Ingeniaritza Konputazionala eta
Sistema Adimentsuak Unibertsitate Masterra

Máster Universitario en Ingeniería Computacional
y Sistemas Inteligentes

Konputazio Zientziak eta Adimen Artifiziala Saila

Departamento de Ciencias de la Computación e Inteligencia Artificial

Master Tesia
Tesis de Máster

Evaluation and development of deep neural networks for super-resolution of microscopy and astrophysics images

Pablo Alonso Pérez

Zuzendaritza
Dirección

Ignacio Arganda-Carreras
Euskal Herriko Unibertsitatea (EHU)
Universidad del País Vasco (UPV)

Marcos Pellejero-Ibáñez
Donostia International Physics Center (DIPC)

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Konputazio Ingeniaritza eta Sistema Adimentsuak Unibertsitate Masterra
Máster Universitario en Ingeniería Computacional y Sistemas Inteligentes

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Trabajo Fin de Máster

Máster Universitario en Ingeniería Computacional y Sistemas Inteligentes

Evaluation and development of deep neural networks for super-resolution of microscopy and astrophysics images

Pablo Alonso Pérez

Advisors
Ignacio Arganda-Carreras
Marcos Pellejero-Ibáñez

September 2021

Acknowledgements

I would like to express my gratitude to Ignacio Arganda-Carreras for his kindness and support throughout the whole project. It has been a pleasure to have him as an advisor.

To Marcos Pellejero-Ibañez, Jens Stücker and Raul Angulo from the *Donostia International Physics Centre,* who were there from the beginning of the project, ready to help when it was needed and guiding me through the process. They were a great team to work with, and this project wouldn't have been possible without them.

And finally, to my family and friends who have supported me; for their care, patience and encouragement. I couldn't have done this without their unconditional support.

Abstract

Due to physical constrains of an Electron Microscope, capturing high-resolution scans of a subject takes a very long time. On the other hand, running a Gravitational N-body simulation of hundreds of millions of particles, required for state-of-the-art research, takes millions of CPU hours. Thus, in this work we propose a new Image Super-Resolution framework based on Generative Adversarial Networks to super-resolve both images scanned by a microscope and snapshots of gravitational N-body simulations. We incorporate techniques from residual neural networks to increase the learning capabilities, and introduce the Wasserstein GAN training method to improve stability. Comparisons have shown that our model performs equally or better than state-of-the art methods in both of these use cases, and provides balanced results that are realistic but don't have much distortion.

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

1.1 Image Super-Resolution

Image Super-Resolution (SR) is the process of recovering high-resolution (HR) images from low-resolution (LR) images (see Figure 1.1). It is an important class of image processing techniques in computer vision which has a wide range of real-world applications such as medical imaging, surveillance, security, astronomical imaging, among others [1].

Image Super-Resolution is a notoriously challenging ill-posed problem because there are multiple possible HR reconstructions of an LR image (see Figure 1.2), and the HR space that we intend to map to the LR input is usually intractable [2].

Image Super-Resolution techniques can be applied to Singe Image Super-Resolution (SISR), which aims to recover a HR image using a single LR image; Multi Image Super-Resolution (MISR), which combines the information of multiple images of the same scene to produce a HR image; or Video Super-Resolution, which can use information from previous and following frames to reconstruct a HR video. In this thesis we will focus on SISR.

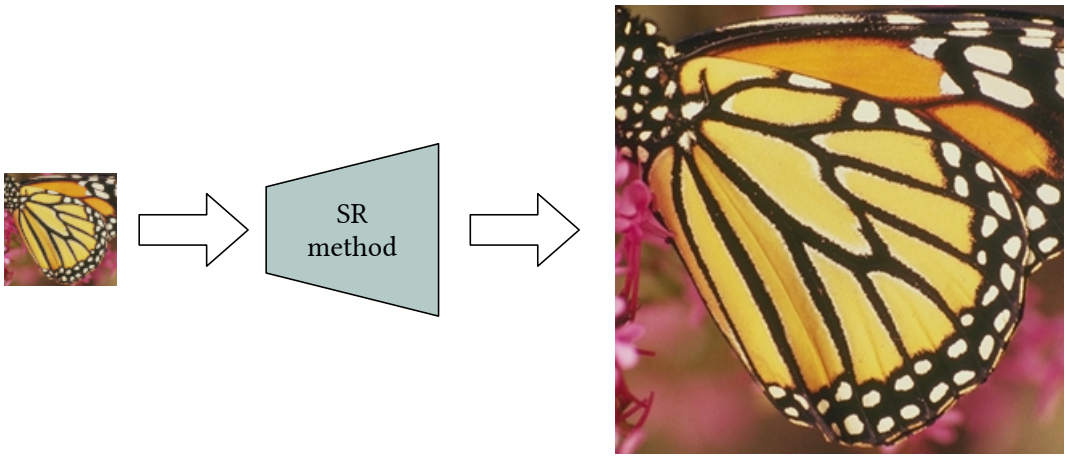


Figure 1.1: Basic structure of Image Super-Resolution methods, in which a LR image (left) is processed in a SR method to create a HR reconstruction of the same image (right).

Figure 1.2: Many high-resolution images can be downsampled to a single low-resolution image. Super-resolution is thus an ill-posed problem. Source: [3]

1.1.1 Problem definition

Image SR aims to recover HR images from LR images. Generally, the LR image z is modelled like this:

$$z = \mathcal{D}(x; \delta), \tag{1.1}$$

where $\mathcal{D}$ is a degradation function, x is the HR image and δ are the parameters of the degradation function.

In real applications, the degradation process ($\mathcal{D}$ and δ) are unknown. However, researchers try to model the degradation mapping in order to easily obtain larger datasets. Some directly model it as a single downscaling operation:

$$\mathcal{D}(x; \delta) = x \downarrow_s, \{s\} \subset \delta, \tag{1.2}$$

where $\downarrow_s$ is a downscaling operator with the scaling factor s.

However, in most real-world applications the LR images are not simple downscaled versions of the HR counterparts, so the degradation operation can be defined as a combination of multiple operations:

$$\mathcal{D}(x; \delta) = (x \otimes \kappa) \downarrow_s + n_\varsigma, \{\kappa, s, \varsigma\} \subset \delta, \tag{1.3}$$

where $(x \otimes \kappa)$ indicates the convolution between a blur kernel κ and the HR image, and n_ς is additive Gaussian noise with variance ς.

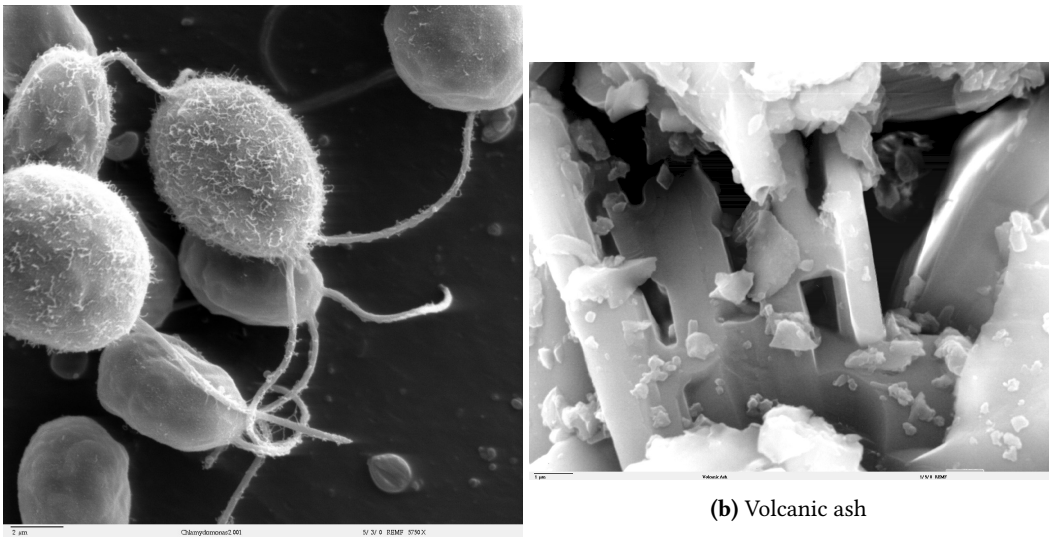
Finally, in the SR process we want to recover a HR approximation $\tilde{x}$ of the ground truth x from the LR image z following:

$$\tilde{x} = \mathcal{F}(z; \theta), \tag{1.4}$$

where $\mathcal{F}$ is the SR model and θ are the parameters of $\mathcal{F}$.

1.2 Electron microscopy

Electron microscopy (EM) is a technique for obtaining high resolution images of biological and non-biological specimens (see Figure 1.3). It is used in biomedical research to investigate the detailed structure of tissues, cells, organelles and macromolecular complexes. The high resolution of EM images results from the use of electrons (which have very short wavelengths) as the source of illuminating radiation [4].



(a) Chlamydomonas

(b) Volcanic ash

Figure 1.3: Scanning Electron Microscope images of Chlamydomonas algae **(a)** and volcanic ash **(b)**. Source: Dartmouth College: Electron Microscopy Facility.

There are two main types of electron microscope – the transmission EM (TEM) and the scanning EM (SEM). The transmission electron microscope is used to view thin specimens (tissue sections, molecules, etc) through which electrons can pass generating a projection image. However, there is an increasing need for large area imaging or even volume imaging of biological tissues at nanoscopic resolution comprising billions of pixels [5].

To image surfaces, scanning electron microscopes (SEMs) need to be used, which depend on the emission of secondary electrons from the surface of a specimen [4]. These microscopes conventionally acquire an image one pixel at a time, so acquiring large amounts of data is very time-consuming. As an example, mapping a 1 mm cube of tissue with an isotropic voxel size of 4 nm will result in almost 16 petabytes of data. Data acquisition at 20 MHz would require a total acquisition time of almost 25 years, even before taking into account overhead times such as those due to stage movements [5].

An obvious way of increasing a SEM's throughput would be to increase the data acquisition rate, therefore taking less time to scan each pixel. However, the signal-to-noise ratio in an SEM image depends on beam current, pixel dwell time, sample contrast and detection efficiency [5]. If we want to lower the time of acquisition by lowering the pixel dwell time, we would need to increase the electron beam current in order to maintain signal-to-noise ratio, thus keeping the specimen we want to scan visible. Increasing the beam current will lead to increasing Coulomb interactions between the electrons, thereby blurring the electron beam and reducing the resolution.

As it is not possible to increase the throughput of the microscope, and the time that can be spent using it for each project is limited, there is a need for SR methods in this field. Super-resolution techniques are very important in their workflow, as they can shorten the time spent using the microscope by lowering the scanning resolution and then upscaling the results using SR methods.

1.3 Gravitational N-body simulation

Gravitational N-body simulations are a widely used theoretical tool in astrophysics and cosmology – the study of the origins of the universe, its large-scale structures and dynamics, and the ultimate fate of the universe [6].

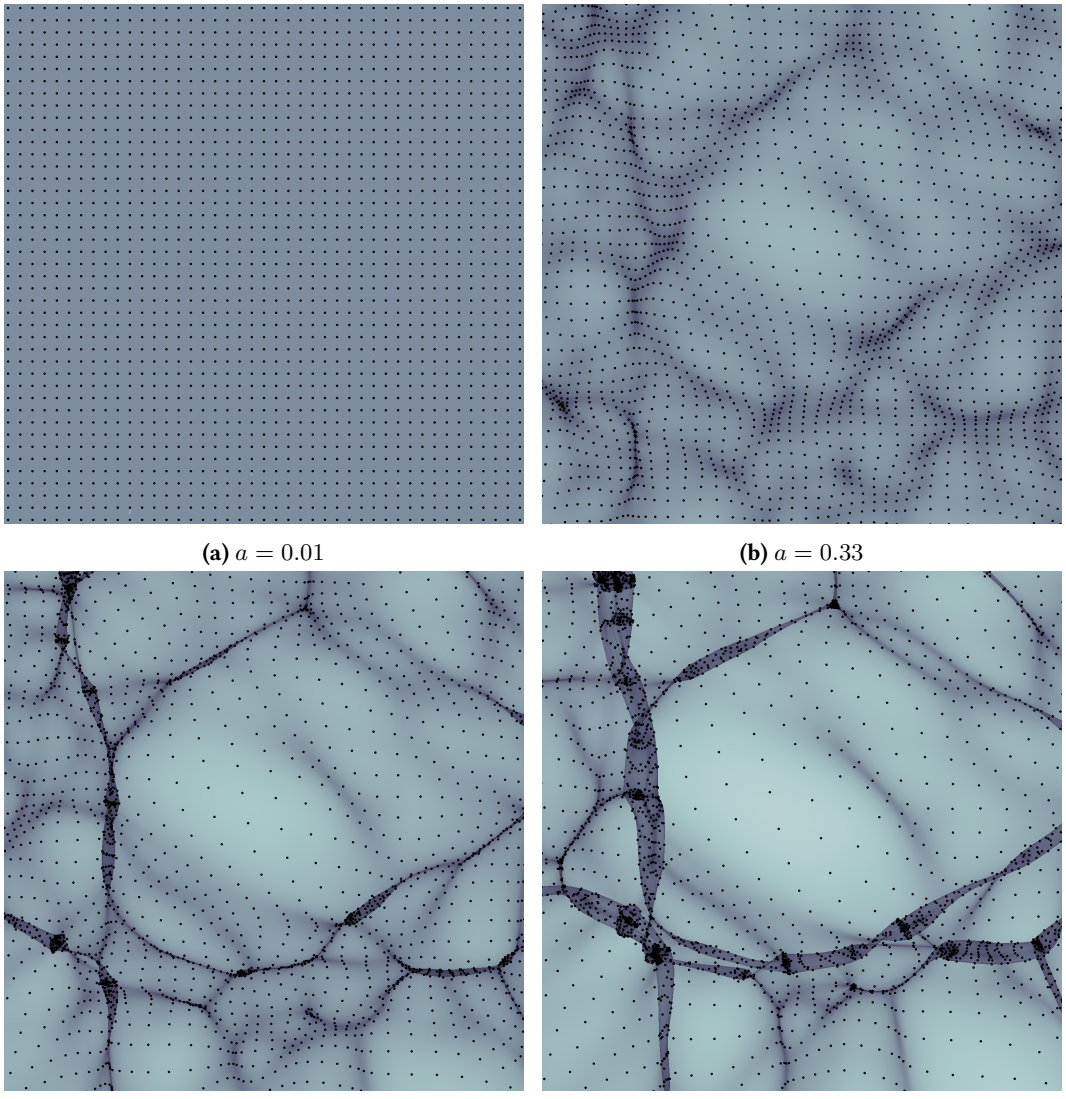
Cosmologists believe that most of the mass in the universe may be in the form of some unknown and invisible particles collectively called "dark matter". Normal nuclear matter forms the luminous stars produced after matter collapsed into galaxies. The dark matter is believed to have no significant interactions except gravity, and it is thought to dominate the mass everywhere except in the stellar cores of galaxies. To achieve a basic understanding of galaxy formation and clustering it may be sufficient only to follow the gravitational interactions of dark matter.

