Diese Arbeit wurde vorgelegt am
Lehr- und Forschungsgebiet Informatik 8 (Computer Vision)
Fakultät für Mathematik, Informatik und Naturwissenschaften
Prof. Dr. Bastian Leibe

Masterarbeit

Super-resolution of Sentinel-2 Images with Generative Adversarial Networks

vorgelegt von: Run Zhang

Matr.- Nr: 383507

Aachen, den 10.03.2020

Erstgutachter:	Prof. Dr. Bastian Leibe
Zweitgutachter:	Prof. Dr. Morris Riedel
Berater:	Dr. Gabriele Cavallaro, Dr. Jenia Jitsev

Contents

List of Figures

List of Tables

List of Abbreviations

RS	Remote Sensing
GSD	Ground Sampling Distance
HR	High Resolution
LR	Low Resolution
GANs	Generative Adversarial Networks
CNNs	Convolution Neural Networks
MSI	Multiple Spectral Instrument
ROI	Region of interests
IR	Infrared
NIR	Near Infrared
SWIR	Short Wave Infrared
MWIR	Middle Wave Infrared
LWIR	Long Wave Infrared
VLWIR	Very Long Wave Infrared
ToA	Top of Atmosphere

BoA	Bottom of Atmosphere
SNR	Signal to Noise Ratio
DL	Deep Learning
HPC	High Performance Computing
JSC	Juelich Supercomputing Center
JSD	Jsensen-Shannon Divergence.
RSA	Residual Self-Attention.

List of Symbols

$\mathcal{S}_{2x}^{L1C}$	Model for 20m $\mapsto$ 10m super-resolution.
$\mathcal{S}_{6x}^{L1C}$	Model for 60m $\mapsto$ 10m super-resolution.
$\mathcal{S}_{2x}^{L2A}$	Model for 20m $\mapsto$ 10m super-resolution.
$\mathcal{S}_{6x}^{L2A}$	Model for 60m $\mapsto$ 10m super-resolution.
A	Set of 20m bands
B	Set of 10m bands
C	Set of 60m bands
$\mathcal{L}_G$	Loss function for optimizing a generator
$\mathcal{L}_D$	Loss function for optimizing a discriminator

Acknowledgements

First, I would like to thank my loving parents Baorong Zhang and Chunfeng Zhang. Thanks for supporting me to finish my study in these years. Education is the best gifts of all you have given to me.

Second, I am very grateful to Prof. Bastian Leibe and Prof. Morris Riedel. Thanks for issuing this project and working as my thesis examiner. Without you, this thesis would never have materialized. This is definitely the most challenging but also the most rewarding year during my studies.

Next, I cannot express my gratitude enough to my supervisors, Dr. Gabriele Cavallaro and Dr. Jenia Jitsev. Thanks for continuous feedback and ideas, keeping me on the track and clearing my doubts. Thanks for being an amazing person and a true inspiration for me. Without your persistent support and encouragement, I would never have finished this thesis successfully.

Then, thank Hans Hermann Voss-Stiftung for your financial support, it has made my life and study in Germany much easier. And thank my lab mates Rocco Sedona, Tomasz Zbudowski and all other colleagues. Thanks for sharing your life and ideas with me during this project. I have had a wonderful year in Forschungszentrum Juelich and have never had an opportunity to learn so much in one year.

Chapter 1

Introduction

Remote sensing (RS) has always been an important field of study. Especially with the development of aviation technologies and growing industrial demands in recent years, RS has become an increasingly popular field in the modern society. From thematic maps in smartphones, to weather forecasts or early warnings on natural disasters, RS has exceedingly influenced our everyday life. It is estimated that over 2 billion people on the planet are users of remote sensing data nowadays. Moreover, RS is the cornerstone of many essential industrial or scientific applications. For instance, RS plays a vital role in dealing with agricultural, ecological and socio-economic issues such as assessing urban growth or economic levels [PYB11, MA05], geological or climate changes surveying, terrain or atmosphere analysis [MB09, BBH+92], assessing ecosystem status and biodiversity [TSG+03, SHM+04], food risk monitoring or resource exploration [RVR11, BD05], *etc.*. Owing to the critical role and high potential benefits, numerous space agencies and commercial industries have continuously invested a lot in RS, and have made significant advances in earth system products and RS methodologies, which are bound to drive the development of all related disciplines in the future.

RS image sensors on-board satellites and airborne platforms are required to provide high-quality earth observation data. However, due to the limitation of sensor resolution, satellite orbital altitudes, space-ground communication bandwidth, *etc.*, the raw acquisitions are often of low resolution (LR) and low signal to noise ratio (SNR). One important indicator to measure the quality of a RS image is ground sampling distance (GSD), *i.e.*, the distance between two consecutive pixel centers measured on the ground, *e.g.*, a GSD of 10m means that one pixel in the image represents linearly 10m on the ground (an area of $10^2 m^2$). Small GSD means a finer representation with more useful information of the earth surface. However, operational satellites can not meet the growing GSD requirements of many new generation scientific applications. Therefore, it is urgent to propose a post-correction method to enhance the GSD of raw RS acquisitions, or improve their spatial resolution.

In these years, deep learning (DL) has regularly redefined what the state-of-the-art is in many scientific problems, such as image classification [KSH12], image semantic segmentation [LSD15], object detection [RHGS15], *etc.* In particular, the generative adversarial networks (GANs) [GPAM+14], rising in recent years, have provided a new learning framework with the help of game theory beyond the previous deep learning methods. Furthermore, unlike classical machine learning tasks, *i.e.*, classification or regression, GANs are extensively exploited to generate and resemble real-world data, *e.g.*, images, videos, audios or 3D shape *etc.*. Before the prosperity of DL, scientists adopted classical feature extraction and selection approaches to generate meaningful features. This has become more challenging with the rising data complexity and data volume, especially in the field of RS. For instance, compared with natural images, RS images have finer radiometric and spectral resolution, and because of having broader land coverage, a RS image always contains a wide variety of ground scenes. Therefore, it is more challenging to describe the hidden knowledge behind RS data with only handcraft features or rules. However, as written in the book *Deep learning* [GBC16], "Deep learning enables us to build complex concepts out of some simple concepts." DL makes it possible to learn complex representations and absorb hidden knowledge automatically by just feeding a well-trained model with raw data. For this reason, the main objective of this thesis is to verify the potentials of DL methods, in particular, GANs, on generating high-quality RS images.

Apart from the image quality, one more aspect to assess a RS image processing algorithm is its data scalability. Through utilization of satellites, we now have continuous monitoring of the entire world with short revisit period. The continuous proliferation and improvement of remote sensing platforms yield a high volume of the earth observation data, *e.g.*, the combined fleet of Sentinel-1, Sentinel-2, and Sentinel-3 (satellites in the European Copernicus space mission) produce an estimated data volume of $\simeq$20TB per day [1]. This data explosion makes both storage and manipulation of the data much more challenging. In addition, one essential reason why deep learning-based methods can surpass the non-learning based methods is that deep models can take advantage of big data and continues to improve accuracy long after non-deep learning algorithms reaching data saturation[2]. Therefore, the algorithm presented in this thesis is scaled up to high performance computing (HPC) systems installed at Juelich Supercomputing Center, so that the explosive growing RS data can be processed with a significantly improved speed and a deep model with better generalization can be trained with large amount of RS data.

[1] https://sentinels.copernicus.eu/web/sentinel/news/-/article/2018-sentinel-data-access-annual-report

[2] https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu

1.1 Problem Statement and Contributions

Different satellites, due to different launch time and missions, often have very distinct system configurations, thus the format of the obtained observation data also varies a lot. To simplify this problem, the super-resolution model proposed in this thesis is only designed for Copernicus Sentinel-2 mission, the detailed configuration of which is introduced in Chapter 2. In short, Sentinel-2 acquires multi-resolution and multi-spectral images. The problem studied in this paper is to super-resolve the multiple LR spectral bands (GSD=20m or 60m) to 10m GSD (the maximal sensor resolution of Sentinel-2), so as to obtain a high resolution (HR) data cube.

The main contributions in this thesis are as the following:

- A deep model based on self-attention mechanism and residual learning for the super-resolution of multi-resolution multi-spectral RS images is proposed. State-of-the-art performance is achieved on several evaluation metrics.

- The effects of adversarial losses on the super-resolution of large-scale multi-spectral RS observations are studied. Quantitative evaluation and visual assessment are applied to compare the impact of different GAN losses, including WGAN-GP, relativistic GAN, GAN with hinge loss, *etc.*.

- A comprehensive evaluation framework is proposed. In this thesis, our model and other comparison methods are evaluated by extensive experiments with many different settings, and to the best of our knowledge, we are the first to evaluate a learning-based Sentinel-2 RS super-resolution model with both synthetic and consistency properties on both Level-1C and Level-2A Sentinel-2 data. Furthermore, in addition to the common metrics, *e.g.*, RMSE, PSNR, SSIM *etc.*, we also propose a new brightness invariant image quality metric, bPSNR, to evaluate the super-resolved output.

- We scale up the process of model training and prediction to HPC clusters installed at JSC. Ring-reduction mechanism and synchronous data parallelism is applied to make the leaning process faster, and the effect of scaled learning rate and scaled mini-batch size is investigated on a large RS dataset. Experiments show that distribute learning can keep the performance intact while significantly speeding up the training and predicting process. The code of this thesis work is publicly available in this repository [3]. With this code, community can train their own distributed super-resolution or GAN applications with a significantly increased speed.

[3] https://gitlab.version.fz-juelich.de/cavallaro1/gan_superesolution

1.2 Thesis Outline

In Chapter 2, we introduce the background knowledge relevant to this project, including Convolution Neural Networks (CNNs), RS images, image super-resolution, and GANs. In Chapter 3, we summarize the related works in the context of GANs stability, natural image or RS image super-resolution. Chapter 4 formally defines the problem of the Sentinel-2 multi-resolution multi-spectral image super-resolution. Chapter 5 discusses the network architectures, the loss functions, and the distributed machine learning mechanism adopt in this thesis. In chapter 6, we start with the procedure of dataset preparation, and descriptions of the used computing resources and evaluation metrics. Then, we describe in detail the performance of our model. Finally, we show the benefits of scaling with distributed learning. In Chapter 7, we conclude all discoveries in this project, discuss the impact of our research and provide a roadmap for possible future directions.

Chapter 2

Background

This chapter provides the background information and knowledge pertaining to the work presented in this thesis. First, Section 2.1 introduces the differences between natural and RS images, configurations of the Sentinel-2 mission and Sentinel-2 MSI products. Next, Section 2.2 provides the basic idea of CNNs and four of their important variants. Then, a brief introduction into the world of GANs is given in Section 2.3. Finally, Section 2.4 presents the concept of super-resolution, the four paradigms when designing a super-resolution model, and some of its applications.

2.1 Remote Sensing Images

RS is the acquisition of information about an object or phenomenon without contact [LS76]. Generally, it refers to the use of satellite- or aircraft-based sensing technologies to detect and record objects on the Earth. According to propagated signals, *e.g.*, electromagnetic radiation, RS can be categorized into active and passive. In this thesis, we only focus on images acquired by passive RS sensor.

- Active RS: The sensor emits radiation in the direction of the target to be investigated, then detects and measures the radiation that is reflected or backscattered from the target.

- Passive RS: The sensor only gather radiation that is emitted or reflected by the target.

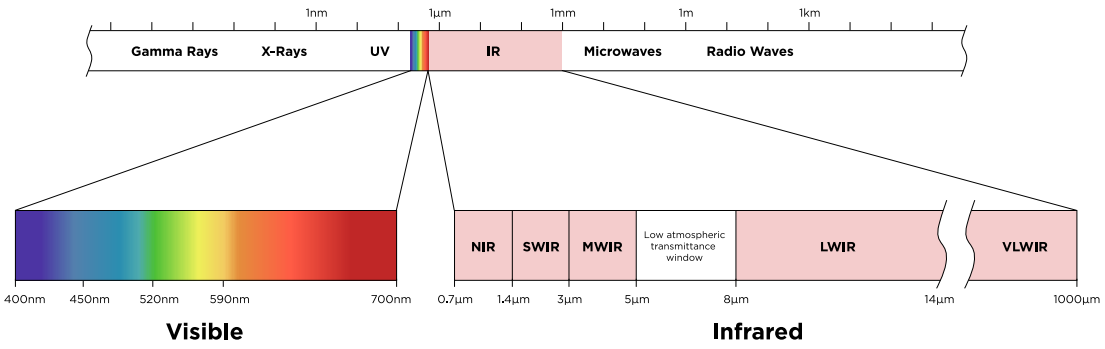


Figure 2.1: Electromagnetic spectrum [IEO19]. The wavelength of visible lights ranges from 400nm to 700nm. The electromagnetic in the range of 700nm to 1mm is called by infrared (IR), including near infrared (NIR), short wave infrared (SWIR), middle wave infrared (MWIR), long wave infrared (LWIR) and very long wave infrared (VLWIR).

2.1.1 Comparison with natural images

In general, digital images can be visually assessed by comparing their size, color fidelity, brightness, sharpness *etc.*. To make the assessment more objective, the following five resolution are adopted to evaluate the quality of an image.

1. **Pixel resolution**: the height and width of an image. No conclusions on pixel resolution can be made to distinguish between natural and RS images, because pixel resolution highly depends on configurations of the image sensor.

2. **Spatial resolution**: It is determined by the size of a square area corresponding to a pixel recorded in an image. In general, natural images have much higher spatial resolution because of shorter shooting distance. In the field of RS, it is also known as GSD.

3. **Spectral resolution**: the number of spectral bands recorded in a RS image. A natural color image has 3 spectral bands (RGB) and a natural gray image has only 1 band. For RS images, the spectral resolution depends on the type of sensor instruments. A panchromatic image only contains 1 band, a multi-spectral image has four or more spectral bands, and a hyper-spectral image can have hundreds of bands.

4. **Radiometric resolution**: the number of different intensities of radiation that the sensor is able to distinguish. In general, in a natural image, the value of each pixel in each channel can be represent with 8 bits. In a RS image, it can range from 8 to 14 bits.

5. **Temporal resolution**: It refers to the time elapsed between two consecutive RS acquisitions over the same region of interests (ROI).

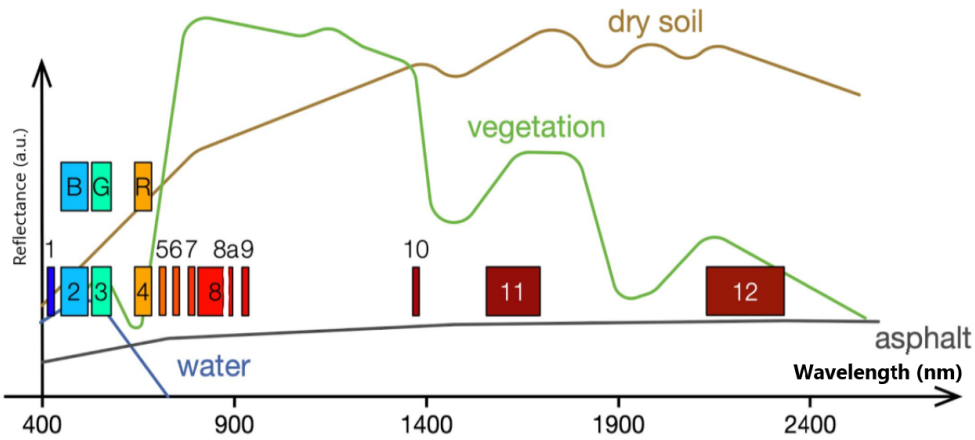


Figure 2.2: Reflectance curve of dry soil, water, vegetation and asphalt [LR18], which can be interpreted as their spectral signatures. The horizontal axis is the electromagnetic wavelength and the vertical axis is the reflectance intensity. The wavelength interval of 13 Sentinel-2 spectral bands are also marked in the coordinate, bands (4, 3, 2) are RGB bands respectively.

One more difference between natural and RS images is that RS image sensors are able to measure the electromagnetic waves in a wider spectrum. A general color digital camera is only sensitive to visible light, whereas a RS camera can also record IR, including NIR, SWIR, MWIR and LWIR, see Figure 2.1. Furthermore, a RS image sensor can filter more narrow wavelength intervals thus sensing images with more finer spectral bands. Wider and finer spectral information is a core characteristic of RS images. For instance, by comparing the signal intensity in each band, a spectral distribution (a.k.a., spectral signature) can be generated to identify objects on the Earth, because different materials define different spectra (*i.e.*, electromagnetic reflection), see Figure 2.2.

2.1.2 Satellites: Sentinel-2A&2B

In the last decades, many RS missions were started, *e.g.*, Quickbird $1-2$, Worldview $1-4$, Landsat $1-8$, Gaofen $1-11$, Sentinel $1-3\&5p$ *etc.* They all have different technical configurations and generate heterogeneous RS data. A good review [TJ16] for the current RS platforms is available. Understanding how to jointly leverage these complementary sources from different satellites in an efficient way is still a challenge in RS, (*i.e.*, multi-temporal and multi-source data fusion). But in this thesis, we only focus on images acquired by Sentinel-2A&2B due to its relatively high spatial, spectral and temporal resolution.

Sentinel-2 is an Earth observation mission, part of the European Space Agency's Copernicus program, comprising a constellation of two polar-orbiting satellites placed in the same sun-synchronous orbit and phased at 180 degree to each other. With a wide swath width of 290 km, the Sentinel-2 mission can monitor large portions of the Earth's surface. Its coverage limits ranges from latitudes 56 degree south to 84 degree

Band	B1	B2	B3	B4	B5	B6	B7	B8	B8a	B9	B10	B11	B12
Center wavelength(nm)	443	490	560	665	705	740	783	842	865	945	1380	1610	2190
Spatial resolution(m)	60	10	10	10	20	20	20	10	20	60	60	20	20
Band width	20	65	35	30	15	15	20	115	20	20	30	90	180
Purpose	AD	B	G	R	VC	VC	VC	NIR	VC	WV	cirrus	SIC	SIC

Table 2.1: The center wavelength, bandwidth, spatial resolution and purpose of 13 Sentinel-2 spectral bands. Bands B5, B6, B7 and B8a are used for vegetation classification (VC); Bands B11 and B12 are for snow, ice, cloud discrimination (SIC); Band B8 is for near infrared (NIR); Band B1 is for aerosol detection (AD); Band B9 is for water vapour (WV).

north. The temporal resolution, *i.e.*, the revisit period, of the combined constellation is only five days and ten days for a single satellite.This enables the monitoring of land-cover and land-use dynamics. A multi-spectral instrument (MSI) is mounted on each satellite, which works passively by collecting the sunlight reflected by the Earth. The reflected light that is recorded is split by a filter and focused onto two separate focal plane assemblies within the MSI, one for Visible and Near-Infra-Red (VNIR) bands and another for SWIR bands. The spectral separation of each band into individual wavelengths is accomplished by stripe filters mounted on top of the detectors. Thus, the Sentinel-2 satellite mission can deliver multi-spectral imagery (MSI products) with 13 spectral bands, see Figure 2.2 and Table 2.1. Furthermore, the MSI products delivered by Sentinel-2 have multiple processing levels (level 0, 1A, 1B, 1C and 2A, see Figure 3.1) and this thesis only investigate the super-resolution of data of format level-1C and -2A.

A common problem in RS products is the trade-off between spatial, spectral and temporal resolutions. Due to the multiple spectral bands and high revisit frequency, Sentinel-2 can produce high volume of data every second but the down-link bandwidth to ground station is comparatively the bottleneck. In order to relief the ground-space bandwidth burden, the spatial resolution of some spectral bands is degraded before transmission. Therefore, the Sentinel-2 MSI products are images composed of bands with multiple resolutions, see Table 2.1.

2.2 Convolution Neural Networks

Convolution neural network (CNN) is an popular deep learning framework which has made major advances in many visual computing tasks, including object classification, tracking, segmentation *etc.*. And it has also been extensively exploited in RS image processing. Next we will briefly explain the main idea of CNN and some of its variants.

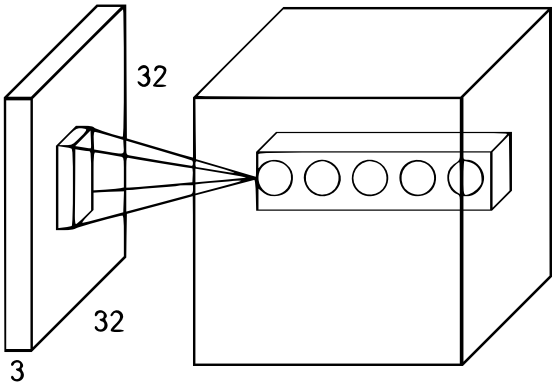


Figure 2.3: Illustration of convolution operation in a CNN. The left is the input to a convolution layer which is a tensor of shape (32, 32, 3), where 32×32 are height and width and 3 is the number of input channels. Each filter is only connected to a local patch. The right is the output feature map. The number of output feature map channels is determined by the number of filters in the convolution layer.

2.2.1 Four key ideas of CNN

CNN relies on local connections, shared weights, pooling and the use of many layers [LBH15]. A CNN is the composition of a number of convolution, activation and pooling layers. The input to a convolution layer is a feature map of one or more channels, shown in Figure 2.3. Each convolution layer has a set of weights called a filter bank and each filter is connected to a local patches, also known as receptive field [HW62]. Each filter receives input from a receptive field and output the weighted sum. By sliding the receptive field, each filter can scan the entire input feature map. Since the filters share their weights and bias over all shifts, they can detect the same pattern at different locations. The reason for this local connection is because local group of values in a reception field are often highly correlated and its local statistics are invariant to location.

Between convolution layers there can be pooling layers, which sub-sample the feature maps according to some function, *e.g.*, maximum value (max-pooling). This simplifies the feature maps and significantly reduces the number of parameters in the network as the number of convolution layers grows. The main benefit of sharing weights and feature pooling is to alleviate the computation burden, *i.e.*, fewer parameters to be learned and smaller feature maps. This enables the construction of deeper networks that can learn faster without sacrificing performance.

The main hyper-parameters of a convolution layer are the following:

1. **Kernel size**: Kernel size defines the size of reception field, *e.g.*, (3,3) is a common kernel size in a 2D convolution layer. For 3D convolution, the reception field is a 3D cube, typically with the size of (3,3,n), where n is the depth of the reception field.

2. **Stride**: Stride defines the step size when a filter traverses the input feature map. A stride larger than 1 is usually used to down-sample an input feature map.

3. **Padding mode**: Padding mode defines how the boarder of feature map is handled. A padded convolution will keep the size of output the same with input, whereas a unpadded convolution will crop away some of the boarders if it is not a 1×1 convolution.

4. **Number of filters**: A convolution layer take the convolution of input feature maps with some filters and output a feature cube with multiple channels. The number of output channels is determined by the number of filters in this convolution layer.

2.2.2 Variants of CNN

Despite that most CNN layers have almost the similar basic components, numerous variants that are suitable to various tasks has been proposed [GWK+18]. Below are some popular variants in the field of visual computing.

2.2.2.1 Dilated convolution

Dilated convolution introduces one more hyper parameter called dilation rate, which defines the size of blank spaces between the values in a filter. For instance, a 3×3 filter with a dilation rate of 2 has the same field of view as a 5×5 filter, *i.e.*, a 5×5 filter with every second column and row deleted, see Figure 2.4a.

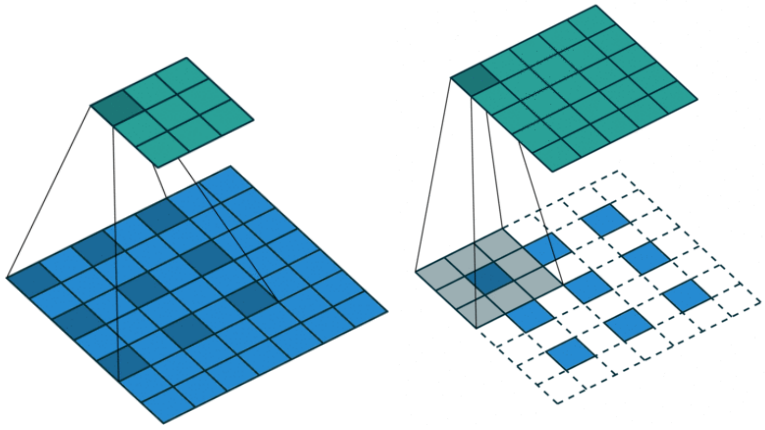
Note the number of all parameters in a convolution layer is invariant to the dilated rate. That means a dilated convolution layer can deliver a wider field of view at the same computational cost. It is particularly popular in the problem of real-time semantic segmentation [ZQS+18].

2.2.2.2 Transposed convolution

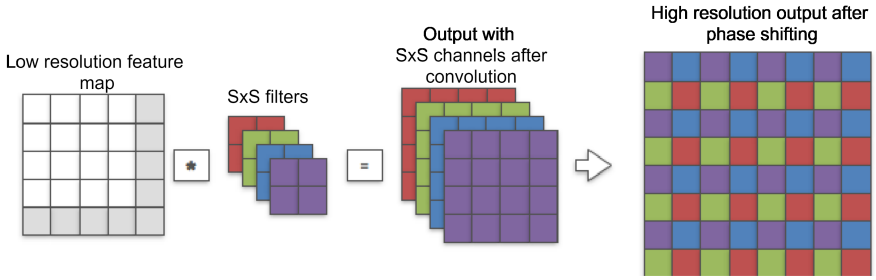
A transposed convolution layer carries out a regular convolution operation but reverts its spatial transformation. Below is a concrete example to show its main idea. Suppose that an image of 5×5 is fed into a regular convolution layer, where the stride is 2, the padding is activated and the kernel size is 3×3, it is easy to find that this layer will yield a feature map of 3×3. The transposed convolution can do is to reverse this process. Unlike strides in a regular convolution layer, strides in a transposed convolution layer determines its up-sampling scale. When the stride is set to be 2 and fed with an input

of 3×3, a transposed convolution layer will first perform padding around each pixel, then conduct a regular convolution which yields an output of 5×5 , see Figure 2.4b.

