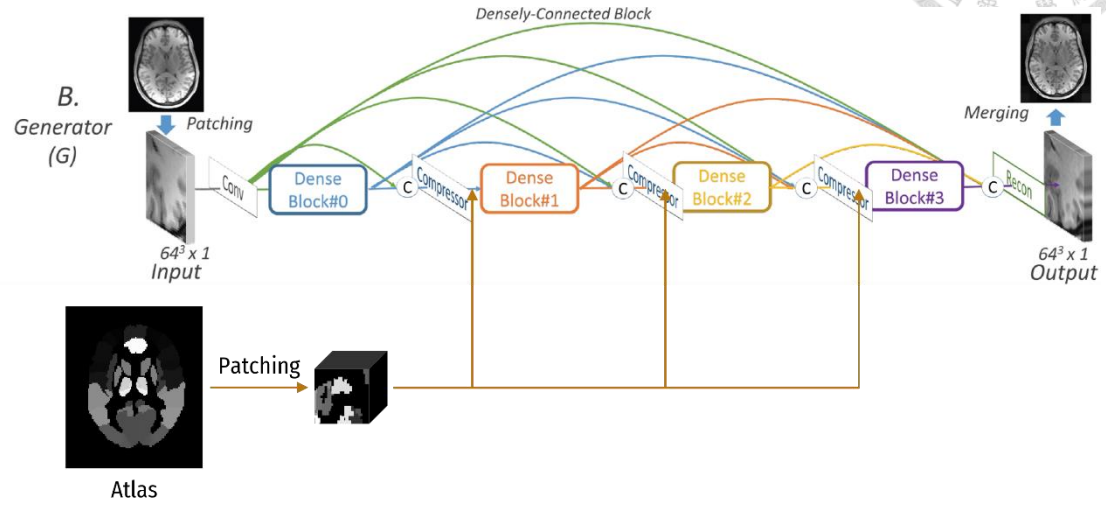


Figure 18. The model architecture of experiment 5.

3.6.6 Experiment 6 (residual learning using energy-based GAN)

Model performance of GAN architecture depends on the interaction between generators and discriminators. The result may be poor image quality if one of the networks is not good enough. Therefore, we refer to the [21] method for redesigning models. It considers discriminator to be an energy function. As such, let's call this model Energy-based Generative Adversarial Network (EBGAN). Here's the loss function:

$$Loss_G = D\big(G(LR)\big)$$

$$Loss_D = D(HR) + max\Big(0,\, m - D\big(G(LR)\big)\Big)$$

G represents the generator network, D represents the discriminator network, and m

doi:10.6342/NTU202303956

represents the margin. However, the discriminator loss should be intuitively understood as follows.

$$Loss_D = D(HR) - D\big(G(LR)\big)$$

For real data, the discriminator output should be small, but for fake data (generator output), the output should be large. In reality, zero is the best discriminator output for real data. In addition, it is difficult to reconstruct. The discriminator output of fake data may be infinitely enlarged under this condition. For this reason, we need a margin to control the second term of the formula in a range. In our experiments, we set it to 2 based on the number of patch inputs.

When real and generator data distributions are the same, then $D(HR)$ is the same as $D\big(G(LR)\big)$. Under the assumption that the value is γ, the loss is m when γ is between zero and m. After training, γ should be between zero and m for real data, but not for generator data. As EBGAN's discriminator is an auto-encoder, the region with low reconstruction error is finite. It is only a small percentage of inputs that reconstruct correctly after passing through the bottleneck. In other words, only a small part of the region is low energy, and the rest is high energy. Therefore, using EBGAN does not result in failure training as in original GAN.

Figure 19 shows the changed architecture of discriminator. In EBGAN discriminator, there is an encoder, a decoder, and a critic. Similar to the auto-encoder, the encoder maps

doi:10.6342/NTU202303956

the input image to a latent space, and the decoder reconstructs the image from the latent space. Moreover, the critic scores the images generated by the decoder. During our experiment, the critic evaluated score using Mean Square Error (MSE). The critic's score is then used to optimize the parameters of the encoder and decoder. The optimization of the critic encourages the encoder and decoder to generate images similar to the input image. This helps to improve the quality of the generated images.

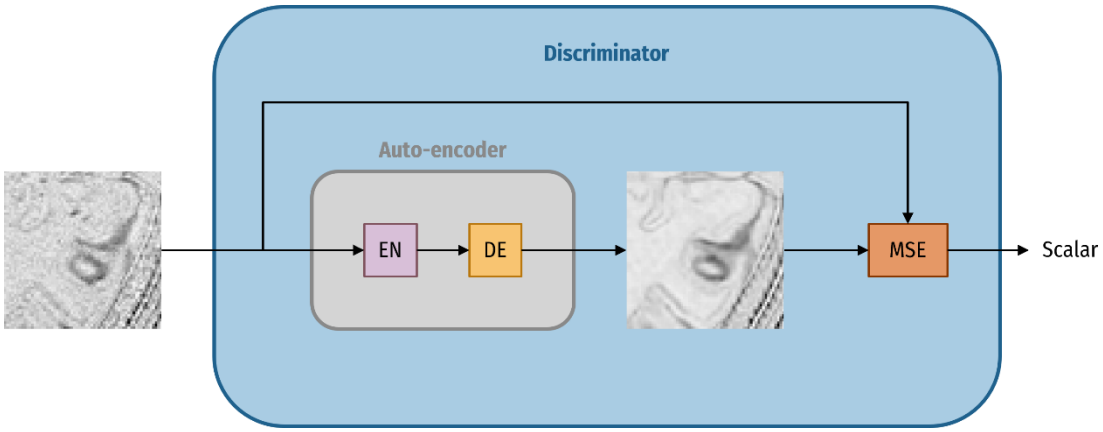


Figure 19. The discriminator architecture of experiment 6.

3.6.7 Experiment 7 (position-aware residual learning using energy-based GAN)

This experiment combines experiments 5 and 6. The purpose of experiment 7 is to confirm whether EBGAN training with brain atlas produces better results than without.

doi:10.6342/NTU202303956

3.6.8 Experiment 8 (position-aware residual learning with two-step training)

This experiment combines experiments 2 and 5. Using a two-stage training method, experiment 2 generates residuals from a super-resolution image generated by a pretrained model. Experiment 8 is designed to determine whether training with the brain atlas produces better results than training without. In this case, super-resolution images obtained from pre-trained models are used to generate atlases.