The evolution of perturbations in a nonrelativistic collisionless gas, based on the evolution of the phase space – space of positions, $\vec{x}$, and momenta, $\vec{p} = am \cdot d\vec{x}/dt$, of particles in a physical system – distribution, $f(\vec{x}, \vec{p}, t)$ is governed by the Vlasov equation [7],

$$\frac{\partial f}{\partial t} + \frac{\vec{p}}{am} \cdot \frac{\partial f}{\partial \vec{x}} - am\vec{\nabla}\phi \cdot \frac{\partial f}{\partial \vec{p}} = 0 \ . \tag{1.5}$$

Here m is the mass of the particle, a is the cosmic scale factor (parameter that measures its relative expansion) and ϕ is the gravitational potential. Note that the quantities defined in the last equation refer to "comoving" coordinates – those in which distances do not change in time due to the expansion of space.

This equation cannot be solved analytically in general. Here is where the N-body simulations come into play (see Figure 1.4 for a small example of a simulation). They solve this equation for as many particles as possible. Modern simulations use millions of particles to follow thousands of galaxies in a large volume of space. These simulations are traditionally run in 3D boxes with periodic boundary conditions. Keeping the volume of the box fixed, the amount of particles that are simulated define the resolution of the simulation. Thus, one particle can represent hundreds of galaxies (if the resolution is low) or even a tiny part of a galaxy (if the resolution is high). As a rule of thumb, the bigger the



(a) $a = 0.01$ **(b)** $a = 0.33$

(c) $a = 0.66$ **(d)** $a = 1.00$

Figure 1.4: Gravitational N-body simulation in a two-dimensional universe. The dots represent the particles and the colour is the density field that can be reconstructed from the particles: the darker the color, the higher the density of particles is in that area. The first frame **(a)** is the initial condition of the simulation, with only a slight perturbation, and those perturbations grow over time in frames **(b-d)**. a denotes the expansion factor of the universe, where $a = 0.01$ indicates that the universe is $100\times$ smaller than what it is today.

simulation and the resolution, the better. Note that those two quantities go against each other, the bigger the simulation, with a fixed number of particles, the smaller the resolution.

In order to run N-body simulations of hundreds of millions of particles (usually required for state-of-the-art research) millions of CPU hours are needed. Moreover, new cosmological observations will require to have thousands of such simulations. Hence super-resolution techniques become very important to help save millions of CPU hours (see, e.g., [8, 9]).

CHAPTER 2

Objectives

Due to the necessity of powerful super-resolution algorithms in both microscopy and astrophysics, the main goal of this project is to provide an algorithm that can deal with both types of images.

The objectives that we will try to fulfil in this project are the following:

1. Study the state of the art of general image super-resolution methods, as well as methods specific to microscopy and N-body simulation images.
2. Explore the available image quality assessment metrics for evaluating the results.
3. Inspired by those state-of-the-art methods, design a solution that can be applied to both EM and gravitational N-body simulation images.
4. Develop the new solution, train it to super-resolve images from a real microsocpy dataset and a real gravitational N-body simulation dataset, and compare with state-of-the-art methods.

CHAPTER 3

State of the art

3.1 Super-resolution

There are multiple techniques in the literature for performing image upscaling or super-resolution. A wide variety of classical methods have been proposed, such as prediction-based methods [10], edge-based methods [11], etc.

In recent years, with the evolution of deep learning techniques and Convolutional Neural Networks (CNN), deep learning based SR algorithms have been widely explored and often achieve state-of-the-art performance on various benchmarks of SR [1].

From the simplest bilinear interpolation to the most complex CNN based algorithms, all of them have their use cases and limitations.

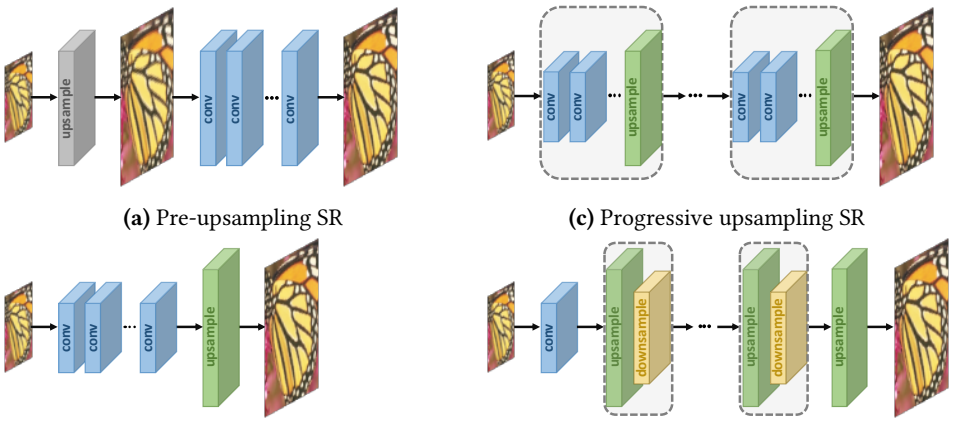
In this thesis, I will focus in exploring the deep learning algorithms that have been used for SR tasks, from the early Convolutional Neural Network based methods [12, 13] to the more recent approaches using Generative Adversarial Networks [14, 15].

3.1.1 Deep CNNs for super-resolution

Since image super-resolution is an ill-posed problem, how to perform upscaling is the key issue. Depending on the architecture of the network, different variants have been defined, such as:

- Pre-upsampling. These methods upsample the image at the beginning, typically with a fixed function such as bilinear interpolation, and then refine it using the neural network (see Figure 3.1a).
- Post-upsampling. With this method, the network extracts features in the low-resolution space and then performs the upscaling at the end using a learnable layer, for example, transpose convolution. (see Figure 3.1b).
- Progressive upsampling. These methods are based on a cascade of CNNs that progressively reconstruct higher-resolution images. They perform a smaller upsampling at each step, upsampling the image step-by-step until the desired upsampling factor is reached (see Figure 3.1c).

- Iterative up-and-down sampling. This SR framework tries to iteratively apply back-projection refinement, i.e., computing the reconstruction error then fusing it back to tune the HR image intensity. It connects upsampling and downsampling layers alternately and reconstructs the final HR result using all of the intermediate reconstructions (see Figure 3.1d).



(a) Pre-upsampling SR

(c) Progressive upsampling SR

(b) Post-upsampling SR

(d) Iterative up-and-down Sampling SR

Figure 3.1: Super-resolution model frameworks based on deep learning. The cube size represents the output size. The grey ones denote predefined upsampling, while the green, yellow and blue ones indicate learnable upsampling, downsampling and convolutional layers, respectively. And the blocks enclosed by dashed boxes represent stackable modules. Source: [1]

The Super-Resolution Convolutional Neural Network (SRCNN), proposed by Dong et al. [12] (see Figure 3.2 and 3.3), which was among the first that used CNNs for SR, uses a pre-upscaling method (Figure 3.1a). This means that the image is first upscaled using a classical, non-learnable method (in this case, bicubic interpolation). After this pre-processing step, a CNN is used to refine that simple upscaling and add finer details to the image. Since the CNN only needs to refine coarse images, this approach helps to reduce the learning difficulty, and the models created can take images with arbitrary resolutions and scaling factors. However, as most operations are performed with the higher size image, the time and memory cost are quite high.

To overcome that cost, Fast Super-Resolution Convolutional Neural Network (FSRCNN) [13] was proposed, which was a faster version of SRCNN (see Figure 3.3). They achieved this by removing the pre-processing step from SRCNN and adding a transpose convolution layer at the end of the network, as a learnable upsampling layer. This post-upscaling method (Figure 3.1b) greatly reduces the computational cost (it is more than 40 times faster [13]), but higher upscaling factors are difficult to train, and a new network would have to be trained for each scaling factor [12].

Following the SRCNN work, Jiwon et al. proposed the Very Deep Super-Resolution (VDSR) approach [16], which uses a very deep convolutional neural network inspired by VGG [17]. They use a 20-layer network with small filters, compared to the three layers used by SRCNN. However, as deep networks are harder to converge, the network learns residuals (the difference between the low resolution and the high resolution image), inspired by the popular ResNet architecture (see Figure 3.4).

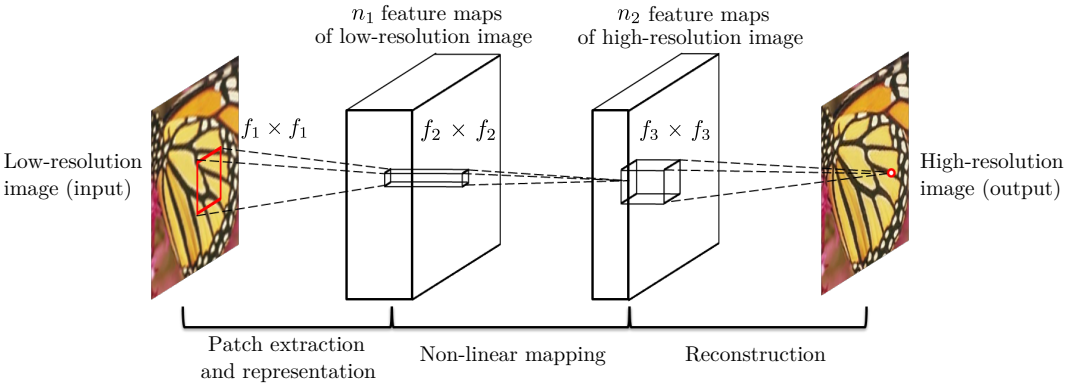


Figure 3.2: SRCNN network architecture. Given a low-resolution image, the first convolutional layer extracts a set of feature maps. The second layer maps these feature maps nonlinearly to high-resolution patch representations. The last layer combines the predictions within a spatial neighbourhood to produce the final high-resolution image. Source: [12]

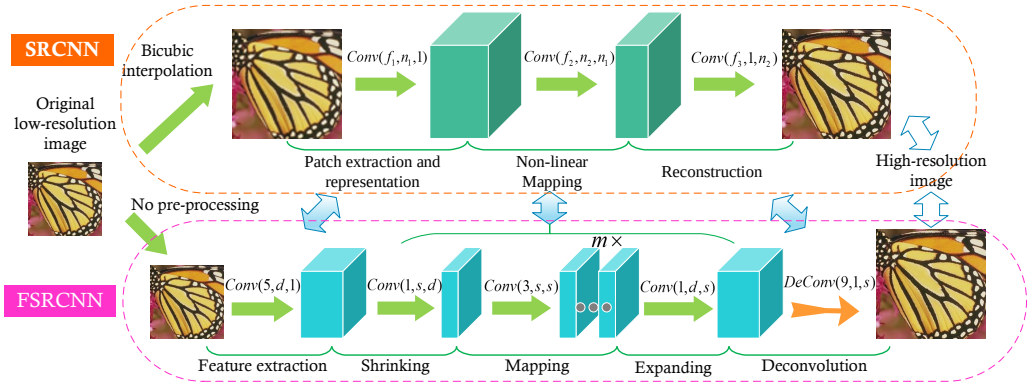


Figure 3.3: Network structures of the SRCNN and FSRCNN. In FSRCNN, the original low-resolution image is input without bicubic interpolation, and a deconvolution layer is introduced at the end of the network to perform upsampling. The non-linear mapping step in SRCNN is replaced by three steps in FSRCNN, namely the shrinking, mapping, and expanding step. Finally, FSRCNN adopts smaller filter sizes and a deeper network structure. Source: [13]

Another method that leverages residual learning is the Residual Channel Attention Network (RCAN), proposed by Zhang et. al. [18]. This network uses a post-upscaling architecture, with a residual-in-residual structure. This approach combines long skip connections over larger parts of the network with short skip connections (see Figure 3.5), which facilitate learning of very deep networks. On the other hand, in order to make the network focus on more informative features, they exploit the interdependencies among feature channels, resulting in a channel attention (CA) mechanism. This is achieved by scaling each channel by a learnable value.

More recently and based on RCAN, Qiao et. al. proposed a super-resolution method for optical microscopy called Deep Fourier Channel Attention Network (DFCAN) [19]. This method leverages the frequency content difference across distinct features to learn precise hierarchical representations of high-frequency information about diverse biological structures.