Due to the ability of feature map up-sampling, transposed convolution are extensively exploited in image super-resolution. But transposed convolution have a natural disadvantage. The uneven overlapping when sliding filters over the entire input feature map may cause a checkerboard pattern of artifacts in the output [ODO16]. Note that these artifacts can be alleviated by using filters of size divisible by stride, or always using more than one transposed convolution layers in a series fashion, but they can not be canceled out totally.



(a) Dilated convolution [DV16] **(b)** Transposed convolution [DV16]



(c) Sub-pixel convolution [SCH+16]

Figure 2.4: Illustration of three variants of convolution layer. In both (a) and (b), the blue grids in the bottom are inputs and green grids on the top are outputs. In (a) dilated convolution, the kernel size is 5×5 but only 3 (darker blue) of them are effective. In (b) transposed convolution, the white dotted grids padding. In (c) sub-pixel convolution, the input is a Low resolution feature map of 4×4 where the gray grids are padding. The aim of (c) is to up-sampling the input with the scale of 2 ($S = 2$). After the regular convolution, the output with 2×2 channels (illustrated with different color) are resize to a high resolution feature map by depth (phase) shift.

2.2.2.3 Sub-pixel convolution

Apart from transposed convolution, sub-pixel convolution [SCH+16] is another common approach to up-sample feature map and execute image super-resolution. Essen-

tially, a sub-pixel convolution layer has the same operation with a regular convolution layer, except the subsequent feature map reshaping (a.k.a. phase shift). As shown in Figure 2.4c, after performing a regular convolution with $S \times S$ filters (S is up-sampling scale), the output feature map is resized into a up-scaled image by shifting multiple feature maps to the same depth.

In sub-pixel convolution, checkerboard artifacts can be significantly alleviated but can not totally removed. The sub-pixel convolution is more efficient than the transposed convolution since it calculates more convolutions on smaller feature maps and reshapes the output in the last step. This avoids the use of extra informationless padding in-between pixels and can accelerate the computation.

2.2.2.4 Separable convolution

Separable convolution splits the convolution with filters into multiple steps. Supposing that we have a convolution operation $y = x \circledast k$, where x, y, k is input feature map, output feature map, filter respectively. If k can be calculated by convolution of multiple sub-filters, *e.g.*, $k = k_1 \circledast k_2$, then y can also be calculated by $y = (x \circledast k_1) \circledast k_2$ according to the commutative property of convolution $(f \circledast g) \circledast h = f \circledast (g \circledast h)$. In this case, y is called a separable convolution.

Depth-wise separable convolution is a representative example. It performs a spatial convolution first by keeping the channels separate. Afterwards, a depth-wise convolution is applied. Assume a regular convolution layer with 32 3×3 filters is applied to an input with 16 channels, each of these 16 channels will be traversed by the 32 kernels, resulting in 512 (16×32) feature maps. Next, we merge feature maps out of every filter by adding them up. Since we can do it 32 times, we get 32 feature maps in the end. On the same example what happens in a depth-wise separable convolution is, we traverse the 16 channels with a single 3×3 filter each, giving us 16 feature maps. Then, before merging anything, we traverse these 16 feature maps with 32 1×1 filters each and only then start to merge the feature maps from each 1×1 filter together. This results in a much efficient convolution layer with only 656 ($16 \times 3 \times 3 + 16 \times 32 \times 1 \times 1$) parameters opposed to the 4608 ($16 \times 32 \times 3 \times 3$) parameters in the regular convolution layer above. Because it can significantly reduce the model size, it has been widely used in deep models designed for mobile devices, *e.g.*, MobileNets [HZC+17].

2.3 Generative Adversarial Networks

Generative adversarial networks(GANs) are continuously soaring to greater and greater popularity. There has been a tremendous increase in the number of papers being published on GANs over the last several years. Further, GANs have been applied to a

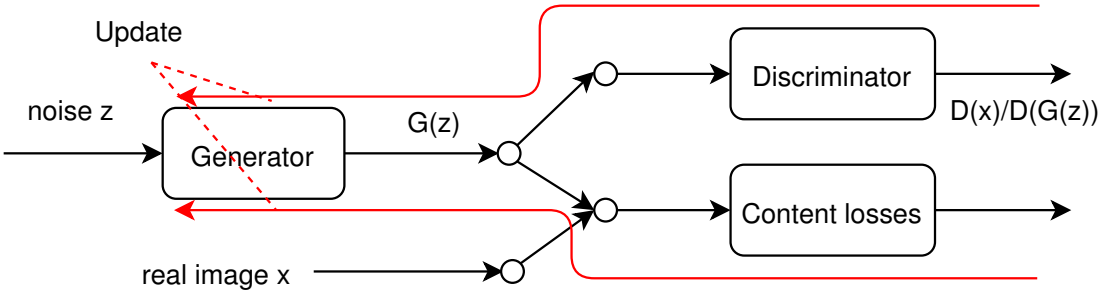


Figure 2.5: The process of updating the generator in a GAN. The architecture of GANs varies a lot. This figure illustrates a GAN architecture widely used in image super-resolution. As the red arrow shows, the generator G is penalized by both losses from Discriminator D and content losses. It means the generator G learns not only to fool the discriminator D, but also to minimize the distance between its output $G(z)$ and the real image x.

great variety of problems, *e.g.*, image generation [ZGMO18], text to image synthesis [RAY+16], 3D shape [WZX+16] or videos generation [VPT16] and in addition to data generation, GANs can also be used to do style transformation [WXWT18], robots imitation learning [HCS+17] *etc.*. We can not deny that GANs have set off a new round of research boom in the field of deep learning.

Goodfellow *et al.* [GPAM+14] first proposed GAN. It is a deep neural network that learns to generate data looking like real-world data. Given a set of training data, GANs can be trained to estimate its underlying probability distribution. Conceptually, distributions means the latent features of the training data and a deep model is expected to approach it automatically during the training process with lots of optimization iterations. Then according to the learned probability distribution, GANs can generate data which is not present in the original training dataset.

GAN is a combination of two deep networks that compete with each other. Given some random noise $z \sim p_z$, the generator $G(z)$ has to generate fake data resembling a real data distribution $x \sim p_r$. A discriminator D, is a binary classifier that tries to distinguish between the real data x and the fake data $G(z)$ generated by the generator G.

2.3.1 Alternating updates inside a GAN

This section formally defines how a GAN works. The discriminator of a GAN outputs a value $D(x) \in [0, 1]$ indicating the chance that its input is real. To improve the ability of recognizing real data x, the discriminator has to maximize $D(x)$. Similarly, $1 - D(G(z))$ has to be maximized to improve the ability of recognizing fake data $G(z)$. Therefore, the objective function of a discriminator can be given by

$$\max_D V(D) = \mathbb{E}_{x \sim p_r} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] .$$

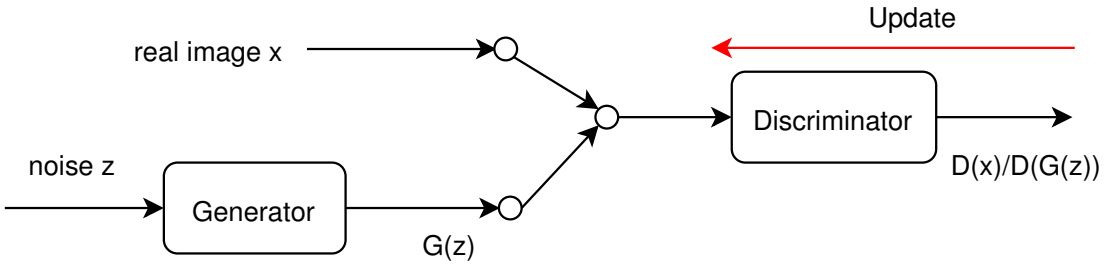


Figure 2.6: Illustration of updating the discriminator in a GAN. The discriminator D learns to distinguish the generated data $G(z)$ and the real data x. So in each iteration, the discriminator is updated to enlarge the difference between $D(x)$ and $D(G(z))$.

On the generator side, the objective function aims to generate data with highest $D(G(z))$ to fool the discriminator, so its objective function can be expressed with

$$\min_G V(G) = \mathbb{E}_{z\sim p_z}\left[\log(1 - D(G(z)))\right].$$

With these two step, the GAN V can be defined as a *minmax* game shown as follows,

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_r}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z}\left[\log(1 - D(G(z)))\right].$$

For real application problems, we can not get all possible values of the distributions p_x or p_z for training, *i.e.*, we can only provide some samples from p_x or p_z. Further, a model is usually optimized by minimizing instead of maximizing losses. So the objective function of discriminator D and generator G used in real applications are often transformed to $\hat{J}$ and $\hat{L}$.

$$\hat{J}(D) = -\frac{1}{N}\sum_{n=0}^{N}\left[\log D(x_n)\right] - \frac{1}{N}\sum_{n=0}^{N}\left[\log(1 - D(G(z_n)))\right]$$

$$\hat{L}(G) = \frac{1}{N}\sum_{n=0}^{N}\left[\log(1 - D(G(z_n)))\right]$$

The training procedure of a GAN consists of alternating stochastic gradient descent update to the discriminator and generator as shown in Figures 2.5 and Figure 2.6. The generator tries new tricks all the time to provide fake data and fool the discriminator. The discriminator also needs to update all the time to keep up with the generator. Ultimately, if everything goes well, the generator learns the true distribution of the training data and becomes good at generating real-looking data (*i.e.*, the discriminator can no longer distinguish between real data and generated data).

GANs can be more than a deep learning framework only used for generating data, since they are based on the concept of self-learning discriminator that can be applied

to many machine learning applications. The discriminator acts like a critic and can be exploited as an approach to provide a better self-feedback.

2.3.2 Optimal solution of a GAN

The optimal solution of a GAN is called *Nash equilibrium*, where $p_r = p_g$ (p_g is the learned distribution of generator). In *Nash equilibrium*, the discriminator cannot distinguish the real from the fake, *i.e.*, for any $x \sim p_r$, $z \sim p_z$, $D(x) = D(G(z)) = \frac{1}{2}$. The original GAN paper has given the detailed prove for *Nash equilibrium* is GAN's optimal solution. The main idea can be divided to two parts. First, supposing a generator is fixed, the optimal discriminator is:

$$D^\star(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}.$$

If the discriminator can reach this optimal point $D^\star$ in every iteration, the optimal value for $V(G, D^\star)$ is

$$V^\star(G, D^\star) = JSD(p_r, p_g) - 2\log 2.$$

where JSD refers to Jsensen-Shannon divergence. Since the Jensen-Shannon divergence has a unique minimum at $p_r = p_g$, so $p_r = p_g$ is GAN's optimal solution, where the output of generator perfectly replicates the real data distribution.

For now, learning-based methods typically use gradient descent to optimize a model. In case of that the loss landscape is global (or local) convex, a model can finally converge to a global (or local) optimal solution. However, gradient descent algorithms can not guarantee a GAN model to converge to the *Nash equilibrium*. This can be demonstrated by a trivial example. Considering two players a and b who control the value x and y respectively, player a wants to maximize the value of $x \cdot y$ while b wants to minimize it. A *Nash equilibrium* happens when one player will not change its action regardless of what the opponent may do. Obviously, $x = y = 0$ is the *Nash equilibrium*, where any opponent actions will not change the game outcome. But this is not the minimal value of $x \cdot y$ where a gradient descent algorithm will converge to.

2.3.3 Failure modes in GAN training

Although GANs have been successfully applied and achieved unprecedented results in many areas, it should not be ignored that a GAN is usually more unstable and difficult to train. Three common failure modes usually occur are as the following:

- **Diminished gradient**: During the training, a discriminator usually wins early against a generator. A too successful discriminator will break the balance in the

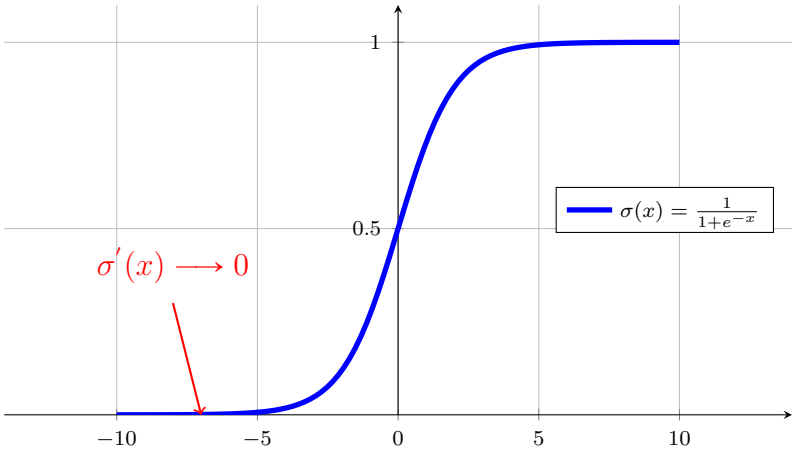


Figure 2.7: Illustration of sigmoid activation σ . When $\sigma(x)$ approaches 0, $\sigma^{'}(x)$ approaches 0, too.

game with a generator. In this case, the discriminator can not provide useful information to lead the generator to update. This can be shown by the gradient of a generator's loss function

$$\begin{aligned}\bigtriangledown_G \mathbb{E}_{z\sim p_z}\left[\log(1 - D(G(z)))\right] &= \frac{1}{1 - D(G(z))}(-D^{'}(G(z))) \\ &= \frac{1}{1 - \sigma(G(z))}(-\sigma^{'}(G(z))).\end{aligned}$$

The output of a discriminator D is probability. A discriminator is too good means $D(G(z)) \longrightarrow 0$ whatever the output of G is. Because D is usually calculated by Sigmoid σ, when $D(G(z)) \longrightarrow 0$, $D^{'}(G(z)) \longrightarrow 0$, see Figure 2.7. This makes the gradient of generator loss $\bigtriangledown_G \mathbb{E}_{z\sim p_z}\left[\log(1 - D(G(z)))\right]$ approach 0 and slows down the optimization process.

- **Mode collapse**: A generator can be restrained in a small subspace and only produces samples with low diversity. *e.g.*, with a plausible output that fools the discriminator, the generator learns to produce only this output. This is unacceptable because a GAN must be able to address the diversity of real-world data.

- **Non-convergence**: The generator and discriminator compete with each other for a *Nash equilibrium*, yet there is no theoretical guarantee of *Nash equilibrium* can be found. Sometimes the model parameters just oscillate, destabilize and never converge. This is elusive and can be caused by many possible reasons, *i.e.*, imbalance between the generator and discriminator, hyper parameter selections, poor training dataset *etc.*.

2.4 Image Super-resolution

Image super-resolution is the technique to construct HR images from input LR images. It can also be interpreted as image up-sampling, up-scaling or enhancing the details within blurry images caused by LR cameras. It has been one of the most active research areas due to the growing desire of HR images in most of modern visual applications or visual computing tasks, *e.g.*, Earth observation [LBDG+18, LWL18], surveillance and security [LFCS07, ZZSL10, RUEA16], regular video enhancement [TMPE09,STW+11], bio-metric information identification [BDX16], or even life-saving medical diagnosis [Gre08,IK15,HSF17] *etc.*. With small modifications, super-resolution techniques can also be used to image restoration or image in-painting.

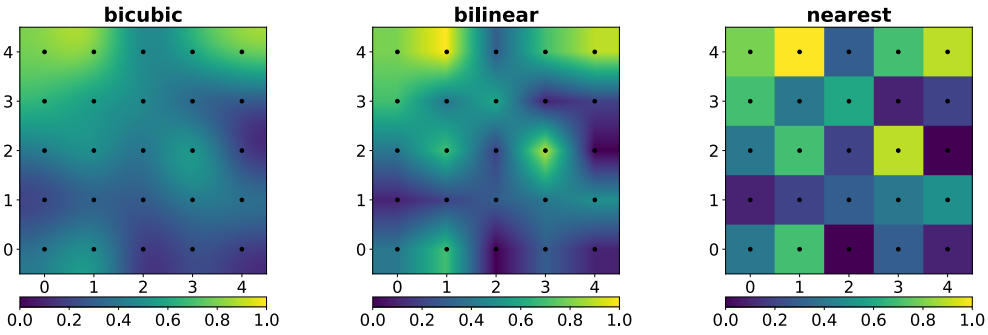
Many methods to super-resolve images or videos have been proposed over the last two decades. Depending on whether machine learning is used or not, super-resolution can be divided to non-learning based super-resolution and learning-based super-resolution.

2.4.1 Non-learning based super-resolution

A natural idea to super resolve an image is to estimate the value of some additional pixels according to prior knowledge, and insert them in-between the pixels of original images. One representative non-learning based super-resolution is interpolation-based methods, which applies naive interpolation methods, *i.e.*, nearest neighbor, bilinear, bicubic, Lanzos [Duc79] *etc.*, to interpolate and enlarge the input LR images, see Figure 2.8. Because of easy to implement and high efficiency, interpolation-based methods are widely used in research or many applications. For instance, most web browsers support to use one of those interpolation methods to resize an image when rendering a website. However, interpolation-based super-resolution often produces blurry results with aliasing artifacts [YH10]. More non-learning based super-resolution methods can refer to the recent surveys [WCH19, NM14, AKB19].

2.4.2 Learning-based super-resolution

Learning-based super-resolution has become a new trend in the field of super-resolution in recent years. Compared to interpolation-based super-resolution, learning-based super-resolution is more adaptive because the network can be tailored to different domain by training using different dataset, *i.e.*, Specialized super-resolution models can be trained for different applications, *e.g.*, natural image, RS, fingerprint, face super-resolution models *etc.*. In Section 2.2, we have introduced two kinds of convolution layers (transposed convolution and sub-pixel convolution) to up-sampling an image. They are both widely used in learning-based super-resolution models. In addition to different up-



(a) Bicubic **(b)** Bilinear **(c)** Nearest neighbor

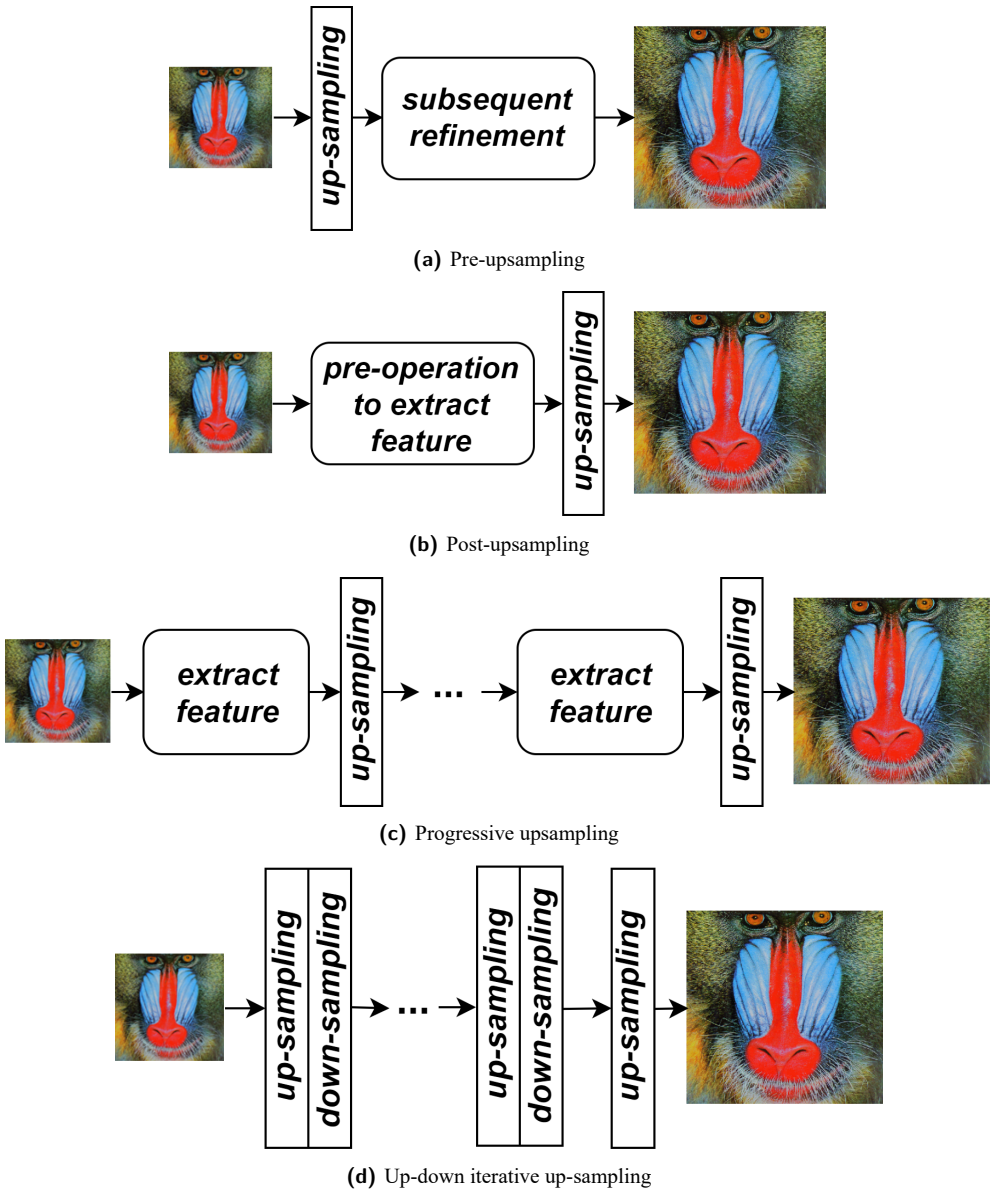
Figure 2.8: Illustration of bicubic, bilinear, nearest neighbor interpolation [Wik19]. Figiure (a), (b), (c) shows respectively the effect of the three interpolation on a $[0, 4] \times [0, 4]$ square consisting of 25 unit squares patched together. Color indicate the pixel values, the black dots are the location of the prescribed data being interpolated.

sampling layers, learning-based super-resolution can also be classified to supervised super-resolution and unsupervised super-resolution.

1. Supervised super-resolution: By using HR images as references and LR images as inputs, the super-resolution model can be trained in a supervised manner. One drawback of this method is that the HR reference does not exist for an image in reality. (LR, HR) training pairs have to be created manually by some generally adopted protocol, *e.g.*, Wald protocol explained in section 4. Additionally, because of the missing HR in reality, more than one HR solution for each LR input is possible. Effective metrics to evaluate the proposed HR solutions are required. Human can justify the quality of images intuitively but the mechanism of human perception is still unknown, which makes the problem even more challenging.

2. Unsupervised super-resolution: Unsupervised super-resolution is not affected by the missing HR references. It can use unpaired LR and HR to train a model, which encourage the model generate image not only matching the corresponding HR but also the general distribution of HR images.

Furthermore, according to the number of input LR images, super-resolution can also be framed into single image super-resolution (SISR) and multiple image super-resolution(MISR). Unlike SISR constructs HR output only based on one LR input, MISR can combine the non-redundant information contained in multiple low resolution frames to generate HR output. In addition, according to the position of up-sampling operation in the entire super-resolution pipeline, super-resolution can also be categorized to pre-upsampling super-resolution, post-upsampling super-resolution, progressive super-resolution and up-down sampling super-resolution, see Figure 2.9.



(a) Pre-upsampling

(b) Post-upsampling

(c) Progressive upsampling

(d) Up-down iterative up-sampling

Figure 2.9: Architecture of four super-resolution pipelines. (a) Pre-upsampling: first, the LR input is up-sampled with adaptive or non-adaptive operations, *e.g.*, naive interpolation, transposed convolution, sub-pixel convolution *etc.*, then, more subsequent operations learn to refine the up-sampling and yield the HR output in the end. (b) Post-upsampling: instead of up-sampling the LR input directly, Post-upsampling up-samples the high-level features extracted by some pre-operations. (c) Progressive upsampling: a up-sampling with large scale is factorized to many child post-samplings with smaller scale. (d) Up-down iterative sampling: iteratively apply back-projection [IP91] to mining the LR-HR reconstruction relationship to super-resolve an LR input.

Chapter 3

Related works

In this chapter, we introduce the related works surrounding the super resolution of RS images. In the beginning, we familiarize the reader with the latest learning-based techniques in natural image super resolution. Then, we continue with the cutting-edge progresses in RS image super resolution. In the end, some techniques to stabilize the training of GANs are described.

3.1 Natural Image Super-resolution

In recent years, learning-based super-resolution has become more dominant and achieved unprecedent results. Next we will briefly introduce representative works of applying deep learning to natural image super resolution. Note all methods below are only for single image super-resolution (a.k.a., example-based super resolution).

Dong *et al.* [DLHT14] proposed SRCNN, that first applied convolution neural networks to natural image super-resolution by learning an end-to-end mapping from LR to HR images. Afterwards, more learning-based methods with deeper architectures were proposed, *e.g.*, DRCN [KKLML16b], VDSR [KKLML16a] *etc.*. DRCN has tried recursive or shared weights to reduce the number of parameters inside a model and created a deeper network with 20 layers. Instead of learning a direct LR $\mapsto$ HR mapping, VDSR learned the residual between LR inputs and HR outputs. This can make the layered feature in the network more sparse and enable a much deeper architecture with high convergence speed and better performance.

In addition to increasing the depth of network, various network architectures have also been proposed. Huang *et al.* [HLVDMW17] first introduced densely connected convolution network (DenseNet) and Tong *et al.* [TLLG17] first exploited it to image super resolution. For each layer within a DenseNet, the outputs of all preceding

layers are used as inputs, and its outputs are used as inputs into all subsequent layers. Further, the concept of residual learning [HZRS16] is first introduced to solve the degradation problem in deep network training (*i.e.*, with the network depth increasing, accuracy gets saturated and then degrades rapidly, reported in [HS15, HZRS16]) and has become one of the most popular deep learning frameworks. In the field of super-resolution, there exists many deep architectures with residual skip connections [LTH+17,ZTK+18,ZLL+18,KKLML16a]. More methods tried to combine DenseNets and ResNets together, representative works are RDN [ZTK+18], RRDB [WYW+18], DCAN [JP19] *etc.* Furthermore, Zhang *et al.* introduced channel attention blocks to ResNets to adaptively rescale channel-wise output of each residual block [ZLL+18].