3.6.9 Experiment 9 (residual learning using perceptual loss)

The flow chart for experiment 9 is shown in Figure 20. There is a modification made to the objective function of experiment 3. Originally, generator and discriminator losses were calculated based on generator output. To improve the model's performance, we input generator output into VGG and compute generator and discriminator losses using VGG's output. Currently, there is no 3D VGG architecture, so we use 2D VGG [22] instead. The 3D patches were converted to slices to be input into the 2D VGG. With VGG, generator output improved in terms of perceptual quality. Consequently, the model generalized better and achieved better results. This experiment's objective function formula is as follows:

$$Loss_G = -D\left(VGG\left(G(LR)\right)\right)$$

$$Loss_D = D\left(VGG\left(G(LR)\right)\right) - D\left(VGG(HR)\right)$$

doi:10.6342/NTU202303956

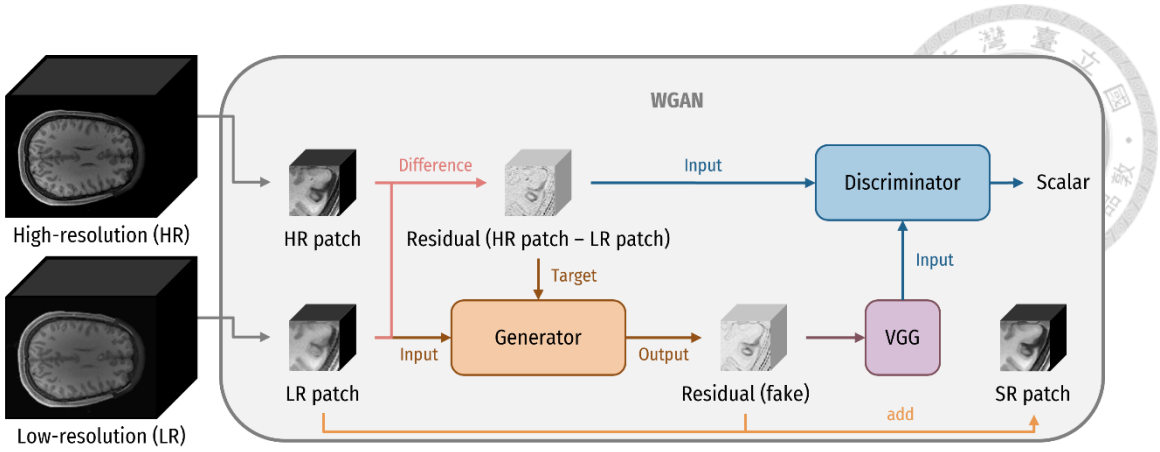


Figure 20. Training process for experiment 9.

3.6.10 Experiment 10 (position-aware residual learning using energy-based GAN and perceptual loss)

This experiment is based on experiment 7, with the difference being that the loss is calculated using VGG in the same manner as experiment 9. It is intended to determine whether combining EBGAN with perceptual loss can improve performance in comparison to utilizing only perceptual loss.

3.6.11 Experiment 11 (position-aware residual learning with different atlas)

The experiment 5 uses the AAL brain atlas to incorporate location information into the model training process. Therefore, we changed the atlas to AAL3 which contains more regions in this experiment. Determine whether the performance increases along with the regions. In AAL3, there are 166 parcellations with a maximum label number of 170,

doi:10.6342/NTU202303956

covering the whole brain. AAL3 is an improved version of AAL2 and provides more accurate labeling of brain regions. It has been widely used in neuroimaging studies for clinical analysis. Table 2 shows the regions of AAL3. Numbers used in AAL2 remain largely the same in AAL3. From number 121 onwards, AAL3 adds new areas.

Table 2. Regions of AAL3. [20]

NO.	Anatomical Description	NO.	Anatomical Description
1, 2	Precentral gyrus	3, 4	Superior frontal gyrus, dorsolateral
5, 6	Middle frontal gyrus	7, 8	Inferior frontal gyrus, opercular part
9, 10	Inferior frontal gyrus, triangular part	11, 12	IFG pars orbitalis
13, 14	Rolandic operculum	15, 16	Supplementary motor area
17, 18	Olfactory cortex	19, 20	Superior frontal gyrus, medial
21, 22	Superior frontal gyrus, medial orbital	23, 24	Gyrus rectus
25, 26	Medial orbital gyrus	27, 28	Anterior orbital gyrus
29, 30	Posterior orbital gyrus	31, 32	Lateral orbital gyrus
33, 34	Insula	35, 36	Anterior cingulate & paracingulate gyri
37, 38	Middle cingulate & paracingulate gyri	39, 40	Posterior cingulate gyrus
41, 42	Hippocampus	43, 44	Parahippocampal gyrus
45, 46	Amygdala	47, 48	Calcarine fissure and surrounding cortex
49, 50	Cuneus	51, 52	Lingual gyrus
53, 54	Superior occipital gyrus	55, 56	Middle occipital gyrus
57, 58	Inferior occipital gyrus	59, 60	Fusiform gyrus
61, 62	Postcentral gyrus	63, 64	Superior parietal gyrus
65, 66	Inferior parietal gyrus, excluding supramarginal and	67, 68	SupraMarginal gyrus

doi:10.6342/NTU202303956

	angular gyri		
69, 70	Angular gyrus	71, 72	Precuneus
73, 74	Paracentral lobule	75, 76	Caudate nucleus
77, 78	Lenticular nucleus, Putamen	79, 80	Lenticular nucleus, Pallidum
81, 82	Thalamus	83, 84	Heschl's gyrus
85, 86	Superior temporal gyrus	87, 88	Temporal pole: superior temporal gyrus
89, 90	Middle temporal gyrus	91, 92	Temporal pole: middle temporal gyrus
93, 94	Inferior temporal gyrus	95, 96	Crus I of cerebellar hemisphere
97, 98	Crus II of cerebellar hemisphere	99, 100	Lobule III of cerebellar hemisphere
101, 102	Lobule IV, V of cerebellar hemisphere	103, 104	Lobule VI of cerebellar hemisphere
105, 106	Lobule VIIB of cerebellar hemisphere	107, 108	Lobule VIII of cerebellar hemisphere
109, 110	Lobule IX of cerebellar hemisphere	111, 112	Lobule X of cerebellar hemisphere
113	Lobule I, II of vermis	114	Lobule III of vermis
115	Lobule IV, V of vermis	116	Lobule VI of vermis
117	Lobule VII of vermis	118	Lobule VIII of vermis
119	Lobule IX of vermis	120	Lobule X of vermis
121, 122	Thalamus, Anteroventral Nucleus	123, 124	Lateral posterior
125, 126	Ventral anterior	127, 128	Ventral lateral
129, 130	Ventral posterolateral	131, 132	Intralaminar
133, 134	Reuniens	135, 136	Mediodorsal medial magnocellular
137, 138	Mediodorsal lateral parvocellular	139, 140	Lateral geniculate
141, 142	Medial Geniculate	143, 144	Pulvinar anterior
145, 146	Pulvinar medial	147, 148	Pulvinar lateral
149, 150	Pulvinar inferior	151, 152	Anterior cingulate cortex, subgenual
153, 154	Anterior cingulate cortex, pregenual	155, 156	Anterior cingulate cortex, supracallosal

doi:10.6342/NTU202303956

157, 158	Nucleus accumbens	159, 160	Ventral tegmental area
161, 162	Substantia nigra, pars compacta	163, 164	Substantia nigra, pars reticulata
165, 166	Red nucleus	167, 168	Locus coeruleus
169	Raphe nucleus, dorsal	170	Raphe nucleus, median

3.7 Evaluation methods

The models we develop are evaluated using three different methods. To determine the difference between the reconstructed image and the ground truth, metrics are used. Lesion segmentation tool is used to determine whether lesion data can be segmented correctly after reconstruction. Furthermore, bootstrap uncertainty is used to assess the stability of models.

3.7.1 Image evaluation metrics

Metrics for evaluating images include the Structural Similarity Index (SSIM), Peak Signal to Noise Ratio (PSNR), and Normalized Root Mean Square Error (NRMSE).

SSIM determines the similarity between two images based on their structural information. It takes into account the perception of the human visual system and puts more emphasis on subtle details, such as the texture, contrast and luminance of the image. SSIM is considered to be a more reliable measure of image similarity compared to other metrics.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

doi:10.6342/NTU202303956

As shown in the formula, μ_x and μ_y are the means of x and y. σ_x and σ_y are the standard deviations of x and y, and σ_{xy} is the covariance. As constants, c_1 and c_2 prevent errors when the denominator is zero.