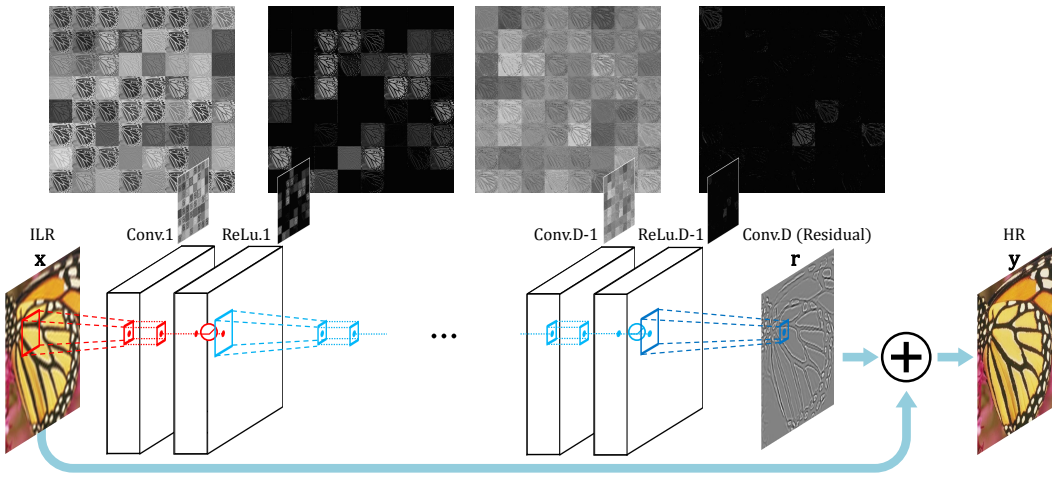


Figure 3.4: VDSR network architecture. An interpolated low resolution image (**x**) is passed through various convolutional layers and is transformed into a residual image (**r**). The element-wise addition of this residual image with the low resolution image produces the final high resolution image (**y**). Source: [16]

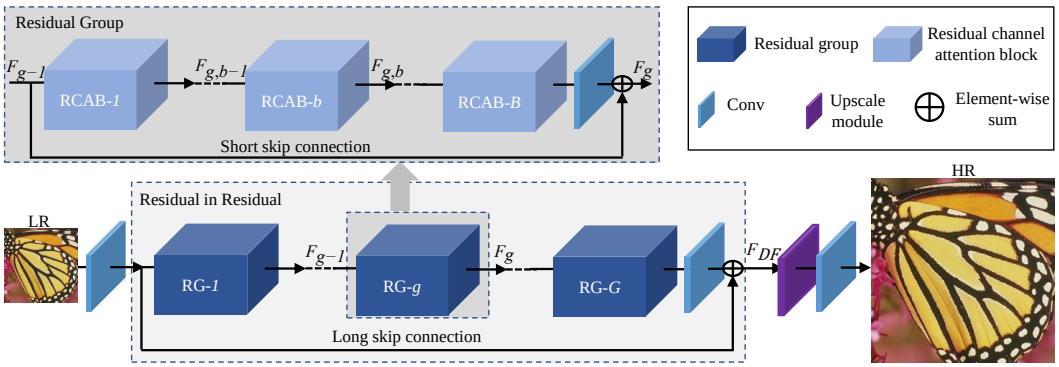


Figure 3.5: RCAN network structure. It consists on long skip connections over multiple residual groups (bottom), and short skip connections within each residual block (top). Source: [18]

3.1.2 Generative Adversarial Networks for Super-Resolution

Generative Adversarial Network (GAN) is a class of machine learning methods where two neural networks contest with each other in a zero-sum game [20]. In super-resolution, it is straightforward to use adversarial training: a SR model is trained as a generator, and a discriminator is defined to determine if the input image is generated or not. The generator then tries to "fool" the discriminator into thinking the fake images are actually real by making them realistic. In order to make the generated images as close to the original as possible, the loss function of the generator is usually composed of the weighted sum of the adversarial loss and a content loss, like such:

$$\mathcal{L}_G(\tilde{I}, I) = \alpha \mathcal{L}_C(\tilde{I}, I) + \beta \mathcal{L}_D(\tilde{I}) \tag{3.1}$$

where $\mathcal{L}_C$ is the content loss and $\mathcal{L}_D$ is the adversarial or discriminator loss. In this framework, the discriminator is trained to maximise its output ($\mathcal{L}_D$) for fake images and minimise it for real images. Hence the generator is trying to create fake images that minimise

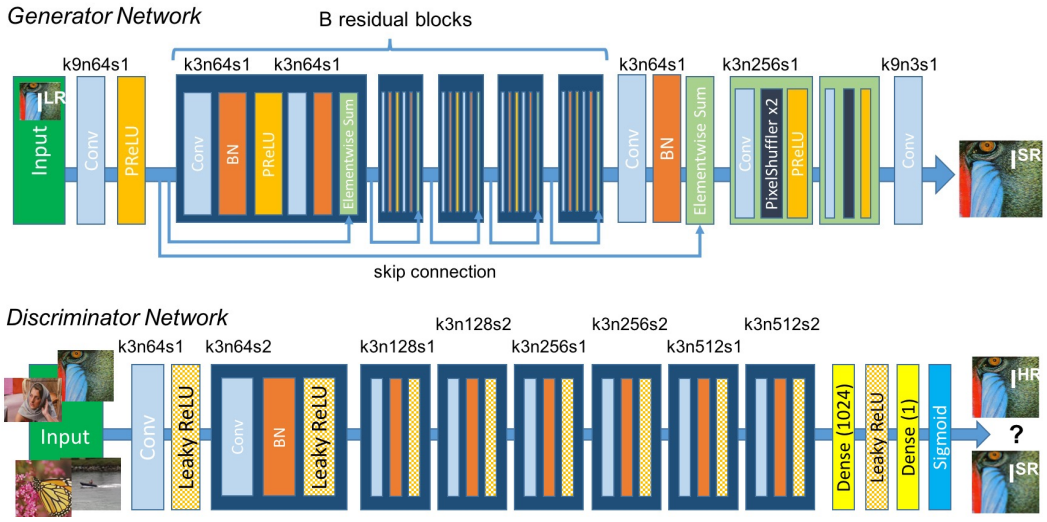


Figure 3.6: Architecture of SRGAN's Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer. Source: [14]

the discriminator's output. I and $\tilde{I}$ are the original HR image and its reconstruction, respectively. α and β are coefficients given to both losses.

This type of training has been used in some SR methods, which produce more realistic looking images [1].

The Super-Resolution Generative Adversarial Network (SRGAN), proposed by Ledig et al. [14] is a GAN-based framework (see Figure 3.6) that produced photo-realistic images with a scaling factor of $4\times$. In this solution, the generator uses a network based on ResNet with residual-in-residual skip connections. The adversarial loss uses a discriminator network based on VGG [17] to differentiate between the super-resolved images and original photo-realistic images.

For the content loss, they tested their network with both pixel-wise mean squared error (MSE) loss and VGG loss. For the VGG loss they extract the image features of both the super-resolved and original images using a pretrained VGG as described by Simonyan in [17], and then calculate the MSE between those features. The latter loss produced more realistic images that achieved better Mean Opinion Scores.

In the literature, there have been multiple improvements over SRGAN's work. One notable example is the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN), proposed by Wang et. al. [15]. They improved three key components from SRGAN: network architecture, adversarial loss and perceptual loss. First, they introduce the Residual-in-Residual Dense Block (RRDB) without batch normalisation as the basic network building unit. Second, they improve the discriminator using Relativistic average GAN (RaGAN) [21], which learns to judge "whether one image is more realistic than the other" rather than "whether one image is real or fake". Lastly, regarding perceptual loss, they extract the features from VGG *before* activation, instead of after like in SRGAN. All of these improvements consistently produced better visual quality with more realistic and natural textures than SRGAN and won the first place in the PIRM2018-SR Challenge [22] (see Figure 3.7).

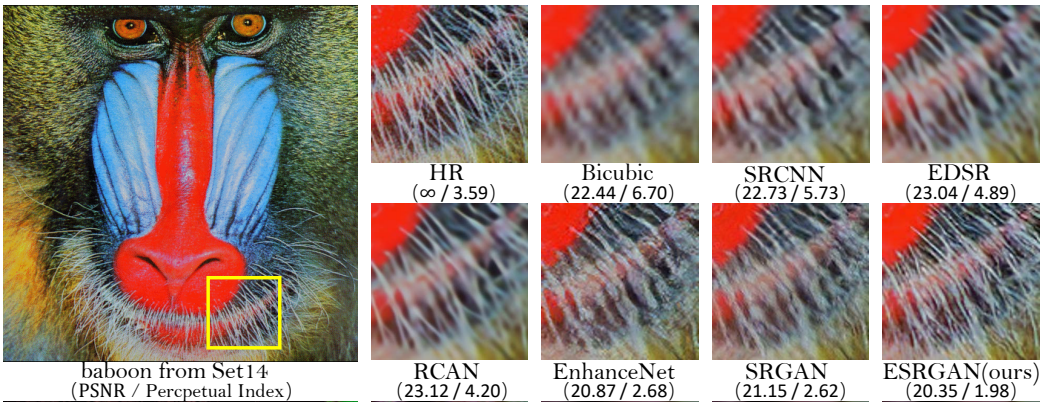


Figure 3.7: Comparison of various SR methods in a picture from Set14 with $4\times$ upscaling. ESRGAN produces more natural textures and less artefacts. Source: [15]

One more interesting GAN-based approach that can be applied for super-resolution is SinGAN, proposed by Shaham et. al. [23]. This algorithm can effectively learn the mapping between low-resolution and high-resolution images using just one reference image for training. In this method, they use an architecture composed of multiple generators and discriminators that work on different scales. The main idea is to reconstruct the coarser or low-frequency details in the smallest scale, then upscale and refine those images in subsequent scales, trying to reconstruct higher frequency details (see Figure 3.8). Despite having been trained with only one image, this approach achieves comparable results to SRGAN.

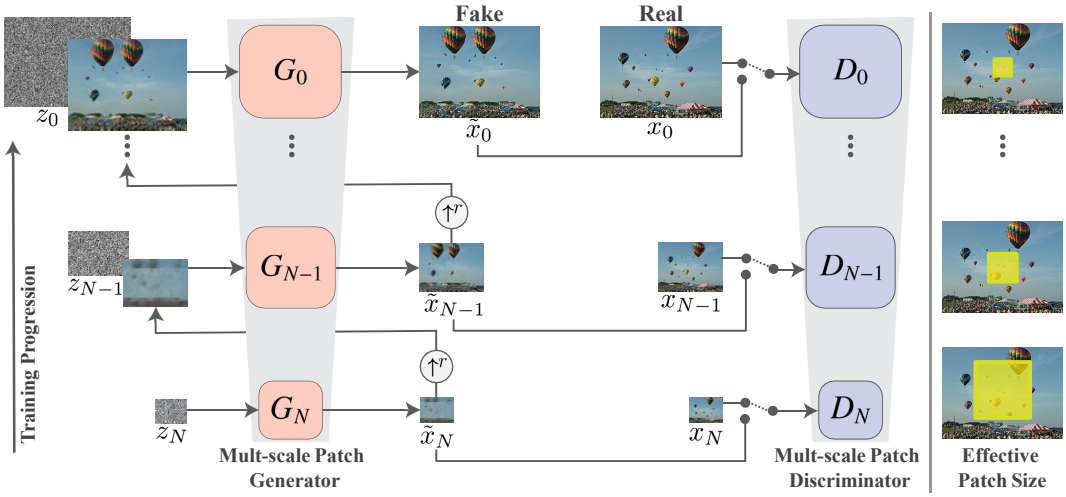


Figure 3.8: SinGAN's multi-scale pipeline. The model consists of a pyramid of GANs, where both training and inference are done in a coarse-to-fine fashion. At each scale, G_n (left) learns to generate image samples in which all the overlapping patches cannot be distinguished from the patches in the down-sampled training image, x_n, by the discriminator D_n (right). When tuning the network for SR, the low resolution image is also input at the coarser layer, in conjunction with the noise. Source: [23].

In cosmological N-body simulations, there have been efforts to produce a higher resolution simulation from a low resolution one. Li et. al. [24] use a Wasserstein GAN

with an architcture inspired by StyleGAN2 [25] to enhance the simulation by adding more particles to an existing simulation and predicting their displacement. This way, what they generate is a new 3D simulation, instead of projections of the density. In contrast, Kodi Ramanah et. al. [26] map the distribution of the low-resolution cosmic density field to the space of the high-resolution small-scale structures.

3.2 Wasserstein GAN

In spite of the promising results that GAN-based methods produce, currently the training process is still difficult and unstable [1]. GAN-based SR methods usually need more time to converge and produce good results, and balancing the training of the generator and discriminator is often difficult as one of them may overfit or underfit.

One attempt in stabilising the training of Generative Adversarial Networks is Wasserstein GAN (WGAN) [27], later revised with WGAN with Gradient Penalty (WGAN-GP) [28]. This paper proposes a new cost function used in the generative model, to replace the more commonly used in GAN *Kullback-Leibler* (KL) and *Jensen-Shannon* (JS) divergences.

Suppose we have a real data distribution p with mean 0, which we assume is Gaussian, and a few q distributions estimated from the model with means ranging from 0 to 35. When $p = q$, the divergency is 0, and as the mean of q increases, the divergency increases. However, the gradient of this divergency eventually diminishes, which makes gradient-descent learning very difficult (see Figure 3.9). In general terms, this means that if the generator is not doing a good job yet, the gradient for the generator diminishes and the generator learns nothing.

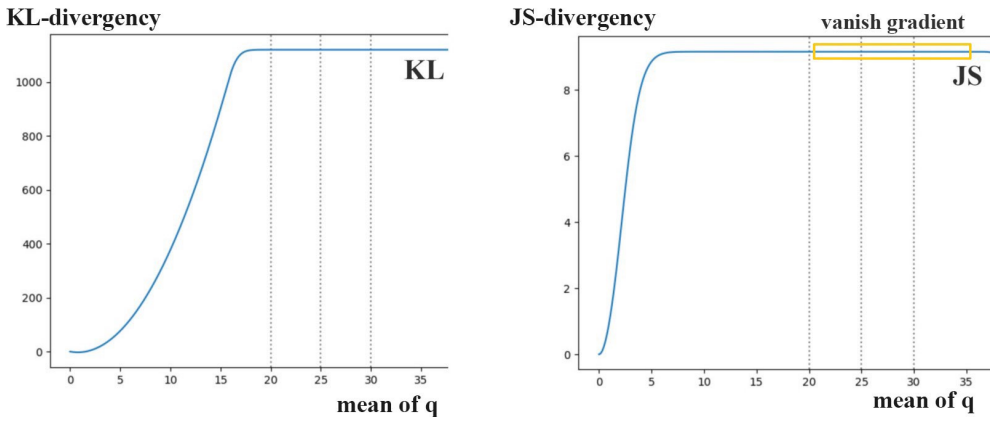


Figure 3.9: Plot of Kullback-Leibler (KL) and Jensen-Shannon (JS) divergencies for q with means ranging from 0 to 35. As the mean of q increases, the gradient tends to 0. Source: [29]

Wasserstein GAN proposes a new cost function that uses the Wasserstein distance, or Earth Mover's Distance. Informally, if the distributions are interpreted as two different ways of piling up a certain amount of earth (dirt) over the region, the Earth Mover's Distance is the minimum cost of make one pile equal to the other; where the cost is assumed to be the amount of dirt moved times the distance by which it is moved [30]. The Wasserstein distance cannot be analytically solved, so the discriminator takes the role of estimating this distance.

This way, the weights have a smoother gradient, no matter if the generator is performing or not (see Figure 3.10).

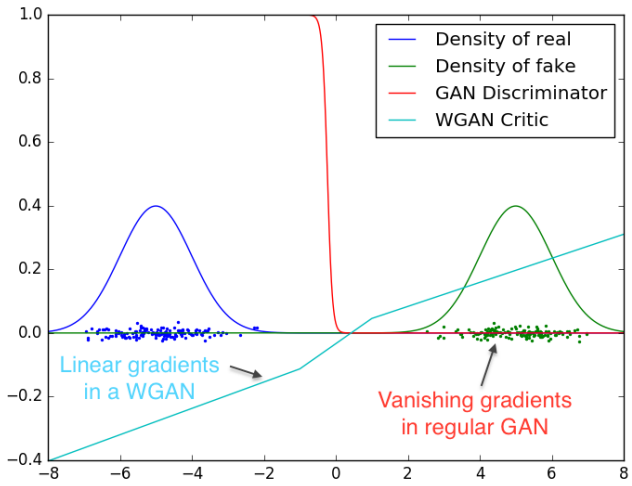


Figure 3.10: Optimal discriminator and critic when learning to differentiate two Gaussians. Source: [27]

In terms of its implementation, the network design is the same, except the discriminator does not have an output activation function.