In addition to exploring different network architectures, there are also many attempts on various loss functions. Because the widely-used evaluation metric, peak signal to noise ratio (PSNR), is highly correlated with pixel-wise difference, most of recent works optimized the model with pixel-wise loss, including L1 loss [WYW+18, JP19,ZLL+18,ZTK+18] and L2 loss [DLHT14,KKLML16a]. What's more, Sajjadi *et al.* [SSH17] proposed a texture matching loss function to create realistic textures rather than optimizing for a pixel-accurate reproduction of ground truth images. Johnson *et al.* [JAFF16] introduced *perception similarity measure* to calculate the distance in deep feature space (*e.g.*, VGGNet layered feature) instead of in image space. And it has become a popular loss function for image super resolution [LTH+17, WYW+18, SSH17].

Recent literature has shown that GANs can be used to generated natural images of unprecedented quality. It has also been used to super-resolve LR images [LTH+17, WYW+18]. Compared with methods without GANs, those GAN-based methods can generate images with more high frequency details, making them have better perceptual quality. But, better perceptual quality does not mean higher reconstruction accuracy to the ground truth, and more and more recent works have shown that reconstruction accuracy and perceptual quality are typically in disagreement with each other, *e.g.*, although SRGAN [LTH+17] can produce high frequency edges, some of those edge actually do not exist in the reference HR. The trade off between the perception and distortion are discussed in those works [BMT+18], [BM18].

3.2 Remote Sensing Image Super-resolution

For RS image super-resolution, deep learning-based methods has also become an important research direction. However, due to many differences between natural and RS images, see Section 2.1, a model designed for natural image may not perform well on RS super-resolution. Moreover, because of the various configurations of satellite image sensors, the formats of RS images also vary a lot, thus making the task of RS super-resolution more complex. In general, when studying a RS super-resolution model, it is necessary to figure out which satellites the method works on or what is the image format. Below we will introduce the cutting-edge progresses of RS super-resolution,

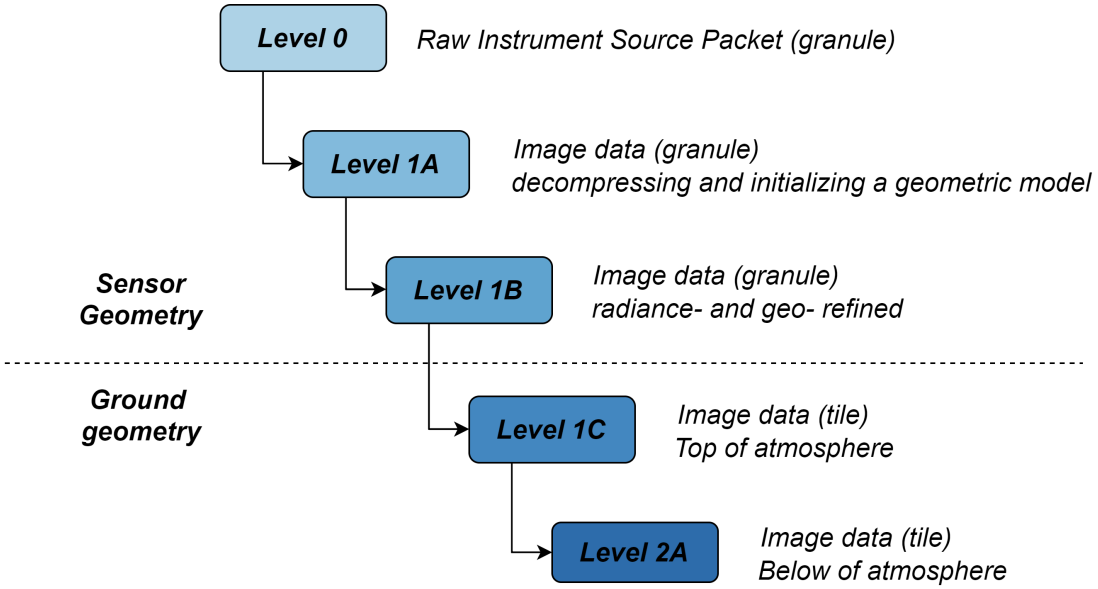


Figure 3.1: Five pre-processing levels of Sentinel-2 MSI products. In this figure, the output of preceded level is also the input of proceeded level.

including panchromatic sharpening (Pan-sharpening), multi-resolution multi-spectral super-resolution, super-resolution of transformed RGB image or original reflectance.

Pan-sharpening means use a panchromatic (single band) to increase the spatial resolution of multi-spectral images. This is based on the assumption that a panchromatic image with high spatial resolution exists in the original observation, *e.g.*, the RS images acquired by Ikonos, GeoEye-1, WorldView-2, Quickbird Lansat-8 *etc.* Yang *et al.* [YFH+17] first proposed PanNet to do pan-sharpening with deep learning. It applied residual networks and high-pass filters to extract high-frequency details in the panchromatic and fuse them to the other naively interpolated LR bands. Liu *et al.* [LWL18] introduced PSGAN based on adversarial learning to fuse the panchromatic and LR multi-spectral bands and generate high quality RS images.

However, not all satellites can create such HR panchromatic, *e.g.*, the MSI products from Sentinel-2A&2B have only 13 multi-resolution multi-spectral bands without panchromatic. For those MSI products, the mainstream method is multiple image super-resolution, that super-resolve multiple LR bands by blending in information from multiple HR bands, representative works include [LBDG+18, LBDBS17, KBP+19, PLPD18]. More recently, GAN-based [PSU18] or unsupervised [YLZ+18, HFBP+18] models have also been proposed to super resolve the Sentinel-2 MSI products.

One more difference between natural image super-resolution and RS super-resolution is that the later is required to consider more complex input formats. Taking Sentinel-2 MSI products as an example, five processing levels of the raw reflectance exist, see Figure 3.1, and different processing levels generate outputs with different data formats. Therefore, apart from the various satellite sensor configurations, the RS super-resolution can also be classified according to the input processing level. Some methods

simplify the problem by transforming the collection of ground reflectance to color natural images, *i.e.*, 3 channels and each pixel is represented by a integer in $[0, 255]$, representative works are [MPGL18,HFBP+19,LWL18,LSZ17].Recently, methods to super-resolve the Sentinel-2 MSI products are proposed, including both Level-1C [LBDG+18, LBDBS17] or Level-2A [PLPD18] products.

To the best of our knowledge, there are still no widely-used evaluation datasets in RS super-resolution. Most previous models are trained and tested on small datasets or several tiles [MPGL18, HFBP+19, PLPD18]. Recently, Lanaras *et al.* [LBDG+18] trained and tested a simple residual model with 45 and 15 tiles respectively randomly selected all over the world, which significantly improved the RS image super-resolution performance.

3.3 Techniques to Stabilize GAN Training

GANs have boosted the generation of high quality images or video in spites of their instability, fragile training, and high sensitivity to network architectures or hyper-parameters. In addition to exploring the potentials of GANs in various practical applications, to improve the stability of training a GAN is also a active research topic. Below are some of representative works published recently.

First, various loss functions have been proposed to stabilize the GAN training process, *e.g.*, least square GAN [MLX+17], Wasserstein GAN [ACB17], Wasserstein GAN with gradient penalty [GAA+17], Relativistic GAN [JM18], Energy-based GAN [ZML16] and its variant [BSM17], Cramer GAN [BDD+17] *etc.*. Although cost functions are one major research area in GANs, it is still too early to make any conclusion on which cost function performs best. Indeed, no evidence can prove that any of those cost functions consistently outperforms the original GAN [GPAM+14], and there is still no single cost function that achieves the best performance among all common test datasets yet. However, those cost functions provide us with more choices when designing our own GAN applications, *i.e.*, when the performance of a GAN model plateaus, we can try to improve it with different cost functions.

In addition, many strategies to coordinate the balance between a generator and a discriminator have been proposed, For example, two time-scale update rule (TTUR) [HRU+17] differentiates the learning rate of the generator and the discriminator in a GAN to balance their update speed. Progressive augmentation GAN [ZK19a] was proposed to progressively augment the input space of the discriminator during training, which is based on the assumption that a discriminator is easy to outperform a generator.

Apart from the methods above, various methods to stabilize the GAN training has been proposed and hard to categorize. For instance, big-GAN [BDS18] proved

that GAN training benefits from scaling. And many regularizer have been proposed to speed up the convergence of a GAN [CDLH18, ZK19b, RLNH17, WGL+18].

Chapter 4

Problem formulation

The 13 spectral bands in a Sentinel-2 MSI product can be divided to three sets according to their GSD, $20m$ bands: $A = \{B5, B6, B7, B8a, B11, B12\}$, $10m$ bands: $B = \{B2, B3, B4, B8\}$ and $60m$ bands: $C = \{B1, B9\}$. $B10$ has comparatively poor radiometric quality and exhibits across-track striping artifacts , so only $B1$ and $B9$ are considered as $60m$ bands in this thesis. All bands in a same MSI product are acquired over the same area. In this thesis, the pixel resolution of a $10m$ band in B is denoted as $W \times H$, the pixel resolution of the corresponding $20m$ or $60m$ bands in A or C can be depicted as $W/2 \times H/2$, $W/6 \times H/6$ respectively. In this thesis, we tackle the problem of super-resolving the spatial resolution of LR bands in A and C to $10m$ GSD, *i.e.*, the pixel resolution of all super-resolved output bands has to be $W \times H$.

This thesis is devoted to super-resolve the Sentinel-2 MSI products with format of both level-1C and level-2A. Because of different data characteristics (level-1C is top of atmosphere (ToA) and level-2A is bottom of atmosphere (BoA), see Figure 3.1), the two formats of data are considered separately when training a super-resolution model. For each format, two models, $\mathcal{S}_{2\times}^{1C}$ and $\mathcal{S}_{6\times}^{1C}$ (or $\mathcal{S}_{2\times}^{2A}$ and $\mathcal{S}_{6\times}^{2A}$) are developed for $2\times$ and $6\times$ super-resolution correspondingly.

$\mathcal{S}_{2\times}^{1C}$ or $\mathcal{S}_{2\times}^{2A}$, super-resolves the $20m$ bands in A with the input from both A and B. $60m$ bands C is excluded because Lanaras *et al.* [LBDG+18] have proved that they do not contribute to $20m \mapsto 10m$ super-resolution.

$$\mathcal{S}_{2\times}^{1C},\ \mathcal{S}_{2\times}^{2A}:\ \ \mathbb{R}^{W\times H\times 4} \times \mathbb{R}^{W/2\times H/2\times 6} \mapsto \mathbb{R}^{W\times H\times 6}$$

$\mathcal{S}_{6\times}^{1C}$ or $\mathcal{S}_{6\times}^{2A}$, super-resolves the $60m$ bands in C with the input from A, B and C.

$$\mathcal{S}_{6\times}^{1C},\ \mathcal{S}_{6\times}^{2A}:\ \ \mathbb{R}^{W\times H\times 4} \times \mathbb{R}^{W/2\times H/2\times 6} \times \mathbb{R}^{W/6\times H/6\times 2} \mapsto \mathbb{R}^{W\times H\times 2}$$

Therefore, the super-resolution model in this thesis not only learns the LR↦HR mapping, but also can blend in information from high spatial resolution bands in a

disjoint spectral domain. The investigation of spectral distortion caused by band fusion is out of the scope of this thesis.

To supervise the training of a super-resolution model, the corresponding HR reference for each LR input is required. But these HR references don't exist in reality. In RS super-resolution, one wide-accepted strategy to circumvent missing HR references is to perform training and testing at a degraded scale, *i.e.*, using the degraded observed data as LR inputs and the original data as HR references. This is known as Wald's protocol and first proposed by Zeng *et al.* [ZHL+10] to quantitatively assess the performance of multi-spectral image fusion. The assumption being made here to use this protocol is that the relationship learned from the degraded resolution level also applies to the original level [PSUB15], *i.e.*, the up-sampling from $20m \mapsto 10m$ GSD, can be learned from ground truth images at $40m$ and $20m$ GSD; and similarly for the $60m \mapsto 10m$ case.

The way to apply Wald's protocol to Sentinel-2 MSI products is shown in Figure 4.1. The objective is to super-resolve the raw A and C to $A^{'}$ and $C^{'}$ respectively. To achieve this, 1) the complex mixture of correlations across spectral bands is learned at a degraded scale by Equation (4.1) and (4.2).

$$\mathcal{S}_{2\times} : (A_{2\times}, B_{2\times}) \mapsto A \tag{4.1}$$

$$\mathcal{S}_{6\times} : (A_{6\times}, B_{6\times}, C_{6\times}) \mapsto C \tag{4.2}$$

2) According to Wald's protocol [WRM97], the mapping relation learned at degraded scale can also be applied to original data, see Equation (4.3) and (4.4).

$$A^{'} = \mathcal{S}_{2\times}(A, B) \tag{4.3}$$

$$C^{'} = \mathcal{S}_{6\times}(A, B, C) \tag{4.4}$$

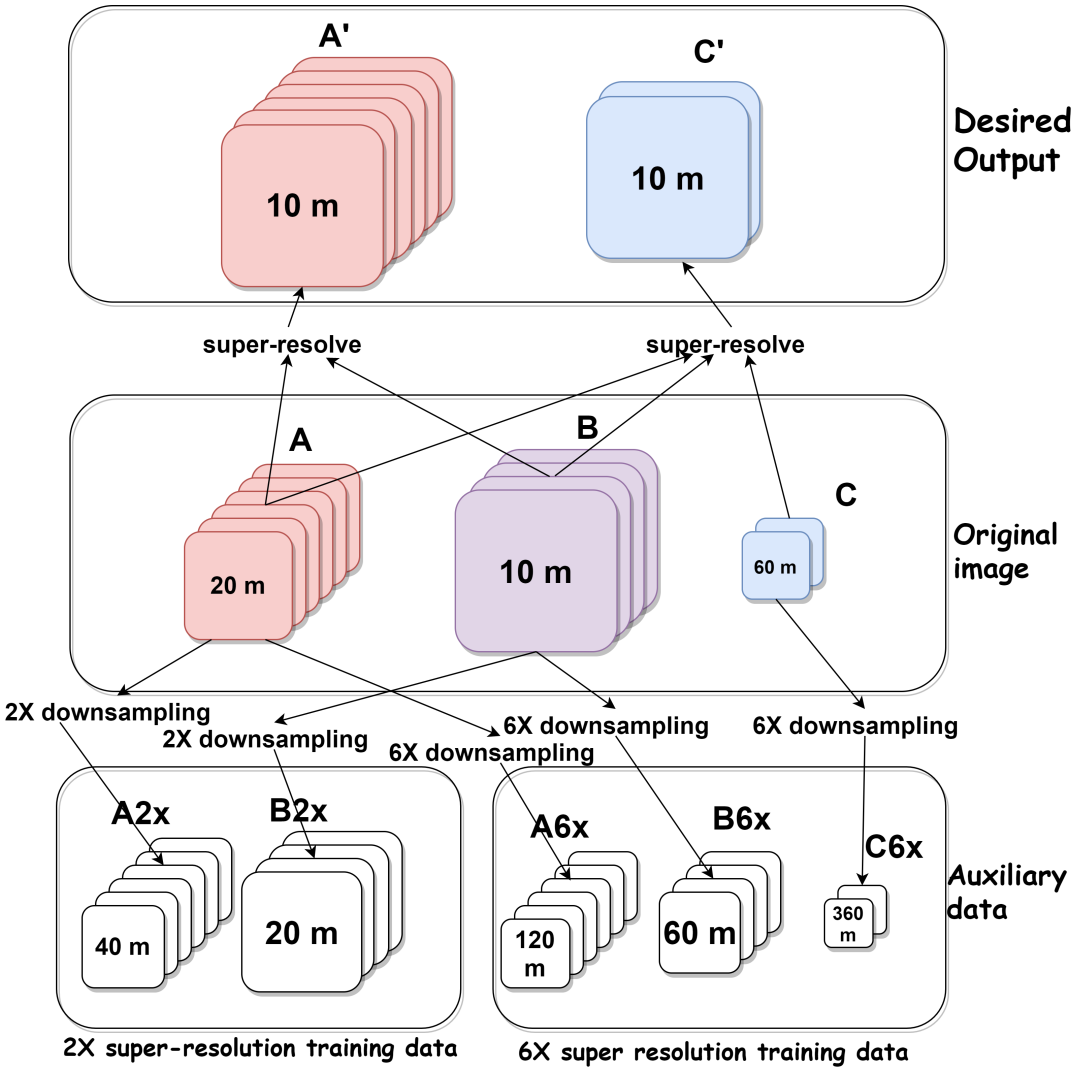


Figure 4.1: Illustration of applying Wald's protocol to Sentinel-2 MSI products to create training datasets. Middle: A, B, C are three categories of original bands of Sentinel-2 MSI products. Bottom left: $A_{2\times}$ and $B_{2\times}$ are bands degraded 2 times from A and B respectively. Bottom right: $A_{6\times}$, $B_{6\times}$ and $C_{6\times}$ are bands degraded 6 times from A, B and C respectively. Top: $A^{'}$ and $C^{'}$ are the desired super-resolved output with $10m$ GSD.

Chapter 5

Methodology

This chapter describes the methods that have been developed for the work of this thesis. First, the network architectures of both generator and discriminator are described in detail in Section 5.1. Then, the loss functions to penalize and guide the model training are introduced in Section 5.2. In the end, the methods to parallelize and speed up the training process are explained in Section 5.3.

5.1 Network Architecture

A GAN-based super-resolution model is made of two interacting modules, a *generator* G and a *discriminator* D. The generator G is tasked with super-resolving the LR inputs, whereas the discriminator D learns to distinguish outputs of G and the ground truth.

5.1.1 Architecture of the generator

First we start with the $2\times$ super-resolution. As shown in Equation (5.1), the training pairs for $\mathcal{S}_{2\times}$ can be given by

$$\left[(a^i_{2\times} \in A_{2\times}, b^i_{2\times} \in B_{2\times}), b^i \in B\right], i \in [0, N] \tag{5.1}$$

where N is the number of pairs in the entire training dataset.

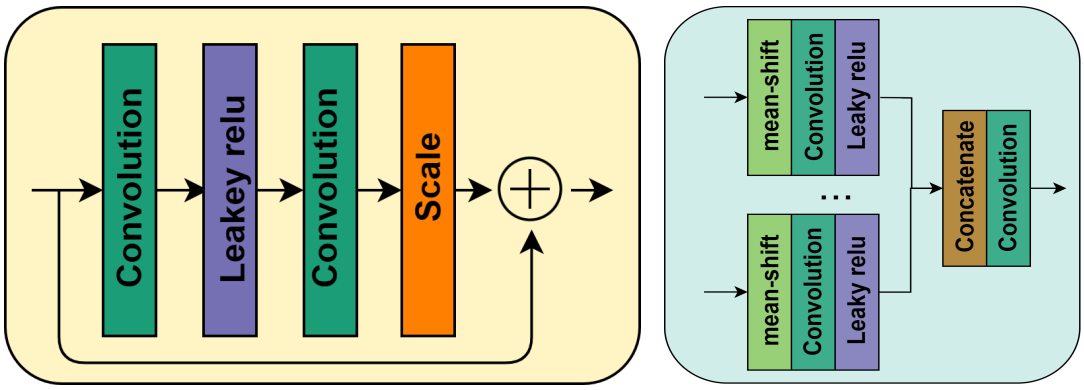
This thesis applied the pre-upsampling framework, illustrated in Figure 2.9, to design the architecture of generator. Pre-upsampling enables the super-resolution model to learn only the residual between inputs and outputs and makes the layered feature in

the network more sparse thus accelerating the training speed. Therefore, the degrade $20m$ bands data $b^i_{2\times}$ is first up-sampled 2 times by a up-sampling module $H_{2\uparrow}$.

$$F_{b^i_{2\times}} = H_{2\uparrow}(b^i_{2\times})$$

As explained in Chapter 4, the super-resolution model tries to learn the mixture correlations of multiple spectral bands. Therefore, the up-sampled $20m$ bands $F_{b^i_{2\times}}$ proceed to fuse with $10m$ bands $a^i_{2\times}$ through a band fusion module H_{fusion}, see Figure 5.1b.

$$F_{fusion_{2\times}} = H_{fusion}(a^i_{2\times}, F_{b^i_{2\times}})$$



(a) Residual block **(b)** Band fusion module

Figure 5.1: (a) Architecture of residual blocks in generator. The output is the sum of original input and the feature map learned by a series of layers. Both of two convolution layers in this block have kernel of size (3, 3). An activation layer, leaky relu, in-between them is used to speed up training. Instead of batch normalization, we multiply the output of second convolution layer with a small constant (0.1), which can also significantly improve the training speed of deep neural networks. This is inspired by some state-of-the-art super-resolution methods [WYW+18, LBDG+18] that also exclude batch normalization layers in their model. (b) Architecture of band fusion module. Each branch of bands first goes through the operation mean-shift to move the mean of input to 0 to suppress the impact of large brightness changes in all training and testing patches, and then a convolution layer and a activation layer are applied to extract low-level feature, then those output of all branches are stacked up together with a concatenation layer and mixed up by one more convolution layer.

Then the output $F_{fusion_{2\times}}$ goes through a residual self-attention module (RSA) and a final convolution layer to learn the difference between the ground truth and the up-sampled $F_{b^i_{2\times}}$.

$$F_{diff_{2\times}} = F_{conv}(F_{RSA}(F_{fusion_{2\times}}))$$

In the end, the output of the generator $\mathcal{S}_{2x}$ can be represented by

$$F_{\mathcal{S}_{2\times}} = F_{b^i_{2\times}} + F_{diff_{2\times}}$$

The RSA model is made of a series of residual blocks (see Figure 5.1a) with a self-attention module (see Figure 5.2) in the middle. Because of spatial limitations of receptive fields, the convolution layer usually cannot explore the global structures

inside an image. The attention mechanism is proposed to capture the long-range dependencies over the entire input feature map. The self-attention architecture was first proposed by Zhang *et al.* [ZGMO18] to generate high quality images. The generator architecture in this thesis is inspired by second-order attention network (SAN) [DCZ+19] which has achieved the state-of-the-art performance in the problem of single image super-resolution with the variant of self-attention mechanism.

The overall architecture of $20m \mapsto 10m$ generator $\mathcal{S}_{2\times}$ is shown in Figure 5.3a. Similarly, the $60m \mapsto 10m$ generator $\mathcal{S}_{6\times}$ can be represented by the following equations and shown in Figure 5.3b.

$$\left[(a^i_{6\times} \in A_{6\times}, b^i_{6\times} \in B_{6\times}), c^i_{6\times} \in C_{6\times}), c^i \in C\right], i \in [0, N]$$

$$F_{b^i_{6\times}} = H_{2\uparrow}(b^i_{6\times}), F_{c^i_{6\times}} = H_{6\uparrow}(c^i_{6\times})$$

$$F_{fusion_6\times} = H_{fusion}(a^i_{6\times}, F_{b^i_{6\times}}, F_{c^i_{6\times}})$$

$$F_{\mathcal{S}_{6\times}} = F_{c^i_{6\times}} + F_{conv}(F_{RSA}(F_{fusion_2\times}))$$

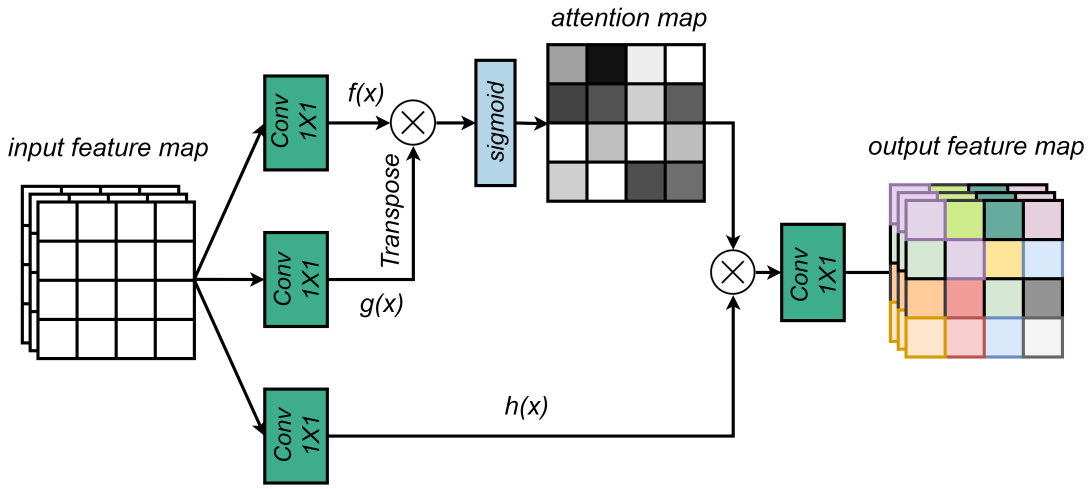
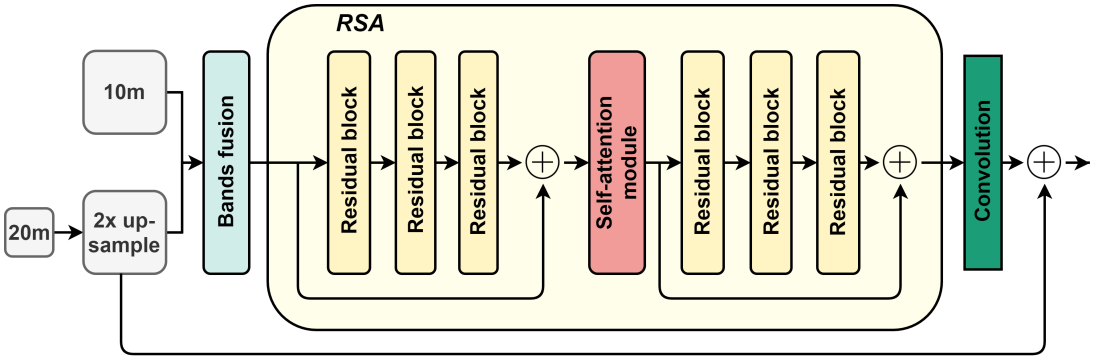


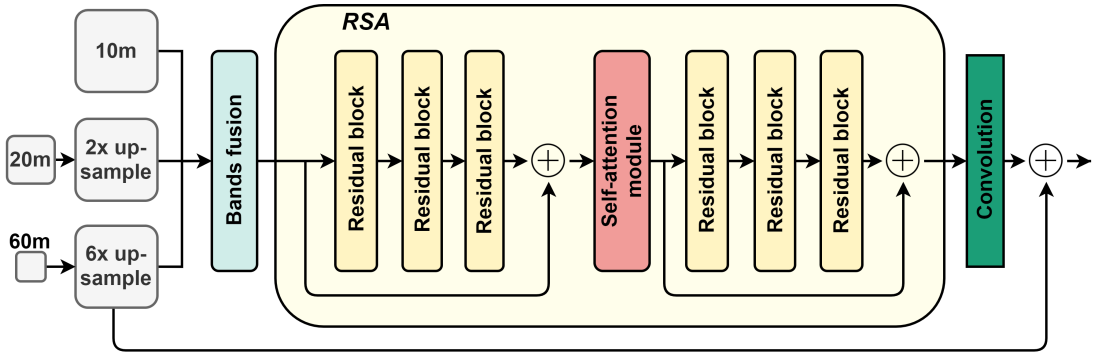
Figure 5.2: Illustration of the architecture of self-attention module. The input and output of this module are both feature maps and the symbol $\otimes$ denotes matrix multiplication. The top two branches, $f(x)$ and $g(x)$, learn an attention map. Then $h(x)$ is multiplied with this attention map and a final 1×1 convolution layer is applied in the end to yield the output feature map. All convolution layers in this paper have 128 filters except $f(x)$ and $g(x)$ has 16 filters.