PSNR measures the quality of a reconstructed signal compared to the original signal. It is calculated by comparing the peak signal-to-noise ratio of the original and reconstructed signals. A higher PSNR generally indicates that the reconstructed signal is of higher quality.

$$PSNR = 10 \times log_{10}\left(\frac{MAX^2}{MSE}\right)$$

$$MSE = \frac{\Sigma_{i=1}^{N}(I_i - P_i)^2}{N}$$

In the formula, MAX represents the maximum intensity of an image. As an example, the MAX for an 8-bit image is 255. MSE stands for Mean Square Error, which measures the difference between two images. MSE is calculated by summing the squared differences between corresponding pixels in the two images. Lower MSE indicates a closer match between the two images. N is the number of pixels. As a result, when PSNR is higher, the image difference is smaller.

NRMSE is used to measure the accuracy of a prediction model compared to actual values. It is calculated as the root mean squared error divided by the standard deviation of the actual values. It is a good indication of how well a model fits the data.

$$NRMSE = \frac{MSE}{I_{max} - I_{min}}$$

doi:10.6342/NTU202303956

The formula represents I_{max} as the maximum intensity of the image, and I_{min} as the minimum intensity.

3.7.2 MS lesion segmentation tool

Segmenting the lesion from the reconstructed image is performed using [23]. The flow chart of the lesion segmentation tool is shown in Figure 21. The tool needs a T1-weighted and FLAIR image to segment. Segmented lesions of ground truth images are generated using ground truth T1-weighted images and ground truth FLAIR images. Furthermore, segmented lesions of super-resolution images are generated from ground truth FLAIR images and super-resolution images. Finally, compare the two segmented results.

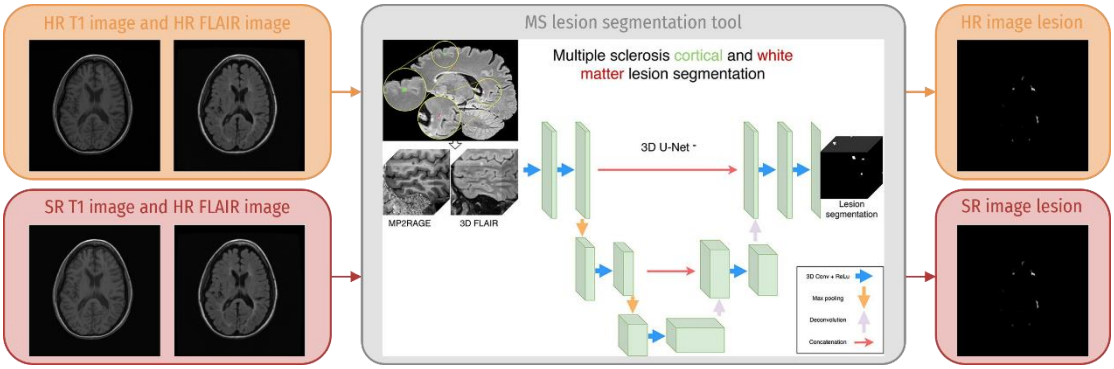
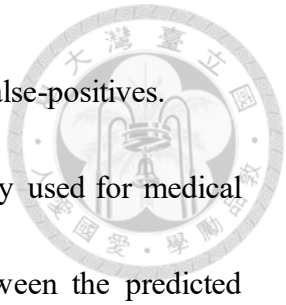


Figure 21. Tool for segmenting lesions.

The accuracy of a prediction is determined by the Dice Coefficient (DSC), the Absolute Volume Difference (AVD), the Positive Predicted Value (PPV), the True Positive Rate (TPR), and the False Positive Rate (FPR). As shown in the formula below, M_R is the mask of a human rater, M_A is the mask generated by an algorithm, and M_R^c is the

doi:10.6342/NTU202303956



complement of M_R that, when intersected with M_A, represents the false-positives.

DSC is a measure of similarity between two samples. It is commonly used for medical image segmentation, where it is used to measure the overlap between the predicted segmentation and the ground truth. The coefficient ranges from 0 to 1, with 1 indicating perfect similarity.

$$DSC(M_R, M_A) = 2\frac{|M_R \cap M_A|}{|M_R| + |M_A|}$$

AVD is a measure of the absolute difference between two samples, meaning the difference between the total volumes of the two samples. This measure is particularly useful in determining the level of similarity between two populations, as it quantifies the amount of variance between the two samples. It can be used to identify any differences in the overall size of the two populations, as well as any differences in the distribution of the populations.

$$AVD(M_R, M_A) = \frac{max(|M_R|, |M_A|) - min(|M_R|, |M_A|)}{|M_R|}$$

PPV is a measure of the accuracy of a diagnostic test. It is the proportion of true positives among all positive test results. PPV is used to determine how reliable a test is at correctly identifying people with a certain condition or disease.

$$PPV(M_R, M_A) = \frac{|M_R \cap M_A|}{|M_R \cap M_A| + |M_R^c \cap M_A|}$$

TPR is a measure of the accuracy of a test in detecting positive cases. It is calculated by dividing the number of true positives (correctly identified positives) by the total number

doi:10.6342/NTU202303956

of actual positives. The higher the TPR, the more accurately the test is able to detect positive cases.

$$TPR(M_R, M_A) = \frac{|M_R \cap M_A|}{|M_R \cap M_A| + |M_R \cap M_A^c|}$$

FPR is the ratio of the number of false positives, meaning incorrect rejections, to the total number of negatives, meaning actual negatives. It is a measure of how reliable a test or algorithm is in correctly identifying true negatives, and thus is an important metric when evaluating a model.

$$FPR(M_R, M_A) = \frac{|M_R^c \cap M_A|}{|M_R^c \cap M_A| + |M_R^c \cap M_A^c|}$$

3.7.3 Bootstrap uncertainty

Bootstrap uncertainty is a statistical technique used to measure model uncertainty. It works by repeatedly sampling data points from a dataset and building models from the sampled data. This allows us to measure the variability of model predictions. Figure 22 illustrates our sampling strategy when evaluating model uncertainty. In this evaluation, the original training set of 990 subjects is divided into 890 subjects for training and 100 subjects for testing. After the training set has been divided, it is folded into 10. Then, take out 9 folds to train the model, and repeat the process 10 times. Following training, test 10 super-resolution models separately. There are 100 super-resolution images generated by each model. In order to combine the images, they are transformed into MNI space and

doi:10.6342/NTU202303956

then mean, and variance are calculated for each pixel.