The main difference between WGAN and WGAN-GP is in the loss functions of the discriminator (Equations 3.2a and 3.3a) and generator (Equations 3.2b and 3.3b). In WGAN, the discriminator is renamed to critic and represented as function f, as its role is now to estimate the Wasserstein distance.

GAN loss functions [20]:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right] \tag{3.2a}$$

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \tag{3.2b}$$

WGAN loss functions [27]:

$$\nabla_{w} \frac{1}{m} \sum_{i=1}^{m} \left[f\left(\boldsymbol{x}^{(i)}\right) + f\left(G\left(\boldsymbol{z}^{(i)}\right)\right) \right] \tag{3.3a}$$

$$\nabla_{\theta} \frac{1}{m} \sum_{i=1}^{m} f\left(G\left(\boldsymbol{z}^{(i)}\right)\right) \tag{3.3b}$$

Here, $\boldsymbol{x}$ is the real, high-resolution image, and $\boldsymbol{z}$ is the downscaled, low-resolution version. G is the generator, D is the discriminator and f is the critic. m is the batch size. ∇_{θ_d} and ∇_{w} are the gradients of the weights of the generators, and ∇_{θ_g} and ∇_{θ} are the gradients of the weights of the discriminator and critic, respectively.

In WGAN's case, f must enforce Lipschitz constraints, therefore the critic's weights are clipped after each gradient update, like so:

$$w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$$
$$w \leftarrow \text{clip}(w, -c, c)$$

where α is the learning rate, w is the weight of the critic, g_w is the gradient of the weight and c is the value to which the weights are clipped.

Lastly, the critic is updated n_{critic} times (5 is recommended by the authors) for every generator update, in order to train the critic close to convergence.

However, in WGAN this weight clipping was used as it was simple and performant, but they don't think it's a good solution. Quoting from [27]:

> Weight clipping is a clearly terrible way to enforce a Lipschitz constraint. If the clipping parameter is large, then it can take a long time for any weights to reach their limit, thereby making it harder to train the critic till optimality. If the clipping is small, this can easily lead to vanishing gradients [...]

Because of this, Gulrajani et. al. propose another solution in [28].

> A differentiable function is 1-Lipschtiz if and only if it has gradients with norm at most 1 everywhere, so we consider directly constraining the gradient norm of the critic's output with respect to its input.

Therefore, to the original critic loss, a gradient penalty is added with coefficient λ (see Equation 3.4, [28]), where we calculate the gradients with reference to an interpolated input. After that, we enforce the 2-norm of the gradient to be equal to 1.

$$\mathcal{L} = \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [D(\boldsymbol{x})] + \lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(||\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})||_2 - 1)^2 \right] \tag{3.4}$$

where $\hat{\boldsymbol{x}}$ is sampled uniformly along a straight line between the generated sample $\tilde{\boldsymbol{x}}$ and the real sample $\boldsymbol{x}$:

$$\hat{\boldsymbol{x}} = \epsilon \tilde{\boldsymbol{x}} + (1 - \epsilon)\boldsymbol{x} \text{ where } 0 \leq \epsilon \leq 1$$

3.3 Image Quality Assessment (IQA)

The best way to assess the quality of an image is perhaps to look at it because human eyes are the ultimate receivers in most image processing environments [31]. Therefore, subjective methods based on humans' perception are more in line with our need.

However, these Mean Opinion Score (MOS) methods are too inconvenient, slow and expensive for practical usage. Because of that, various objective computational methods are used for IQA.

The objective IQA methods are divided into three categories [32]: full-reference metrics that perform assessment using reference images, reduced-reference metrics based on comparisons of extracted features, and no-reference metrics without any reference image.

Peak signal-to-noise ratio (PSNR) is one of the most popular full-reference metrics for IQA. It is defined via the maximum pixel value (255 for 8-bit colour images) and the mean squared error (MSE).

$$\text{PSNR} = 10 \times \log_{10} \left(\frac{L^2}{\frac{1}{N} \sum_{i=1}^{N} (I(i) - \tilde{I}(i))^2} \right) \tag{3.5}$$

where I is the true image, and $\tilde{I}$ is the reconstruction; N is the number of pixels in the image; and L is the maximum pixel value of the images.

Since PSNR only looks at pixel-level MSE, it often leads to poor performance in representing reconstruction quality in real scenes [1]. However, with the need to compare with other works in the literature, it is still a widely used IQA metric.

Another widely used IQA metric is the Structural Similarity index (SSIM) [32], which measures the structural similarity between images based on independent luminance, contrast and structure comparisons. This metric measures the perceptual quality better and thus, it is also popular.

While most super-resolution images are evaluated by full-reference metrics, the effectiveness is not clear and the required ground-truth images are not always available in practice. Due to this issue, some no-reference metrics have been developed to evaluate the perceptual quality of super-resolved images [33, 34].

3.3.1 Perceptual Quality vs. Distortion

Image super-resolution methods, or any image restoration algorithm, are typically evaluated using some distortion measure (e.g. PSNR, SSIM, etc.) or by human opinion scores that quantify perceptual quality. However, Blau and Michaeli mathematically proved [35] that distortion and perceptual quality are at odds with each other.

In that paper, they show that, as distortion decreases, the probability for correctly discriminating generated and real images increases, thus decreasing perceptual quality. Therefore, they deem impossible to create an image that has both low distortion *and* high perceptual quality (see Figure 3.11).

An example of this phenomenon can be seen in Figure 3.12, where an SR algorithm may have a lower PSNR value, but it *looks* better to the human viewer.

Because of this tradeoff, some GAN-based SR methods [14], which often achieve a higher perceptual quality and higher distortion, combine popular metrics like PSNR and SSIM with Mean Opinion Score (MSO) [38], or with a perceptual quality metric like Ma et. al. [33] or Perceptual Index [22], to assess the performance of their solution.

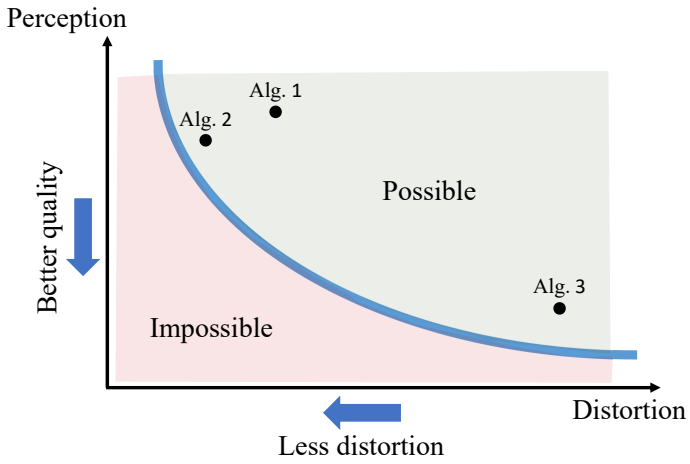
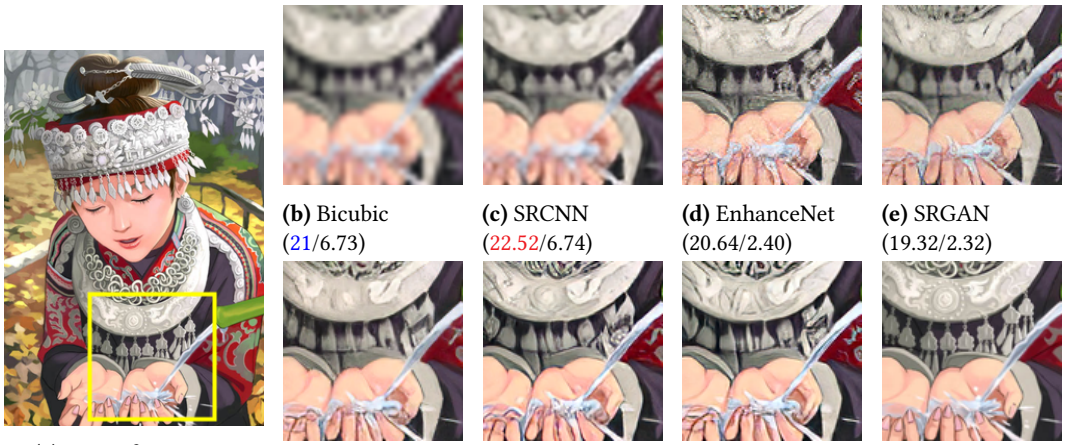


Figure 3.11: The perception-distortion tradeoff. There exists a region in the perception-distortion plane which cannot be attained, regardless of the algorithmic scheme. When in proximity of this unattainable region, an algorithm can be potentially improved only in terms of its distortion or in terms of its perceptual quality, one at the expense of the other. Source: [35]



(a) comic from Set14

(b) Bicubic (21/6.73)

(c) SRCNN (22.52/6.74)

(d) EnhanceNet (20.64/2.40)

(e) SRGAN (19.32/2.32)

(f) ESRGAN (18.87/2.27)

(g) ESRGAN+ (18.06/2.17)

(h) nESRGAN+ (17.76/2.63)

(i) HR (∞/2.76)

Figure 3.12: Comparison of various state-of-the-art super-resolution methods, with a $4\times$ upscaling factor using an image from the Set14 dataset [36]. PSNR (left) and Perceptual Index [22] (right) metrics are shown. The best score for each metric is shown in red, and the second-best is shown in blue. Source: [37]

CHAPTER 4

Methodology

In this chapter we will describe our method used to upsample images from two type of sources: electron microscopy and gravitational N-body simulations. We will explain the network structure in detail, the loss functions used for optimisation and the steps made in the training loop.

4.1 Network Architecture

For the network architecture, we decided to go with a GAN approach, because, as we discussed in Section 3.3.1, they usually produce more realistic and sharper images, albeit at the cost of a higher distortion. This approach would, in theory, minimise the *blurriness* or *haziness* effect that other CNN-based algorithms produce, thus creating an image that *looks* better to the human eye.

These GANs are composed of two networks: a generator, which has the role of upsampling the images, and a discriminator (critic in a Wasserstein GAN), which is trained to discern between generated and real images.

4.1.1 Generator

The generator uses a residual-in-residual model based on the popular ResNet architecture. The model has N residual blocks composed of a 2D Convolutional (Conv2D) layer with 64 filters of size 3×3, a Parametric ReLU (PReLU) layer (see Equation 4.1, α is a learnable value), and another Conv2D layer with 64 filters of size 3×3, with a short residual skip connection. There is also a long residual skip connection over all of the residual blocks.

$$\mathrm{PReLU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{otherwise} \end{cases} \tag{4.1}$$

We use a learnable post-upsampling method for upscaling the images. The upsampling layer is a sub-pixel convolution layer *PixelShuffle*, originally proposed by Shi et. al. [39], which aggregates various feature maps into a single layer (see Figure 4.1). Each upsampling block has a $2\times$ upsampling factor, so we add $\log_2(S)$ upsample blocks, depending on the

desired upscaling factor. This also means that the upscaling is fixed and a new network has to be trained for each factor.

In these upsampling blocks, there is a Conv2D layer with 256 3×3 filters, the PixelShuffle layer with an upscaling factor of $2\times$, which results in 64 feature maps from those 256 in the input, and a PReLU activation layer.

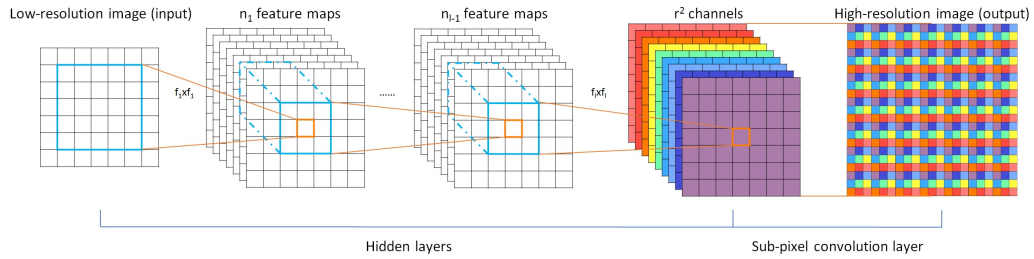


Figure 4.1: Sub-pixel convolution layer (PixelShuffle) operation. Here, it transforms nine 7×7 feature maps into a single 21×21 image. Source: [39]

We also use an initial convolutional layer with 64 5×5 filters to extract features, and a final convolutional layer with one 5×5 filter to reconstruct the single-channel output image. Finally, we use a hyperbolic tangent for the final activation.

In Figure 4.2 there is a visual representation of the generator network.

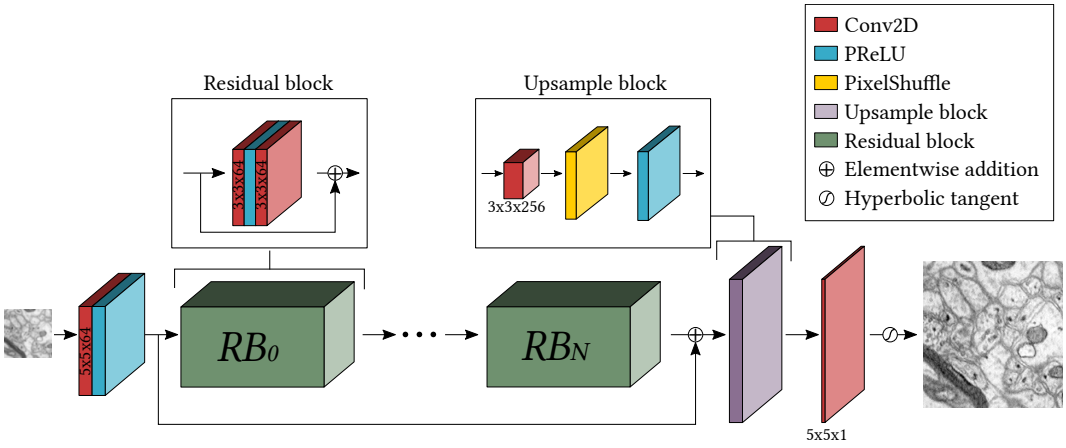


Figure 4.2: Generator architecture. It is composed of N residual blocks and $\log_2(S)$ upsample blocks, where S is the upscaling factor of the image. $S = 2^k; k \in \mathbb{N}$. Numbers in Conv2D blocks indicate [*kernel width*]x[*kernel height*]x[*number of filters*].