5.1.2 Architecture of the discriminator

The architecture of the discriminator is inspired by SRGAN [LTH+17] and shown in Figure 5.4. The input images, (*i.e.*, either the the ground truth or output of the generator), are processed by a series of convolution, batch normalization , activation (leaky relu is used) layers to extract the distinguishable representation. Stride length in each



(a) $20m \mapsto 10m$ super-resolution generator



(b) $60m \mapsto 10m$ super-resolution generator

Figure 5.3: Illustration of architecture of generator, including (a) $20m \mapsto 10m$ super-resolution generator and (b) $60m \mapsto 10m$ super-resolution generator. The two generators have the same RSA module and final convolution layer. One difference is that $60m \mapsto 10m$ generator has three branches of input and the up-sampled $60m$ bands is added up to the output in the end, whereas the $20m \mapsto 10m$ generator has only two branches of input and the up-sampled $20m$ bands is added up to the final output. All convolution layers in this thesis have 128 filters except $f(x)$ and $g(x)$ have 16 filters. 6 residual blocks is used in the RSA module shows the fairness with the comparable method DSen2 in the sense of model complexity.

convolution layer is increased gradually to reduce the feature map resolution thus reducing the representation size. Then the representation is followed by two dense layers and a final Sigmoid function to calculate the probability for fake generation/ground truth classification.

5.2 Model Training

In this section, we presents the process to train a super-resolution model and all used loss functions. The training process can be divided to two phases, the first is to pretrain the generator and the second is to train a GAN initialized by the pretrained generator.

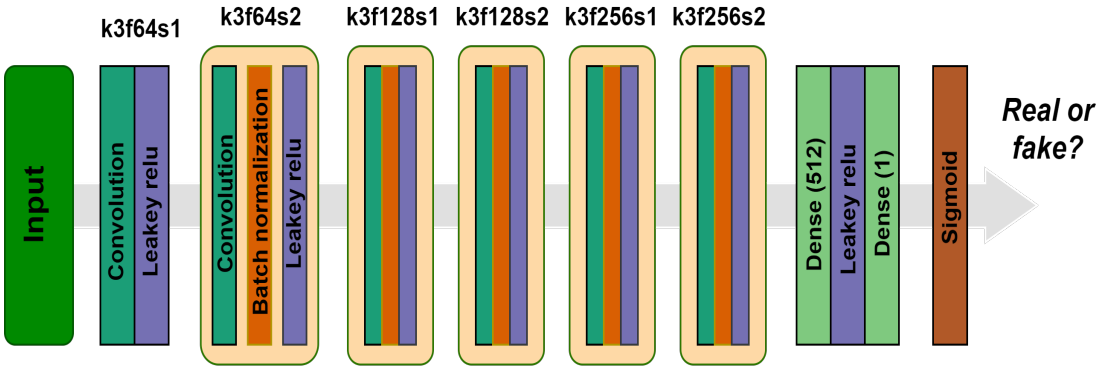


Figure 5.4: Illustration of the architecture of discriminator. k is the side length of a kernel, s is the stride length. f is the number of filters in a convolution layer. For instance, k2f64s1 means that the kernel size is $(3, 3)$, the stride is 1 and the number of filters is 64.

5.2.1 Pretrain the generator

As explained in Section 2.3.3, a discriminator is easier to surpass a generator in an early phase in most cases. To balance the training between generator and discriminator, we first pretrain the generator with L1 loss that is a common pixel-wise loss to evaluate the distance between two images. Compared to L2 loss penalizing larger errors and being more tolerant to small errors, L1 loss is better to preserve high frequency details and has been widely adopted by previous works [WYW+18, JP19, ZLL+18, ZTK+18].

$$\mathcal{L}_{L1}(\hat{HR}, HR) = \frac{1}{HWC}||HR_{i,j,c} - \hat{HR}_{i,j,c}||$$

where HR, $\hat{HR}$ are the ground truth and output of generator respectively and H, W, C means height, width and number of channel. In this thesis, we use Charbonnier loss, which is a variant of L1 loss.

$$\mathcal{L}_{Charb}(\hat{HR}, HR) = \frac{1}{HWC}\sqrt{(HR_{i,j,c} - \hat{HR}_{i,j,c})^2 + \epsilon^2}$$

where $\epsilon = 1e^{-4}$ is added to improved the numerical stability.

As discussed in [WYW+18, PSC+18], by pretraining the generator, the discriminator will be trained with a relatively good generator output in an early phase, so the discriminator can focus more on learning finer differences between the generated output and the ground truth. Further, by pretraining, we can concentrate to training the generator with the entire dataset. Comparing with splitting the dataset for generator training and discriminator training, this way can utilize the dataset more efficiently.

Algorithm 5.2.1 Minibatch stochastic gradient descent training of GANs. The number of steps to update the discriminator, k, is a hyper-parameter. For WGAN and WGAN-gp, $k = 1$ is used. For other types of GAN, $k = 1$ is used.

for number of training epochs **do**
  for k steps **do**
    • Sample a minibatch of HR ground truth $\{x_1, ..., x_n\}$.
    • Run generator to create a minibatch of fake output $\{G(z_1), ..., G(z_n)\}$.
    • Update the discriminator by descending its stochastic gradient: $\mathcal{L}_D$.
  end for
  • Sample a minibatch of LR input $\{z_1, ..., z_n\}$.
  • Update the generator by descending its stochastic gradient: $\mathcal{L}_G$
end for

5.2.2 GAN training

In the chapter of background, we have shown that the original GAN losses are composed of two parts, discriminator loss $\hat{J}(D)$ and adversarial loss $\hat{L}(G)$. And as shown in Figure 2.5, a super-resolution generator is usually penalized by an additional content loss. In this thesis, the generator is first initialized with pre-trained generator above, and then updated by gradient descent of the loss $\mathcal{L}_G$

$$\mathcal{L}_G = \eta \hat{L}(G) + \mathcal{L}_{Charb}$$

where $\mathcal{L}_{Charb}$ is used as content loss and η is a coefficient to balance $\hat{L}(G)$ and $\mathcal{L}_{Charb}$. And the discriminator is updated by loss $\mathcal{L}_D$.

$$\mathcal{L}_D = \hat{J}(D)$$

The alternating update of the generator and discriminator is shown in Algorithm 5.2.1.

As shown in Section 3.3, many GAN losses have been proposed to tackle the instability of GAN training. Although they all advocated they can be used to train a better GAN on some particular datasets, no GAN loss was proven to be able to perform consistently better than all others on all datasets until now. So next, we will present some cutting-edge GAN losses. Their performance on Sentinel-2 MSI product super-resolution will be evaluated in Section 6.6.

1. Least square GAN(LSGAN). The discriminator loss is

$$\hat{J}_{LSGAN}(D) = \frac{1}{2N}\sum_{n=0}^{N}\left[(D(x_n) - b)^2\right] + \frac{1}{2N}\sum_{n=0}^{N}\left[(D(G(z_n) - a))^2\right]$$

   where a and b are the labels denoting fake and real data.

   The generator loss is

$$\hat{L}_{LSGAN}(G) = \frac{1}{2N}\sum_{n=0}^{N}\left[(D(G(z_n) - c))^2\right]$$

where c is a label for fake data that is not necessary to be equal to a. It can be proved that when $(a, b, c) = (-1, 1, 0)$, the optimal value for GAN is $\frac{1}{2}\chi^2(p_r + p_g||2p_g)$ where χ^2 is the Pearson Chi-Square divergence [MLX+17].

2. Wasserstein GAN(WSGAN) [ACB17]. The discriminator loss is

$$\hat{J}_{WGAN}(D) = -\frac{1}{N}\sum_{n=0}^{N}[D(x_n)] + \frac{1}{\sum}_{n=0}^{N}[D(G(z_n))]$$

The generator loss is

$$\hat{L}_{WGAN}(G) = -\frac{1}{N}\sum_{n=0}^{N}[D(G(z_n))]$$

3. Wasserstein GAN with gradient penalty(WGAN-GP) [GAA+17]. The generator loss is the same with WGAN

$$\hat{L}_{WGAN-GP}(G) = \hat{L}_{WGAN}(G) = -\frac{1}{N}\sum_{n=0}^{N}[D(G(z_n))]$$

The discriminator loss is

$$\hat{J}_{WGAN-GP}(D) = \hat{J}_{WGAN}(D) + \lambda GP$$

where GP is the gradient penalty and defined as

$$GP = \lambda\frac{1}{N}\sum_{1}^{N}(\|\nabla_{\tilde{x}_n}D(\tilde{x}_n)\| - 1)^2, \tilde{x}_n = tG(z_n) + (1-t)x_n$$

with t uniformly sampled from $(0, 1)$ and λ is a coefficient to balance $\hat{J}_{WGAN}(D)$ and GP.

4. GAN with Hingle loss. The discriminator loss is

$$\hat{J}_{Hingle}(D) = \frac{1}{N}\sum_{n=0}^{N}[max(0, 1 - D(x_n))] + \frac{1}{N}\sum_{n=0}^{N}[max(0, 1 + D(G(z_n)))]$$

The generator loss is the same with WGAN and WGAN-GP

$$\hat{L}_{Hingle}(G) = \hat{L}_{WGAN-GP}(G) = \hat{L}_{WGAN}(G) = -\frac{1}{N}\sum_{n=0}^{N}[D(G(z_n))]$$

5. Relativistic standard GAN [JM18]. The discriminator loss is

$$\hat{J}_{relativ}(D) = -\frac{1}{N}\sum_{n=0}^{N}[\log(D(x_n) - D(G(z_n)))]$$

The generator loss is

$$\hat{L}_{relativ}(G) = \sum_{n=0}^{N} [\log(D(G(z_n)) - D(x_n))]$$

5.3 Distributed Learning with Horovod

To process of large amounts of Sentinel-2 RS data, we speed up the training process with parallelization. Jeff *et al.* [CDLH18] have proposed two paradigms to parallelize the training of a deep model, including *model parallelism* and *data parallelism*. *Model parallelism* means parallelizing the computation of a single model with multiple processes or over multiple machines, while *data parallelism* means parallelizing the gradient descent by splitting the training dataset to multiple partitions and allocating them to multiple machines. One important difference between the two paradigms is that each machine in a system of *model parallelism* computes only part of the model with the entire dataset, whereas each machine in a system of *data parallelism* computes the entire model with only part of the training dataset.

The *data parallelism* is based on the assumption that the model size can fit in each machine so that all machines can keep the same copy of the entire model. Furthermore, mini-batch SGD is also an important reason to enable *data parallelism*. In general, SGD is too computationally intensive to update model parameters after evaluating each single training sample. To speed up SGD, the concept of mini-batch is proposed, which allows the model to update only once with the average gradients after evaluating all samples in a mini-batch. As a result, the way to utilize *data parallelism* in a distributed computing cluster becomes quite straightforward. Each mini-batch is split into multiple partitions, and then each partition is allocated to one machine to evaluate all samples and compute their gradients. In this thesis, we optimize the training process with *synchronized data parallelism*, more specifically, for t th stochastic gradient descent (SGD) iteration for model θ, a mini-batch M is split into multiple partitions where each partition is consumed by its own worker to calculate gradients, finally we accumulate the gradients from all works and update the model θ with the average gradient $\frac{\sum_i \nabla_\theta \mathcal{L}(\theta, x_i)}{|M|}$, where x_i is the i th instance and $\mathcal{L}$ is the loss function.

In field of natural computer vision, many works have been published to show the relationship between mini-batch size, learning rate and model accuracy resulting from scaling the training. For instance, Priya *et al.* [GDG+17] has empirically showed that there is no loss of accuracy when training a object recognition model with large mini-batch sizes up to 8192 images on Imagenet, and Yang *et al.* [YGG17] proposed layer-wise adaptive learning rate scale that enables a larger mini-batch size up to 32K in Resnet-50. To the best of our knowledge, this thesis first applies distributed deep

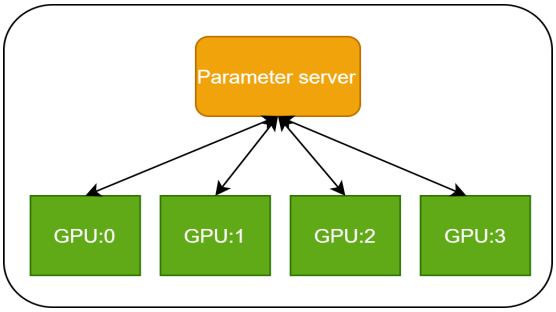


Figure 5.5: Illustration of the architecture of a system with 1 parameter server and 4 GPUs. For each mini-batch training iteration, those 4 GPUs take the responsibility of evaluate each samples in the mini-batch, calculating gradients and sending them to the parameter server. The parameter server collects those gradients, averaging them and sending them back to each GPU. Finally, all 4 GPUs will update their own model with the same gradients.

learning on remote sensing image super-resolution and investigates the impact of scaled batch size and scaled learning rate on model accuracy.

5.3.1 Central parameter server

In a *synchronous data parallelism*, the models in all machines must have the same parameters before and after each mini-batch updating. A straightforward method to synchronize gradients is to set up parameter servers collecting the gradients from all machines and sending back their average, see example with 1 parameter server in Figure 5.5. However, one inevitable disadvantages of this method is that is too hard to decide the optimal number of parameter servers. If only few sever is used, they are easy to become the bottleneck of the whole system for both computation and data transmission. If too many servers are used, it will saturate the network interconnects. Further, the utilization rate of bandwidth is not stable and optimal, *i.e.*, when all gradients have been collected and servers are computing the average, the bandwidth is actually wasted in this case. This becomes unneglected when the model grows extremely large. However, because of easy to use and implement, this is adopt as the built-in data parallelism mechanism in Tensorflow [ABC+16].

5.3.2 Ring-reduction mechanism

Instead of setting up center parameter servers, ring-reduction can distribute the task of gradient synchronization to each child machine. Supposing we are trying to synchronize the gradients on a computing cluster with N machines(GPU), the ring-reduction algorithm will connect all machines as a ring and split all gradients in each machine to N chunks, see Figure 5.6. For each mini-batch training, the ring-reduction algorithm have $2 \times (N-1)$ steps in total. In each step, each machine sends one chunk of data to its right neighbor and receive one chunk of data from its left neighbor. In the first $N-1$

steps, the received chunk is added to the corresponding chunk, see Figure 5.6(a-c). In the second $N-1$ steps, the received chunk will replace the corresponding chunk, see Figure 5.6(d-f). In the end, all machines(GPUs) will have the same sum of gradients, see Figure 5.6(g), so each machine can calculate the average by dividing the sum with N, see Figure 5.6(h).

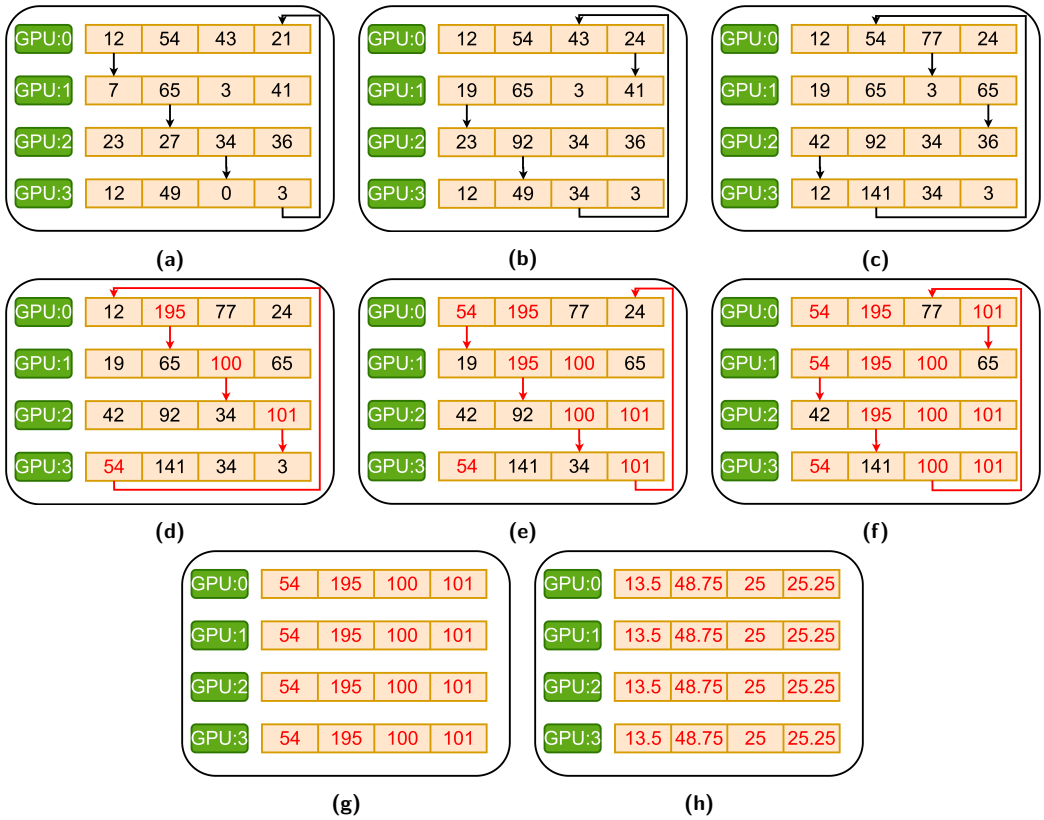


Figure 5.6: Illustration of gradient ring-reduction step by step in a cluster with 4 GPUs. Buffers in each GPU are used to synchronize gradients. In this example, each GPU is equipped with 4 buffers (depicted as yellow grids) and each buffer is of size for 1 gradient. The objective of ring-reduction is to average and synchronize gradients over the 4 GPUs. The address (grid index) of each gradient decides the model parameter it refers to, *e.g.*, $12, 7, 23, 12$ in the first grid in Figure (a) are gradients for a same model parameter. The sum of all gradients is calculated step by step by Figure (a-g), and each GPU calculates the average by dividing with the number of GPUs in Figure (h).

Thus, no obvious bottleneck exists inside the system with ring-reduction and it has been proved to be bandwidth-optimal [PY09]. In this thesis, ring-reduction is used to parallelize the training of our model. It has been included in a popular library Horovod [SDB18] and can be easily plugged into a existing Tensorflow deep model without complex changes.

5.3.3 Modified learning rate scale rule

To make distributed learning efficient, the per-worker workload must be large, which implies a corresponding growth in the SGD mini-batch size with the increase of the

number of workers. However, larger mini-batch size always cause optimization difficulties. To improve the model convergence with the larger mini-batch size, Priya *et al.* [GDG+17] proposed linear learning rate scale rule, but in practice, we found this rule often causing failure of training (loss explodes). Therefore, the following modified linear learning rate scale rule is used.

Modified linear learning rate scale rule: when the mini-batch size is multiplied by k, the initial learning rate is also multiplied by k and decays to half every $\frac{n}{k}$ SGD iterations.

Chapter 6

Experiments

This section presents and discusses the performance of the proposed super-resolution model $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$. First, we start with the experiment configurations, including the used distributed computing power and hyper-parameters settings. Second, the procedure to create training datasets, statistical analysis of the dataset, and evaluation metrics are explained. Then, the experiment results with various configurations are described in detail. In the end, the impact of distributing learning is discussed.

6.1 Experiment Settings

The super-resolution model in this thesis is implemented and evaluated with a popular open-source deep learning framework, Tensorflow 1.13 [ABC+16]. Some of the comparison methods are tested with Keras [C+15] and OpenCV[1]. The overall network architecture and training framework is heavily influenced by the repository[2] and the repository[3].

6.1.1 Computing platforms

All experiments are conducted on the HPC systems installed at the Juelich Supercomputing Center, including two supercomputing systems JUWELS, JURECA, and one pilot system JURON.

[1] https://opencv.org/
[2] https://github.com/xinntao/ESRGAN
[3] https://github.com/taki0112/Self-Attention-GAN-Tensorflow

	JUWELS	JURON	JURECA
No. nodes	58	18	75
No. GPUs per node	4	4	4
Type of GPUs	Tesla P100	Tesla V100	Tesla K80
Mem per GPU(Gb)	16/32	16	24
Peak GPU Performance(PF_LOp/s)	1.6	0.44	0.38

Table 6.1: Configuration of three used super-computing systems, JUWELS, JURON, and JURECA.

JUWELS (Juelich wizard for European leadership science) [Kra19] and JURECA (Juelich Research on Exascale Cluster Architectures) [KT16] are both supercomputing systems that are accessible for European researchers at large. In JUWELS, apart from 2511 CPU-only nodes, there are 56 accelerated computing nodes equipped with four Nvidia Tesla V100 GPUs. Those four GPUs in each node are interconnected via NVLink in an all-to-all topology. In JURECA, there are 1872 computing nodes in total, and 75 of them are accelerated nodes equipped with 2 Nvidia Tesla K80. The GPUs are connected with PCI Ex-press Generation 3.0. The GPU computing peak performance for JUWELS and JURECA are 1.6 and 0.44 $PFlop/s$ respectively.

JUWELS and JURECA have the same software characteristics. Both platforms are running CenOS 7 Linux distribution system and Slurm batch system with Parastation resource management. All the computing nodes are diskless, and all connected to a central GPFS file system. And they both support OpenMP programming model for intra-node parallelization.

JURON (the name is derived from JUelich and neuRON) is a pilot computing system tailored for a human brain project (HBP). It is a POWER8NVL system consisting of 18 nodes developed by IBM and Nvidia. Each node is equipped with 2 IBM POWER8 processors and 4 Nvidia Tesla P100 GPUs(16GB HBM2 memory). In each node, each pair of GPUs are connected to one CPU socket via fast NVlink, and the two GPUs in each pair are connected with NVlink. The compute nodes are connected via Mellanox ConnectX-4 Infiniband EDR network adapters to a single switch that can reach 100Gbps information transmission. The GPU peak performance is 0.38 PFlOp/s. And the batch system running in JURON is LSF.

6.1.2 Hyper-parameter settings

All parameters adopted in this thesis are listed in Table 6.2. Within the phase of GAN training, TTUR [HRU+17], different learning rates for discriminator and generator, is used to balance the training. Furthermore, the number of filters in a convolution layer can increase the model size by a significant amount. However, more filters are usually desirable because using fewer filters, especially in generator, can make the output too blurry. More filters can help the generator to capture additional information, eventually

Hyper-parameter	value
k	For WGAN-GP, $k = 5$ For other types of GAN, $k = 1$
learning rate	the initialized learning rate is $0.0001 * No.GPUs$, and the $n = 64000$ in the modified linear learning rate scale rule.
mini-batch size	$\mathcal{S}_{2\times}$: $No.GPUs \times 128$ $\mathcal{S}_{6\times}$: $No.GPUs \times 32$
optimizer	Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 1e^{-8}$ For WGAN-GP, ADAM with $\beta_1 = 0$, $\beta_2 = 0.9$, $\epsilon = 1e^{-8}$
Up-sampling operation in generator	Bilinear interpolation
number of filters in each generator convolution layer	128
η	$2e^{-4}$
λ	10

Table 6.2: Hyper-parameter settings when training model $\mathcal{S}_{2\times}$, $\mathcal{S}_{6\times}$.

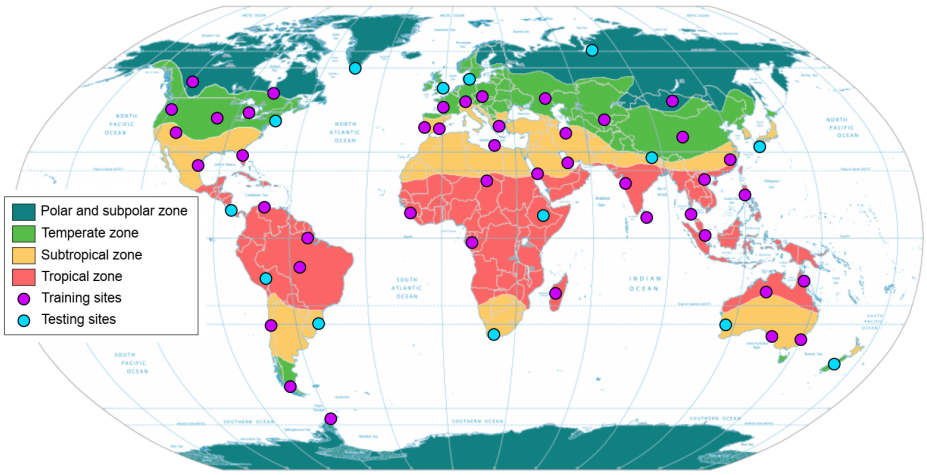


Figure 6.1: Locations of all training and testing Sentinel-2 tiles [LBDG+18]. There are 60 tiles in total, including 45 for training and 15 for testing. All tiles are randomly evenly selected on the globe and cover all climate zones.

adding sharpness to output. All convolution layers in this thesis have 128 filters, which is the same with the comparison method DSen2.