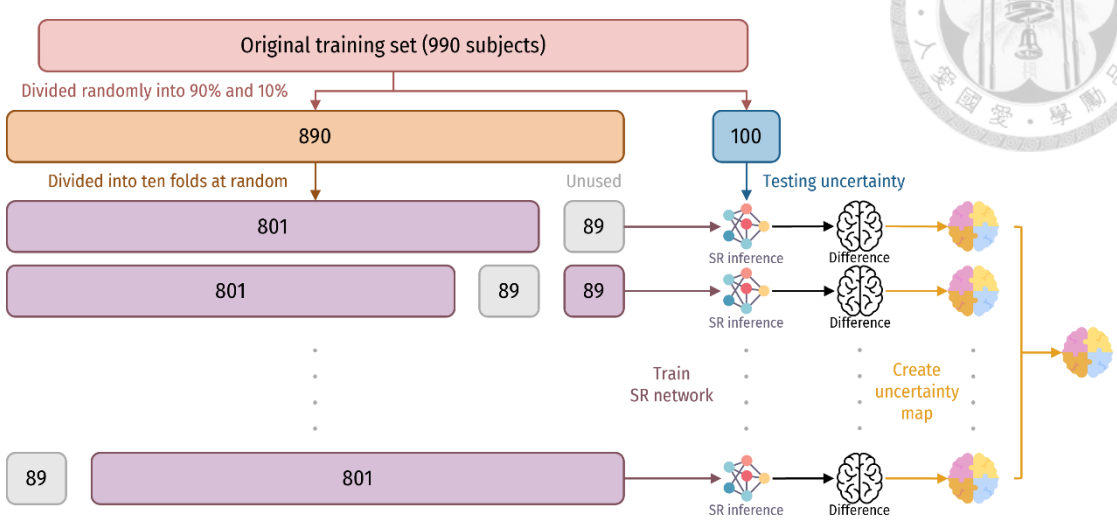


Figure 22. Bootstrap uncertainty sampling strategy.

doi:10.6342/NTU202303956

Chapter 4 Results

Table 3 shows the comparison of all experiments. Upsampling based on spatial domain is denoted by S, and upsampling based on frequency domain is denoted by F. W represents WGAN, and EB represents EBGAN. Yes indicates that a method is being used, while No indicates that it is not being used. Initially, experiment 0 was conducted in accordance with the original paper's methodology. In experiment 1, high-frequency detail is enhanced using edge detection in the loss function. In experiment 2, we use two-step training. The first step is to create an initial super-resolution image that is high-resolution but appears smooth. In the second step, the initial super-resolution images are input into the model to produce residual images, which are then added to the low-resolution images. In experiment 3, residual images are generated directly from low-resolution images in order to improve reconstruction efficiency. In experiment 4, a different upsampling method is applied. In experiment 5, the brain atlas is used to add location information to the model. Based on EBGAN, experiments 6 and 7 modify the discriminator architecture and loss function, while experiment 7 trains with bran atlas. In experiment 8, brain atlas is used as well, whose architecture is derived from experiment 2. In experiment 9, the loss is calculated using VGG. In experiment 10, the brain atlas, EBGAN, and the loss calculated using VGG are all applied. In experiment 11, more regions of the brain atlas are used.

doi:10.6342/NTU202303956

Table 3. The comparison of all experiments.

Experiment	0	1	2	3	4	5	6	7	8	9	10	11
Spatial / Frequency domain up-sampling	S	S	S	S	F	S	S	S	S	S	S	S
LR image histogram matching	No	No	No	No	Yes	No	No	No	No	No	No	No
WGAN / EBGAN	W	W	W	W	W	W	EB	EB	W	W	EB	W
Residual learning	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Two-stage training	No	No	Yes	No	No	No	No	No	Yes	No	No	No
Position-aware	No	No	No	No	No	Yes	No	Yes	Yes	No	Yes	Yes
Edge loss	No	Yes	No	No	No	No	No	No	No	No	No	No
Perceptual loss	No	No	No	No	No	No	No	No	No	Yes	Yes	No

Table 4 presents the results of the evaluation using metrics. SSIM and PSNR are both better when they are larger, while NRMSE is better when it is smaller. Experiment 11 has the best results for each metric. Experiment 5 is the second best. It is only the brain atlas that differs between experiment 5 and experiment 11, the atlas used in experiment 11 contains more regions. Therefore, we can conclude that a more detailed brain atlas can lead to better performance in the evaluation metrics.

doi:10.6342/NTU202303956

Table 4. Evaluation of all experiments using image evaluation metrics in Section 3.7.1.

Experiment	SSIM	SSIM STD	PSNR	PSNR STD	NRMSE	NRMSE STD
0	0.9345	0.0054	30.3714	0.8575	0.2297	0.0098
1	0.9459	0.0041	35.8611	0.7326	0.1221	**0.0045**
2	0.9744	0.0032	35.6009	0.9287	0.1259	0.0077
3	0.9683	0.0040	35.7057	0.7985	0.1243	0.0050
4	0.4181	0.0191	26.1539	**0.1892**	0.3747	0.0370
5	0.9765	0.0030	36.1970	1.2261	0.1179	0.0113
6	0.9725	**0.0026**	35.2638	0.9422	0.1310	0.0097
7	0.9743	0.0029	35.2571	0.8876	0.1310	0.0085
8	0.9757	0.0033	35.8203	1.3362	0.1233	0.0134
9	0.9704	0.0027	34.8682	0.9725	0.1372	0.0111
10	0.9689	0.0035	34.7430	0.9449	0.1389	0.0068
11	**0.9774**	0.0031	**36.3099**	1.1136	**0.1162**	0.0090

Figure 23 illustrates a zoomed-in coronal view of all experiments, cubic interpolation, and ground truth images. Except for experiment 0 and experiment 4, there is good agreement between each super-resolution image and the ground truth image.

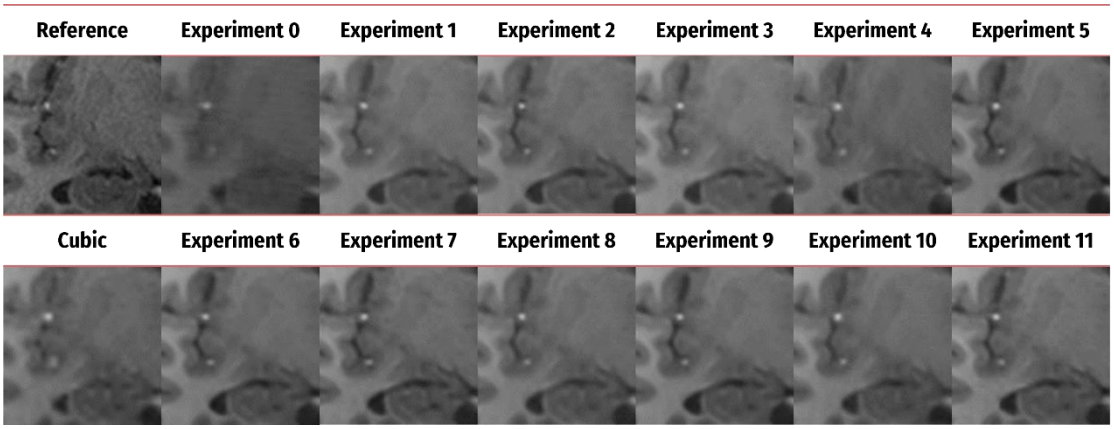


Figure 23. Ground truth image, super-resolution images of all experiments in coronal view, and cubic interpolation image.

To distinguish between the disparities between experiments more clearly, the difference from ground truth is computed and is shown in Figure 24. The color bar is located at the

doi:10.6342/NTU202303956

far right of the figure. Red represents the positive value obtained by subtracting the cubic interpolation or super-resolution image from the ground truth image. In contrast, blue represents a negative value. In the same way as the results of the metric evaluation, experiment 5 and 11 are more reliable.