4.1.2 Critic

For the critic, we use a network design inspired by SRGAN [14], which in turn follows the guidelines of Radford et. al. [40].

This design uses a series of Conv2D layers with a stride of 2, in order to widen the receptive field, and an increasing number of 3×3 sized filters in each step, similar to VGG [17]. We use Instance Normalisation after each convolutional layer, as proposed in [41], to improve stability of the learning. Finally, we apply Leaky ReLU as activation [42].

In contrast with [14] and [40], instead of flattening the last convolutional layer and using fully connected layers as a classifier, we average over the last convolutional layer, which uses only one filter, to output a single value per image. In this particular use case, this produced better results and eased the training.

As this is a critic part of a Wasserstein GAN with Gradient Penalty (WGAN-GP) [28], instead of classifying the image into true or fake it estimates the Wasserstein distance, or the earth mover's distance. This value is not restricted to a defined range, such as $[0 - 1]$ in a traditional GAN. Thus, the critic does not have a final activation function.

A visual representation of the architecture can be found in Figure 4.3.

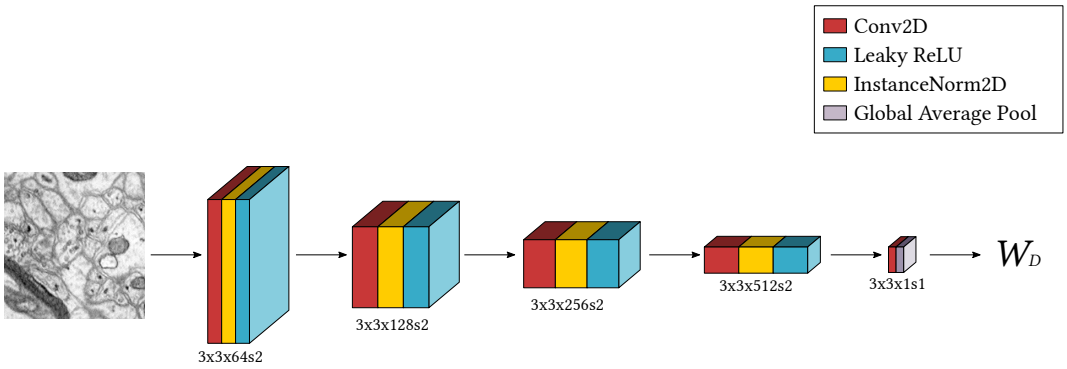


Figure 4.3: Basic architecture of the critic. Numbers below the blocks indicate [*kernel width*]x[*kernel height*]x[*number of filters*]s[*stride*] in the Conv2D layers.

4.2 Loss functions

The loss functions for our GAN are based on WGAN-GP [28], and adapted for the super-resolution use case.

In case of the loss function of the critic, it has not had any modifications from the original WGAN-GP paper, as the critic's purpose remains the same, estimating the Wasserstein distance (see Equation 4.2, [28]).

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m} C(\tilde{\boldsymbol{x}}^{(i)}) - C(\boldsymbol{x}^{(i)}) + \lambda \left[(||\nabla_{\hat{\boldsymbol{x}}^{(i)}} C(\hat{\boldsymbol{x}}^{(i)})||_2 - 1)^2 \right] \tag{4.2}$$

where C is the critic, $\boldsymbol{x}^{(i)}$, $\tilde{\boldsymbol{x}}^{(i)}$ and $\hat{\boldsymbol{x}}^{(i)}$ are the HR image, its reconstruction and an interpolation between them, respectively, and λ is the gradient penalty coefficient.

The loss function of the generator, however, has been changed to fit our needs. The original use-case of GANs was to generate realistic images of a determined style or type from random noise. Therefore, using just the adversarial loss is sufficient.

In contrast, the input in our network is not noise, but the low resolution version of the image we want to restore. Therefore, we need the generated image to be as close to the original as possible, while still looking realistic. Therefore, we add a L1 loss factor to

the loss function with a set coefficient (see Equation 4.3). As we will discuss in Chapter 5, changing this coefficient value greatly affects the image in the output.

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m} -C(\tilde{\boldsymbol{x}}^{(i)}) + \gamma \mathrm{L1}(\boldsymbol{x}^{(i)}, \tilde{\boldsymbol{x}}^{(i)}) \tag{4.3}$$

where C is the critic and L1 is the mean average error. $\boldsymbol{x}^{(i)}$ and $\tilde{\boldsymbol{x}}^{(i)}$ are the original HR image and its reconstruction, respectively. γ is the coefficient for the L1 loss, and m is the batch size.

The network was implemented using PyTorch.

4.3 Training strategy

As we are building a Wasserstein GAN with gradient penalty, the training will be similar to that in the original paper [28].

For each step where we optimise the generator, the critic is optimised n_{critic} steps. This, as the authors of the paper suggest, ensures that the critic is close to converging.

In the generator step, we sample a batch of data, $\boldsymbol{x}$, from the real distribution (the high-resolution image). Then, we get the low resolution version of those images, $\boldsymbol{z}$, using a degradation function $\mathcal{D}$ specific to each dataset. We then use the generator to generate a fake high-resolution version from those generated low resolution images, $\tilde{\boldsymbol{x}}$. Finally, we update the generator with the Adam optimiser based on the loss function in Equation 4.3. As a last step, we update the learning rate of the generator using the One Cycle learning rate scheduler [43].

Similarly, for the critic we sample a batch of real data $\boldsymbol{x}$, degrade it with $\mathcal{D}$ to get $\boldsymbol{z}$, and get the super-resolved image $\tilde{\boldsymbol{x}}$ using the generator. Then, we generate a random number $0 \leq \epsilon \leq 1$. We use that to build an interpolated image between the real image and the generated image: $\hat{x} \leftarrow \epsilon x + (1-\epsilon)\tilde{x}$. After that we calculate the gradient with respect to this interpolated image using the critic, and get its 2-norm. Finally, we update the critic with the Adam optimiser based on the loss function in Equation 4.2. As a last step, we also update the learning rate of the critic using the One Cycle learning rate scheduler.

The whole training process can be found in Algorithm 1.

The training loop was programmed using PyTorch Lightning.

Algorithm 1 WGAN with gradient penalty for Super-Resolution.

Require: A degradation function $\mathcal{D}$, a reconstruction loss function $\mathcal{R}$, the gradient penalty coefficient λ, the reconstruction coefficient γ, the number of critic iterations per generator iteration n_{critic}, the batch size m, number of steps n_{steps}, Adam hyperparameters β_1, β_2.

Require: Initial critic parameters w_0, initial generator parameters θ_0, inital learning rates α_{g0}, α_{g0}

for $s = 1, \dots, n_{\text{steps}}$ **do**
  if $s \mod n_{\text{critic}} = 0$ **then** ▷ Update Generator
    for $i = 1, \dots, m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$
      $z \leftarrow \mathcal{D}(x)$
      $\tilde{x} \leftarrow G_\theta(z)$
      $L^{(i)} \leftarrow -C_w(\tilde{x}) + \gamma \mathcal{R}(x, \tilde{x})$
    end for
    $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, \theta, \alpha_g, \beta_1, \beta_2)$
    $\alpha_g \leftarrow \text{OneCycle}(\alpha_g, s)$
  else ▷ Update Critic
    for $i = 1, \dots, m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, a random number $\boldsymbol{\epsilon} \sim U[0, 1]$.
      $z \leftarrow \mathcal{D}(x)$
      $\tilde{x} \leftarrow G_\theta(z)$
      $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
      $L^{(i)} \leftarrow C_w(\tilde{x}) - C_w(y) + \lambda(||\nabla_{\hat{x}} C_w(\hat{x})||_2 - 1)^2$
    end for
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha_c, \beta_1, \beta_2)$
    $\alpha_c \leftarrow \text{OneCycle}(\alpha_c, s)$
  end if
end for

CHAPTER 5

Results

5.1 Electron microscopy

5.1.1 The dataset

The image data used was produced by Lichtman Lab at Harvard University (Daniel R. Berger, Richard Schalek, Narayanan "Bobby" Kasthuri, Juan-Carlos Tapia, Kenneth Hayworth, Jeff W. Lichtman). Their corresponding biological findings were published in [44].

This electron microscopy (EM) dataset is comprised of 100 training images and 100 evaluation images. They are monochromatic images with size 1024×1024. The training and evaluation data sets are both 3D stacks of 100 sections from a serial section Scanning Electron Microscopy (ssSEM) data set of mouse cerebral cortex. The microcube measures $6 \times 6 \times 3$ microns approx., with a resolution of $6 \times 6 \times 30$ nm/voxel.

An example of this high-resolution dataset can be found in Figure 5.1.

5.1.2 Preprocessing and data augmentation

The used dataset only has high-resolution versions of the images. Therefore, we need to create low-resolution images synthetically.

For downsampling the images, we used the EM "crappifying" method from [45] to synthetically degrade the HR images to LR, which approximates the real-world low resolution images of the same field of view.

In this method, we apply a Gaussian blur filter to the HR image with a standard deviation of $\sigma = 3$, and then scale it down to the desired size with bilinear interpolation. The main factor that we wanted to explore was $4\times$ downscaling on each axis, but we also ran our algorithm with $2\times$ and $8\times$ downscaling factors.

A sample of the effect of this "crappifying" method is shown in Figure 5.2.

After downscaling, both HR and LR variants of the image were divided into 64 patches of sizes 128×128 and 32×32 respectively. This input size has had the best results for training among the ones we tested. These image patches are used only for training, as the images used for calculating evaluation metrics are upsampled using the whole image.

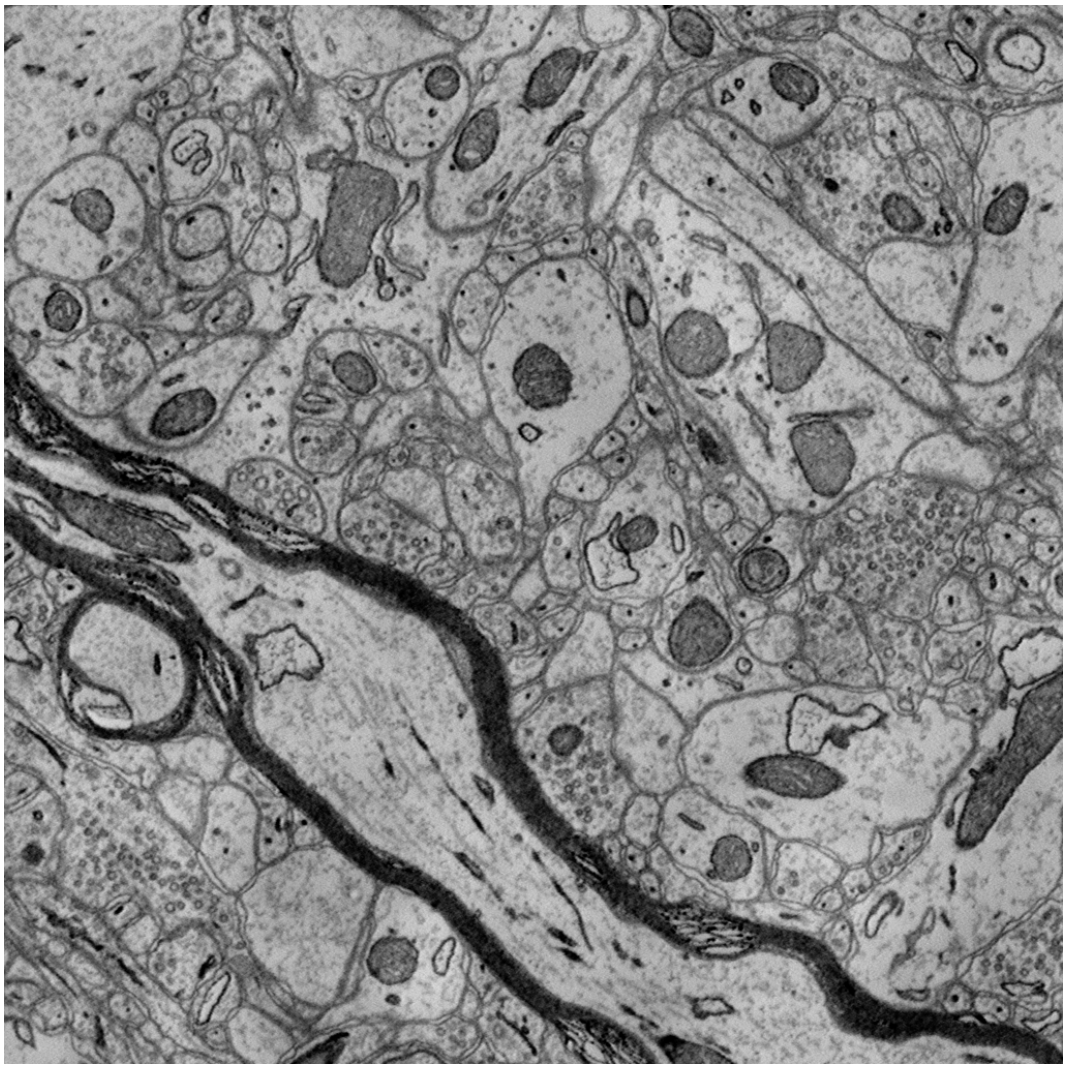
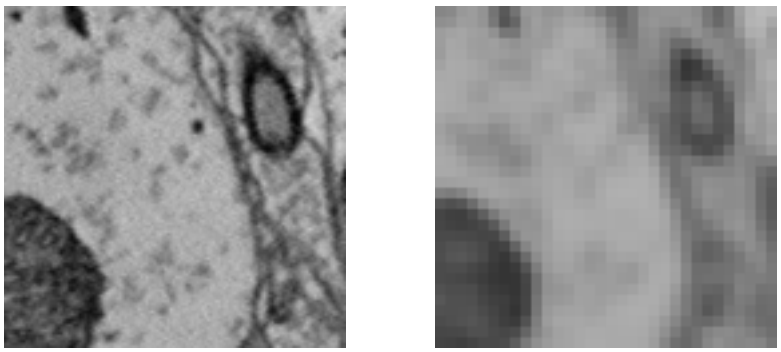


Figure 5.1: Sample image from the electron microscopy dataset.



(a) High-resolution patch **(b)** Low-resolution patch

Figure 5.2: Sample of a patch in the EM dataset, and its "crappified" low-resolution version, upscaled to the same size using nearest-neighbour interpolation.