6.2 Dataset Preparation

6.2.1 Data collection

To train the super-resolution model $\mathcal{S}_{2\times}$, $\mathcal{S}_{6\times}$, the MSI products from Sentinel-2A&2B are used. All available Sentinel-2 MSI products are published on the Copernicus services data hub [4] and can be downloaded free of charge. In this thesis, we mainly super-resolve the data of format both level-lC and level-2A.

All Level-lC MSI products are composed of 100x100 km^2 tiles (ortho-images in UTM/WGS84 projection), which are created by applying a Digital Elevation Model (DEM) to project the image in cartographic geometry. Per-pixel radiometric measurements are provided in ToA reflectances along with the parameters to transform them into radiances. Level-1C products are resampled with a constant Ground Sampling Distance (GSD) of 10m, $20m$, or $60m$ depending on the native resolution of the different spectral bands, see Table 2.1. The relationship between a Level-2A and a Level-1C MSI product is that the former provides BoA reflectance images, which are derived from the later through Sentinel-2 Toolbox[5].

To train and test our model, we downloaded 60 level-1C products as [LBDG+18], including 45 tiles (30 from Sentinel-2A and 15 from Sentinel-2B) for training and 15 tiles (8 from Sentinel-2A and 7 from Sentinel-2B) for testing, see Figure 6.1. Those tiles are evenly distributed on the globe to cover more climate zones or land-cover types. To improve the numerical stability and train the model better, we only select products without undefined (black) pixels. However, we do not exclude those tiles with clouds occlusion (both dense and cirrus clouds) in our training and testing dataset, because usually, a super-resolution model being robust to clouds occlusion is more desired.

6.2.2 Data preprocessing

According to the way to create training datasets by applying Wald's protocol on Sentinel-2 MSI products, explained in Chapter 4, we need to degraded the original Sentinel-2 tiles with a desired scale ratio $S(2$ or $6)$ first. The degradation process can be divided into two stages. First, blurring the image using a decimation filter that can emulate the modulation transform function (MTF) of Sentinel-2 image sensors. This stems from that the scale invariance of super-resolution model (*e.g.*, migrating a $40m \mapsto 20m$ super-resolution model to $20m \mapsto$ 10m tasks) can be strengthened if the decimation filter matches to the sensor's MTF [KAC09]. Gaussian filter f_G with standard deviation

[4] https://scihub.copernicus.eu/
[5] https://sentinel.esa.int/web/sentinel/toolboxes/sentinel-2

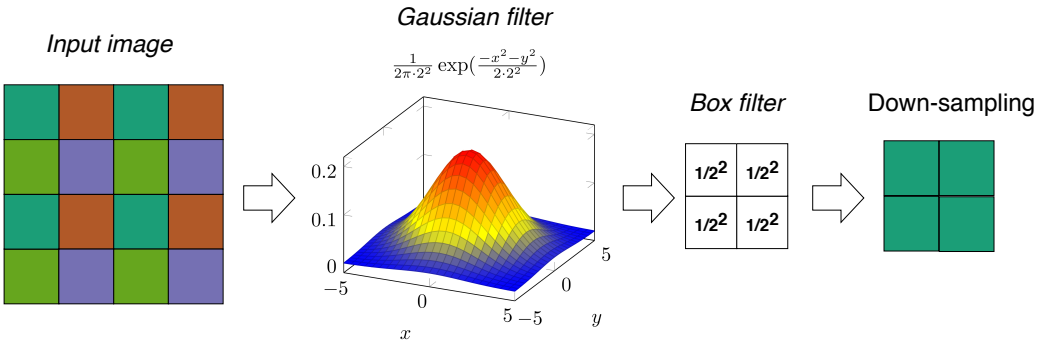


Figure 6.2: Illustration of Sentinel-2 tiles $2\times$ degradation. The input is each single channel of the original Sentinel-2 tile. Each grid represents a pixel and different color is used to distinguish their position. All channels are degraded with the same way. First, a Gaussian filter with $\sigma = 2$ (determined by the degradation scale) is applied to blur the image. Then, a box average filter traverses each image to average the pixel values inside a box of size 2×2 and finally yields the $2\times$ degraded output.

$\theta = \frac{1}{S}$ is used in this thesis. Second, down-sampling each band by averaging over a $S \times S$ box, see Figure 6.2 for an example of down-sampling with $S = 2$.

scale	phase	No. tiles	No. patches in each tile
2x	Train	45	$8000 \times (4 + 6) \times 32 \times 32$
	Test	15	$52441 \times (4 + 6) \times 32 \times 32$
6x	Train	45	$20250 \times (4 + 6 + 2) \times 96 \times 96$
	Test	15	$676 \times (4 + 6 + 2) \times 96 \times 96$

Table 6.3: Configurations of both training and testing dataset. 4, 6, 2 means the number of $10m$, $20m$, $60m$ bands respectively.

Because Sentinel-2 MSI tiles are too large ($\simeq$ 800M per product) to fit into a GPU, we randomly select patches in each tile to create the training dataset. Similarly, in the testing phase, a tile is split into many test patches, and we predict the corresponding HR patches and recompose the entire HR tile with them. To avoid border distortion, those test patches are set to be boarder inter-overlapping. By a smaller patch size, lager effective batch size can be used. Experiments to investigate the effect of batch size on model accuracy is explained in Section 6.7. Further, Zhang *et al.* [ZGMO18] has also tested the effect of patch size on self-attention module: model with self-attention module at middle-to-high level of feature maps (32×32 or 64×64) perform better than at lower level of feature maps (8×8 or 16×16). To summarize, we use 16×16 $20m$ band patches and pre-upsampling them to 32×32 for model $\mathcal{S}_{2\times}$ and 16×16 60m band patches and pre-upsampling them to 96×96 for model $\mathcal{S}_{6\times}$. More detailed configurations of training and testing datasets are shown in Table 6.3.

Sentinel-2 MSI products are published in the format of SAFE[6] where the imagery of each band is put in a separate JPEG2000 file. In this thesis, we use the library

[6] https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/data-formats

Band	B1	B2	B3	B4	B5	B6	B7	B8	B8a	B9	B10	B11	B12
Spatial resolution(m)	60	10	10	10	20	20	20	10	20	60	60	20	20
μ_b^{L1C}	1627	1302	1306	1400	1505	2122	2428	2350	2634	821	-	1985	1298
std_b^{L1C}	1297	1491	1293	1348	1441	1400	1447	1416	1471	702	-	1199	1001
μ_b^{L2A}	-	1271	1171	911	1614	2261	2509	2611	2684	-	-	2192	1571
std_b^{L2A}	-	1776	1676	1671	1749	1641	1634	1695	1622	-	-	1377	1242

Table 6.4: The statistics of 13 Sentinel-2 spectral bands in the level-1C and level-2A training dataset. μ_b^{L1C} or μ_b^{L2A} means the mean intensity of pixels in band b, std_b^{L1C} or std_b^{L2A} means the standard deviation of pixel intensities in band b.

GDAL[7] to extract band data, and the pixel value in each band is the product of raw surface reflectance and a constant 10000, so in most case, the pixel value of each band is in the range $(0, 10000)$. However, some pixels (*e.g.*, pixels of clouds, snow, ice mountain *etc.*) may exceed 10000. With in mind that unscaled inputs always result in a model with large weights that is often unstable and of high generalization error, whereas unscaled references often result in exploding gradients causing the learning process to fail [B+95], standard score is used to normalize each training band.

$$\bar{x}_{bn} = \frac{x_{bn} - \mu_b}{std_b}$$

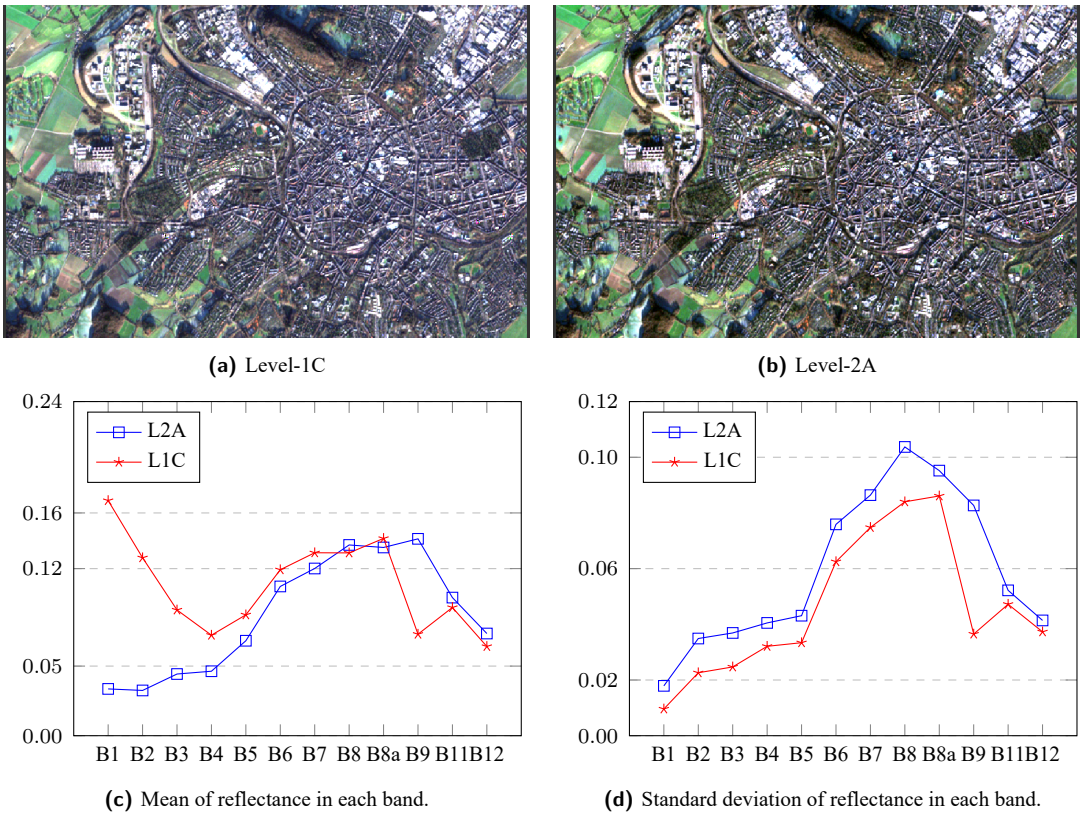
where b is the index of each spectral band, n is index of each pixel in each band, μ_b is the mean value of band b, std_b is the standard deviation of band b. Standard score is used instead of min-max normalization to avoid the affect of too large outliers.

Moreover, because of sensitive to different spectrum intervals, each band in a Sentinel-2 MSI product always have different statistics. For instance, $20m$ bands consistently have higher reflectance and associated larger variance than 10m bands [PLPD18]. The performance of super-resolution model is also highly correlated to the statistics of each band. In general, the higher reflectance intensity and the larger variance a band has, the more difficult the super-resolution is, see Table 6.6. In addition to different spectral responses, the format conversion also has a strong influence on the band statistics. Figure 6.3 use the panorama of Aachen to show the difference between Level-1C and Level-2A Sentinel-2 MSI products.

6.3 Evaluation Criteria

In this section, the metrics to quantitatively evaluate the model are explained. Note that HR reference of original data don't exit in reality, and many HR outputs with different configurations are possible for a same LR input. For this reason, the super-resolution of LR $\mapsto$ HR is an ill-posed (a.k.a. ill-determined) inverse problem. Therefore, many metrics are used in this thesis for a more comprehensive assessment.

[7] https://gdal.org/python/index.html



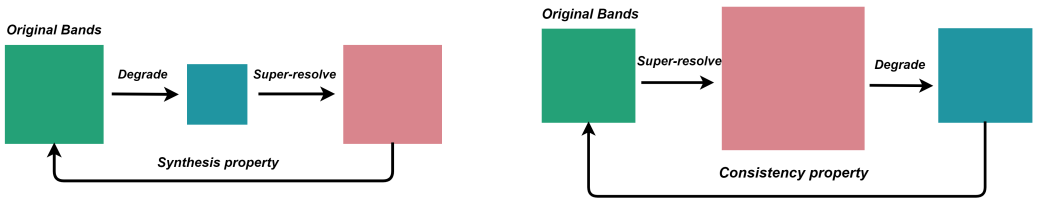
(a) Level-1C

(b) Level-2A

(c) Mean of reflectance in each band.

(d) Standard deviation of reflectance in each band.

Figure 6.3: Comparison between Sentinel-2 tiles of format level-1C and level-2A. (a-b) are both panorama of the city Aachen (B4, B3, B2 as RGB). (b) Level-2A has removed the energy reflected by atmosphere from (a) Level-1C. (c-d) are the statistical comparison between (a) and (b). (c) shows that except B8, B9, B11 and B12, bands of Level-2A have lower reflectance. (d) shows that all bands of level 2A have higher standard deviation.



(a) Synthesis property

(b) Consistency property

Figure 6.4: Illustration of evaluating the synthesis and consistency property of a super-resolution model.

Wald *et al.* [WRM97] has stated that two properties a pan-sharpened image must have, including *Synthesis property* and *Consistency property*. In this thesis, the super-resolved Sentinel-2 tiles are also evaluated based on the two properties, see Figure 6.4a and 6.4b, which can both be quantitatively represented by metrics explained in Section 6.3.1.

1. Synthesis property: The degraded images super-resolved to the original scale should be as identical as possible to the original observation.

2. Consistency property: The super-resolved images degraded to the original scale should be as identical as possible to the original observation.

6.3.1 Quantitative metrics

$\mathcal{RMSE}$ means rooted mean square error. Although a super-resolution model is desired to produce results with high perceptual quality, lower pixel-wise distance to ground truth is also important to avoid information bias or distortions induced by super-resolution model. Therefore, $\mathcal{RMSE}$ is one of the main evaluation metrics in this thesis and can be given by

$$\mathcal{RMSE}(HR, \hat{HR}) = \sqrt{\frac{1}{|N|}\sum_{n\in N}\left(HR_n - \hat{HR}_n\right)^2}$$

where HR is HR reference, $\hat{HR}$ is the output of a super-resolution model, N denotes the number of pixels in HR or $\hat{HR}$.

$\mathcal{PSNR}$ (peak signal to noise ratio) is highly correlated to $\mathcal{RMSE}$ and given by

$$\mathcal{PSNR} = 10 \cdot \log(\frac{MAX^2}{MSE})$$

where MAX is the maximal possible pixel value in an image, *e.g.*, for a nature image (radiometric resolution is 8 bits), $MAX = 255$. MSE is mean square error and equals to $\mathcal{RMSE}(HR, \hat{HR})^2$.

Compared with $\mathcal{PSNR}$, $\mathcal{SSIM}$ (Structural similarity index) is more consistent with human visual perception. It takes luminance, contrast and structure of an image into account and is calculated on sliding windows of both HR and $\hat{HR}$. The measure between two windows x and y is given by

$$\mathcal{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)\cdot(2\theta_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\theta_x^2 + \theta_y^2 + c_2)}$$

where μ_x, μ_y is the mean of window x and y respectively, θ_{xy} is the covariance of x and y, θ_x, θ_y are the variance of x and y, $c_1 = (k_1 \cdot MAX)^2$, $c_2 = (k_2 \cdot MAX)^2$ and $k_1 = 0.01$ $k_1 = 0.03$ by default. Note that UIQ, another common evaluation metric, is a special case of $\mathcal{SSIM}$ when $c_1 = c_2 = 0$.

$\mathcal{SRE}$ is the ratio of power of signal to error, and it is given in decibels by

$$\mathcal{SRE} = 10\log_{10}\frac{\mu_{HR}^2 \cdot N}{\left\|HR_1 - \hat{HR}_2\right\|^2}$$

where μ_{HR} denotes the mean of HR reference.

$\mathcal{ERGAS}$ [Wal00] calculates the amount of spectral/spatial distortion in the enhanced image based on the mean square error and is given by

$$\mathcal{ERGAS} = \frac{100}{r^2}\sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(\frac{\sqrt{\sum_{n=1}^{N}(\hat{HR}_{n,l} - HR_{n,l})^2/N}}{\sum_{n=1}^{N}\hat{HR}_{n,l}/N}\right)^2}$$

where r is the scale ratio of high resolution image to low resolution image, L demotes the number of bands and $\hat{HR}_{n,l}$ denotes the pixel n in band l.

$\mathcal{SAM}$ calculates the spectral similarity between two vectors as an angle. $\mathcal{SAM}$ between two entire images is the average of all the angles for each pixel. It is given in degrees by

$$\mathcal{SAM}(\hat{HR}, HR) = \frac{1}{N}\sum_{n=1}^{N}\arccos\left(\frac{\sum_{l=1}^{L}\hat{HR}_{n,l}HR_{n,l}}{\sqrt{\sum_{l=1}^{L}\hat{HR}_{n,l}^2\sum_{l=1}^{L}HR_{n,l}^2}}\right).$$

However, all metrics above are more or less sensitive to the brightness of input images to evaluate. Sometimes, a constant shift in the intensity of every pixel quickly ramps up and may be even more detrimental than generating images with lots of artifacts. Therefore in this thesis, we propose a metric $\mathcal{BPSNR}$ (brightness invariant PSNR) modified from $\mathcal{CPSNR}$[8]. Before evaluating the similarity between a generated images and a reference image, the intensity of the generated image are equalized, so that the average pixel brightness of both images are equal.

We assume that the pixel-intensities are represented as real numbers $\in [0, 1]$ for any given image. For a pair of images $(HR, \hat{HR})$ to evaluate, the bias in brightness b is give by

$$b = \frac{1}{N}\left(\sum_{x,y\in HR} HR(x,y) - \hat{HR}(x,y)\right).$$

Next, the corrected mean square error cMSE of $\hat{HR}$ w.r.t. HR is

$$cMSE(HR, \hat{HR}) = \frac{1}{N}\sum_{x,y\in HR}\left(HR(x,y) - \left(\hat{HR}(x,y) + b\right)\right)^2$$

which results in a brightness invariant peak signal to noise ratio.

$$\mathcal{BPSNR}(HR, \hat{HR}) = -10\log_{10}\left(\frac{MAX^2}{cMSE(HR, \hat{HR})}\right)$$

For $\mathcal{RMSE}$, $\mathcal{ERGAS}$ and $\mathcal{SAM}$, the optimal value is 0, whereas the higher value means a better performance in $\mathcal{SRE}$, $\mathcal{PSNR}$, $\mathcal{BPSNR}$ and $\mathcal{SSIM}$.

[8] https://kelvins.esa.int/proba-v-super-resolution/scoring/

6.3.2 Visual assessment

Note that visual assessment is also important for a GAN-based super-resolution model besides quantitative evaluation. Unlike a traditional learning-based method optimize the model with PSRN-oriented loss functions, *e.g.*, L1 and L2 losses, a GAN-based model also optimizes its generator to fool the discriminator. Yet, no well-defined metrics can comprehensively show the effect of adversarial loss (calculated by the discriminator) on the generated images. Therefore, visual assessment plays a vital role to evaluate a GAN-based super-resolution model, because the nature of enhancement can be clearly recognized and artifacts visually identified.

6.4 Results of Level-1C Super-resolution

This section shows the performance of $\mathcal{S}_{2\times}^{L1C}$ and $\mathcal{S}_{6\times}^{L1C}$ on Sentinel-2 level-1C tiles super-resolution. Note the two models in this section are both trained with an effective mini-batch size of 512 and the impact of training with larger or smaller effective batch size is discussed in Section 6.7. The comparison methods include: 1) naive bicubic interpolation, 2)ATPRK [WSAPI15], 3) SupReME [LBDBS17], 4) Superres [Bro17], 5) DSen2 [LBDG$^+$18]. The results of ATPRK, SupReME, and Superres refer from the paper [LBDG$^+$18], where some metrics (ERGARS, SSIM, PSNR and bPSNR) are not tested. The performance of naive bicubic interpolation is tested with the build-in image resize function in the library OpenCV. DSen2 is also a pre-upsampling learning-based super-resolution method with the same 6 residual blocks with our method, and its performance is tested using the model published on the repository[9].

6.4.1 Synthesis property evaluation

As explained in Section 6.3, quantitative evaluation of synthesis property is carried out at the degraded scale, *i.e.*, the model $\mathcal{S}_{2\times}$ is evaluated on the task to super-resolve $40 \mapsto 20m$ and $\mathcal{S}_{6\times}$ is evaluated on the task to super-resolve $360 \mapsto 60m$. Therefore, all test patches, see Table 6.3, should be degraded synthetically before evaluation with the same way to generate the training patches shown in Figure 6.2.

- **L1C $2\times$ super-resolution**: Table 6.5 and Table 6.6 shows the synthesis property of $\mathcal{S}_{2\times}^{L1C}$ when super-resolving $2\times$ degraded $20m$ bands to the original scale. Table 6.5 shows the average performance over all 6 $20m$ bands. We can see that our pre-trained $\mathcal{S}_{2\times}^{L1C}$ achieves the best performance consistently over all evaluation metrics. Note because the value of each pixel in a Sentinel-2 tile is

[9] https://github.com/lanha/DSen2/tree/master/models

the product of original radiance reflectance and a constant 10000, a RMSE of 1 in each pixel means a reflectance error of $1e^{-4}$. Compared with the native bicubic interpolation, our model has reduced the RMSE by 72%. Compared with DSen2, the improvement of our method is relatively small. However, a 0.1 improvement of PSNR can be considered as an effective improvement in the problem of image super-resolution. Table 6.6 is the performance per band. The performance of RMSE in each band is highly correlated to the pixel intensity, *i.e.*, usually, the higher pixel value a bands has, the higer RMSE the super-resolved output has, *e.g.*, the RMSE of B6, B7 and B8a is higher than other bands within all methods. The example to statistically compare each bands is shown in Figure 6.3. Compared with RMSE, the performance of SRE in each band is relatively more balanced.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
Bicubic	125.69	25.64	1.22	3.48	0.82	44.9998	45.0003
ATPRK	116.2	25.7	1.68	-	-	-	-
SupReME	69.7	29.7	1.26	-	-	-	-
Superres	66.2	30.4	1.02	-	-	-	-
DSen2	35.85	35.94	0.78	1.07	0.9322	55.5416	55.9317
$\mathcal{S}_{2\times}^{L1C}$	**34.99**	**36.19**	**0.75**	**1.03**	**0.9336**	**55.7756**	**56.3358**

Table 6.5: Average performance of super-resolving 6 $20m$ bands in B by pre-trained generator $\mathcal{S}_{2\times}^{L1C}$ in sense of synthesis property. Our method achieved the best result consistently over all evaluation metrics.

	B5	B6	B7	B8a	B11	B12
	RMSE					
Bicubic	101.23	117.29	129.52	137.73	127.65	118.73
ATPRK	89.4	119.1	136.5	147.4	113.3	91.7
SupReME	48.1	70.2	78.6	82.9	76.5	61.7
Superres	50.2	66.6	76.8	82.0	66.9	54.5
DSen2	27.74	32.68	36.07	38.02	36.22	34.55
$\mathcal{S}_{2\times}^{L1C}$	**27.48**	**32.27**	**35.58**	**37.46**	**35.56**	**33.68**
	SRE					
Bicubic	25.46	25.69	25.69	25.73	25.94	25.70
ATPRK	26.6	26.9	26.7	26.6	24.7	22.7
SupReME	31.2	31.0	31.0	31.2	27.9	26.1
Superres	31.3	31.7	31.4	31.4	29.1	27.2
DSen2	36.15	36.33	36.37	36.49	36.45	35.97
$\mathcal{S}_{2\times}^{L1C}$	**36.26**	**36.44**	**36.49**	**36.62**	**36.66**	**36.22**

Table 6.6: Per-band performance of super-resolving each $20m$ band in B by pre-trained model $\mathcal{S}_{2\times}^{L1C}$ in sense of synthesis property. Our method has achieved the best result consistently over all metrics and over all $20m$ bands.

- **L1C** $6\times$ **super-resolution**: Table 6.7 and Table 6.8 shows the synthesis property of $\mathcal{S}_{6\times}^{L1C}$ when super-resolving $6\times$ degraded $60m$ bands to original scale. Table 6.7 shows the average performance over all $60m$ bands. Our method has

achieved the state-of-the-art performance consistently over all evaluation metrics. Compared with $2\times$ super-resolution, $6\times$ super-resolution has better RMSE. There are two possible reasons: 1) compared with $20m$ bands, $60m$ bands have a smaller average pixel intensity, see Table 6.4. 2) There are only 2 $60m$ bands but there are 6 $20m$ bands. Table 6.8 is the performance per band.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
Bicubic	161.85	19.79	1.78	7.30	0.36	37.6785	37.6785
ATPRK	145.1	20.4	1.62	-	-	-	-
SupReME	85.7	24.8	0.98	-	-	-	-
Superres	100.2	22.8	1.42	-	-	-	-
DSen2	28.11	34.47	0.36	1.38	0.8953	52.4984	52.1305
$\mathcal{S}_{6\times}^{L1C}$	**26.80**	**34.98**	**0.34**	**1.29**	**0.8991**	**52.9451**	**52.2735**

Table 6.7: Average performance of super-resolving 2 $60m$ bands in C by pre-trained generator $\mathcal{S}_{6\times}^{L1C}$ in sense of synthesis property. Our method achieved the best result consistently over all evaluation metrics.

	B1	B9	B1	B9
	RMSE		SRE	
Bicubic	169.54	158.12	22.43	19.79
ATPRK	162.9	127.4	22.8	18.0
SupReME	114.9	56.4	25.2	24.5
Superres	107.5	92.9	24.8	20.8
DSen2	29.28	27.51	37.25	34.44
$\mathcal{S}_{6\times}^{L1C}$	**27.60**	**26.18**	**37.77**	**34.95**

Table 6.8: Per-band performance of super-resolving each $60m$ band by pre-trained model $\mathcal{S}_{6\times}^{L1C}$ in sense of synthesis property, including $B1$ and $B9$. Our method achieved the best result consistently over all evaluation metrics and over all $60m$ bands.