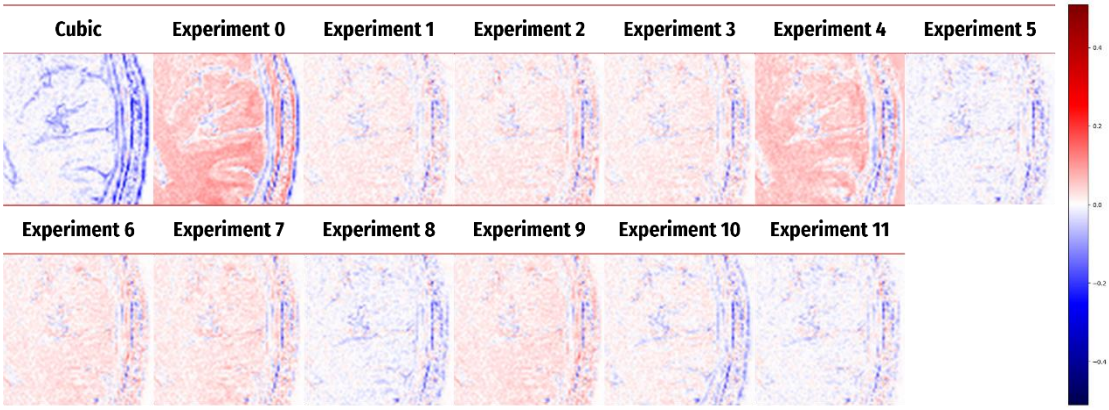


Figure 24. The difference in axial view between ground truth image and all of experiments' images or cubic interpolation image.

To determine whether the model is suitable for clinical application, the data associated with the lesion are analyzed. After downsampling the lesion data, our model is used to super-resolve the data. After that, segment the super-resolution image and the original high-resolution image for comparative purposes. In this situation, the segmented results are almost identical, which indicates that super-resolution does not remove the lesion from the image. Table 5 displays segmentation results for all experiments. The DSC and AVD performed optimally in experiment 5 and the PPV and FPR performed the best in experiment 9. To identify the model with the most accurate super-resolution capability,

doi:10.6342/NTU202303956

the DSC, AVD, PPV, TPR, and FPR are added together. As a result of the addition, the score represents the degree of accuracy of each experiment. Consequently, experiment 5 has received the highest score.

Table 5. MS lesion segmentation results in all experiments.

Experiment	DSC	AVD	PPV	TPR	FPR	Score
0	0.8238	0.2741	0.7559	0.9252	0.0001	2.1308
1	0.9484	0.0608	0.9254	0.9745	2.85e-05	2.7590
2	0.9533	0.0324	0.9494	0.9581	1.88e-05	2.8096
3	0.9556	0.0348	0.9488	0.9637	1.78e-05	2.8155
4	0.8152	0.2774	0.7471	0.9173	0.0001	2.1022
5	**0.9619**	**0.0200**	0.9587	0.9656	1.42e-05	**2.8520**
6	0.9580	0.0422	0.9685	0.9489	9.60e-06	2.8236
7	0.9483	0.0496	0.9620	0.9365	1.15e-05	2.7857
8	0.9508	0.0558	0.9290	**0.9750**	2.52 e-05	2.7738
9	0.9580	0.0434	**0.9735**	0.9443	**8.53e-06**	2.8238
10	0.9285	0.0908	0.8943	0.9683	3.83e-05	2.662
11	0.9585	0.0360	0.9607	0.9576	1.43e-05	2.8265

Figure 25 shows MS lesion segmented images of ground truth and super-resolution images for all experiments. As can be seen in the upper part of the image, the segmented results of experiments 0 and 4 appear to be inaccurate, while experiment 10 also appears to be a little inaccurate. Both experiments 7 and 9 are able to segment the small lesion in the lower left portion of the image. In other experiments, the results are similar.

doi:10.6342/NTU202303956

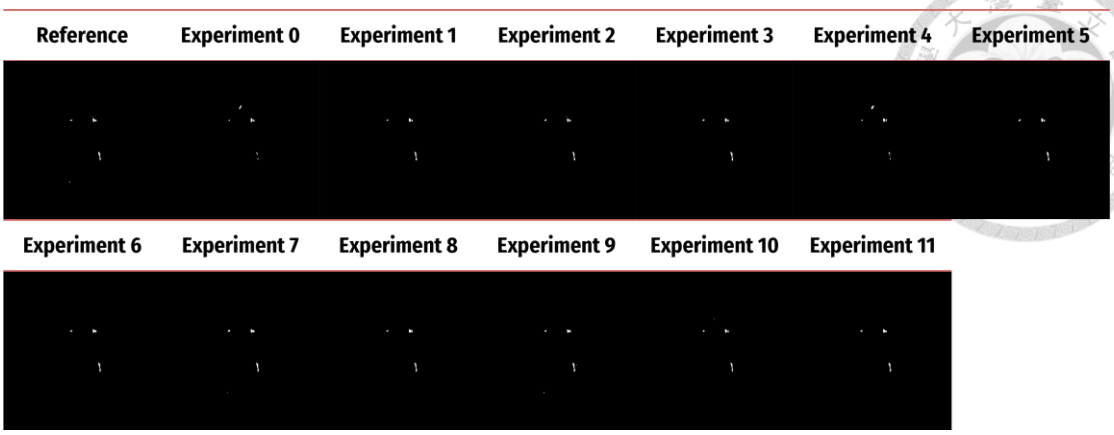


Figure 25. Results of MS lesion segmentation based on ground truth and super-resolution images for all experiments.

On the basis of both metrics and lesion segmentation results, experiment 5 is selected in order to test bootstrap uncertainty. The results of ten fold training are presented in Table 6. Except for fold 5, the other folds' results are fairly similar. As a result, our model appears stable to a certain extent.

Table 6. Results of the evaluation of 10 models trained in bootstrap uncertainty.

Fold	SSIM	SSIM STD	PSNR	PSNR STD	NRMSE	NRMSE STD
1	0.9751	0.0038	35.6138	1.3341	0.1288	0.0174
2	0.9768	0.0038	35.8952	1.3700	0.1247	0.0174
3	0.9698	0.0039	35.2960	1.2959	0.1335	0.0176
4	0.9759	0.0038	35.7165	1.3438	0.1273	0.0174
5	0.7510	0.0161	32.5352	1.1338	0.1836	0.0246
6	0.9766	0.0042	35.8914	1.4999	0.1252	0.0201
7	0.9749	0.0038	35.7797	1.3566	0.1264	0.0175
8	0.9763	0.0039	35.8351	1.3648	0.1256	0.0175
9	0.9720	0.0038	35.2748	1.3165	0.1338	0.0177
10	0.9754	0.0038	35.6336	1.3493	0.1285	0.0178

In order to generate an uncertainty map, accuracy is first calculated. All differences between ground truth and inference are mapped to MNI space and their mean is computed.

doi:10.6342/NTU202303956

The difference mean is then divided by the ground truth. This result can be used to demonstrate the accuracy of the inference. Figure 26 illustrates the accuracy map of all subjects' accuracy means. Only the surrounding area has a low level of accuracy, whereas the majority of the area does not.