Finally, to introduce some variation of the dataset in training, we apply a rotation of the images, with a rotation angle randomly chosen from $\{0, 90, 180, 270\}$. They are also flipped randomly around the horizontal and vertical axis.

5.1.3 Evaluation metrics

As we have shown in Chapter 3, it is difficult to evaluate image quality, as distortion and perceptual quality are at odds with each other, and both cannot be had at the same time.

For that reason, multiple metrics were used to evaluate the quality of the images, both full-reference and no-reference metrics. For full-reference, we used PSNR and SSIM, which are standard in the literature and can be easily compared to other works. These are both distortion measures, and do not represent perceptual quality well.

For the no-reference metric, we used Perceptual Index (PI) [22], which is a weighted sum of Ma et. al. [33] and NIQE [34] metrics. Individually, those metrics try to measure image "realism" by combining various sources of information in the image, such as spatial and frequency information. This measure, as it is a no-reference metric, can't be used by itself to evaluate the reconstruction quality, but it is useful to assess the perceptual quality of such reconstruction.

Lastly, we also conducted a survey (see Figure 5.3) among people who have experience with EM images, to rank some of the methods from best to worst. We randomly chose 10 images from the evaluation dataset and super-resolved them with the following methods:

- Our method, $\gamma = 100$
- Our method, $\gamma = 50$
- ESRGAN+ [15]
- RCAN [18]
- DFCAN [19]

More specifically, we cut a 256×256 patch of the image from the centre, in order to make it more easy to view all of them at once, in full size, in a computer screen.

We presented all of the images anonymised and in a random order to prevent bias in the survey.

To extract a score from the survey, we used the average position that a particular method has had in the ranking, 1 being the best, and 5 being the worst. The survey was filled out by 4 people, so each of the MOS was determined by the average over 40 samples.

5.1.4 Results

All the experiments presented in this section were done with a downsampling factor of $4\times$ in each axis. We ran our images through the state-of-the-art methods ESRGAN+, RCAN and DFCAN for comparison. We also tested our method with various values for its hyperparameter γ, with values of 200, 100 and 50 shown. Higher and lower values were also tried during the investigation. With higher values of 500 and 1000, the adversarial

Figure 5.3: First question of the survey that we conducted in order to evaluate the quality of the images. Users have to drag the images from the left column to the right column, and place them in order from best quality to worst quality.

part of the loss function was almost ignored, and it produced softer images, similar to those of RCAN (see Section 5.1.4.2). With a small value of $\gamma = 10$, we could not get the network to converge with this data.

5.1.4.1 Quantitative analysis

The results of various SR methods on the EM dataset can be found in Table 5.1.

	Loss function	PSNR↑	SSIM↑	PI↓	MOS↓
Reference	N/A	∞	1	3.955	-
Bilinear	N/A	21.25	0.4180	9.539	-
Our method, $\gamma = 200$	Adversarial + L1	25.82	0.7548	4.461	-
Our method, $\gamma = 100$	Adversarial + L1	25.15	0.7331	4.335	**1.375**
Our method, $\gamma = 50$	Adversarial + L1	24.79	0.7143	4.126	2.425
ESRGAN+	Adversarial + VGG + L1	22.49	0.6032	**3.932**	2.35
RCAN	L1	**27.11**	**0.8046**	6.525	4.075
DFCAN	L1	26.44	0.7818	6.871	4.775

Table 5.1: Comparison of various deep CNN-based SR methods' performance on the EM dataset at $4\times$ upscaling factor, compared with the reference image and simple bilinear upscaling. Measures are PSNR, SSIM, Perceptual Index (lower is better) and Mean Opinion Score (lower is better). Best performance on each measure is highlighted in bold. Scores marked with '-' have not been tested. Hyperparameters used in these experiments are in Table 1.

As we see, the algorithm that produced the best Perceptual Index score has been ESRGAN+, with 3.932. Note that this is a no-reference metric, and the original HR image scored worse, 3.955. On the other hand, the PSNR and SSIM of this algorithm are the lowest among the ones we tested, with 22.49 dB and 0.6032 respectively.

On the other end of the spectrum, the algorithm with the highest PSNR and SSIM score has been RCAN, with 27.11 dB and 0.8046 respectively, but it also has the second worst Perceptual Index, at 6.525.

This results further reinforce the point explained in Section 3.3.1: perception and distortion are at odds with each other. As perceptual quality increases, both PSNR and SSIM go down, and vice versa. It's important to find the balance between these two in each use case, as low distortion and higher accuracy might be more important than realism.

In the survey we conducted, four experts ranked 10 images of various super-resolution methods from best to worst. In that survey, our method, in the $\gamma = 100$ variant consistently got a higher rank than the other methods, with an average rank of 1.375. The next pair of methods in the ranking, which have a similar score, are ESRGAN+ and our method with $\gamma = 50$, with average ranks of 2.35 and 2.425. The next algorithm with the best ranking was RCAN, with an average rank of 4.075 and, finally, DFCAN with an average ranking of 4.775.

After completing the survey, some experts commented that "it was complicated and nothing was comparable to the reference image. It was difficult to choose between the three best because each had its own relevant flaws and advantages."

5.1.4.2 Qualitative analysis

If we do a qualitative analysis of the images (see Figure 5.4), we can see that algorithms that are not GANs (RCAN and DFCAN), produce more blurry and washed out images. This is the result of optimising towards pixel accuracy through losses such as MSE or MAE, without taking into account the look of the whole image. On the other hand, these methods, especially RCAN, preserve the structures of the original image better than some GAN-based methods, although that might be the result of masking smaller errors with a blurry image. A good example of this are the membranes of the neurons to the centre-left in Figures 5.4c and 5.4d (red arrows). In RCAN the lines are blurrier but better represent the structure in the original high-resolution image (5.4h). ESRGAN+, in contrast, changes the position and shape of the neurons. This can be verified with the SSIM scores in Table 5.1, as ESRGAN+ has a significantly lower score than the other methods. Lastly, DFCAN can't clearly define these neurons in the image.

Comparing the GAN based methods, ESRGAN+ captures the texture of the original image better, while the others blur the image more, especially in the lightest parts of the image. Because of that, it got a better Perceptual Index than our method (Table 5.1). Focusing on the vesicles at the top-centre of the image and towards the bottom-right (blue arrows), in our method they are very clearly defined, especially in the $\gamma = 200$ variant (Figure 5.4g). ESRGAN+ and the $\gamma = 50$ variant don't show all of the circles, and in ESRGAN+ they are not well defined. In contrast, both DFCAN and RCAN aren't able to recreate those smaller details.

Lastly, the $\gamma = 50$ variant produced some artefacts similar to salt-and-pepper noise, with some pixels turning close to black or white. We couldn't identify what caused this issue.

5.2 Gravitational N-body simulation

5.2.1 The dataset

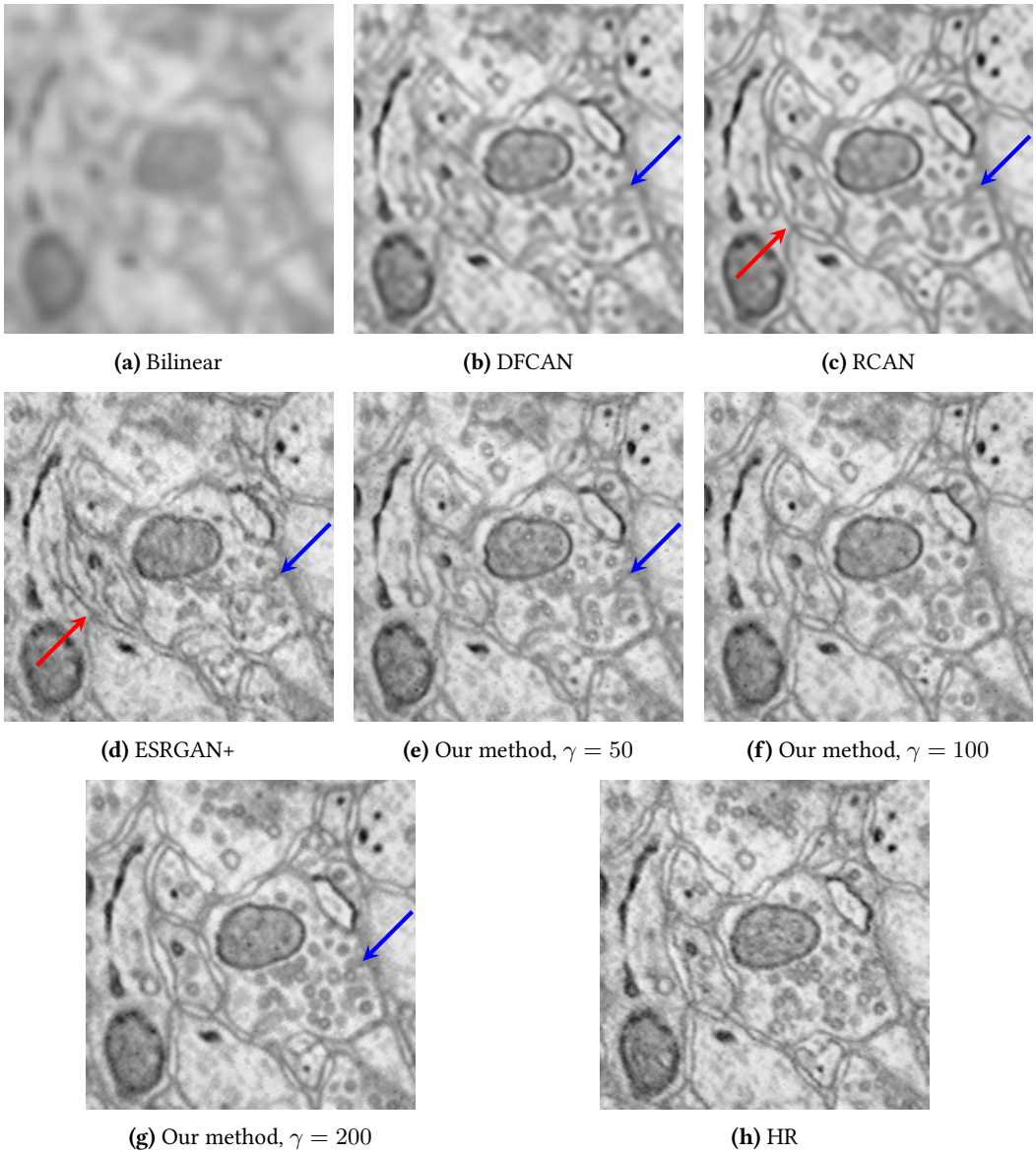
For the astrophysical data set we use an N-body simulation with the characteristics given in Table 5.2. The gravitational evolution was carried out with an updated version of `L-Gadget3` [46, 47]

We take this simulation and create N 2D-images of 200×200 Mpc$^2/h^2$ (remember 1Mpc $\approx 3.086 \times 10^{19}$km and h refers to the reduced Hubble parameter as stated in Table 5.2) by slicing the 3D box at random points along the axes. Each pixel represents the overdensity of that small region of the volume, $\delta_m(x, y, z) = (\rho_m(x, y, z) - \bar{\rho}_m)/\bar{\rho}_m$, where m refers to the matter component and ρ is the density. Thus each pixel is a slice of this δ.

100 images were created of this simulation, so we will use 80 images for training the network, and 20 for evaluation.

5.2.2 Preprocessing and data augmentation

In contrast with the EM dataset, the gravitational N-body simulation dataset is not made of actual images, whose pixels represent light. The pixel values of these images range from



(a) Bilinear **(b)** DFCAN **(c)** RCAN

(d) ESRGAN+ **(e)** Our method, $\gamma = 50$ **(f)** Our method, $\gamma = 100$

(g) Our method, $\gamma = 200$ **(h)** HR

Figure 5.4: Qualitative comparison of various SR methods in an image of the EM dataset with $4\times$ upscaling in each axis. Red arrows point to neuron membranes, and blue arrows point to vesicles.

0 to around 400, thus we need to transform these values in order to be able to run them through our network.

Firstly, we transform the values to a logarithmic scale, using 10-base log. This, on one hand, enhanced the contrast in the "darker" parts of the image, making the details present there more visible to the eye (see Figure 5.5) and easier to predict by a neural network. On the other hand, it lowers the range of values to a range closer to the desired $0 \leq x \leq 1$ range. As some values in the image are 0, it would result in some values being $-\infty$ after applying the logarithm, so we add 2×10^{-2} to every value of the image beforehand.

After taking the logarithm, we normalise the images to a $[0, 1]$ range, using the same maximum and minimum values for all images and all SR methods for consistency.

Parameter	Value
Volume	$(512\ \text{Mpc}/h)^3 \approx (1.67 \times 10^9\ \text{lightyears}/h)^3$
Particle number	1536^3
Mass per particle	$0.32 \times 10^{10} M_{\text{sun}}/h$
Ω_{m}	0.30964
Ω_{b}	0.04897
Ω_{Λ}	0.69036
h	0.6766
n_{s}	0.9665

Table 5.2: Parameters defining the N-body simulation. Mpc stands for "Megaparsecs" and M_{sun} for solar mass. Ω's stand for the ratio of the density of each component of the Universe to the critical density (density that would make the Universe flat): Ω_{m} refers to the amount of cold dark matter, Ω_{b} to the amount of "baryonic" matter (matter that composes dust, stars, planets,...), and Ω_{Λ} to the amount of "dark energy". The reduced Hubble constant h indicates the rate at which the universe is expanding in units of 100 km/s/Mpc. n_{s} is the so called spectral index.

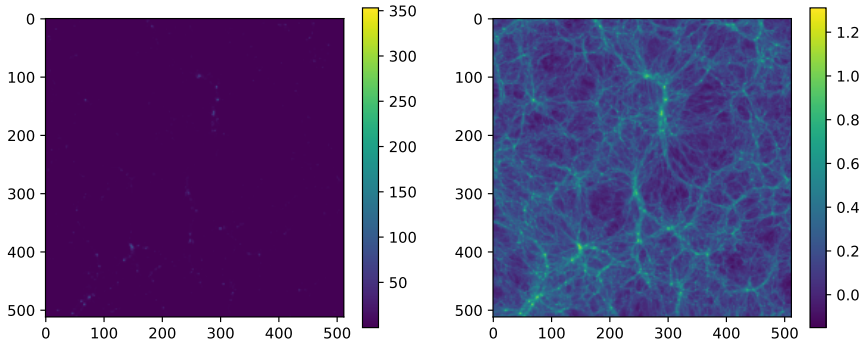


Figure 5.5: Sample visualisation of the gravitational N-body simulation dataset before and after the logarithm is applied to the values of the image.