6.4.2 Consistency property evaluation

Consistency property is quantitatively evaluated at original scale, see Figure 6.4b, which distinguish our method from the previous one that only do visual assessment at original scale.

- **L1C** $2\times$ **super-resolution**: Table 6.9 and Table 6.10 show the consistency property of $\mathcal{S}_{2\times}^{L1C}$ when degrading the $2\times$ super-resolved $20m$ bands back to the original scale. Table 6.10 shows the average performance over 6 $20m$ bands. Our method has achieved the best performance consistently over all evaluation metrics. In particular, compared to Bicubic interpolation, our method has reduced the RMSE by 85%; compared to another learning-based baseline DSen2, our method has reduced the RMSE by 28%. Table 6.9 show the performance per band, our method has achieved the best performance in RMSE and SRE over all bands.

	B5	B6	B7	B8a	B11	B12
	RMSE					
Bicubic	28.50	33.23	36.78	38.95	35.50	32.73
DSen2	6.34	6.20	6.53	6.63	6.10	5.67
$\mathcal{S}_{2\times}^{L1C}$	**4.31**	**4.58**	**4.77**	**4.86**	**4.40**	**4.09**
	SRE					
Bicubic	36.47	36.65	36.63	36.70	37.14	37.01
DSen2	50.06	51.49	51.81	52.21	52.64	52.42
$\mathcal{S}_{2\times}^{L1C}$	**53.21**	**54.20**	**54.63**	**55.00**	**55.61**	**55.43**

Table 6.9: Per-band performance of super-resolving each $20m$ band in B by pre-trained model $\mathcal{S}_{2\times}^{L1C}$ in sense of consistency property.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
Bicubic	34.96	36.70	0.40	0.95	0.9849	56.1555	55.6129
DSen2	5.91	52.10	0.08	0.18	0.9899	71.9820	72.3170
$\mathcal{S}_{2\times}^{L1C}$	**4.28**	**55.11**	**0.07**	**0.1251**	**0.9901**	**74.9196**	**75.4208**

Table 6.10: Average performance of super-resolving each $20m$ bands in B by pre-trained generator $\mathcal{S}_{2\times}^{L1C}$ in sense of consistency property.

- **L1C** $6\times$ **super-resolution**: Table 6.11 and Table 6.12 shows the consistency property of $\mathcal{S}_{6\times}^{L1C}$ when degrading the $6\times$ super-resolved original $60m$ bands back to the original scale. Table 6.11 shows the average performance over 6 $20m$ bands and Table 6.12 is the performance per band. It is easy to find that 1) For $2\times$ super-resolution, learning-based methods can significantly improve both of synthetic and consistency property. 2) For $6\times$ super-resolution, learning-based methods have better synthetic property but worse consistency property than naive Bicubic interpolation. Note that our model has achieved better performance than another learning-based method Dsen2 over all evaluation metrics.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
Bicubic	**14.02**	**41.04**	**0.18**	**0.63**	**0.9919**	**58.9763**	**58.5900**
DSen2	24.69	36.31	0.20	1.03	0.9710	54.1462	54.6915
$\mathcal{S}_{6\times}^{L1C}$	22.35	37.41	0.17	0.92	0.9798	55.3258	55.5437

Table 6.11: Average performance of super-resolving 2 $60m$ bands in C by pre-trained generator $\mathcal{S}_{6\times}^{L1C}$ in sense of consistency property.

	B1	B9	B1	B9
	RMSE		SRE	
Bicubic	**14.49**	**13.68**	**43.71**	**41.04**
DSen2	29.24	23.73	37.64	36.31
$\mathcal{S}_{6\times}^{L1C}$	26.56	21.61	38.72	37.42

Table 6.12: Per-band performance of super-resolving each $60m$ band in C by pre-trained model $\mathcal{S}_{6\times}^{L1C}$ in sense of consistency property.

6.5 Results of Level-2A Super-resolution

This section shows the performance of $\mathcal{S}_{2\times}^{L2A}$ on Sentinel-2 Level-2A tiles super-resolution. Compared with Level-1C tiles, Level-2A tiles are atmospherically corrected. Only $20m$ bands are considered in this section that are enough to show the effectiveness of our methods. Similar to the case of Level-1C in Section 6.4, this section only presents the performance of models trained with an effective mini-batch size 512. The comparison methods include naive bicubic interpolation, DSen2 and DSen2-L2A. The Sentinel-2 L2A dataset is created as explained in Section 6.2. The model DSen2 is downloaded from the repository[10] and model DSen2-L2A is trained with the new Sentinel-2 L2A dataset and the code in the repository[11].

6.5.1 Synthesis property evaluation

- **L2A $2\times$ super-resolution**: Table 6.13 and Table 6.14 shows the synthesis property of $\mathcal{S}_{2\times}^{L2A}$ when super-resolving $2\times$ degraded $20m$ bands to original scale. Table 6.13 shows the average performance over all $20m$ bands. Similar to L1C tiles super-resolution, a RMSE of 1 on the L2A testing tiles means a average reflectance error of $1e^{-4}$. Compared with the native bicubic interpolation, our model has reduced the RMSE by 72%. Compared with the learning-based baseline DSen2-L2A, our model reduces the RMSE by 4%. Compared with super-resolving Level-1C tiles, super-resolving Level-2A tiles are more difficult for both naive Bicubic interpolation and our model. This can be reflected by the worse performance in Table 6.13 compared to Table 6.5. Owing to the performance difference between DSen2 and DSen2-L2A, retraining a model with Level-2A dataset can substantially improve the ability to super-resolve Level-2A tiles. Table 6.14 is the performance of each band. Our pre-trained model has achieved the best performance on RMSE and SRE consistently over all spectral bands.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
Bicubic	149.75	24.51	2.10	3.93	0.8181	41.1114	41.1114
DSen2	47.60	34.27	1.83	1.32	0.91	50.85	52.73
DSen2-L2A	43.00	35.17	1.54	1.19	0.9265	51.7563	**54.1827**
$\mathcal{S}_{2\times}^{L2A}$	**41.45**	**35.55**	**1.46**	**1.14**	**0.9275**	**52.1107**	54.1475

Table 6.13: Average performance of super-resolving 6 $20m$ bands in B by pre-trained generator $\mathcal{S}_{2\times}^{L2A}$ in sense of synthesis property.

[10] https://github.com/lanha/DSen2/model
[11] https://github.com/lanha/DSen2/training

	B5	B6	B7	B8a	B11	B12
			RMSE			
Bicubic	133.87	149.90	160.27	165.71	152.44	142.35
DSen2	41.01	45.58	48.57	50.23	47.91	46.14
DSen2-L2A	37.07	41.99	44.97	46.24	43.51	41.63
$\mathcal{S}_{2\times}^{L2A}$	**36.14**	**40.86**	**43.64**	**44.83**	**42.00**	**40.07**
			SRE			
Bicubic	23.74	24.19	24.34	24.52	24.94	24.84
DSen2	33.91	34.39	34.57	34.73	34.73	34.30
DSen2-L2A	34.59	34.99	35.16	35.39	35.57	35.22
$\mathcal{S}_{2\times}^{L2A}$	**34.83**	**35.26**	**35.45**	**35.69**	**35.91**	**35.59**

Table 6.14: Per-band performance of super-resolving each $20m$ band in B by pre-trained model $\mathcal{S}_{2\times}^{L2A}$ in sense of synthetic property.

6.5.2 Consistency property evaluation

- **L2A** $2\times$ **super-resolution**: Table 6.16 and Table 6.15 shows the consistency property of $\mathcal{S}_{2\times}^{L2A}$ when degrading the $2\times$ super-resolved $20m$ L2A bands back to the original scale. Table 6.15 shows the average performance over 6 $20m$ bands. Compared with naive Bicubic interpolation, our method has reduced the RMSE by 85%. Compared with learning-based DSen2-L2A, our method has reduced the RMSE by 37% Table 6.16 is the performance per band. Our method has achieved the best performance consistently over all evaluation metrics and all spectral bands.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
Bicubic	41.81	35.83	0.72	1.08	52.25	0.9844	52.3023
DSen2-L2A	8.95	50.00	**0.28**	0.23	0.9819	66.1935	66.6093
$\mathcal{S}_{2\times}^{L2A}$	**5.65**	**53.87**	**0.28**	**0.14**	**0.9821**	**70.2786**	**70.8070**

Table 6.15: Average performance of super-resolving 6 $20m$ bands in B by pre-trained generator $\mathcal{S}_{2\times}^{L2A}$ in sense of consistency property.

6.6 Experiment on Adversarial Losses

6.6.1 Quantitative evaluation

This thesis also investigates the effect of adversarial losses on the pre-trained super-resolution model. Based on the pre-trained $\mathcal{S}_{2\times}^{L1C}$, we further train four GANs with four different adversarial losses. Table 6.17 shows their synthesis property and Table 6.18

	B5	B6	B7	B8a	B11	B12
	RMSE					
Bicubic	37.89	42.61	45.66	47.03	42.55	39.36
DSen2-L2A	8.95	9.64	10.14	10.27	9.15	8.32
$\mathcal{S}_{2\times}^{L2A}$	**6.33**	**6.03**	**6.27**	**6.44**	**5.75**	**5.30**
	SRE					
Bicubic	34.75	35.16	35.29	35.49	36.14	36.15
DSen2-L2A	48.08	48.66	48.90	49.21	50.06	50.32
$\mathcal{S}_{2\times}^{L2A}$	**51.49**	**52.96**	**53.16**	**53.30**	**54.15**	**54.20**

Table 6.16: Per-band performance of super-resolving each $20m$ band in B by pre-trained model $\mathcal{S}_{2\times}^{L2A}$ in sense of consistency property.

shows their consistency property when super-resolving $2\times$ degraded $20m$ L1C bands to the original scale.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
pre-trained $\mathcal{S}_{2\times}^{L1C}$	**34.99**	**36.19**	**0.75**	**1.03**	**0.9336**	**55.7756**	**56.3358**
+ WGAN-GP	39.86	35.08	0.88	1.17	0.9269	54.6377	55.7148
+ vanilla GAN	61.28	31.18	1.39	1.93	0.8664	50.9394	50.7452
+ relativistic GAN	98.35	26.99	2.19	3.13	0.7860	46.7731	46.5310
+ hinge loss GAN	56.07	31.95	1.32	1.76	0.8824	51.7295	51.7466

Table 6.17: The effect of four adversarial losses on pre-trained $\mathcal{S}_{2\times}^{L1C}$ in sense of synthetic property. The definition of the adversarial loss in vanila GAN, relativistic GAN, WGAN-GP, and GAN with hinge loss is explained in see Section 5.2.2.

	RMSE	SRE	SAM	ERGAS	SSIM	PSNR	bPSNR
pre-trained $\mathcal{S}_{2\times}^{L1C}$	**4.28**	**55.11**	**0.07**	**0.1251**	**0.9901**	**74.9196**	**75.4208**
+ WGAN-GP	6.56	50.98	0.12	0.19	0.9893	70.8014	71.2999
+ vanilla GAN	45.94	33.67	1.09	1.39	0.9352	53.2586	53.1678
+ relativistic GAN	58.34	31.12	1.47	1.89	0.8979	51.3161	51.3626
+ hinge loss GAN	35.33	35.51	0.94	1.23	0.9528	55.6350	55.7038

Table 6.18: The effect of adversarial losses on pre-trained $\mathcal{S}_{2\times}^{L1C}$ in sense of consistency property.

With super-resolution methods advanced, perceptual quality oriented and reconstruction accuracy oriented methods have become two distinct research trends. Figure 6.5 simply classifies some learning-based super-resolution methods to two clusters, including high PSNR/SSIM methods and high visual quality methods [BMT+18]. GAN-based methods [LTH+17, WYW+18] have achieved visual pleasant results but significantly lower PSNR/SSIM that is even worse than naive interpolation. Our experiments also showed the similar results, *i.e.*, adversarial training with a discriminator does't improve the pretrained generator on both synthesis and consistency property, see Table 6.17 and Table 6.18. Examples used for visual assessment are demonstrated in Section 6.6.2.

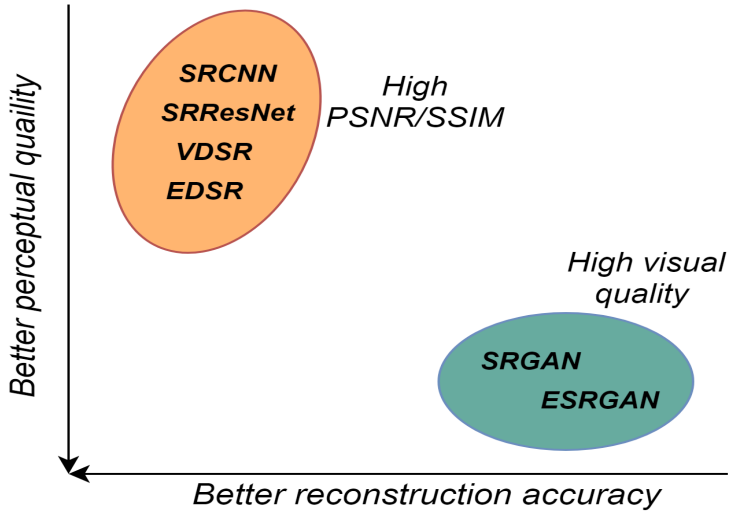


Figure 6.5: Two distinct clusters of learning-based super-resolution methods, high PSNR/SSIM methods and high perceptual quality methods. The plotted methods include SRCNN [DLHT14], SRResNet [LTH+17], VDSR [KKLML16a], EDSR [LSK+17], SRGAN [LTH+17], ESRGAN [WYW+18].

6.6.2 Visual assessment

In addition to quantitative evaluation, visual assessment is also indispensable for the evaluating a image super-resolution model, by which the super-resolution effects and artifacts can be visually identified. Figure 6.6 (or Figure 6.7) shows some super-resolved example patches of level-1C (or level-2A) Sentinel-2 $20m$ bands. Figure 6.10 shows some super-resolved example patches of level-1C Sentinel-2 $60m$ bands.

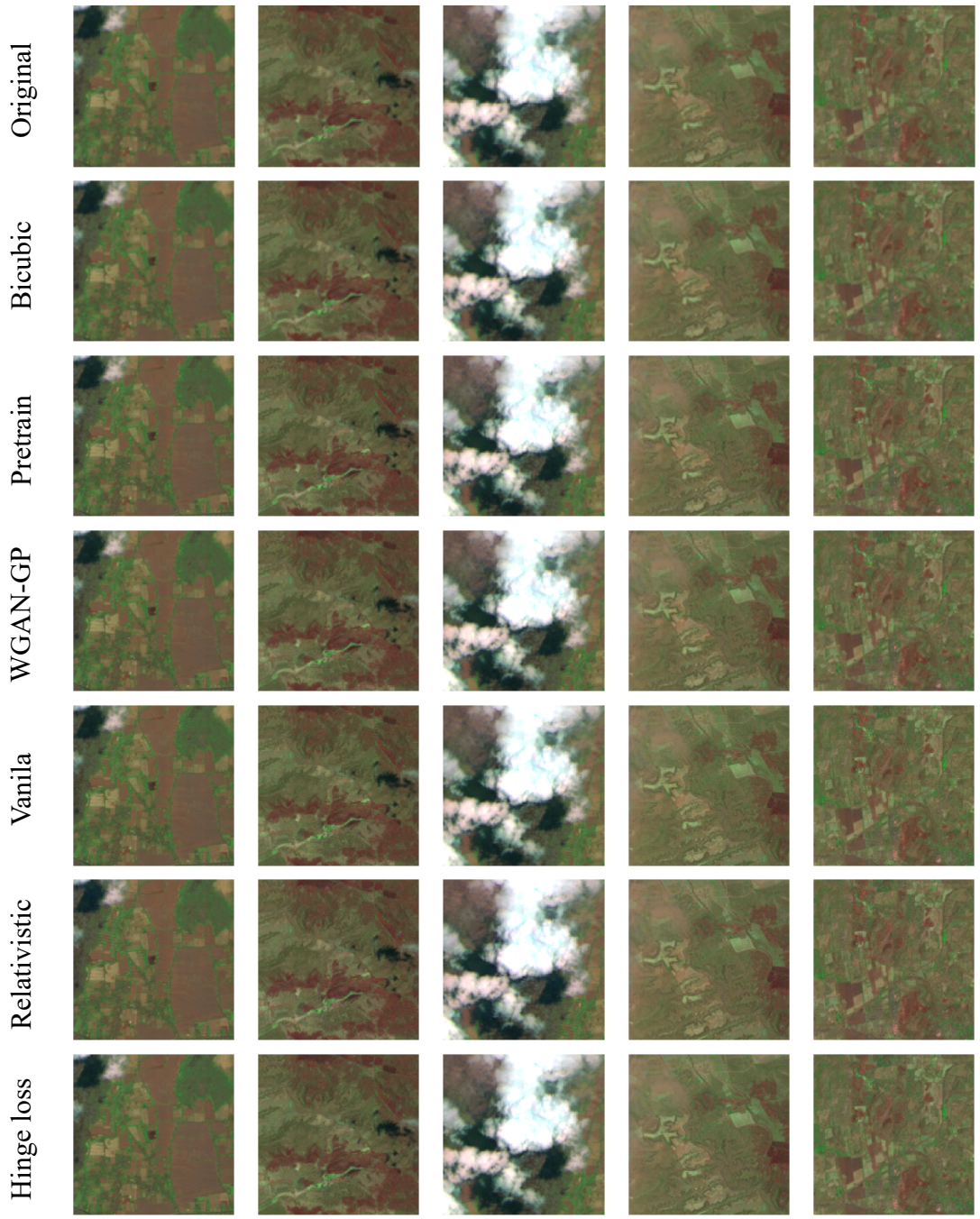


Figure 6.6: Examples of super-resolving the original Sentinel-2 level-1C $20m$ bands to 10m GSD. Bands B12, B8a and B5 are used as RGB bands for visualization. From top to bottom are example patches of original Sentinel-2 level-1C bands, super-resolution output of Bicubic interpolation, pre-trained $\mathcal{S}_{2\times}^{L1C}$, vanila GAN, WGAN-GP, relativistic GAN, GAN with hinge loss.

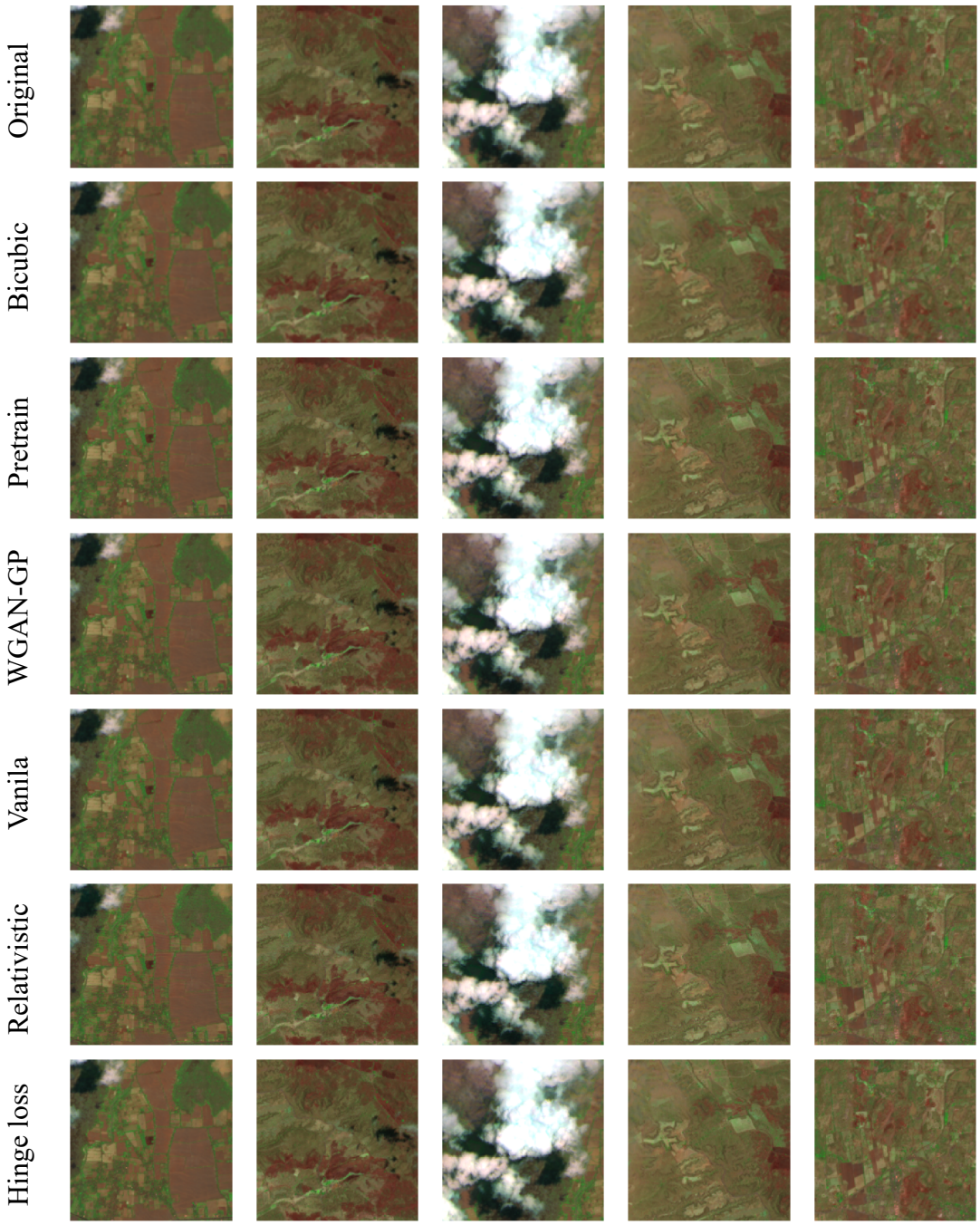


Figure 6.7: Examples of super-resolving the original Sentinel-2 level-2A $20m$ bands to $10m$ GSD. Bands B12, B8a and B5 are used as RGB bands for visualization. From top to bottom are example patches of original Sentinel-2 level-2A bands, super-resolution output of Bicubic interpolation, pre-trained $\mathcal{S}_{2\times}^{L1C}$, vanila GAN, WGAN-GP, relativistic GAN, GAN with hinge loss.

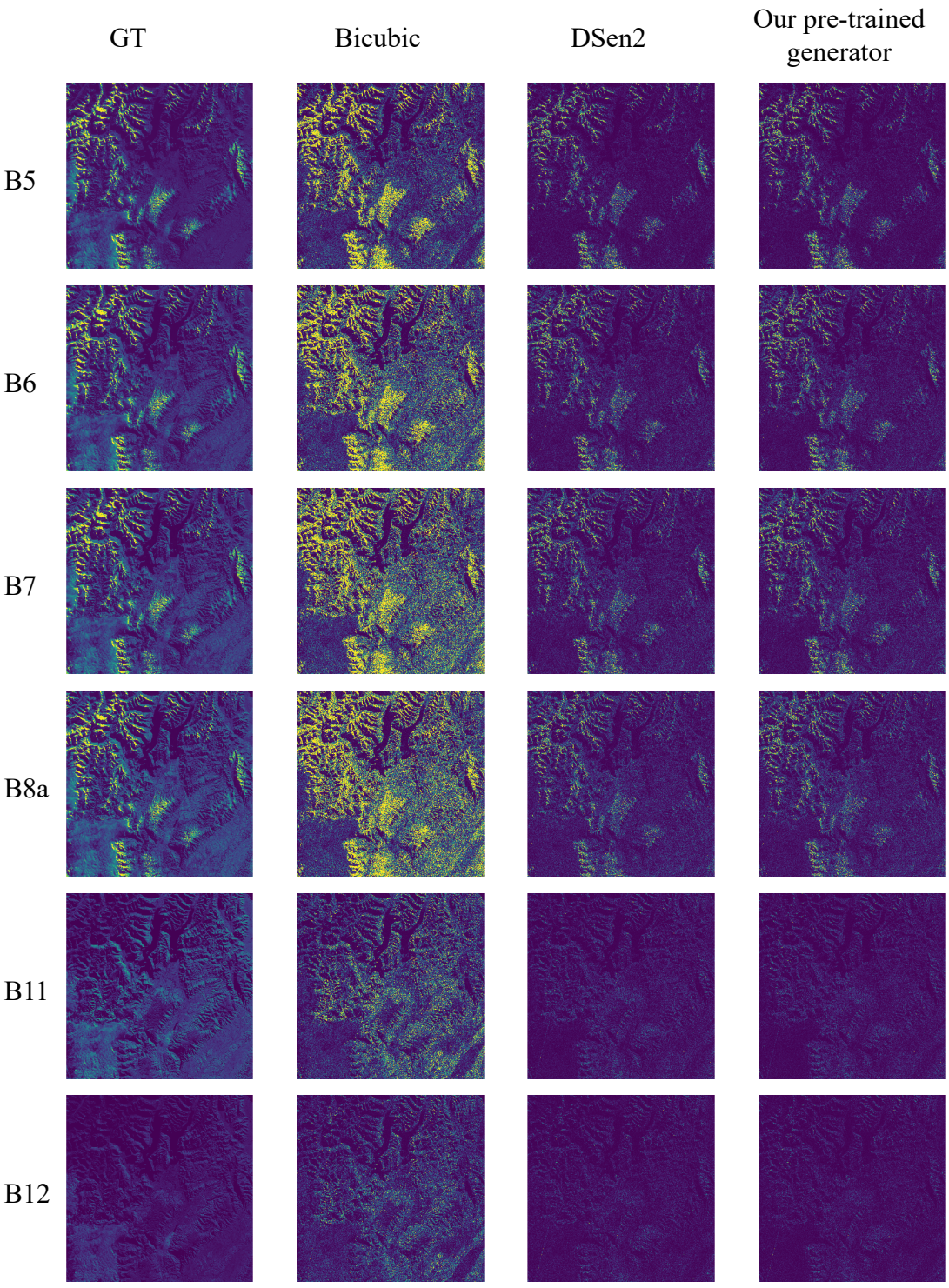


Figure 6.8: Absolute pixel differences between $2\times$ super-resolved results at degraded scale and the original $20m$ bands for L1C tile super-resolution. The value of difference is indicated by the pixel brightness, *i.e.*, the brighter a pixel is, the larger the super-resolution error is.