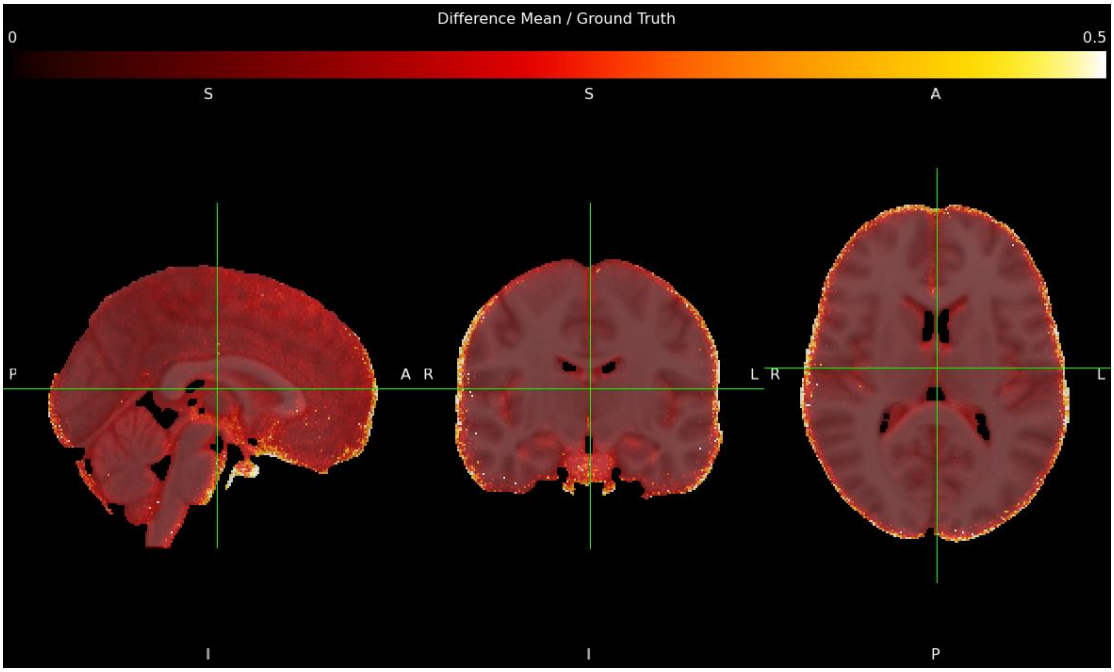


Figure 26. Accurate map of each area.

The second step in generating uncertainty maps is to determine whether the inferences are consistent. As a result, variance of inference is calculated. Following this, the variance is divided by the ground truth, in the same manner as the accuracy. The consistency map in Figure 27 shows the average consistency of all subjects. This map is very similar to the accuracy map.

doi:10.6342/NTU202303956

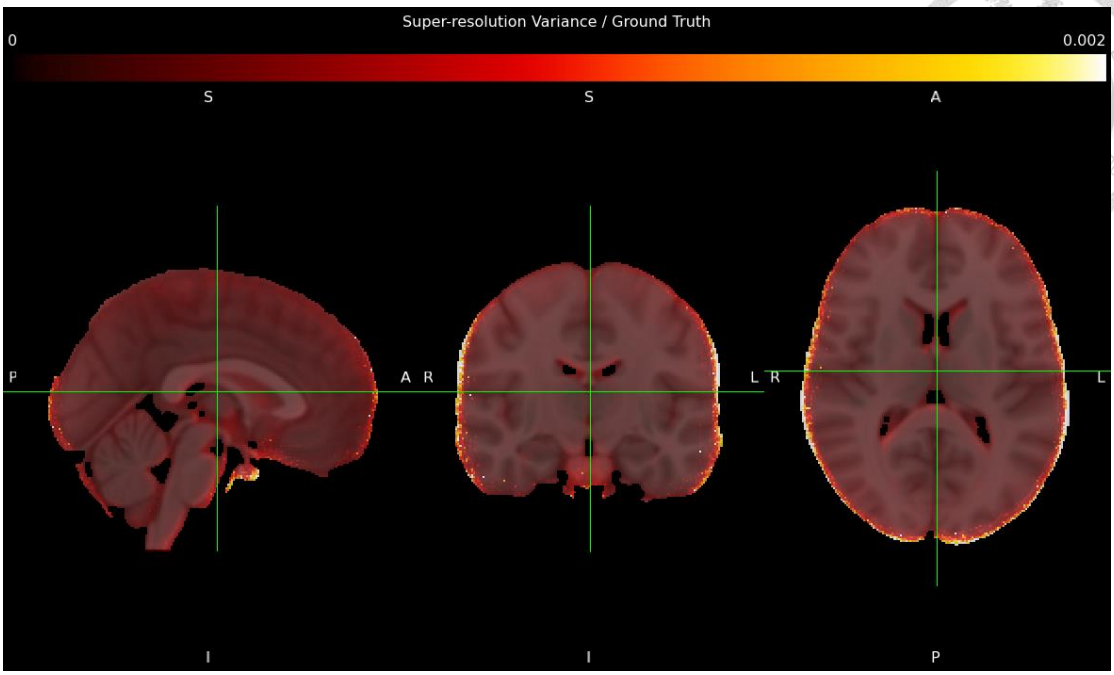


Figure 27. Consistency map of each area.

Finally, multiply the accuracy map by the consistency map in order to obtain the uncertainty map. The uncertainty map for experiment 5 is shown in Figure 28. There is a low level of uncertainty in the main areas of the brain.

Figure 28. Uncertainty map.

doi:10.6342/NTU202303956

In addition, we compare our method with another recently developed method. Least Squares GAN (LSGAN) [24] represents an improved version of GANs that addresses the problem of mode collapse. It uses a least-squares loss function instead of the original cross-entropy loss function, which helps to stabilize the training process and reduces the risk of mode collapse. Table 7 presents the training parameters for experiment 5 and LSGAN. There is no difference in any of the parameters. The evaluation metrics are shown in Table 8. We have a clear advantage in terms of performance.

Table 7. Experiment 5 and LSGAN training parameters.

Parameters	**Experiment 5**	**LSGAN**
Training set	801	801
Validation set	111	111
Testing set	100	100
Patch size	64x64x64	64x64x64
Training step	327084	327084
Learning rate	0.0001	0.0001
Batch size	2	2

Table 8. Comparatively to other methods currently in use.

Fold 1	**Experiment 5**	**LSGAN**
SSIM	**0.9751**	0.9155
SSIM STD	**0.0038**	0.0118
PSNR	**35.6138**	28.0192
PSNR STD	1.3341	**1.1552**
NRMSE	**0.1288**	0.3076
NRMSE STD	**0.0174**	0.0293

A second evaluation method is also used in order to compare LSGAN with our results. Figure 29 shows the subtractive value between experiment 5 and the LSGAN difference

doi:10.6342/NTU202303956

mean. The red color indicates that LSGAN's inference has a much greater difference from ground truth than our method. By contrast, blue represents our method has a much greater difference. According to the results, our method has a much lower difference in all areas. As well, accuracy maps are also used to evaluate the results. Figure 30 shows a comparison of accuracy maps. There is no doubt that our method performs better.