When this transformations are done, we create the LR images of the desired size by taking the local mean of the pixels we want to reduce. So, when making images with a $4\times$ factor, we take the average of each 4×4 block of the original image and set the mean of those values as the value in the LR image.

For data augmentation, similarly to the EM dataset, we apply a random rotation to the images, with angles chosen from $\{0, 90, 180, 270\}$. We also flip the images in both the x and y axes with a probability of 0.5.

5.2.3 Evaluation metrics

In the gravitational N-body simulations dataset we also used some of the more widely used evaluation metrics in the literature, both full-reference and no-reference. Similarly to the EM version, we used PSNR, SSIM and PI for our evaluation metrics, which help us compare the results with other methods in the literature. These metrics are designed for images which have a fixed value range, therefore we calculate these metrics *after* applying

the logarithm and normalising the values.

To also evaluate how well the algorithm reconstructs the details of the original image in all of the spatial frequencies or wavenumbers, we also calculate the power spectrum of the image. The power spectrum is mathematically defined as the spherically averaged mean squared amplitude of the coefficients of the Fourier transform of the density field. Power spectra are widely used in dark matter simulations [48, 49, 50], and they tell how much the image varies at different wavenumbers. If there are more smaller details in the image, the amplitude of the power spectrum in the higher wavenumbers will be higher. This is very important as lowering the resolution of an image removes smaller details first.

After getting the power spectrum of both the true image and the generated image, we can compute the MSE of those spectra to get a value we can easily and objectively compare.

As you will see in the next section, the values of this MSE are quite high, so, as well as the raw MSE, we also provide a score based on this error, following this formula:

$$s = 1 - \frac{\text{MSE}(P(k)_{hr}, P(k)_g)}{\text{MSE}(P(k)_{hr}, P(k)_{lr})}, \tag{5.1}$$

where $P(k)_{hr}$, $P(k)_{lr}$, $P(k)_g$ are the power spectrum of the HR image, the power spectrum of the LR image upsampled to the same size as HR using nearest-neighbour, and the power spectrum of the generated image, respectively. Therefore, a score of 1 would be the perfect reconstruction, and values below 0 means they performed worse than the LR image directly.

5.2.4 Results

All the experiments presented in this section were done with a downsampling factor of $4\times$ in each axis. ESRGAN+, RCAN and DFCAN were used for comparison. Our method was trained with γ values of 10, 50 and 100. For this dataset, a higher learning rate in the critic was required for it to converge, especially in the $\gamma = 10$ test. More details about the hyperparameters used can be found in Table 2.

5.2.4.1 Quantitative analysis

The metrics of the gravitational N-body simulation dataset in various SR methods are shown in Table 5.3.

Similar to what we saw in Section 5.1.4, the methods that only use a pixel loss for their loss function, L1 in this case, obtained the best results in regards to PSNR and SSIM metrics. Between RCAN and DFCAN, the former performed slightly better with 29.4dB and 0.836 in PSNR and SSIM respectively, but the scores are really close.

On the other end, there are the GAN methods that include adversarial loss in their loss functions. The method that had the worst performance among them, and overall, was ESRGAN+, with PSNR and SSIM values of 25.85dB and 0.3925. In the case of our method, these values went up as the value of the hyperparameter γ increased, with values ranging from 25.89dB to 26.93dB in PSNR, and from 0.4947 to 0.5690 in SSIM.

Taking a look at the perceptual index, GAN methods performed better than RCAN and DFCAN. The best scoring method was ours, with hyperparameter γ set to 50 and a score of 2.7024, followed by the $\gamma = 10$ variant with 2.7103.

	Loss function	PSNR	SSIM	PI	PS MSE	Rel. MSE
Reference	-	∞	1	2.6566	0	1
LR	-	27.34	0.7843	11.9682	6.2746×10^{13}	0
Our method, $\gamma = 100$	Adversarial + L1	26.93	0.5690	2.9362	$\mathbf{4.3991 \times 10^{12}}$	**0.9298**
Our method, $\gamma = 50$	Adversarial + L1	26.61	0.5469	**2.7024**	8.1325×10^{12}	0.8703
Our method, $\gamma = 10$	Adversarial + L1	25.89	0.4947	2.7103	5.2181×10^{13}	0.1683
ESRGAN+	Adversarial + VGG + L1	25.85	0.3925	2.9873	1.6163×10^{14}	-1.5759
RCAN	L1	**29.40**	**0.8360**	5.9558	2.3942×10^{13}	0.6184
DFCAN	L1	29.32	0.8338	5.8925	3.7357×10^{13}	0.4046

Table 5.3: Comparison of the performance of various SR models in the Gravitational N-body simulation dataset. Metrics shown are PSNR, SSIM, PI, MSE of the power spectra, and relative MSE of the power spectra. PSNR, SSIM and PI are calculated *after* taking the logarithm and normalising the values to $[0, 1]$. Hyperparameters used in these experiments are shown in Table 2.

Finally, we need to take a look at the power spectra of these resulting images. The LR image, when upscaled to the HR size with nearest neighbour, produced a power spectrum with a MSE of 6.2746×10^{13} with respect to the spectrum of the original HR image. We will use this as a baseline value for the relative MSE of the power spectra. As the raw values are very high, we will only compare the relative values, but they are still shown in Table 5.3.

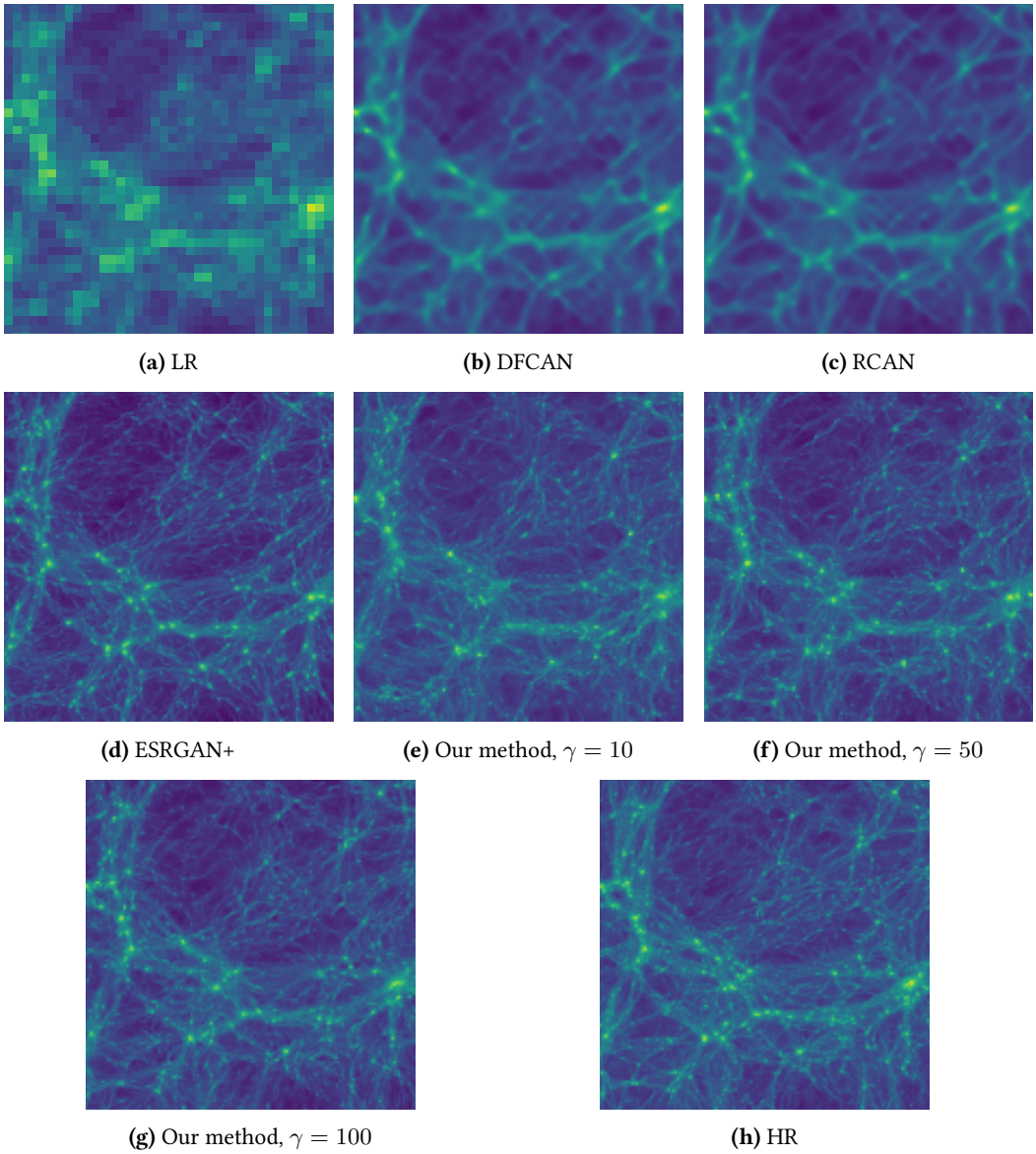
Among the ones we tested, the better performing algorithm was our method, in the $\gamma = 100$ variant, with a relative MSE between its power spectrum and the HR's power spectrum of 0.9298. The second best score also came from our method, in this case with the hyperparameter $\gamma = 50$, and a score of 0.8703. The next best scoring methods are RCAN and DFCAN, with scores of 0.6148 and 0.4046 respectively. After those is our method with the hyperparameter $\gamma = 10$, scoring 0.1683. Finally, ESRGAN+ had the worst score among the ones we tested, with a value of -1.5759. This means that its power spectrum was further from the original HR spectrum than that of the LR image. A graph of the power spectra can be found in Figure 5.7.

5.2.4.2 Qualitative analysis

If we do a qualitative analysis of the results (see Figure 5.6), the results share similar characteristics and artifacts to those in the EM dataset (Section 5.1.4.2). In DFCAN (Figure 5.6b) and RCAN (Figure 5.6c), the images look blurry and washed out and, as a consequence, many smaller, higher frequency details in the "darker" parts of the image are not present. In contrast, the other GAN-based methods, ESRGAN+ and our method, produce sharper images that have a realistic look: they are different to the HR version, but without the reference they could all be plausible reconstructions.

Among these GAN-based methods it is difficult to discern significant differences in plain sight, so we will compare their power spectra instead.

Looking at the power spectra in Figure 5.7, we can see that RCAN and DFCAN have a significantly lower spectrum in the high-frequency areas. This drop can be verified with the images in Figure 5.6, where these algorithms can't reproduce higher frequency details correctly.



(a) LR **(b)** DFCAN **(c)** RCAN

(d) ESRGAN+ **(e)** Our method, $\gamma = 10$ **(f)** Our method, $\gamma = 50$

(g) Our method, $\gamma = 100$ **(h)** HR

Figure 5.6: Detailed view of an image from the gravitational N-body simulation dataset **(h)**, its LR downscaled version with a factor of $4\times$ **(a)**, and reconstructions using various SR algorithms **(b-g)**.

Another spectrum that stands out is ESRGAN+, which is lower than the original HR throughout the whole spectrum. In contrast, all other methods produce a similar spectrum to the HR in the lower frequencies. This could be because ESRGAN+ produces results with lower contrast, resulting in a lower general power. In fact, if we multiply the values in the image produced by ESRGAN+ by 1.1 and then reverse the normalization and logarithm preprocessing steps, the spectrum produced is more similar to the HR version, with slightly lower power in the lower frequency area, and higher power amplitude in the higher frequency area (see Figure 5.8). The average MSE of the spectra if we apply this transformation is 3.5688×10^{13}, with a relative value of 0.4312.

If we focus on our method, there are some differences in the spectrum with the change of the hyperparameter γ. When we train the model with $\gamma = 10$, the resulting images have

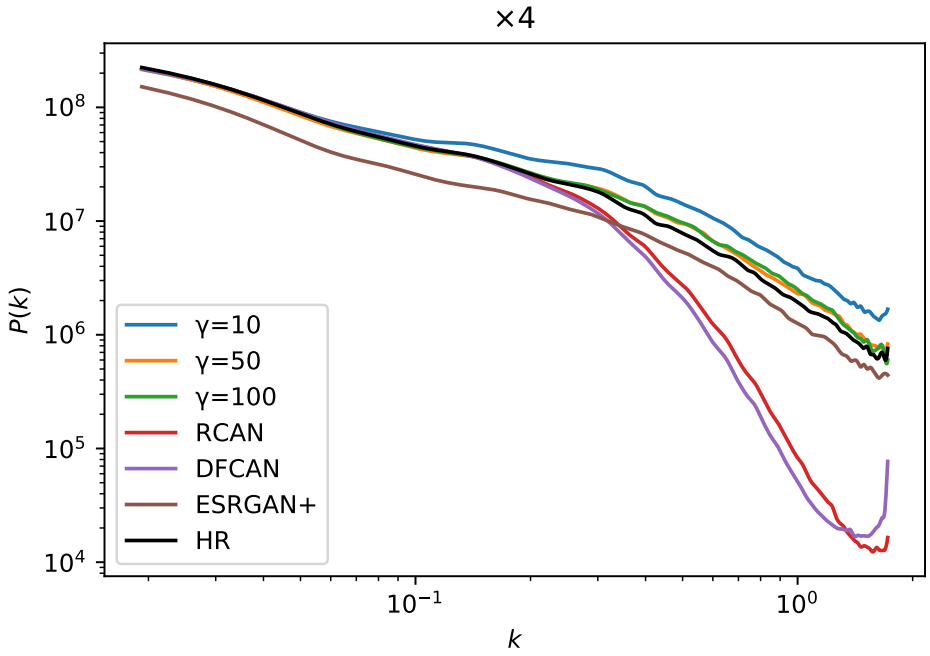


Figure 5.7: Power spectrum of the results of various SR algorithms on a single image from the gravitational N-body simulation dataset. Both axes are in logarithmic scale.