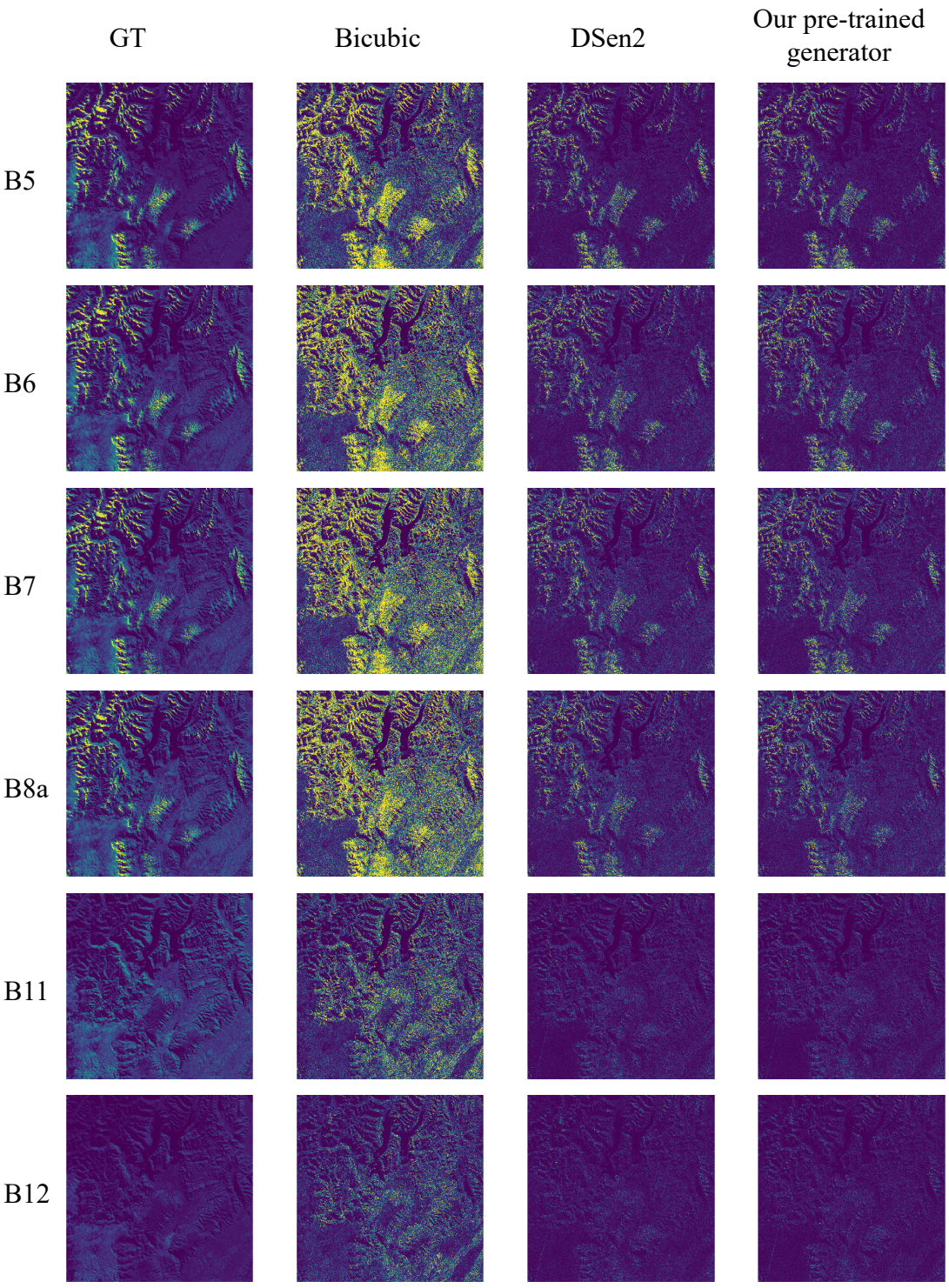


Figure 6.9: Absolute pixel differences between $2\times$ super-resolved results at degraded scale and the original $20m$ bands for L2A tile super-resolution. The value of difference is indicated by the pixel brightness, *i.e.*, the brighter a pixel is, the larger the super-resolution error is.

Figure 6.6, Figure 6.7 and Figure 6.10 are all visualization of super-resolved original band patches. Apart from this, synthetic reconstruction error can be visualized at

degraded scale. Figure 6.8, Figure 6.9 and Figure 6.11 show the absolution pixel residual between the super-resolved synthetic band and the original one. Yellow denotes higher pixel values and dark blue means lower pixel values. Therefore, it is obvious that learning-based methods can significantly reduce the reconstruction error and high-contrast edges reconstruction is difficult for both naive interpolation and learning-based super-resolution (DSen2 and our method).

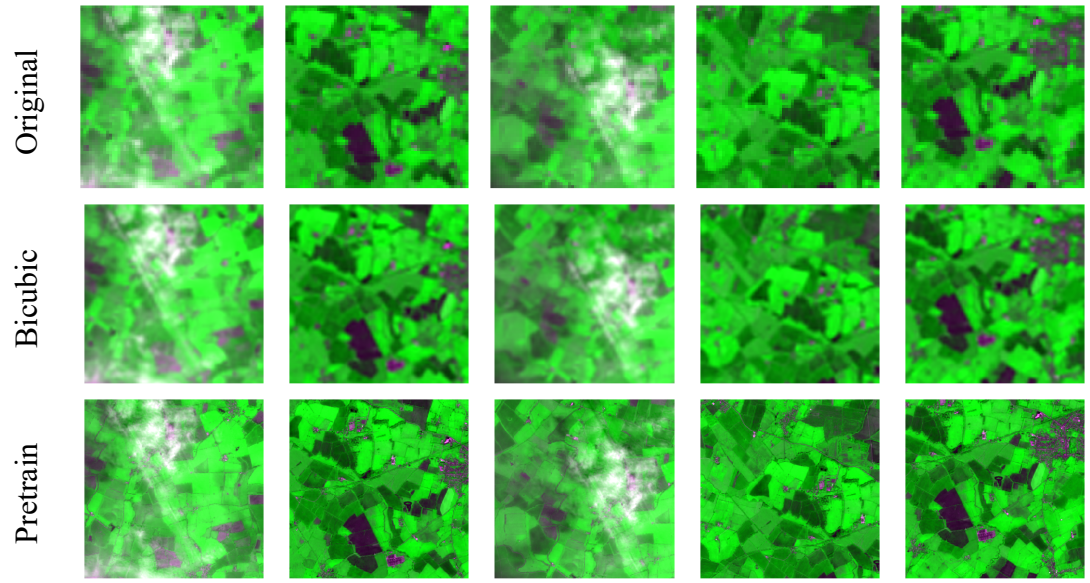


Figure 6.10: Examples of super-resolving the original Sentinel-2 level-1C $60m$ bands to $10m$ GSD. Bands B1, B9 and B1 are used as RGB bands for visualization. From top to bottom are example patches of original Sentinel-2 level-1C bands, super-resolution output of Bicubic interpolation, pre-trained $\mathcal{S}_{6\times}^{L1C}$.

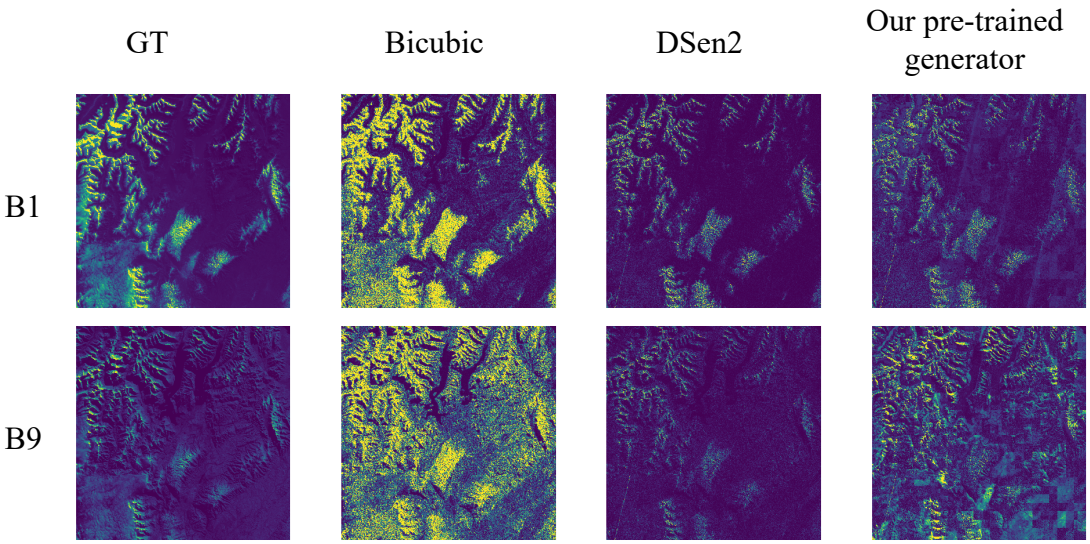
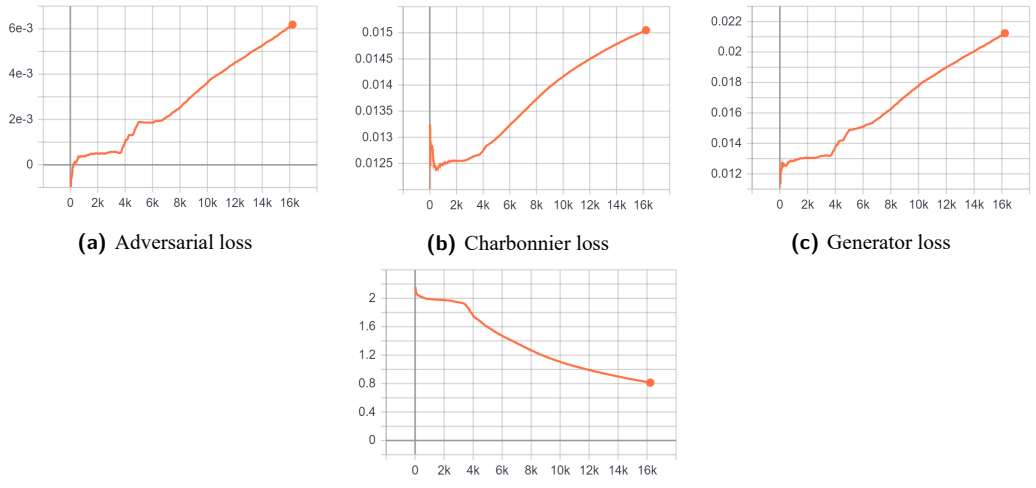


Figure 6.11: Absolute pixel differences between $6\times$ super-resolved results at degraded scale and the original $60m$ bands for L1C tile super-resolution. The value of difference is indicated by the pixel brightness, *i.e.*, the brighter a pixel is, the larger the super-resolution error is.

6.6.3 GAN training process profiling

Instead of simple gradient descent of objective loss functions, the optimization of a GAN is achieved by competence in sync between a generator and a discriminator. With a multitude of parameters guiding the adversarial optimization, it is a challenging task to reach the equilibrium between a generator and a discriminator.

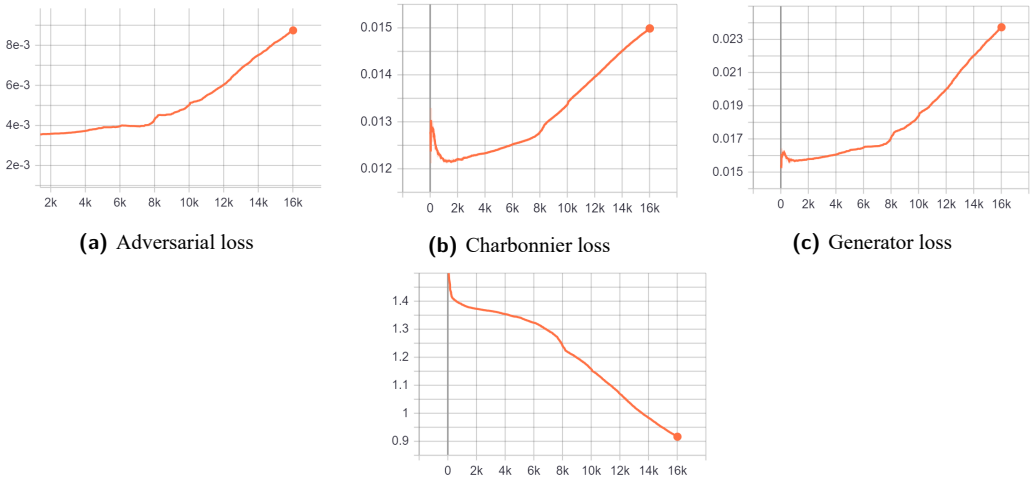
In this thesis, Tensorboard is used to profile the GAN loss dynamics. The four loss components when training a GAN with hinge loss is shown in Figure 6.12. Similarly, Vanila GAN is shown in Figure 6.13, WGAN-GP shown in Figure 6.14 and relativistic GAN shown in Figure 6.15. A coefficient η has been multiplied to the adversarial loss to balance it with the Charbonnier loss, and note that this thesis only test the case when $\eta = 0.0002$. Large amount of time and resources required to run each experiment prevents us to test other possible settings. The generator loss is the weighted sum of adversarial loss and the Charbonnier loss. From the loss dynamics, we can see that all discriminator losses in the four cases steadily decrease, which mean the discriminator become stronger and stronger to distinguish the generator outputs. Before adversarial losses get involved, the generators are already pre-trained with pixel-wise loss, so the adversarial loss and Charbonnier loss start with a very low value. The adversarial loss in WGAN-GP fluctuates throughout the training process, possible caused by the different learning pace of the generator and the discriminator.



(a) Adversarial loss **(b)** Charbonnier loss **(c)** Generator loss

(d) Discriminator loss

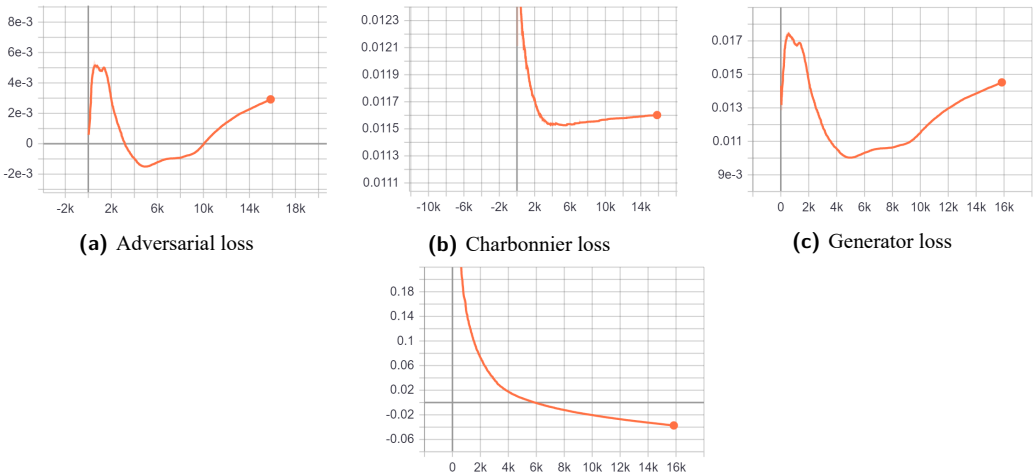
Figure 6.12: Loss profiling of GAN with hinge loss



(a) Adversarial loss **(b)** Charbonnier loss **(c)** Generator loss

(d) Discriminator loss

Figure 6.13: Loss profiling of Vanila GAN



(a) Adversarial loss **(b)** Charbonnier loss **(c)** Generator loss

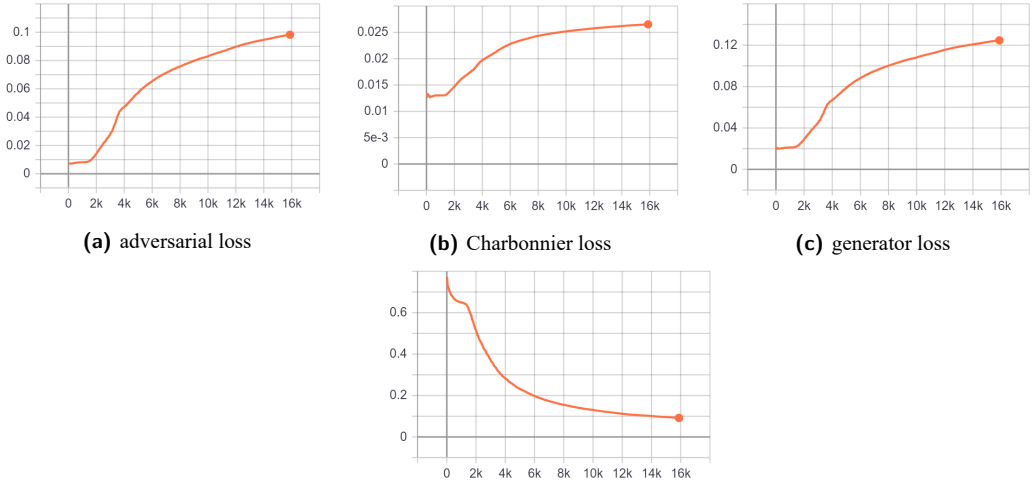
(d) Discriminator loss

Figure 6.14: Loss profiling of WGAN-GP

6.7 Experiments on Distributed Learning

6.7.1 Multi-nodes multi-GPUs training

The two models, $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$, are trained on the GPU clusters, JURON and JUWELS. Their connection topology and architectures are explained in Section 6.1.1 and the largest mini-batch size when training $\mathcal{S}_{2\times}$ (or $\mathcal{S}_{6\times}$) on them with a single GPU is 128 (or 32). Table 6.19 and Table 6.20 shows the synthetic performance of $\mathcal{S}_{2\times}$, $\mathcal{S}_{6\times}$ with scaled mini-batch size and scaled learning rate trained on JURON when super-resolving the degraded Sentinel-2 patches to original scale. When training with four GPUs in 24 hours, $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$ achieved better performance than the learning-based baseline,



(a) adversarial loss **(b)** Charbonnier loss **(c)** generator loss

(d) discriminator loss

Figure 6.15: Loss profiling of relativistic GAN

DSen2. When scaling up to train with 16 GPUs, $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$ can converge faster in four hours and have no severe performance loss. Distributed learning (specifically, synchronous data parallelism) is thus proven to speed up the learning substantially while keeping super-resolution performance intact.

Method	No. GPU	Batch size	Training time	RMSE	SRE	SAM	ERGAS	SSIM	PSNR
Bicubic	-	-	-	125.69	25.64	1.22	3.48	0.82	44.9998
DSen2	1	128	96h	35.85	35.94	0.78	1.07	0.9322	55.5416
$\mathcal{S}_{2\times}^{L1C}$	1	128	24h	36.42	35.83	0.78	1.08	0.9320	55.4393
$\mathcal{S}_{2\times}^{L1C}$	2	256	24h	35.67	36.03	0.77	1.06	0.9329	55.6199
$\mathcal{S}_{2\times}^{L1C}$	4	512	24h	**34.99**	**36.19**	**0.75**	**1.03**	**0.9336**	**55.7756**
$\mathcal{S}_{2\times}^{L1C}$	8	1028	12h	35.61	36.05	0.76	1.05	0.9329	55.6393
$\mathcal{S}_{2\times}^{L1C}$	16	2056	**4h**	38.58	35.27	0.81	1.16	0.9291	54.9243

Table 6.19: The synthetic performance of model $\mathcal{S}_{2\times}^{L1C}$ with scaled batch size and scaled learning rate. The learning rate of the experiment in each row is initialized with $0.0001\times$ No. GPUs

Method	No. GPU	Batch size	Training time	RMSE	SRE	SAM	ERGAS	SSIM	PSNR
Bicubic	-	-	-	161.85	19.79	1.78	7.30	0.36	37.6785
DSen2	1	128	96h	28.11	34.47	0.36	1.38	0.8953	52.4984
$\mathcal{S}_{6\times}^{L1C}$	1	32	24h	29.20	34.00	0.37	1.43	0.8917	51.9991
$\mathcal{S}_{6\times}^{L1C}$	2	64	24h	27.23	34.69	0.35	1.32	0.8959	52.7027
$\mathcal{S}_{6\times}^{L1C}$	4	128	24h	**26.80**	**34.98**	**0.34**	**1.29**	**0.8991**	**52.9451**
$\mathcal{S}_{6\times}^{L1C}$	8	256	12h	27.74	34.54	0.36	1.36	0.8959	52.5506
$\mathcal{S}_{6\times}^{L1C}$	16	512	**4h**	32.28	32.97	0.42	1.62	0.8828	50.9784

Table 6.20: The synthetic performance of model $\mathcal{S}_{6\times}^{L1C}$ with scaled batch size and scaled learning rate. The learning rate of the experiments in each row is initialized with $0.0001\times$ No. GPUs

Figure 6.16 and Figure 6.17 shows the per-second throughput of Sentinel-2 tile patches when training $\mathcal{S}_{2\times}^{L1C}$ and $\mathcal{S}_{6\times}^{L1C}$ on JURON and JUWELS with different number of GPUs. It is obvious that the training speed can not grow linearly. This mainly stems from two reasons: 1) when training with more GPUs, the communication cost to aggregate gradients on all GPUs also increases, 2) the learning speed is influenced

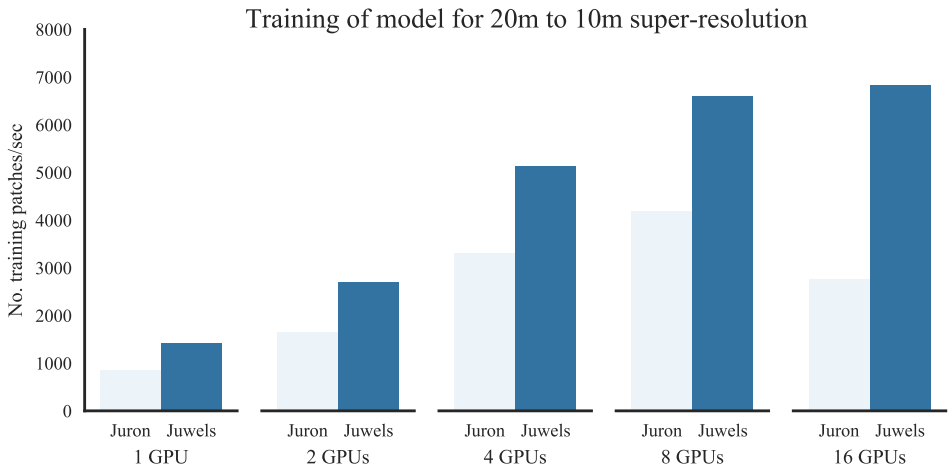


Figure 6.16: Data throughput when training $\mathcal{S}_{2\times}^{L1C}$ on Juron and Juwels.

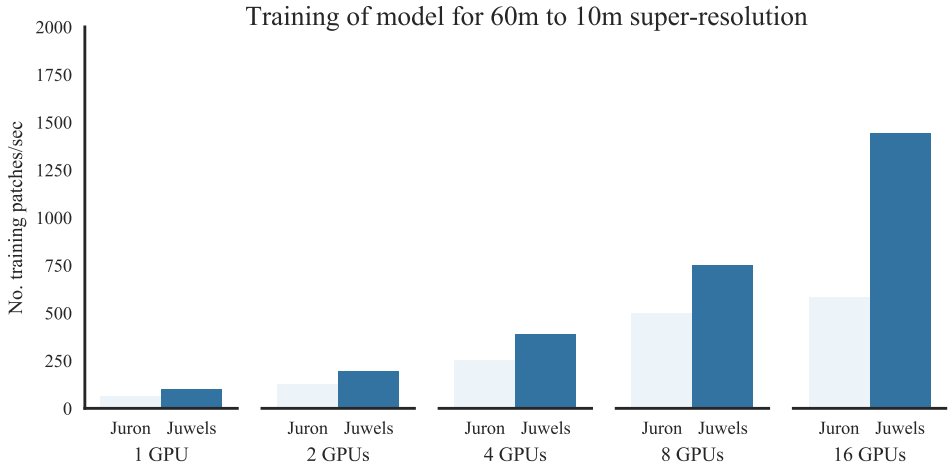


Figure 6.17: Data throughput when training $\mathcal{S}_{6\times}^{L1C}$ on Juron and Juwels.

by both mini-batch size and infrastructure architecture, reflected by the different data throughput of model $\mathcal{S}_{2\times}^{L1C}$ and $\mathcal{S}_{6\times}^{L1C}$, and the performance difference of JURON and JUWELS.

6.7.2 Horovod activity profiling

Horovod has the ability to record the timeline of its activity, called *Horovod Timeline*. It is important for debugging a distributed machine learning system with Horovod. Figure 6.18 shows an example of recording the *Horovod Timeline* when training $\mathcal{S}_{2\times}^{L1C}$ for hundreds of iterations. There are two major phases for each tensor reduction (or parameter optimization iteration):

- **Negotiation**: a phase when all workers send to rank 0 signal that they're ready for synchronization. Each worker reports readiness by **NEGOTIATE_ALLREDUCE**

and immediately after negotiation, rank 0 sends all other workers signal to start synchronization.

- **Processing**: a phase when the operation actually happens represented by **ALL_REDUCE**.