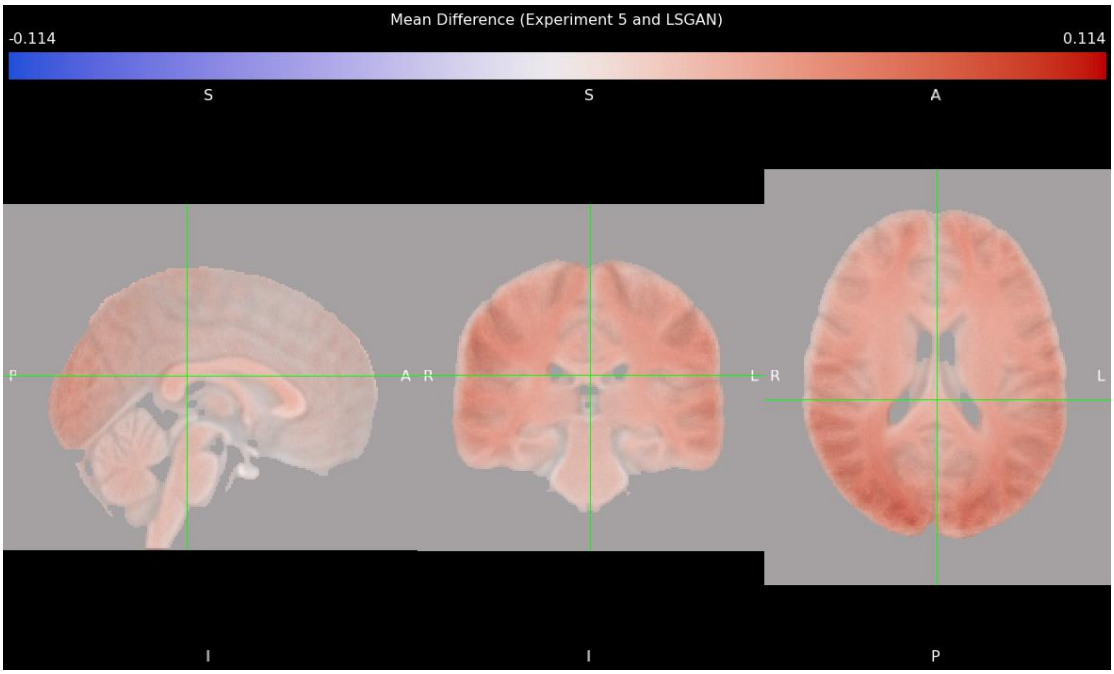


Figure 29. Subtractive value between experiment 5 and LSGAN difference mean.

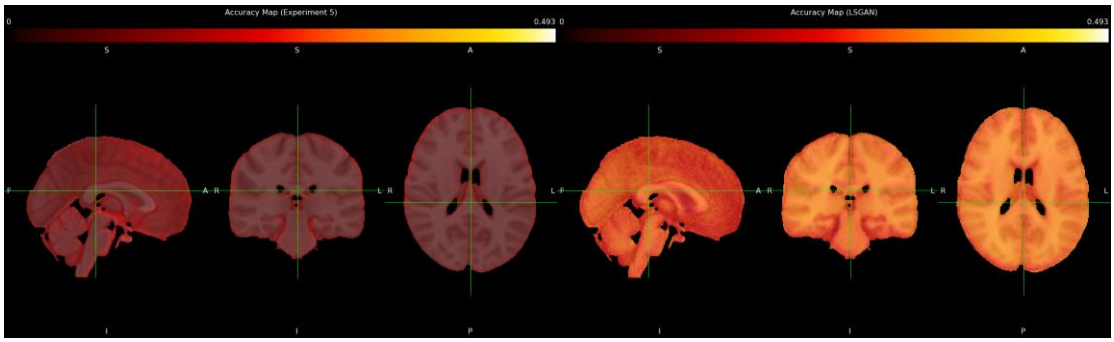


Figure 30. Comparing the accurate maps in fold 1 between experiment 5 and LSGAN.

Finally, we compare training and inference times for all experiments. Training consists of

doi:10.6342/NTU202303956

three parts, namely generator pre-training, discriminator pre-training, and WGAN training. The training steps that are used to calculate training time are 100, 100, and 70, respectively. Based on the actual number of training steps, the calculated time should be increased. The GPU used is GeForce RTX 3090. Table 9 shows the training and inference times. Due to edge loss, experiment 1 requires the most training time. In each step of training, edge detection is used to calculate loss. Due to its two-step design, experiment 2 and 8 take the longest to infer. In comparison with the original reference method (experiment 0), residual learning reduced training and inference time (experiment 3 and 4). Despite the fact that atlas can improve the results, training and inference time increase (experiment 5, 7, 8, 10 and 11). Based on the model architecture, EBGAN and perceptual loss using VGG take the shortest time to train (experiment 6 and 9). However, the model can be used directly for inference after the training process is complete, therefore the inference time is of greater significance. With the exception of experiment 2 and 8, the inference time of the other experiments is short.

doi:10.6342/NTU202303956

Table 9. The training and inference time for each experiment.

Experiment	Total training time (seconds)	Total training time (hours)	Inference time (seconds per subject)
0	323792	89.94	135.78
1	645996	179.44	134.07
2	516497	143.47	274.39
3	306911	85.25	125.24
4	301770	83.82	**124.17**
5	372077	103.35	140.76
6	**283646**	**78.79**	137.33
7	357753	99.37	133.27
8	551144	153.09	276.17
9	289842	80.51	139.19
10	331028	91.95	134.99
11	357609	99.33	132.75

doi:10.6342/NTU202303956

Chapter 5 Discussion

5.1 Preprocessing of patching and merging data

In our experiment, we found that the merged image appears noisy at the edges of the patch when the overlap size is 6. By using a larger overlap size 16, the noise can be removed. We suggest that a larger overlap size should be used when merging images to reduce noise. This will help to preserve the integrity of the image and prevent degradation of the image due to noise.

5.2 Edge-based loss

In experiment 1, edge-based loss is applied to enhance the edge detail of the images. The experiment results showed that this method achieved better performance in terms of edge detail compared to the baseline method. Moreover, the edge-based loss improved the visual quality of the generated images. However, the edge-based loss method also has some drawbacks. The biggest drawback is that each training step involves computing the edge of the image. Compared to the baseline method, it may result in additional computing costs. Additionally, some areas of the images that were trained with edge-based loss still appear smooth. Consequently, edge-based loss only enhances edge resolution to a limited extent, and it also requires additional computing resources.

doi:10.6342/NTU202303956

5.3 Residual learning

As stated in [5], the experimental results are consistent with the arguments. Using residual learning, neural networks can be trained more efficiently. It facilitates the learning process by reducing the network complexity. Further, the information contained in the model input is not lost during the convolution process. This allows the model to accurately capture both low-level and high-level features from the input. It significantly reduces the risk of overfitting and allows for faster training and better generalization.

5.4 Different methods of upsampling

Prior to reconstruction, low-resolution images should be upsampled. To test different methods, we upsample in both the frequency and spatial domains. In comparison, upsampled images in the frequency domain are capable of obtaining detailed images, but they are subject to aliasing. Even when compressed sensing is applied, images still exhibit aliasing. As a result of these aliased parts, the reconstruction process may be affected. Therefore, our results indicate that upsampling in spatial domain is a more effective method.

doi:10.6342/NTU202303956

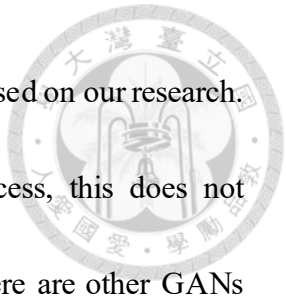
5.5 Position-aware

In the course of training, images are divided into patches. These patches are then used to train the model. However, the model does not contain information regarding patch locations. As a result, brain atlases are used to add location information to the model. This helps the model better understand the relationship between different parts of the image. Learning different features according to the region of the brain in which they are located. This helps to improve the accuracy of the model by allowing it to better identify meaningful patterns in the data. Additionally, it's beneficial for improved generalization, as the model is able to better identify patterns across different datasets. The results of [25] indicate that cardiac MRI image training with phase information can improve the image quality. The argument is similar to ours, which is that training with additional information from an image can improve performance.