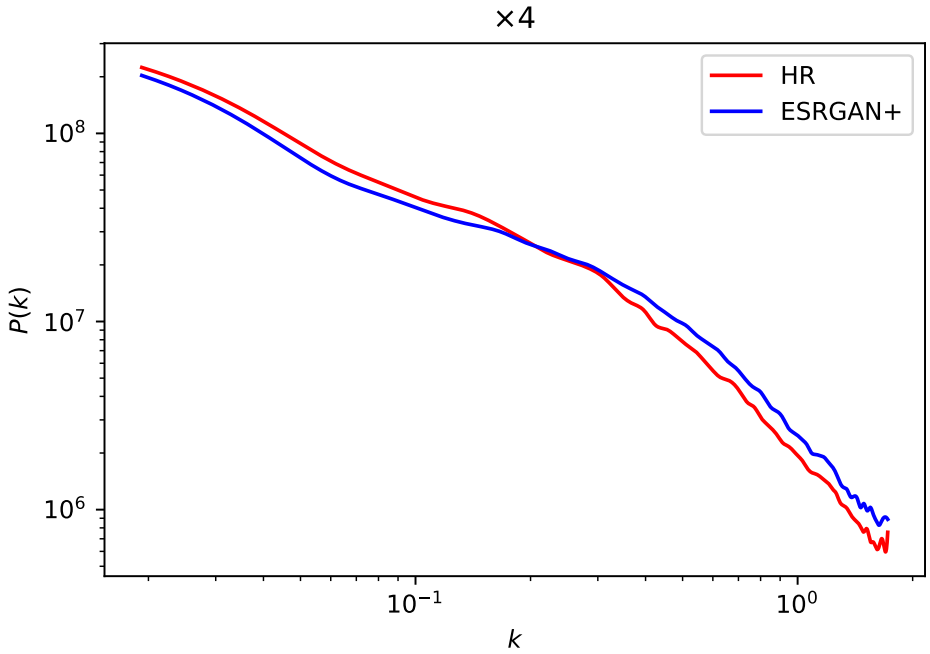


Figure 5.8: Comparison of the ESRGAN+ spectrum when the resulting image is multiplied by 1.1 before calculating the spectrum, and the original image's spectrum.

significantly more mid- and high-frequency detail than the original image.

The other 2 variants, with γ values of 50 and 100, are very similar, and both produce more high-frequency details than the original image, but it's less noticeable than that of the $\gamma = 10$ variant.

CHAPTER 6

Conclusions

Electron microscopy, and the Scanning Electron Microscope are an excellent tool that enables scientist to capture the shapes of objects in three dimensions and high resolution. However, due to physical limitations of the process, taking high resolution images of large objects take a very long time. Hence, there is a need for a tool that can enhance faster, lower-resolution scans and recover small details.

Gravitational N-body simulations are a widely used theoretical tool in astrophysics and cosmology - the study of the origins of the universe, its large-scale structures and dynamics, and the ultimate fate of the universe. However, similarly, in order to run N-body simulations of hundreds of millions of particles (usually required for state-of-the-art research) millions of CPU hours are needed. Moreover, new cosmological observations will require to have thousands of such simulations.

Thus, the aim of this project has been to research and create a super-resolution method that can upscale images taken with an electron microscope and 2D visualisations of the gravitational N-body simulations.

We have explored the state of the art in SR methods using deep convolutional neural networks. Two main types of methods were found that provide the most difference in the results they achieve: GANs and CNNs. GANs have a promising solution that provides more realistic results, but many current evaluation metrics that are used in research aren't able to measure that realism. There is a need for a reliable evaluation metric that scores this characteristic, and it is an interesting topic of research.

For its characteristics, we chose a GAN approach to our solution. We incorporated Wasserstein GAN technology in our solution in order to stabilise training and produce a more reliable solution. We also used residual learning for the generator, which improves training efficiency as the image we aim to super-resolve is similar to the target image.

In the EM images, we achieved results that correctly reproduce the details that are present in the original high-resolution image, without diverging too much from the original image in favour of realism, or producing an image that is blurry and washed out. We couldn't find any evaluation metric that could correctly evaluate this balance between perceptual quality and distortion, but experts ranked our method's results higher than other state-of-the-art methods. However, some of our results produced some artefacts

that would need further investigation in order to find the source of the issue. Finally, the experts concluded that none of the methods tested were as close to the original image as they would like, so further improvements can still be made.

In the gravitational N-body simulation images, GAN approaches are the most appropriate solution, as these were the only ones among those that were tested in this project that could effectively reproduce the high-frequency details present in the image. Both the state-of-the-art ESRGAN+ approach and our method produced realistic images that could be plausible reconstructions of the original image, albeit not being exactly equal. When comparing the power spectra, we noticed ESRGAN+ produced images with lower contrast, resulting in an overall lower power spectrum. Further investigation into the ESRGAN+ implementation could be done in order to find the source of this issue. On the other hand, our method produced results that were closest to the original image's power spectrum. In cosmological context, it is generally more important that the reconstructions have correct statistics such as the spectrum, rather than having a correct reconstruction on the MSE level. Therefore, GANs are the best option in this case.

Ultimately, our method produced comparable results to other state-of-the-art SR methods and, in some instances, the balance that we achieved between distortion and perceptual quality can produce images that are more useful than others, depending on the use case.

As future work, we would like to see how the model performs in real low resolution images. In this project, all the LR images were synthetically generated from their HR counterparts. Creating LR images in a real way (with a lower resolution EM scan, and gravitational simulations with a smaller number of particles) and seeing how they perform in the models trained with synthetic images would be insightful. On the other hand, it would also me interesting to create a user-friendly program that microscopy lab researchers and astrophysicists can use for training and testing the model, so they can more easily incorporate it into their workflow.

Appendix

EM experiment hyperparameters

Network	γ	Optimizer	LR_G	LR_D	Scheduler	Batch	Patch size	Down. Factor	Steps
15 gen. RB (1.4M + 1.6M)	200	Adam $\beta = (0.5, 0.9)$	0.0005	0.0005	OneCycleLR	8	128x128	4	108,000 (18,000 gen. steps)
15 gen. RB (1.4M + 1.6M)	100	Adam $\beta = (0.5, 0.9)$	0.0005	0.0005	OneCycleLR	8	128x128	4	108,000 (18,000 gen. steps)
15 gen. RB (1.4M + 1.6M)	50	Adam $\beta = (0.5, 0.9)$	0.0005	0.0005	OneCycleLR	8	128x128	4	108,000 (18,000 gen. steps)
RCAN 16 (0.98M)	-	RMSProp	0.001	-	OneCycleLR	8	128x128	4	21,600
DFCAN 16 (2.4M)	-	Adam $\beta = (0.9, 0.999)$	0.0003	-	OneCycleLR	8	128x128	4	21,600
ESRGAN+ (16M + 14M)	-	Adam $\beta = (0.9, 0.999)$	0.0001	0.0001	MultiStepLR	8	128x128	4	100,000

Table 1: Hyperparameters for various experiments of SR algorithms with the EM dataset. Numbers in brackets in the network column are the number of trainable parameters in the network, or the ones in the generator + in the discriminator/critic. LR_G and LR_D are the learning rates of the generator and the discriminator/critic (where applicable).

Gravitational N-body simulation experiment hyperparameters

Network	γ	Optimizer	LR_G	LR_D	Scheduler	Batch	Patch size	Down. Factor	Steps
15 gen. RB (1.4M + 1.6M)	100	Adam $\beta = (0.5, 0.9)$	0.0005	0.001	OneCycleLR	8	128x128	4	160,002 (26,667 gen. steps)
15 gen. RB (1.4M + 1.6M)	50	Adam $\beta = (0.5, 0.9)$	0.0005	0.001	OneCycleLR	8	128x128	4	160,002 (26,667 gen. steps)
15 gen. RB (1.4M + 1.6M)	10	Adam $\beta = (0.5, 0.9)$	0.0005	0.001	OneCycleLR	8	128x128	4	160,002 (26,667 gen. steps)
RCAN 16 (0.98M)	-	RMSProp	0.001	-	OneCycleLR	8	128x128	4	8,000
DFCAN 16 (2.4M)	-	Adam $\beta = (0.9, 0.999)$	0.0003	-	OneCycleLR	8	128x128	4	8,000
ESRGAN+ (16M + 14M)	-	Adam $\beta = (0.9, 0.999)$	0.0001	0.0001	MultiStepLR	8	128x128	4	100,000

Table 2: Hyperparameters for various experiments of SR algorithms with the Gravitational N-body simulation dataset. Numbers in brackets in the network column are the number of trainable parameters in the network, or the ones in the generator + in the discriminator/critic. LR_G and LR_D are the learning rates of the generator and the discriminator/critic (where applicable).

Bibliography

[1] Z. Wang, J. Chen, and S. C. H. Hoi. Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. See pages 1, 9, 10, 13, 15, and 18.

[2] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, Dec 2019. See page 1.

[3] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. NTIRE 2021 Learning the super-resolution space challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 596–612, June 2021. See page 2.

[4] What is Electron Microscopy? - UMASS Medical School, Jun 2018. See page 3.

[5] A.L. Eberle, S. Mikula, R. Schalek, J. Lichtman, M.L. Knothe Tate, and D. Zeidler. High-resolution, high-throughput imaging with a multibeam scanning electron microscope. *Journal of Microscopy*, 259(2):114–120, 2015. See pages 3, 4.

[6] Michele Trenti and Piet Hut. N-body simulations (gravitational). *Scholarpedia*, 3(5):3930, January 2008. See page 4.

[7] F. Bernardeau, S. Colombi, E. Gaztañaga, and R. Scoccimarro. Large-scale structure of the Universe and cosmological perturbation theory. *Physics Reports*, 367(1-3):1–248, September 2002. See page 4.

[8] Nathanaël Perraudin, Ankit Srivastava, Aurelien Lucchi, Tomasz Kacprzak, Thomas Hofmann, and Alexandre Réfrégier. Cosmological N-body simulations: a challenge for scalable generative models. *Computational Astrophysics and Cosmology*, 6(1):5, December 2019. See page 6.

[9] Siyu He, Yin Li, Yu Feng, Shirley Ho, Siamak Ravanbakhsh, Wei Chen, and Barnabás Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Science*, 116(28):13825–13832, July 2019. See page 6.

[10] Claude E. Duchon. Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology*, 18(8):1016–1022, August 1979. See page 9.

[11] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. See page 9.

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016. See pages 9, 10, and 11.

[13] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network, 2016. See pages 9, 10, and 11.

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. See pages 9, 13, 18, 22, and 23.

[15] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018. See pages 9, 13, 14, and 29.

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016. See pages 10, 12.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. See pages 10, 13, and 22.

[18] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. See pages 11, 12, and 29.

[19] Chang Qiao, Di Li, Yuting Guo, Chong Liu, Tao Jiang, Qionghai Dai, and Dong Li. Evaluation and development of deep neural networks for image super-resolution in optical microscopy. *Nature Methods*, 18(2):194–202, Feb 2021. See pages 11, 29.

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press. See pages 12, 16.

[21] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan, 2018. See page 13.

[22] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 PIRM Challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018. See pages 13, 18, 19, and 29.

[23] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. See page 14.

[24] Yin Li, Yueying Ni, Rupert A. C. Croft, Tiziana Di Matteo, Simeon Bird, and Yu Feng. AI-assisted superresolution cosmological simulations. *Proceedings of the National Academy of Sciences*, 118(19):e2022038118, May 2021. See page 14.

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN, 2020. See page 15.

[26] Doogesh Kodi Ramanah, Tom Charnock, Francisco Villaescusa-Navarro, and Benjamin D Wandelt. Super-resolution emulator of cosmological simulations using deep physical models. *Monthly Notices of the Royal Astronomical Society*, 495(4):4227–4236, May 2020. See page 15.

[27] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. See pages 15, 16, and 17.

[28] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. See pages 15, 17, 23, and 24.

[29] Jonathan Hui. GAN - Wasserstein GAN and WGAN-GP, Jun 2018. See page 15.

[30] Nikos Drakos. The earth mover's distance. *University of Edinburgh*, 1998. See page 15.

[31] Zhou Wang, Alan C. Bovik, and Ligang Lu. Why is image quality assessment so difficult? In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–3313–IV–3316, 2002. See page 17.

[32] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. See page 18.

[33] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16, 2017. See pages 18, 29.

[34] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. See pages 18, 29.

[35] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. See pages 18, 19.

[36] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry Schumaker, editors, *Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. See page 19.

[37] Nathanaël Carraz Rakotonirina and Andry Rasoanaivo. ESRGAN+ : Further improving enhanced super-resolution generative adversarial network. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3637–3641, 2020. See page 19.

[38] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016. See page 18.

[39] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CoRR*, abs/1609.05158, 2016. See pages 21, 22.

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks. *arXiv e-prints*, November 2015. See pages 22, 23.

[41] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. See page 22.

[42] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. See page 22.

[43] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019. See page 24.

[44] Narayanan Kasthuri, Kenneth Jeffrey Hayworth, Daniel Raimund Berger, Richard Lee Schalek, José Angel Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis Raymond Jones, Mike Roberts, Josh Lyskowski Morgan, Juan Carlos Tapia, H. Sebastian Seung, William Gray Roncal, Joshua Tzvi Vogelstein, Randal Burns, Daniel Lewis Sussman, Carey Eldin Priebe, Hanspeter Pfister, and Jeff William Lichtman. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015. See page 27.

[45] Linjing Fang, Fred Monroe, Sammy Weiser Novak, Lyndsey Kirk, Cara R. Schiavon, Seungyoon B. Yu, Tong Zhang, Melissa Wu, Kyle Kastner, Yoshiyuki Kubota, Zhao Zhang, Gulcin

Pekkurnaz, John Mendenhall, Kristen Harris, Jeremy Howard, and Uri Manor. Deep learning-based point-scanning super-resolution imaging. *bioRxiv*, 2019. See page 27.

[46] Volker Springel. The cosmological simulation code gadget-2. *Monthly Notices of the Royal Astronomical Society*, 364(4):1105–1134, 12 2005. See page 32.

[47] Raul E. Angulo, Matteo Zennaro, Sergio Contreras, Giovanni Aricó, Marcos Pellejero-Ibañez, and Jens Stücker. The BACCO Simulation Project: Exploiting the full power of large-scale structure for cosmology. *Monthly Notices of the Royal Astronomical Society*, July 2021. See page 32.

[48] Daniel J. Eisenstein and Wayne Hu. Power spectra for cold dark matter and its variants. *The Astrophysical Journal*, 511(1):5–15, jan 1999. See page 35.

[49] Stephen D. Landy, Stephen A. Shectman, Huan Lin, Robert P. Kirshner, Augustus A. Oemler, and Douglas Tucker. The Two-Dimensional Power Spectrum of the Las Campanas Redshift Survey: Detection of Excess Power on 100 h^{-1} Mpc Scales. *The Astrophysical Journal*, 456(1), jan 1996. See page 35.

[50] Kevin M. Huffenberger and Uroš Seljak. Halo concentration and the dark matter power spectrum. *Monthly Notices of the Royal Astronomical Society*, 340(4):1199–1204, 04 2003. See page 35.