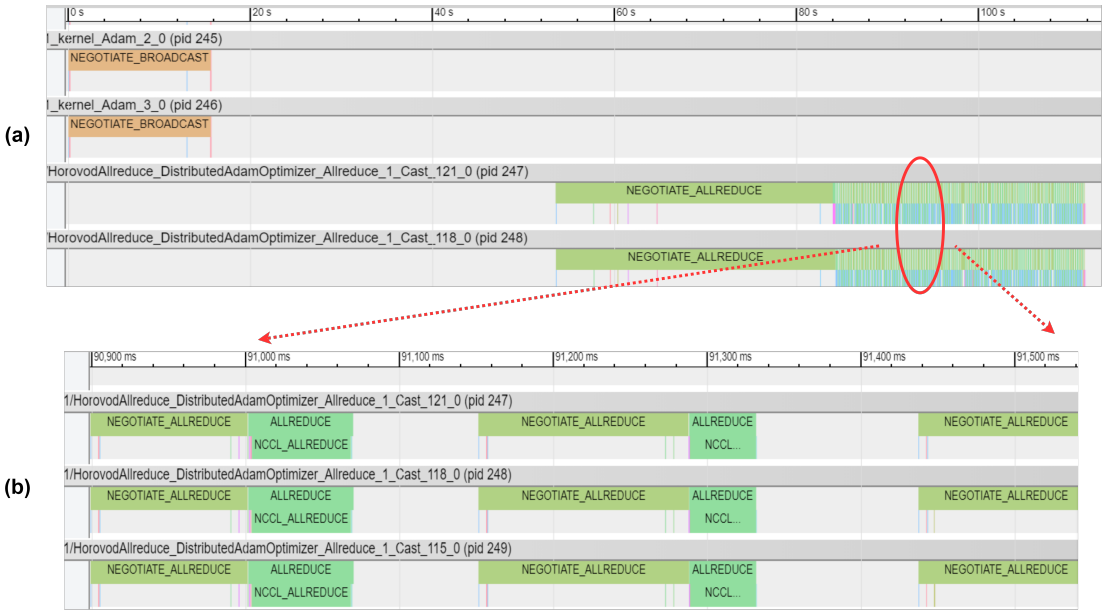


Figure 6.18: Example of Horovod activity profiling. This figure shows the example Horovod timeline when pre-training the $\mathcal{S}_{2\times}^{L1C}$ for hundred of iterations. Figure (a) shows the global trace, where 0 84s is the preparation phase for initializing devices, synchronizing the model parameter, preparing the feeding data *etc.*. From 84s to 112s are model optimization iterations, some of which are expanded in Figure (b). Note that this figure only shows the timing of gradient synchronization (or reduction) over all devices, forwarding the data through each network layer and calculating the gradient of each model parameter are not included in this timeline.

Chapter 7

Discussion and future work

7.1 Discussion

This thesis has worked on the problem of super-resolving the original Sentinel-2 RS images (MSI products) to higher spatial resolution and GSD, due to the high demanding of finer earth surface representations in both scientific and industrial applications in the modern society. To achieve this goal, a deep model based on self-attention mechanism and residual learning is proposed, which has been proved to be effective by extensive experiments and achieved the state-of-the-art performance demonstrated in Section 6.4 and Section 6.5. Residual learning means that this deep model learns the residual between HR references and naively up-sampled inputs (implemented with Bilinear interpolation), instead of learning the direct low $\mapsto$ high resolution mapping. This is applied because residual learning can make the feature space sparser thus the model converging faster. Furthermore, during the experiments, adaptive up-sampling layers (*i.e.*, transposed convolution and sub-pixel convolution) have been tested to replace Bilinear interpolation, but experiments show that they both perform worse and can slow down the training process. More experiments are planned to be conducted to show how to utilize those modules that have been proven to be effective in natural image super-resolution to super-resolve RS images. This thesis also proposed a band fusion module to capture the correlations of multiple spectral bands. This is based on the discovery of Lanars *et al.* [LBDG$^+$18] that higher-resolution bands contribute to super-resolving low-resolution bands even though they are in disjoint spectral bands (*i.e.*, $10m$ bands contribute to $20m$ bands super-resolution, similarly, $10m$ and $20m$ bands are also used as input when super-resolving $60m$ bands). In contrast, previous pan-sharpening methods highly relies on the assumption that HR panchromatic bands overlap at least partially with the LR spectral bands. Last, self-attention module is plugged into a series of residual blocks, which is designed to capture the long-range dependencies over the entire feature map, because a common convolution layer cannot explore the global structures inside an image due to the limitation of receptive fields.

Experiments showed that the two skip connections in the RSA module are helpful to speed up the training process.

This thesis also presents the entire landscape of applying GANs on RS image super-resolution. In this thesis, the generator is first pre-trained by pixel-wise losses with the entire dataset created in Section6.2. A discriminator is then plugged to improve the pre-trained generator further by providing adversarial loss. In this stage, the dataset is split for generator and discriminator training. This is a common paradigm to apply GANs to image super-resolution and mainly stems from three reasons as following: 1) Utilize the dataset more efficiently. In the beginning, the generator can learn the low $\mapsto$ high resolution mapping with the entire dataset and converge to a preliminary satisfying solution faster. 2) Because the pre-trained generator can provide a better output, the discriminator can learn the high frequency difference between the output of generator and the ground truth in a relatively early phase. 3) Usually, a discriminator can easily surpass the generator and cause the failure of training because of diminished gradients. Starting with a pre-trained generator to some extent can help to stabilize the training of GAN. The architecture of discriminator in this thesis is highly influenced by the SRGAN [LTH+17] that is also used by ESRGAN [WYW+18].

As for the stability of GANs for image super-resolution, there are mainly two concerns: 1) The balance between a generator and a discriminator. The complexities (*i.e.*, number of layers and number of parameters) of both generator and discriminator are desired to be in a balance, *i.e.*, a far deeper discriminator and a shallower generator or vice versa has to be avoided in most cases. In addition, the learning rate of a generator and a discriminator can also be considered separately to balance the learning progress. 2) The balance between the content loss and the adversarial loss. Two representative GAN-based super-resolution methods, SRGAN [LTH+17] and ESRGAN [WYW+18], update the generator with a weighted sum of perceptron loss, pixel-wise loss and adversarial loss, where perceptron loss is the most dominant one and calculated with the layered feature in a well-known object classification deep model VGG-19 (pre-trained on Imagenet). Because there is no widely-used RS dataset and pre-trained model in the field of RS to provide perceptron losses, this thesis only updates the generator with the weighted sum of pixel-wise loss and adversarial loss. Because the adversarial loss can be calculate with multiple formulas that has difference numerical scale (see Section 5.2.2), the weight to balance those various adversarial losses is an also important concern when design a GAN-based super-resolution model.

GANs are able to add more high-frequency details to the super-resolved results compared with traditional models without adversarial losses, thus making the output have better perceptual quality. To reconstruct the high-resolution images that is as close as possible to the ground truth is also an important objective in image super-resolution. However, better perceptual quality does not mean higher reconstruction accuracy, *i.e.*, high-frequency details that improve the image perceptual quality can be distortions that don't exist in reality, which is reflected by the cutting-edge GAN-based super-resolution models, SRGAN [LTH+17] and ESRGAN [WYW+18]. Both

of the two methods can produce more visual pleasant results, but perform worse in the sense of quantitative metrics (*e.g.*, RMSE, PSNR and SSIM). This is acceptable when we only desire a photo-realistic output for natural image super-resolution. But for RS super-resolution, the earth observations are expected to be clear, precise and real, that can reveal the actual situation on the earth surface. This is an important concern that should be kept in mind before applying GANs to super-resolve the RS images. Our experiments has also shown the consistent discovery with SRGAN and ESRGAN, that four adversarial losses lead to a worse performance in quantitative evaluation. As for visual assessment, because the raw earth observations are far blurry compared with natural images, the effect of GAN is not very obvious on our Sentinel-2 MSI products.

Another important perspective of this thesis is that the differences between RS and natural images are analyzed, which can explain why RS image super-resolution is distinct from natural image super-resolution. This is also the reason why some state-of-the-art natural image super-resolution algorithms cannot be directly applied to super-resolve the Sentinel-2 MSI products. 1) RS images has more complex data formats. The raw observation is the electromagnetic reflection from the earth surface. Those reflected energy is multiplied with a constant 10000 when converted to geometry images. Compared with natural images, the pixel value in a RS image has wider data range and larger deviation. 2) A RS image has more spectral bands. And in some satellites, *e.g.*, Sentinel-2A&2B, different spectral bands have different spatial resolution. 3) Due to the broader satellite land coverage, a variety of ground scenes are usually contained in one single RS image. In addition, because of the occlusion of clouds and mountains, a RS image is more likely to have large brightness changes inside a single tile. 4) Even with the most advanced sensor, the spatial resolution of a RS image is far coarse than a natural image, *e.g.*, WorldView-4 is one of the satellites equipped with the most sophisticate image sensor and the largest spatial resolution of all bands is $0.31m$[1]. So a RS images is usually more blurry than a natural image. 5) In a standard natural image super-resolution dataset (*e.g.*, DIV2k [AT17]), synthetic LR training patches are created by only Bicubic down-sampling. However, for RS super-resolution, Gaussian filter is used to blurry the image before down-sampling to emulate the modulation transfer function of a RS sensor. All those difference make RS image super-resolution even more challenging.

Thanks to the support from Juelich Supercomputing Center, our model is scaled uo to HPC systems, thus we can train and evaluate our model faster with massive RS observation data. Experiments have shown that our distributed model can achieve the state-of-the-art performance and significantly reduce the training time from several days to several hours. To exploit the multiple GPUs in multiple nodes efficiently, synchronous data parallelism is used to scale-up the training process, so a non-trivial growth of mini-batch size happens when training with more GPUs. This thesis proved that when training $\mathcal{S}_{2\times}$ with mini-batch size up tp 2056 or training $\mathcal{S}_{6\times}$ with mini-batch size up to 512, the model can be trained successfully and has no severe performance loss. Distributed training is thus shown to speed up learning substantially while keeping

[1] https://www.euspaceimaging.com/about/satellites/

performance intact. Furthermore, instead of setting up central parameter severs, ring-reduction mechanism (embedded in the library Horovod) is applied to aggregate and average the gradients over multiple machines. This distributed training framework has been tested by extensive experiments in two high-performance computing systems, *i.e.*, JURON and JUWELS, which are constructed with different GPU connection topology. One more concern in distributed machine learning is the communication costs to aggregate the gradients grows with the increase of mini-batch size and number of used GPUs, thus the training speed usually cannot increase linearly with the number of used machines. To back up this point, the data throughput when training our super-resolution model is recorded on both of the two HPC systems, JURON and JUWELS.

In this thesis, a comprehensive assessment of a RS super-resolution deep model has been proposed. First, because there is no standard dataset for RS image super-resolution, most of previous RS super-resolution methods are only tested with few samples. Our model is evaluated with a much larger dataset that covers various climate zones and terrains, which can reflect the performance of a deep model better. Secondly, because of the missing high-resolution references, most previous methods can only do quantitative evaluation at degraded scale. At original scale, only visual assessment is provided in their methods. This thesis evaluates our model by calculating both synthesis and consistency property, which enables us to do quantitative evaluation at both degraded and original scale. By extensive experiments, two conclusions can be made: 1) the model in this thesis have achieved the state-of-the-art performance for $20m \mapsto 10m$ super-resolution in the sense of both synthesis and consistency property; 2) As for $60m \mapsto 10m$ super-resolution, our model has achieved state-of-the-art synthetic property. For consistency property, our model $\mathcal{S}_{6\times}$ is shown to be better than the learning-based method DSen2 (widely-used as baseline in this problem), but worse than the naive interpolation. More experiments are planned to study the reason and the consistency property in large scale ($6\times$ or even lager) super-resolution. Last, this thesis also evaluated our model with multiple input formats (*i.e.*, Level-1C and Level-2A). The statistical difference is illustrated with a simple example of the panorama of the city Aachen. It has been shown that a model trained with data of format level-1C cannot super-resolve the level-2A tiles well, and the model in this thesis has also achieved a better performance than DSen2 for L2A super-resolution in sense of both synthetic and consistency property.

7.2 Future Work

In Section 4.1, this thesis only presents a super-resolution model based on pre-upsampling paradigm. But some architectures with post-upsampling paradigm have also been implemented during the project, that can also yield comparable performance. In the future, more experiments can be done to find a better generator and study the difference of the four paradigms, see Figure 2.9, on RS image super-resolution. Then, with the difference between natural and remote sensing images in mind, we can migrate the modules

widely used in natural image super-resolution to RS image super-resolution, *e.g.*, shared global connection, channel attention, second order channel attention, normalization techniques including instance normalization, layer-wise normalization, spectral normalization, densely connected network *etc.*. Their implementation are also included in our project repository[2]. With those modules, community can construct their own super-resolution generators.

This thesis has proposed a deep model to super-resolve Sentinel-2 tiles, evaluated by extensive experiments in the sense of both synthetic and consistency property. But more experiments can be done to show the benefits from super-resolution. For example, we can super-resolve the earth observation to higher resolution and test if it contribute to land cover classification, cloud removal *etc.*.

In the perspective of GANs, this thesis focuses on the way to embed adversarial losses more stably to a super-resolution model, *e.g.*, testing four kinds of ways to calculate adversarial loss, using different learning rate and comparable model complexity for generator and discriminator, pre-training generator before discriminator getting involved. However, we do not research much on the architecture of discriminator. Based on the differences between natural and RS image super-resolution (RS images has higher representation complexity and higher signal reconstruction accuracy demand), one possible direction to complement this work in the future can be to design a particular discriminator for the RS earth observation super-resolution. Besides, one more future direction is addressing the reconstruction distortions introduced by adversarial losses. For example, some distortion measures [BMT+18] can be embed to the generator optimization to improve the reconstruction accuracy while keeping the better perceptual quality.

Furthermore, in the perspective of distributed learning, modified linear learning rate scale rule is used to resolve the loss explosion when linear learning rate scale rule is used. But more strategies or heuristics can be tried to improve the model convergence rate on HPC clusters in the future, *e.g.* learning rate restart, warmup or distillation [GKXS18].

[2] https://gitlab.version.fz-juelich.de/cavallaro1/gan_superesolution

Acknowledgements

Research leading to these results has in parts been carried out on the Human Brain Project PCP Pilot Systems at the Juelich Supercomputing Centre, which received co-funding from the European Union (Grant Agreement no.604102)

Bibliography

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

[Ada95] D. Adams. *The Hitchhiker's Guide to the Galaxy*. San Val, 1995.

[AKB19] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *arXiv preprint arXiv:1904.07523*, 2019.

[AT17] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[B+95] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[BBH+92] Michael Bevis, Steven Businger, Thomas A Herring, Christian Rocken, Richard A Anthes, and Randolph H Ware. Gps meteorology: Remote sensing of atmospheric water vapor using the global positioning system. *Journal of Geophysical Research: Atmospheres*, 97(D14):15787–15801, 1992.

[BD05] DS Boyd and FM Danson. Satellite remote sensing of forest resources: three decades of research development. *Progress in Physical Geography*, 29(1):1–26, 2005.

[BDD+17] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[BDS18] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[BDX16] Weixin Bian, Shifei Ding, and Yu Xue. Fingerprint image super resolution using sparse representation with ridge pattern prior by classification coupled dictionaries. *IET Biometrics*, 6(5):342–350, 2016.

[BM18] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6228–6237, 2018.

[BMT+18] 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)*, pages 0–0, 2018.

[Bro17] Nicolas Brodu. Super-resolving multiresolution images with band-independent geometry of multispectral pixels. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8):4610–4617, 2017.

[BSM17] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[C+15] François Chollet et al. Keras, 2015.

[CDLH18] Yanshuai Cao, Gavin Weiguang Ding, Kry Yik-Chau Lui, and Ruitong Huang. Improving gan training via binarized representation entropy (bre) regularization. *arXiv preprint arXiv:1805.03644*, 2018.

[DCM+12] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231, 2012.

[DCZ+19] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.

[DGI17] Neil Patrick Del Gallego and Joel Ilao. Multiple-image super-resolution on mobile devices: an image warping approach. *EURASIP Journal on Image and Video Processing*, 2017(1):8, 2017.

[DLHT14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[Duc79] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of applied meteorology*, 18(8):1016–1022, 1979.

[DV16] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[GAA+17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[GDG+17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[Gib17] Andrew Gibiansky. Bringing hpc techniques to deep learning.(2017). *URL http://research. baidu. com/bringing-hpc-techniques-deep-learning*, 2017.

[GKXS18] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gre08] Hayit Greenspan. Super-resolution in medical imaging. *The Computer Journal*, 52(1):43–63, 2008.

[GWK+18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[HCS+17] Karol Hausman, Yevgen Chebotar, Stefan Schaal, Gaurav Sukhatme, and Joseph J Lim. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1235–1245. Curran Associates, Inc., 2017.

[HFBP+18] Juan Mario Haut, Ruben Fernandez-Beltran, Mercedes E Paoletti, Javier Plaza, Antonio Plaza, and Filiberto Pla. A new deep generative network for unsupervised remote sensing single-image superresolution. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11):6792–6810, 2018.

[HFBP+19] Juan Mario Haut, Ruben Fernandez-Beltran, Mercedes E Paoletti, Javier Plaza, and Antonio Plaza. Remote sensing image superresolution using deep residual channel attention. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[HRU+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[HS15] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

[HSF17] Yawen Huang, Ling Shao, and Alejandro F Frangi. Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6070–6079, 2017.

[HW62] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The*

Journal of physiology, 160(1):106–154, 1962.

[HZC+17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[IEO19] Inc Infiniti Electro-Optics, a division of Ascendent Technology Group. https://www.infinitioptics.com/technology/multi-sensor. 2019.

[IK15] Jithin Saji Isaac and Ramesh Kulkarni. Super resolution techniques for medical image processing. In *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, pages 1–6. IEEE, 2015.

[IP91] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[JM18] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[JP19] Dong-Won Jang and Rae-Hong Park. Densenet with deep residual channel-attention blocks for single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[KAC09] Muhammad Murtaza Khan, Luciano Alparone, and Jocelyn Chanussot. Pansharpening quality assessment using the modulation transfer functions of instruments. *IEEE transactions on geoscience and remote sensing*, 47(11):3880–3891, 2009.

[KBP+19] Michal Kawulok, Pawel Benecki, Szymon Piechaczek, Krzysztof Hrynczenko, Daniel Kostrzewa, and Jakub Nalepa. Deep learning for multiple-image super-resolution. *arXiv preprint arXiv:1903.00440*,

2019.

[KKLML16a] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[KKLML16b] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.

[Kra19] Dorian Krause. Juwels: Modular tier-0/1 supercomputer at the jülich supercomputing centre. *JLSRF*, 5:A135, 2019.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KT16] Dorian Krause and Philipp Thörnig. Jureca: general-purpose supercomputer at jülich supercomputing centre. *Journal of large-scale research facilities JLSRF*, 2:62, 2016.

[LBDBS17] Charis Lanaras, José Bioucas-Dias, Emmanuel Baltsavias, and Konrad Schindler. Super-resolution of multispectral multiresolution images from a single sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.

[LBDG+18] Charis Lanaras, José Bioucas-Dias, Silvano Galliani, Emmanuel Baltsavias, and Konrad Schindler. Super-resolution of sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:305–319, 2018.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[LFCS07] Frank Lin, Clinton Fookes, Vinod Chandran, and Sridha Sridharan. Super-resolved faces for improved face recognition from surveillance video. In *International Conference on Biometrics*, pages 1–10. Springer, 2007.

[LK16] Lukas Liebel and Marco Körner. Single-image super resolution for multispectral remote sensing data using convolutional neural networks. *ISPRS-International Archives of the Photogrammetry, Remote*

Sensing and Spatial Information Sciences, 41:883–890, 2016.

[LKM+18] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 700–709, 2018.

[LR18] Jens Leitloff and Felix M. Riese. Examples for CNN training and classification on Sentinel-2 data. http://doi.org/10.5281/zenodo.3268451, 2018.

[LS76] J Lintz and DS Simonett. Remote sensing of environment addision wesley. *Rading mars*, 1976.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[LSK+17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[LSZ17] Sen Lei, Zhenwei Shi, and Zhengxia Zou. Super-resolution for remote sensing images via local–global combined network. *IEEE Geoscience and Remote Sensing Letters*, 14(8):1243–1247, 2017.

[LTH+17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[LWL18] Xiangyu Liu, Yunhong Wang, and Qingjie Liu. Psgan: a generative adversarial network for remote sensing image pan-sharpening. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 873–877. IEEE, 2018.

[MA05] Charles N Mundia and Masamu Aniya. Analysis of land use/cover changes and urban expansion of nairobi city using remote sensing and gis. *International journal of Remote sensing*, 26(13):2831–2849, 2005.

[MB09] Sabyasachi Maiti and Amit K Bhattacharya. Shoreline change analysis and its application to prediction: A remote sensing and statistics based approach. *Marine Geology*, 257(1-4):11–23, 2009.

[MLX+17] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[MPGL18] Wen Ma, Zongxu Pan, Jiayi Guo, and Bin Lei. Super-resolution of remote sensing images based on transferred generative adversarial network. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1148–1151. IEEE, 2018.

[NM14] Kamal Nasrollahi and Thomas B Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6):1423–1468, 2014.

[ODO16] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[PLPD18] Darren Pouliot, Rasim Latifovic, Jon Pasher, and Jason Duffe. Landsat super-resolution enhancement using convolution neural networks and sentinel-2 for training. *Remote Sensing*, 10(3):394, 2018.

[PSC+18] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee. Srfeat: Single image super-resolution with feature discrimination. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 439–455, 2018.

[PSU18] Frosti Palsson, Johannes R Sveinsson, and Magnus O Ulfarsson. Single sensor image fusion using a deep convolutional generative adversarial network. In *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5. IEEE, 2018.

[PSUB15] Frosti Palsson, Johannes R Sveinsson, Magnus Orn Ulfarsson, and Jon Atli Benediktsson. Quantitative quality evaluation of pansharpened imagery: Consistency versus synthesis. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1247–1259, 2015.

[PY09] Pitch Patarasuk and Xin Yuan. Bandwidth optimal all-reduce algorithms for clusters of workstations. *Journal of Parallel and Distributed Computing*, 69(2):117–124, 2009.

[PYB11] Hai Minh Pham, Yasushi Yamaguchi, and Thanh Quang Bui. A case study on the relation between city planning and urban growth using remote sensing and spatial metrics. *Landscape and Urban Planning*, 100(3):223–230, 2011.

[RAY+16] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[RLNH17] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, pages 2018–2028, 2017.

[RUEA16] Pejman Rasti, Tonis Uiboupin, Sergio Escalera, and Gholamreza Anbarjafari. Convolutional neural network super resolution for face recognition in surveillance monitoring. In *International conference on articulated motion and deformable objects*, pages 175–184. Springer, 2016.

[RVR11] Oscar Rojas, Anton Vrieling, and Felix Rembold. Assessing drought probability for agricultural areas in africa with coarse resolution remote sensing imagery. *Remote sensing of Environment*, 115(2):343–352, 2011.

[SCH+16] 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[SDB18] Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799*, 2018.

[SHM+04] Douglas A Stow, Allen Hope, David McGuire, David Verbyla, John Gamon, Fred Huemmrich, Stan Houston, Charles Racine, Matthew Sturm, Kenneth Tape, et al. Remote sensing of vegetation and land-cover change in arctic tundra ecosystems. *Remote sensing of environment*, 89(3):281–308, 2004.

[SKYL17] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[SSH17] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500, 2017.

[STW+11] Heng Su, Liang Tang, Ying Wu, Daniel Tretter, and Jie Zhou. Spatially adaptive block-based super-resolution. *IEEE Transactions on Image Processing*, 21(3):1031–1045, 2011.

[TJ16] Charles Toth and Grzegorz Jóźków. Remote sensing platforms and sensors: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:22–36, 2016.

[TLLG17] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4799–4807, 2017.

[TMPE09] Hiroyuki Takeda, Peyman Milanfar, Matan Protter, and Michael Elad. Super-resolution without explicit subpixel motion estimation. *IEEE Transactions on Image Processing*, 18(9):1958–1975, 2009.

[TSG+03] Woody Turner, Sacha Spector, Ned Gardiner, Matthew Fladeland, Eleanor Sterling, and Marc Steininger. Remote sensing for biodiversity science and conservation. *Trends in ecology & evolution*, 18(6):306–314, 2003.

[VPT16] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[Wal00] Lucien Wald. Quality of high resolution synthesised images: Is there a simple criterion? 2000.

[WCH19] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *arXiv preprint arXiv:1902.06068*, 2019.

[WGL+18] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv preprint arXiv:1803.01541*, 2018.

[Wik19] Wikipedia contributors. Bicubic interpolation — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Bicubic_interpolation&oldid=906667716, 2019. [Online; accessed 28-February-2020].

[WRM97] Lucien Wald, Thierry Ranchin, and Marc Mangolini. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogrammetric Engineering and Remote Sensing*, 63:691–699, 11 1997.

[WSAPI15] Qunming Wang, Wenzhong Shi, Peter M Atkinson, and Eulogio Pardo-Igúzquiza. A new geostatistical solution to remote sensing image downscaling. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):386–396, 2015.

[WXWT18] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing*, 27(8):4066–4079, 2018.

[WYW+18] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[WZX+16] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016.

[YFH+17] Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John Paisley. Pannet: A deep network architecture for pan-sharpening. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[YGF+13] Jun Yang, Peng Gong, Rong Fu, Minghua Zhang, Jingming Chen, Shunlin Liang, Bing Xu, Jiancheng Shi, and Robert Dickinson. The role of satellite remote sensing in climate change studies. *Nature climate change*, 3(10):875–883, 2013.

[YGG17] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[YH10] Jianchao Yang and Thomas Huang. Image super-resolution: Historical overview and future challenges. *Super-resolution imaging*, pages 20–34, 2010.

[YLZ+18] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018.

[ZGMO18] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[ZHL+10] Yongnian Zeng, Wei Huang, Maoguo Liu, Honghui Zhang, and Bin Zou. Fusion of satellite images in urban area: Assessing the quality of resulting images. In *2010 18th International Conference on Geoinformatics*, pages 1–4. IEEE, 2010.

[ZK19a] Dan Zhang and Anna Khoreva. Pa-gan: Improving gan training by progressive augmentation. *arXiv preprint arXiv:1901.10422*, 2019.

[ZK19b] Brady Zhou and Philipp Krähenbühl. Don't let your discriminator be fooled. In *International Conference on Learning Representations*, 2019.

[ZLL+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)*, pages 286–301, 2018.

[ZML16] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[ZQS+18] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[ZTK+18] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.

[ZZSL10] Liangpei Zhang, Hongyan Zhang, Huanfeng Shen, and Pingxiang Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010.