5.6 Different types of GAN

In order to further improve the performance of the model, we try using different architectures of GAN. After reviewing previous papers, EBGAN was selected due to its ability to generate higher quality images. It also has the ability to constrain the generator to produce images from a given set of classes. Finally, it has a more stable learning process compared to other GANs. However, despite the fact that EBGAN is capable of generating

doi:10.6342/NTU202303956



more detailed images, some of the generated parts appear to be fake based on our research. In addition, while EBGAN may have a more stable learning process, this does not necessarily mean that it is the best option for image generation. There are other GANs that may be more effective in generating realistic images.

5.7 Perceptual loss

After computing loss using features generated by the VGG, the result has a more pleasing visual appearance. As discussed in [26], this is the same result. This is because the VGG captures more fine details, such as edges and color, which in turn improves the overall quality of the generated image. Thus, the VGG is a useful tool for image generation tasks.

5.8 Clinical feasibility testing using lesion segmentation

Our study also uses MS lesion data to perform super-resolution. With the aid of a lesion segmentation tool, we were able to confirm that the lesion does not disappear after super-resolution. This confirms that our super-resolution technique is effective in preserving lesion information. This information is vital for clinicians in diagnosing and treating diseases. Therefore, our method has great potential in the medical field.

doi:10.6342/NTU202303956

5.9 Uncertainty assessment

With the accuracy map, we can determine which areas have a high difference from ground truth. In addition, we may be able to determine the stability of super-resolution images in each area by using a consistency map. Lastly, uncertainty maps combine both to determine the degree of uncertainty. Using both accuracy and consistency maps, we can identify regions where the super-resolution image is reliable, and regions which are less reliable. This allows us to make decisions about how to use the super-resolution image effectively.

5.10 Training and inference time

As a final step, we calculate the training and inference time of each experiment. The purpose of this is to confirm whether the time increases as a result of adding other techniques. For most experiments, it does not appear to have much effect on inference time. However, for certain experiments, the training time can increase significantly due to the added complexity. In such cases, it is important to consider the trade-offs between accuracy and efficiency. It is also important to compare the results to other techniques to ensure that the best possible solution is chosen.

doi:10.6342/NTU202303956

Chapter 6 Conclusion

Our method performs better than the method described in the reference paper. SSIM is 0.97, PSNR is 36.30, and NRMSE is only 0.1162. The results of segmentation using super-resolution on clinical images with MS lesions are almost identical to those obtained using high-resolution images. In experiment 5, DSC reaches 0.96, AVD is only 0.02, and TPR reaches 0.96. Lastly, bootstrap is used to test uncertainty. In our method, most regions have a low variance and the difference from the ground truth is small, which indicates that these regions are credible. As a result of these evaluation results, we believe that super-resolution can indeed be used in MRI. It is possible to obtain low-resolution images with less scanning time and then reconstruct them with super-resolution. This method can reduce the likelihood of moving noise and increase the number of cases that can be scanned within a given period of time.

This study aims to reconstruct brain T1-weighted images. It is possible that this technique could be applied to other brain imaging sequences, which would enhance the value of the research. Clinically, it will also be more useful.

doi:10.6342/NTU202303956

References

[1] Dong, C., C.C. Loy, K. He, and X. Tang, *Image Super-Resolution Using Deep Convolutional Networks.* IEEE Trans Pattern Anal Mach Intell, 2016. **38**(2): p. 295-307.

[2] Goodfellow, I.J., et al., *Generative adversarial nets*, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. 2014, MIT Press: Montreal, Canada. p. 2672–2680.

[3] Ledig, C., et al., *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.* arXiv e-prints, 2016: p. arXiv:1609.04802.

[4] He, K., X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition.* arXiv e-prints, 2015: p. arXiv:1512.03385.

[5] Pham, C.H., et al., *Multiscale brain MRI super-resolution using deep 3D convolutional networks.* Comput Med Imaging Graph, 2019. **77**: p. 101647.

[6] Lyu, Q., et al., *Multi-Contrast Super-Resolution MRI Through a Progressive Network.* IEEE Trans Med Imaging, 2020. **39**(9): p. 2738-2749.

[7] Gulrajani, I., et al., *Improved Training of Wasserstein GANs.* arXiv e-prints, 2017: p. arXiv:1704.00028.

[8] Shan, H., et al., *3-D Convolutional Encoder-Decoder Network for Low-Dose CT via Transfer Learning From a 2-D Trained Network.* IEEE Trans Med Imaging, 2018. **37**(6): p. 1522-1534.

[9] Gatys, L.A., A.S. Ecker, and M. Bethge. *Image Style Transfer Using Convolutional Neural Networks*. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[10] Sajjadi, M.S.M., B. Schölkopf, and M. Hirsch, *EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis.* arXiv e-prints, 2016: p. arXiv:1612.07919.

[11] Li, Z., et al., *DeepVolume: Brain Structure and Spatial Connection-Aware Network for Brain MRI Super-Resolution.* IEEE Trans Cybern, 2021. **51**(7): p. 3441-3454.

[12] Çiçek, Ö., et al., *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.* arXiv e-prints, 2016: p. arXiv:1606.06650.

[13] Tao, X., et al., *Detail-revealing Deep Video Super-resolution.* arXiv e-prints, 2017: p. arXiv:1704.02738.

[14] Zhang, K., et al., *SOUP-GAN: Super-Resolution MRI Using Generative Adversarial Networks.* Tomography, 2022. **8**(2): p. 905-919.

[15] Chen, Y., et al., *Efficient and Accurate MRI Super-Resolution using a Generative*

doi:10.6342/NTU202303956

Adversarial Network and 3D Multi-Level Densely Connected Network. arXiv e-prints, 2018: p. arXiv:1803.01417.

[16] Chen, Y., et al., *Brain MRI Super Resolution Using 3D Deep Densely Connected Neural Networks.* arXiv e-prints, 2018: p. arXiv:1801.02728.

[17] Blasche, M. and C. Forman. *Compressed Sensing – the Flowchart.* 2016; Available from: https://www.mriquestions.com/uploads/3/4/5/7/34572113/siemens_mri_magnetom-world_compressed-sensing_compressed-sensing-flowchart_blasche-03520147.pdf.

[18] Canny, J., *A computational approach to edge detection.* IEEE Trans Pattern Anal Mach Intell, 1986. **8**(6): p. 679-98.

[19] Krishna Pandey, R., N. Saha, S. Karmakar, and A.G. Ramakrishnan, *MSCE: An edge preserving robust loss function for improving super-resolution algorithms.* arXiv e-prints, 2018: p. arXiv:1809.00961.

[20] Rolls, E.T., et al., *Automated anatomical labelling atlas 3.* Neuroimage, 2020. **206**: p. 116189.

[21] Zhao, J., M. Mathieu, and Y. LeCun, *Energy-based Generative Adversarial Network.* arXiv e-prints, 2016: p. arXiv:1609.03126.

[22] Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition.* arXiv e-prints, 2014: p. arXiv:1409.1556.

[23] La Rosa, F., et al., *Multiple sclerosis cortical and WM lesion segmentation at 3T MRI: a deep learning method based on FLAIR and MP2RAGE.* NeuroImage: Clinical, 2020. **27**: p. 102335.

[24] Sanchez, I. and V. Vilaplana, *Brain MRI super-resolution using 3D generative adversarial networks.* arXiv e-prints, 2018: p. arXiv:1812.11440.

[25] Lin, J.-Y., Y.-C. Chang, and W. Hsu, *Efficient and Phase-Aware Video Super-Resolution for Cardiac MRI*. 2020. p. 66-76.

[26] Nasser, S.A., et al., *Perceptual cGAN for MRI Super-resolution.* 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2022: p. 3035-3038.

doi:10.6342/NTU202303956